DEVELOPMENT OF NONLINEAR FINITE ELEMENT CODES FOR STABILITY STUDIES OF WEB MATERIAL WITH STRAIN STATE DEPENDENT PROPERTIES

By

HASAN H.YURTCU

Bachelor of Mathematical Engineering
Yildiz Technical University
Istanbul, Turkiye
2003

Master of Science Mathematical Engineering
Yildiz Technical University
Istanbul, Turkiye
2006

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2011

DEVELOPMENT OF NONLINEAR FINITE ELEMENT
CODES FOR STABILITY STUDIES OF WEB
MATERIAL WITH STRAIN STATE
DEPENDENT PROPERTIES

Dissertation Approved:

Dr. J.K. Good

Dissertation Adviser

Dr. R.P.Singh

Dr .A.K.Kalkan

Dr.B.Binegar

Outside Committee Member

Dr.Sheryl A.Tucker

Dean of the Graduate College

TABLE OF CONTENTS

Page

LIST OF SYMBOLS

a	Web Span Length
E	Young's Modulus
FE	Finite Elements
Fig.	Figure
MD	Machine Direction
CMD	Cross Machine Direction
R	Radius of the roller
t	Web Thickness
Tw	Web line tension
X,Y,Z	Coordinates
σ	Stress
σ_{ycr}	Critical Buckling Stress
σ_{md}	Machine Direction Stress
σ_{cmd}	Cross Machine Direction Stress
ν	Poison's Ratio
ε	Strain
μ	Coefficient of Friction
I, I_z	Area Moment of Inertia
m_{cr}	Critical Roller Taper
T	Web Tension
T_{slack}	Tension Where Slack Edge Occurs

V, V_{avg}	Velocity, Average Velocity
ω	Angular Velocity
σ_1, σ_2	Principal Normal Stresses
$F_{friction}$	Friction Force
FE	Finite Elements
x,y,z	Coordinate Directions
σ_y	Normal Stress
ε_x, ε_y	Normal Strains
ε_1, ε_2	Principal strains

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

1. CHAPTER I

1.1 INTRODUCTION

A web is any flexible thin material. Webs are made in continuous production processes. The webs are often stored in wound roll form, since this is the only convenient means of storing long lengths of flexible material. A production roll of polyester film that is 48 gages (0.00048”) in thickness might be 4 feet in diameter. If the web is wound on a 6” diameter core there can be nearly 60 miles of web length in the roll. That polyester web could be nearly 300 inches wide when made, but costumer rolls to be shipped could be only 6 inches to 60 inches in width depending on the products to be made. Thus a particular web might be 60 miles long, 60 inches wide, and 0.00048 inches in thickness. Plastic films, papers, foils, and thin metal sheets are examples of webs. Web handling can be defined as all processes employed during the transportation of webs. Cutting, coating, slitting, printing, laminating and drying are some of the processes that add value to the web. During these processes, web materials must travel around several rollers in the process equipment.

During the transportation of webs trough process machinery, compressive stresses are induced in webs and these compressive stresses may cause instabilities. If these instabilities occur in free spans, which are the sections of web between two rollers, they are called *troughs* and if they occur at rollers, they are called *wrinkles* (Figure 1). Wrinkles and troughs result in loss of value and quality of the webs. For instance, if wrinkles may cause permanent damages to the webs the result is wasted web and highly priced downtime of web lines. The processes such as laminating, printing on paper, metalizing of films or coating require web to be planar. Troughs may affect these processes and like wrinkles the result is waste of material and time. As a result, wrinkles and troughs are two important engineering problems in the web handling industry.

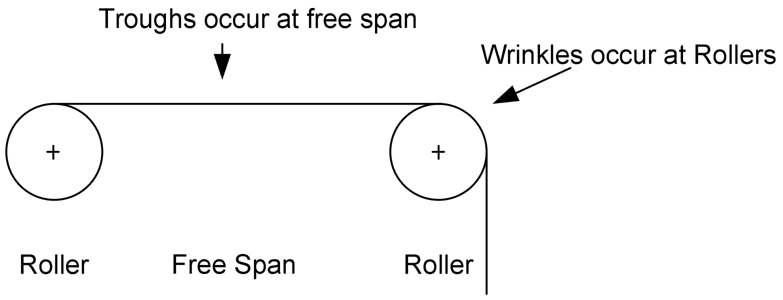


Figure 1. 1 Troughs and Wrinkles

There are three types of wrinkles. Machine direction wrinkles, cross machine direction wrinkles and shear wrinkles. The machine direction [MD] is the direction of the web travel through the web process machine and the cross machine direction [CMD] is the direction that is perpendicular to the machine direction (i.e. across the web width). MD wrinkles occur because of the compressive forces in the CMD. CMD wrinkles are produced during winding or unwinding processes due to interlayer slippage that may be the result of air entrainment [1]. Shear forces due to the imperfect rollers, misaligned rollers and non uniform webs may result in shear troughs and wrinkles.

Shear wrinkles can result from misaligned or tapered rollers [1, 2]. Shear wrinkles can be classified as regime one wrinkles and regime two wrinkles.

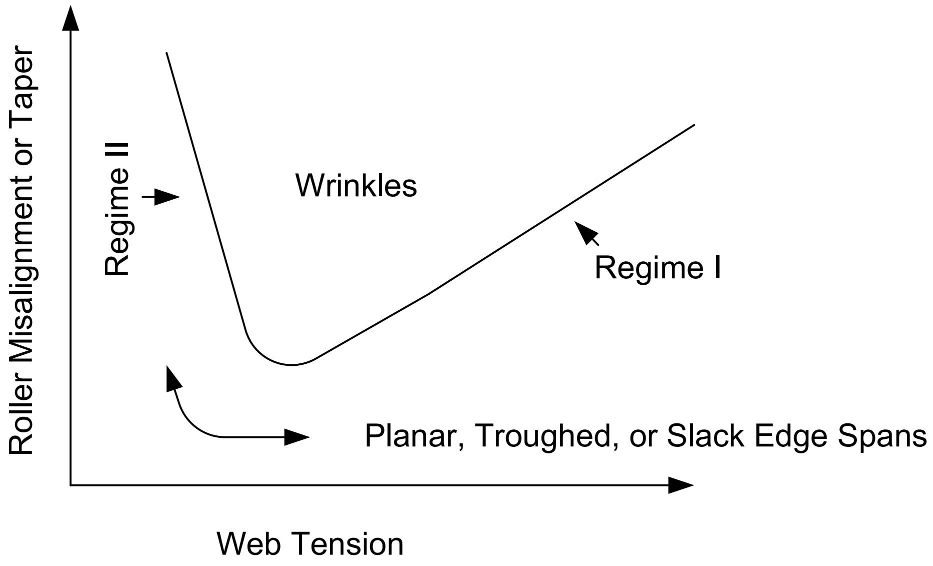

Roller Misalignment or Taper
Regime II
Wrinkles
Regime I
Planar, Troughed, or Slack Edge Spans
Web Tension

Figure 1. 2 Regime I and Regime II Wrinkles

Regime one wrinkles are diagonal wrinkles which occur because of the presence of a lateral shear force in the web. The shear force might be due to a misaligned or tapered roller. The shear forces in a web which are necessary to cause troughs or wrinkles are very small. The misalignments and tapers which cause the shear forces are unintentional. For regime one troughs and wrinkles to occur there must be sufficient traction between the web and the rollers to react the shear forces. The shear forces occur because when traction is adequate the web will attempt to gain normal entry to the downstream roller. The concept or law of normal entry of a web to a roller is attributed to Lorig [3]. Regime two wrinkles are dependent on traction and velocity between the web and the roller. In this case, if traction is insufficient, CMD compressive stresses cannot develop in the web. Frictional restraint forces between the web and the roller are required to sustain wrinkles. Increases in web tension result in increased normal forces between the web and roller and act to decrease the air films which develop between web and roller. Air films can develop between webs and rollers due to the hydrodynamic entrainment of air. Increased web tension and normal force act to decrease the thickness of the air films. The air films result in decrease in traction between the web and rollers. Thus an increase in web tension will increase the potential frictional forces that would be needed to sustain a wrinkle on a roller. The normal entry rule may be violated as a result [3]. Good et al [4] showed that regime two wrinkles could be predicted. In this proposal it will be assumed that sufficient traction exist between webs and rollers such that only regime one wrinkles need to be considered.

Wrinkles are defined as buckling of webs around rollers. Webs have very small resistance to compression in free spans. Webs may withstand more compressive stress on the rollers than free span, because the web has greater stability in the form of a cylindrical shell than it has a flat plate. The critical CMD compressive stress needed to wrinkle the web can be predicted using classical cylindrical shell buckling expressions.

Previous research by Webb[9], Beisel[12] and Mallya[15] have shown that web wrinkles on rollers can be predicted by analyzing the compressive CMD stresses that form in the web as it contacts rollers. These compressive stresses arise due to troughs that have already formed in the free span prior to the rollers. The troughs themselves may be the result of many disturbances that exist in web lines. Misaligned rollers, tapered rollers and web non uniformities (such as holes) are examples of such disturbances. Thus, it could be stated that web wrinkles on rollers are the result of two instabilities. A disturbance is first required to induce a trough instability in the free span. The first instability has now occurred. The web that was planar in the free span has now troughed. After the troughs appear CMD compressive stresses will arise in the web entering the roller downstream of that span.

When the troughs first appear, the CMD stress in the web entering the roller may be small. Thus, whatever disturbance produced the trough initially may have to become yet larger before sufficient CMD compressive stress can be generated in the web on the roller to produce an MD wrinkle. At this point, the second instability has occurred. The web on the roller that earlier had the shape of a sector of a cylindrical shell has now buckled. Thus, it has been proven that the prediction of wrinkles upon a roller involves:

(1) An instability (troughs) occurring in the span upstream of the roller.

(2) As whatever disturbance increases that produced the troughs in step (1), a post buckling analysis must be undertaken. It will bill seen that as the disturbance increases that a CMD compressive stress will arise in the web on the downstream roller.

(3) As the disturbance yet increase further the CMD compressive stress in the web on the

downstream roller is also increasing. As this CMD stress increases it will finally surpass the cylindrical shell buckling stress. At this point, with sufficient friction between the web and the roller, a wrinkle will form in the web on the roller.

The purpose of this study is to develop efficient computational tools that can accomplish the analyses required in step (1), (2), and (3).

2. CHAPTER II

2.1 LITERATURE REVIEW

The literature has been reviewed and the findings will be broken into two sections. First the basic theories of membrane instability will be reviewed. Second, those studies which examine web instability will be reviewed. At the close of this chapter a final summary of the findings will be included and a statement of proposed research will be presented.

2.2 Theory of Membrane Instability

Wagner [18] prepared a treatise on sheet metal girders with very thin webs. Probably this study is the earliest investigation of the mechanics of wrinkling membranes. He worked to develop the structural method of sheet metal girders. His methods were based on the assumption of the low stiffness in bending of the metal web. He worked to explain the behavior of the thin metal webs in beams carrying a shear load well in excess of the initial buckling value. He proposed tension field theory. Tension field theory helps to analyze flexible structures that can support tension, but have no ability to resist compression. This theory was further developed by Reissner [23], Stein-Hedgepeth [19] and Mikolas [20, 21]. Miller-Hedgepeth [5] and Miller et al. [6, 22] adopted this theory to the finite element method. Tension field theory can be applied to the web lines due to the fact that web lines can support tension but cannot carry compression and also web materials are flexible structures.

Stein and Hedgepeth [19] suggested a particularly useful approach concerning partly wrinkling membranes. This work is a seminal work in this field. They derived a theory to predict the stresses and deformations of stretched membrane structural components for loads under which part of the membrane wrinkles. Their theory was based on the basic assumption that a membrane has no bending stiffness and because of this can carry no compressive stress. They applied their theory to in-plane bending of a stretched rectangular membrane, a pressurized cylinder, and to the rotation of a hub in a stretched circular membrane. They presented stresses and deformations in equation form for the wrinkled and unwrinkled regions. The membrane they considered is elastic, isotropic, has no bending stiffness, and cannot carry compressive stress. In their work they considered average strains and displacements of the wrinkling material rather than detailed deformations of each wrinkle. In terms of the wrinkling equations given, their theory was limited in the sense that average strains must be small compared with unity. They started to investigate the wrinkling region by looking the principal stresses. They used a criterion that if both principal stresses are zero, the membrane is unloaded and thus will not wrinkle. Their criteria for a wrinkled membrane is one principal stress must be zero and other principal stress is non zero and tensile. The nonzero principal stress may be assumed to act along the crest of the wrinkle.

The approach that was developed by Stein and Hedgepeth [19] was further developed and applied by Mikolas [20, 21]. He presented experiments and analysis for the wrinkling behavior of stretched membranes under the influence of a torque loading through an attached hub. He found that theory and experimental results were in a very good agreement. The work done by Stein and Hedgepeth [19] and Mikolas [20, 21] were closed form solutions and did not involve the finite element method. The principal stress criterion that they used is employed in our current approach.

Miller and Hedgepeth [5] developed a new algorithm for finite element analysis based on the same assumptions and field equations after finding some critical disadvantages connected with the Stein and Hedgepeth [19] approach. This work may be the most important study in this field. In their algorithm the element stiffness is dependent on the current principal strains. Wrinkling membrane elements can have either taut behavior, wrinkled behavior or slack behavior. In taut behavior both principal stresses are larger than zero, in wrinkled behavior one of the principal stresses is greater than zero the other is equal to zero and in slack behavior both principal stresses are equal to zero. In their algorithm, in the first load step all elements are assumed to taut behavior. In the consecutive steps, element behaviors are calculated with respect to strain states of previous steps. In other words, the decision on the stress state is made using the criteria based upon principal strains. In the algorithm they apply load step by step and they continue to solve for a particular load step until the convergence is achieved for that load step. In the proposed research, the algorithm that is employed is similar to Miller and Hedgepeth’s algorithm.

Miller et al. [6, 22] investigated the algorithm further. They presented the efficiency and accuracy of the algorithm by applying it to the problems Stein and Hedgepeth [19] studied. They described the algorithm more detailed in these studies. They also described how Miller and Hedgepeth [5] derived the D-taut, D-slack and D-wrinkled constitutive matrices in detail.

2.3 Applications of Instability Analyses to Webs

Shelton [17] studied the steering effects of downstream misaligned roller. He modeled web span as a beam. His work helped us to justify boundary conditions which should be associated with a straight uniform web approaching a misaligned roller when there is sufficient friction to enforce the kinematic boundary conditions.

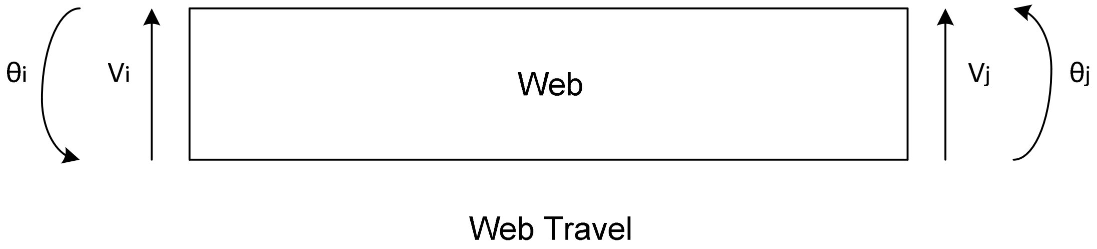


Figure 2.1 Boundary Conditions for Misaligned Roller

He determined that the concept of normal entry that had been used in the drive belt industry can also be used in web guiding systems. He stated that V_i and θ_i can be arbitrary set equal to zero without effecting the relative lateral deformation within the span.

θ_j is the misalignment of the downstream roller at j, and due to the law of normal entry the slope of the web at j will be θ_j. Shelton proved that the final boundary condition was the downstream moment being zero. Thus $M_j(L) = V_j''(L) = 0$. In our problem these boundary conditions will be used while modeling the instability of a web due to a downstream misaligned roller.

Gehlbach, et al. [32] proposed buckling criteria for troughed webs in a free span. They showed experimental verification for downstream misaligned roller case. In their work, isotropic web properties were considered.

Gopal and Kedl [33] were the fist people who used finite element analysis and a commercial FE code to study trough formation in the web span between rollers. They modeled a web span by using triangular plate elements and they used ABAQUS FE commercial code. They were successful to predict deformations of a web due to the misaligned roller.

Benson et al. [8] developed a finite element model for wrinkling analysis and called this code FEWA. They used this code to make calculations in their paper. Their aim was to better understand the conditions which cause wrinkle formation. They worked to predict locations where troughs would form and predict magnitudes of compressive stresses. They compared some of their results with the results from the nonlinear version of commercial code ABAQUS. In their code instead of using Wagner’s tension field theory they used the tension field theory that Wu [24, 25, and 26] introduced first. In this method, it is assumed that out of plane deflection relieves compressive stress across a wrinkle and that there is an associated strain with this deflection.

Roisum [28, 29, 30, and 31] described wrinkling phenomena in detail. He explained wrinkling, air entrainment, tension control, roller design, problems associated with profile variations, why and how wrinkles form, the types of wrinkles, troubleshooting techniques. He explained the importance of the problem due to the material cost and waste for the producers. While doing this he explained the subject in a simple way rather than an academic way.

Shelton [14] worked on buckling of webs. He modeled a web as a buckled plate or a shell. He studied buckled wavelengths of webs in free spans and on rollers. He predicted the wavelength of the buckled form and he compared these results with experimental data.

Good et al. [4] worked on velocity independent and velocity dependent wrinkles. Velocity dependence occurs when velocities are high enough and web tension is low enough to allow sufficient air entrainment between webs and rollers that Shelton's boundary conditions [17] are no longer valid. They found that velocity dependent wrinkles can be avoided by using enough traction and suitable web line conditions.

Hashimoto [34] worked on the studies done by Beisel [2, 10], which is a theoretical model for predicting the web wrinkling due to the misaligned roller. The theoretical model was established upon the experimental results. The experiments which he did were for non uniform webs with different Young's modulus in MD and CMD directions. He compared these experimental results with the model and he verified the accuracy of the model.

Jones and McCann [35] studied wrinkling of webs on rollers and drums. They used a new variational analysis based on the method of Rayleigh-Ritz. They modeled the wrinkling as a continuous sine wave in a web on a roller or a drum. They reported that a shell buckling on a rigid support in an outward mode is similar to wrinkling on rollers.

Beisel [2, 10] studied wrinkling phenomena due to the misaligned roller. In his study, he considered orthotropic material properties. He showed that while wrinkle formation would require post buckling analysis, prediction of trough formation can be expressed by closed form solution.

Papandreadis [7] employed finite element methods to predict troughs in the webs. He studied effects of several parameters on the amount of lateral contraction of the web. He examined the effects of web material properties (Poisson's ratio, modulus of elasticity) and web geometry (various length-to-width ratios), web thickness, loading conditions (tensile loading, combination of tension and shear forces) on the wrinkling phenomenon. He used a finite element code, named NASTRAN (**Na**sa **St**ructural **An**alysis), to analyze the buckling of panels and the resulting shapes (like wavelength of the corrugations, number of waves).

Webb [9] was the first person who tried to couple the behavior of the web in the free span to the web on the roller. Probably, the most important finding in his work was that CMD compressive stresses were forming in the web on the roller due to the troughs that had already formed in the free spans. He used quadrilateral elements within the commercial finite element code COSMOS to predict wrinkles due to the downstream misaligned roller.

He observed that there is a linear relationship between $\theta_{wrinkle}$ and θ_{trough} . He performed experiments to measure the deflection as a result of a misaligned roller. He found that the misalignment required to form a wrinkle was roughly twice that to form a trough over a wide range of parameters. He tested a wide range of web material properties, web thickness, and span length to width ratio (often called span ratio). He tried to find a relationship between the width reduction of the web and web buckling. He modeled the web crossing the roller using regular elastic elements. He used wrinkling membrane elements for the web in free span. In Fig.2.2 Webb's approach to the problem is presented.

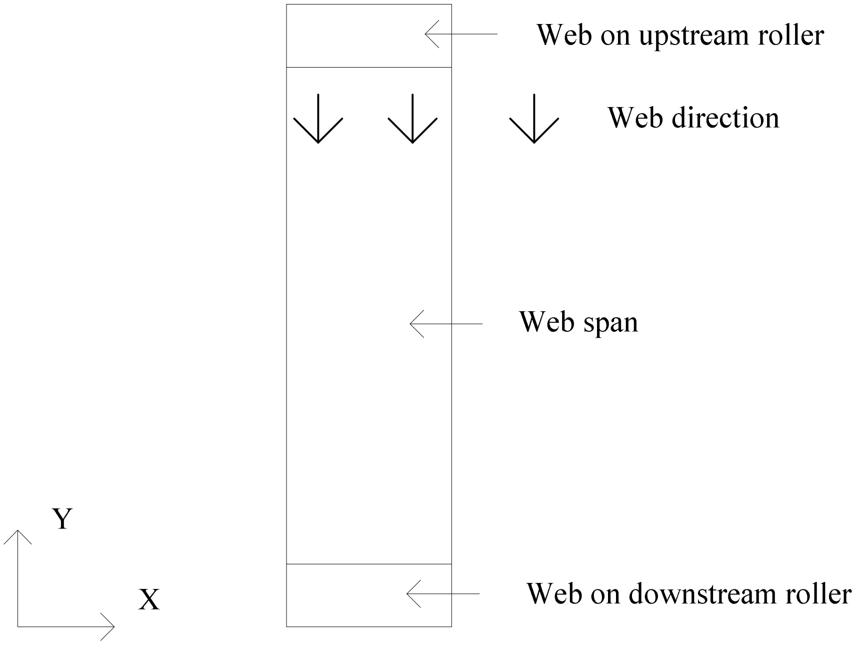


Figure 2.2 Webb's Model

In Fig.2.3 the boundary conditions and the applied loads in Webb's model is shown.

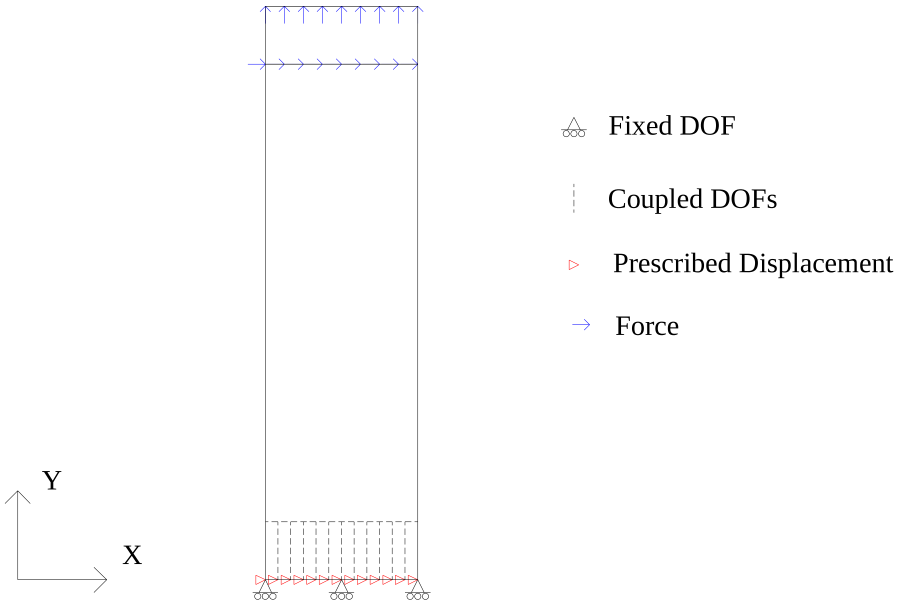


Figure 2.3 Webb’s Model: Boundary Conditions and Applied Loads

Timoshenko [11] showed the axial buckling stress for a sector of a cylinder is the same per unit circumferential length as that of a complete cylinder. He found that σ_{ycr} is

$$\sigma_{ycr} = -\frac{h}{R} \cdot \frac{E}{\sqrt{3 \cdot (1-\nu^2)}} \tag{2.1}$$

The web wrapped around the roller can be assumed to be sector of a cylindrical shell. Shelton [14] studied with the mechanics of buckling and he hypothesized that the web wrapped around the roller is shows similarity to an internally pressured thin cylinder vessel. He found that the tension in the web performs like the hoop stress in a pressurized cylinder and that a pressure results between the web and roller. He discussed that Eq.2.1 may indeed be appropriate to use in modeling a web buckling on a roller.

Good and Beisel [10] developed an instability criterion for orthotropic web in shell form in order to predict wrinkling of webs:

$$\sigma_{ycr} = -\frac{t}{R}\frac{\sqrt{E_x E_y}}{\sqrt{3(1-v_x y_y)}} \tag{2.2}$$

where *R* is the radius of the cylinder, *t* is the thickness of the web, *Ex and Ey are* the elastic modulus in x and y directions respectively, and *v* is Poisson's ratio. Beisel performed many tests of aluminum soda cans which were near perfect in geometry. He found that the buckling stress approached that given by Timoshenko's expression and concluded that the earlier disagreement being due to shell imperfections must have been correct. Thus expressions 2.1 and 2.2 appear to be applicable to sectors of web transiting rollers. If the compressive stresses in the web on the roller reach the σ_{ycr}, the web will buckle.

Webb [9] increased the shear forces used to simulate the misalignment of the roller until the compressive stress induced in the web on the roller reaches the σ_{ycr} value in (Eq.2.2). When the critical compressive CMD stress was reached, he determined the rotation of the downstream roller from the displacements output by the finite element code. The experimental critical misalignment of the roller and the rotation of the web at the entry of the roller computed using finite element model did not match well. He proposed that the experimental critical misalignment of the roller θ_{cr} is the sum of the critical angles predicted by the model for troughing, $\theta_{cr,trough}$ and for wrinkling, $\theta_{cr,wrinkle}$ in verifying his model. This was found later to be incorrect by Beisel [12].

Webb's work was carried forward by Beisel [12]. Beisel studied troughing and wrinkling due to roller misalignment, roller taper, and roller crown. There is some similarity between the effects of roller misalignment and taper on troughs and wrinkles. In both cases the web is steered laterally in the machine and a shear stress results in the web. The misaligned roller induces the misalignment angle and a lateral deformation at the misaligned roller. The tapered roller induces a lateral deformation and a bending moment at the tapered roller. Beisel's method of modeling the problem with COSMOS was different from the Webb's method of modeling the problem. Beisel achieved good agreement with his models and the experimental results. His model and his results will be introduced in Chapter IV and Chapter V.

Swift [39] examined steering of drive belts. He worked with the concept of a couple developed in a web approaching a tapered or crowned roller and the resulting steering of the web. He suggests the minimum amount of taper or crown which should be employed to control the web with minimum interference of stresses in the web. He gave experimental results to support his suggestions for corrective measures.

Shelton [44] discussed misaligned and tapered rollers, and he mentioned multiple web span interactions and moment transfer. In his thesis, Beisel [12] compared his model with Shelton's model for the critical taper required to cause troughs to form in a web span approaching a tapered roller.

Good and Beisel [45] worked on the formation of troughs due to tapered rollers. They attempted to determine whether the procedure that was employed for misaligned rollers would be applicable to the case of downstream tapered rollers. This work was a part of Beisel's thesis [12].

Brown [46] presented a new method for modeling the elastic behavior of webs conveyed over rollers. He worked on lateral displacement of a misaligned roller and lateral displacement of a tapered roller. He suggested two modifications of the web boundary conditions. One was a generalization of the normal entry rule and the other was the addition of what he named the normal strain rule. With a numerical partial differential equation solver, he solved two dimensional plane stress equations and compared the results with earlier models.

Shimizu et al. [47] and Shimizu [48] worked on plates which have holes and are subjected to tensile load. They used the finite element method in their work. They investigated the effects of aspect ratios (height/width) and shapes of holes to the *k* which is the buckling coefficient of the plate described by Timoshenko and Gere [11]. They determined that with that curvatures on corners of holes have little effect in improving the tension buckling strength and that the buckling coefficient increases corresponding to the increasing aspect ratio.

El-Sawy and Martini [49] studied the effects of plate aspect ratio (height/width), hole size, hole location and loading ratio on the buckling coefficient k of rectangular perforated plates subjected to uniform end compression in the x direction and compression or tension in the y direction.

Mallya [15] was the first person to examine the effects of holes in process machinery in web handling industry. He applied Beisel's method of modeling to web wrinkles due to circular and elliptical discontinuities in the web. He studied the behavior of webs with voids traveling over a roller. He compared experimental results and finite element model results that he modeled using commercial FE code COSMOS. He studied elliptical voids and circular holes in terms of generating wrinkles. His FE model was similar in form to Beisel's FE model with respect to boundary conditions and using five panels. His model and results will be introduced in Chapter VI.

Kara [16] also used similar modeling method to predict the occurrence of wrinkles due to length variation across the web width. He attempted to find the critical conditions that would induce wrinkles. He heated the center of the web during his experiments to achieve length variation. He also used COSMOS FE commercial code to model his case, and used experimental findings to confirm his FE model.

Relevant books include, ‘Theory of Elastic Stability’ written by Timoshenko and Gere [11], ‘Introduction to Finite Elements in Engineering’ written by Chandrupatla and Belegundu [36], ‘Finite Elements and Approximation’ written by Zeinkiewicz and Morgan [37] and Visual basic Excel for Dummies [27]. Also ‘Finite Element Analysis Class Notes’ from Good [38] is used for finite element part of the study.

2.4 Summary

The tension field theory can be applied to webs in web lines because the web can support tension but cannot carry compression. Wagner [18] proposed tension field theory. His theory was further developed by Raissner [23], Stein and Hedgepeth [19] and Mikolas [21, 22]. Miller and Hedgepeth [5] developed a new algorithm for finite element analysis based on this field equation. This algorithm is the most common algorithm used to examine wrinkles.

Lorig [3] developed the normal entry rule for a web approaching a roller. Shelton [17] found that the moment in a web is zero when it approaches a misaligned roller. In the web handling area, Gopal and Kedl [33] were the first to employ a commercial finite element code to study trough formation in the web span between the rollers.

Webb [9] attempted to model wrinkle formation due to a misaligned roller. He used partly wrinkling membrane elements while modeling the web span. He used the commercial finite element code called COSMOS while modeling his work, and shell buckling criteria to determine whether wrinkling occurred.

Beisel [12] made the most recent attempts to model wrinkle formation due to the misaligned roller and the tapered roller. He modeled the web between the rollers by wrinkling membrane elements and linked it to the classical shell buckling criteria as Webb [9] did. He studied webs approaching misaligned rollers and tapered rollers. He used the commercial finite element code COSMOS to model these cases. He compared his model results with his experimental results .These results showed good agreement.

Mallya [15] was the first person who investigated the effects of voids on the stability of webs. He performed experiments and modeled circular and elliptical discontinuities in webs in web lines. He also used the commercial finite element code COSMOS in his models and his model results and test results matched well.

2.5 Statement of Proposed Research

In the literature no citations were found for the wrinkling instability of moving webs that did not involve the use of commercial finite element codes for solution. Use of commercial finite element codes by novice users to solve nonlinear problems associated with web instability is difficult.

The objective of the proposed research will be to develop user friendly finite element codes that will solve nonlinear instability problems associated with strain state dependent material properties and boundary conditions of moving webs. This code will be unique and will have economic value by helping minimize web material losses as described in the introduction.

3. CHAPTER III

3.1 FINITE ELEMENT EQUATIONS

In this chapter, the finite element equations will be described in detail. Displacement, strain and stress equations, the element stiffness matrix, meshing, banded matrix and their relations will be studied. A discussion of the solution method for cases in which the elasticity matrix [D] is not constant will be given.

3.2 Two Dimensional Four Node Quadrilateral Elements

In this study, two dimensional *four node quadrilateral elements* are used. In this section the equations and properties of four node quadrilateral elements will be given briefly.

The stiffness matrix for quadrilateral elements can be found from the strain energy term in a continuum.

$$U = \int_V \frac{1}{2} \sigma^T \varepsilon dV \tag{3.1}$$

In finite elements we consider the total strain energy to be sum of the strain energies from each element. Eq.3.2 can be obtained by replacing dV by $t\,dA$ in Eq.3.1, where t is the uniform thickness of an element.

$$U = \sum_e t_e \int_e \frac{1}{2} \sigma^T \varepsilon dA \tag{3.2}$$

The small strain displacement relations for two dimensional problems can be written as:

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} \tag{3.3}$$

Where *u* and *v* are the deformations in the *x* and *y* directions respectively. In two-dimensional fields, the displacement components at each point in the domain of the finite element can be represented as functions of two coordinate directions.

$$\mathbf{u} = [u(x, y), v(x, y)]^T \tag{3.4}$$

For the general quadrilateral element shown in Fig.3.1, the nodal displacement vector is

q = $[q_1, q_2, q_3, ..., q_8]^T$

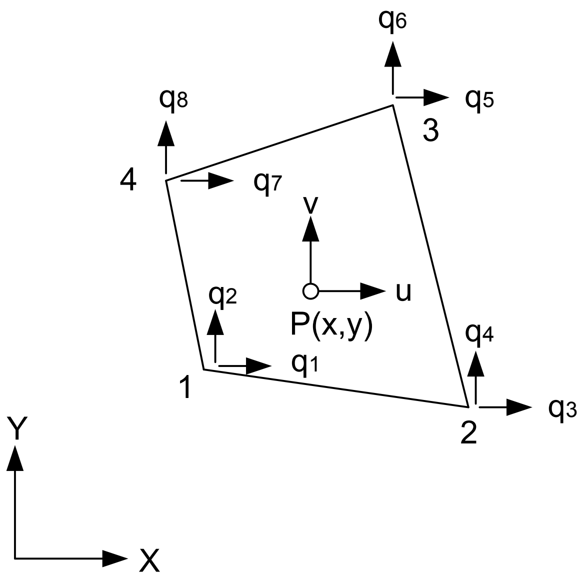


Figure 3.1 Four Node Quadrilateral Element

The finite element method uses concept of shape functions to develop interpolations systematically.

According to the concept, the shape functions must be developed for the master element. The master element is defined in natural coordinates (ξ, η) and has a square shape in the natural coordinate system (Fig.3.2). The Lagrange shape functions are N_1, N_2, N_3 *and* N_4. Shape functions take the value of unity at the node where they are defined and the value zero at the other nodes. For example N_1 takes the value one of unity at node one and takes the value zero at the other nodes.

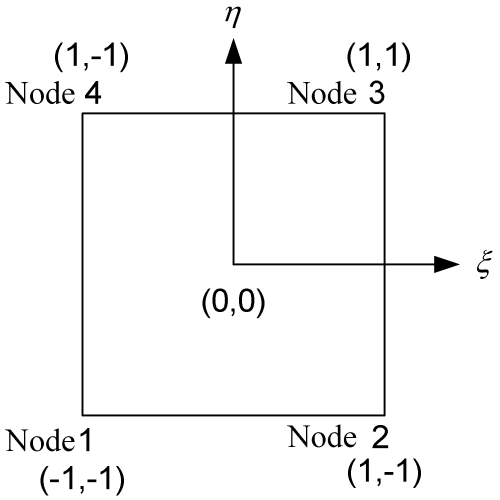


Figure 3. 2 Quadrilateral Elements In ξ,η Space (Master Element)

At the edges $\xi=+1\,and\;\eta=+1$ N_1 is equal to zero. So, N_1 must be a function like,

$$N_1=c(1-\xi)\,(1-\eta) \tag{3.5}$$

c is a constant that can be determined easily. Since N_1 is equal to one at node one, where $\xi=-1\,and\;\eta=-1$.If we put these values at Eq.3.5,

$$1=c(1-(-1))\,(1-(-1)) \tag{3.6}$$

yields $c=1/4$.Finally N_1 can be written as,

$$N_1=\frac{1}{4}(1-\xi)\,(1-\eta) \tag{3.7}$$

By using the same procedure N_1, N_2, N_3 *and* N_4 can be written as,

$$\begin{aligned}
N_1 &= \frac{1}{4}(1-\xi)\,(1-\eta) \\
N_2 &= \frac{1}{4}(1+\xi)\,(1-\eta) \\
N_3 &= \frac{1}{4}(1+\xi)\,(1+\eta) \\
N_4 &= \frac{1}{4}(1-\xi)\,(1+\eta)
\end{aligned} \tag{3.8}$$

Now, the shape functions can be used to interpolate the displacement at any point within the domain of the element using the equations:

$$\begin{aligned}
u &= N_1q_1 + N_2q_3 + N_3q_5 + N_4q_7 \\
v &= N_1q_2 + N_2q_4 + N_3q_6 + N_4q_8
\end{aligned} \tag{3.9}$$

if N is,

$$\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \tag{3.10}$$

then the displacement can be written as:

$$[\mathrm{u}] = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} = \left[N\right]\left\{q\right\} \tag{3.11}$$

with the help of shape functions a point in the element can be described as,

$$\begin{aligned} x &= N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 \\ y &= N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 \end{aligned} \tag{3.12}$$

The shape functions can also be used to generate a map between the cartesian coordinates (*x*,*y*) and the natural coordinates (ξ,η). Since the same shape functions have been used to interpolate deformation within an element and to generate the coordinate map equations this is called an isoparametric formulation. The strain relationships (3.3) require derivatives with respect to cartesian coordinates. We currently have the deformations *u and v* defined as shape functions, which are functions of the natural coordinates ξ *and* η, multiplied by nodal deformations that are constants. So, to determine the strains in cartesian coordinates we must first relate the derivatives of deformations in natural coordinates (ξ,η) to derivatives in cartesian coordinates(*x*,*y*). If a displacement function in x, y coordinates is $u = u(x, y)$ then this function can be considered to be an implicit function of ξ *and* η as $u = u[x(\xi,\eta), y(\xi,\eta)]$. Differentiation due to the chain rule,

$$\begin{aligned} \frac{\partial u}{\partial \xi} &= \frac{\partial u}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \xi} \\ \frac{\partial u}{\partial \eta} &= \frac{\partial u}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \eta} \end{aligned} \tag{3.13}$$

If we define the Jacobian matrix as,

$$\mathrm{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \tag{3.14}$$

We can rewrite Eq.3.13 as,

$$\begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} = \mathrm{J} \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} \tag{3.15}$$

It seems that the Jacobian can transform derivatives in cartesian coordinates to derivatives with respect to natural coordinates. By using Eq.3.8, Eq.3.12 can be written as,

$$\begin{aligned} x &= \frac{1}{4}(1-\xi)\,(1-\eta)x_1 + \frac{1}{4}(1+\xi)\,(1-\eta)x_2 + \frac{1}{4}(1+\xi)\,(1+\eta)x_3 + \frac{1}{4}(1-\xi)\,(1+\eta)x_4 \\ y &= \frac{1}{4}(1-\xi)\,(1-\eta)y_1 + \frac{1}{4}(1+\xi)\,(1-\eta)y_2 + \frac{1}{4}(1+\xi)\,(1+\eta)y_3 + \frac{1}{4}(1-\xi)\,(1+\eta)y_4 \end{aligned} \tag{3.16}$$

With the help of Eq.3.16 Jacobian term Eq.3.14 can be written as,

$$\mathrm{J} = \frac{1}{4}\begin{bmatrix} -(1-\eta)x_1 + (1-\eta)x_2 + (1+\eta)x_3 - (1+\eta)x_4 & -(1-\eta)y_1 + (1-\eta)y_2 + (1+\eta)y_3 - (1+\eta)y_4 \\ -(1-\xi)x_1 - (1+\xi)x_2 + (1+\xi)x_3 - (1-\xi)x_4 & -(1-\xi)y_1 - (1+\xi)y_2 + (1+\xi)y_3 + (1-\xi)y_4 \end{bmatrix} \tag{3.17}$$

this equation can be written as,

$$\mathrm{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{3.18}$$

Now the Jacobian can be inverted and rewrite Eq.3.15 to produce derivatives that are related to strains:

$$\begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} = \mathrm{J}^{-1} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} \tag{3.19}$$

by using Eq.3.17, Eq.3.19 can be written as

$$\begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathrm{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \end{Bmatrix} \tag{3.20}$$

by following the same procedure *v* displacements can be obtained as,

$$\begin{Bmatrix} \dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial y} \end{Bmatrix} = \frac{1}{\det \mathrm{J}} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} \tag{3.21}$$

The equation $\partial x \, \partial y = \det J \, \partial \xi \, \partial \eta$ has a proof in reference [36].

By using Eq.3.3, Eq.3.20, Eq.3.21 and defining A matrice as,

$$A = \frac{1}{\det \mathrm{J}} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{22} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \tag{3.22}$$

The strain displacement relations can be written as

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} = \frac{1}{\det \mathrm{J}} \begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{22} & J_{11} & J_{22} & -J_{12} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} = A \begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} \tag{3.23}$$

with the help of 3.17 G matrice can be defined as,

$$G=\frac{1}{4}\begin{bmatrix} -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) & 0 \\ -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) & 0 \\ 0 & -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) \\ 0 & -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) \end{bmatrix} \qquad (3.24)$$

and using the displacement vector Eq.3.9 the derivatives of *u and v* can be written in the natural coordinates as a function of the nodal deformations,

$$\begin{Bmatrix} \dfrac{\partial u}{\partial \xi} \\ \dfrac{\partial u}{\partial \eta} \\ \dfrac{\partial v}{\partial \xi} \\ \dfrac{\partial v}{\partial \eta} \end{Bmatrix} = \frac{1}{4}\begin{bmatrix} -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) & 0 \\ -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) & 0 \\ 0 & -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) \\ 0 & -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) \end{bmatrix} q = G\,q \qquad (3.25)$$

If B is defined as **B=AG,** by using Eq.3.23 and Eq.3.25 strain term can be written as,

$$\varepsilon = \frac{1}{4\det \mathrm{J}}\begin{bmatrix} J_{22} & -J_{12} & 0 & 0 \\ 0 & 0 & -J_{21} & J_{11} \\ -J_{22} & J_{11} & J_{22} & -J_{12} \end{bmatrix}\begin{bmatrix} -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) & 0 \\ -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) & 0 \\ 0 & -(1-\eta) & 0 & (1-\eta) & 0 & (1+\eta) & 0 & -(1+\eta) \\ 0 & -(1-\xi) & 0 & -(1+\xi) & 0 & (1+\xi) & 0 & (1-\xi) \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix}$$

$$= B\,q \qquad (3.26)$$

two dimensional constitutive relations will be used to relate stress to strain $\sigma = D\varepsilon$ and now,

$$\sigma = D\,B\,q \qquad (3.27)$$

By using Eq.3.26 and Eq.3.27 the strain energy term$U = \sum_e t_e \int_e \frac{1}{2} \sigma^T \varepsilon dA$, becomes,

$$\begin{aligned} U &= \sum_e \frac{1}{2} q^T \; [t_e \int_{-1}^{1} \int_{-1}^{1} B^T DB \det J \; d\xi d\eta] \, q \\ &= \sum_e \frac{1}{2} q^T k^e q \end{aligned} \tag{3.28}$$

where k is 8x8 element stiffness matrix:

$$k^e = t_e \int_{-1}^{1} \int_{-1}^{1} B^T DB \det J \; d\xi d\eta \tag{3.29}$$

These integrals can be evaluated by using numerical integration methods. The Gaussian approach will be considered for this purpose.

3.3 Numerical Integration by Gaussian Approach

By integrating in natural coordinates the bounds of integration are much simplified. In cartesian coordinates the *y*-bounds will be functions of *x* and the *x*-bounds will be functions of *y*. In natural coordinates our bounds are from -1 to1 for both ξ *and* η . Series can be used to take the integrals (Eq.3.29).By using the Gaussian quadrature approach; integration can be evaluated using weights and points. These points are also called *Gauss points*.

$$I = \int_{-1}^{1} f(\xi) \, d\xi \;\approx\; \omega_1 f(\xi_1) + \omega_2 f(\xi_2) + \cdots + \omega_n f(\xi_n) \tag{3.30}$$

Eq.3.30 is an example of *n*- point approximation. Here $\omega_1, \omega_2 \ldots \omega_n$ are weights and $\xi_1, \xi_2 \ldots \xi_n$ are the Gauss points. If a one point formula is employed, the integral becomes:

$$I = \int_{-1}^{1} f(\xi)\, d\xi \;\approx\; \omega_1 f(\xi_1) \tag{3.31}$$

If a two point formula is employed, the integral becomes:

$$I = \int_{-1}^{1} f(\xi)\, d\xi \;\approx\; \omega_1 f(\xi_1) + \omega_2 f(\xi_2) \tag{3.32}$$

The finite element method naturally incorporates some error as a numerical approximation. The complex continuums were modeled with many finite elements with simple shape functions to represent the element deformations (Eq.3.8). Hence, it would be undesirable to incorporate additional error in the integration of stiffness terms. It is desirable that the integration be exact. In a one point formula two parameters are considered $(\omega_1 \; and \; \xi_1)$. Suppose that our integration is required be exact when $f(\xi)$ a polynomial of is order one. So, suppose $f(\xi)$ is a function $f(\xi) = a_0 + a_1 \xi$. If we select $\omega_1 = 2 \, and \; \xi_1 = 0$ Gaussian quadrature will yield an exact result.

In a two point formula there are four parameters to choose $(\omega_1, \omega_2, \xi_1 \; and \; \xi_2)$.Suppose that $f(\xi)$ must be exact for a cubic polynomial, $f(\xi) = a_0 + a_1 \xi + a_2 \xi^2 + a_3 \xi^3$. The error term will be,

$$Error = \int_{-1}^{1} (a_0 + a_1 \xi + a_2 \xi^2 + a_3 \xi^3)\, d\xi \; - [\, \omega_1 f(\xi_1) + \omega_2 f(\xi_2)] = 0 \tag{3.33}$$

Solution yields four nonlinear equations and they have the unique solution,

$$\begin{aligned} &\omega_1 = \omega_2 = 1 \\ &\xi_1 = -\xi_2 = -1/\sqrt{3} \end{aligned} \tag{3.34}$$

As a result by using two Gauss points and by using the values in Eq.3.34 a cubic expression or less can be integrated exactly. By increasing the number of Gauss points, different weights (ω) and different locations (ξ) can be found. In the FE algorithm that is developed in this study two Gauss points are used.

Two Dimensional Integrals

The equation 3.29 for our stiffness terms involves two-dimensional integrals. So we need to extend Gaussian quadrature to the two dimensional integral form:

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\, d\xi d\eta \tag{3.35}$$

If *I* is in a form like Eq.3.35, *I* can be written as,

$$\begin{aligned} I &\approx \int_{-1}^{1} [\sum_{i=1}^{n} w_i f(\xi_i,\eta)]\, d\eta \\ &\approx \sum_{j=1}^{n} w_j [\sum_{i=1}^{n} w_i f(\xi_i,\eta_j)] \\ &\approx \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j\; f(\xi_i,\eta_j) \end{aligned} \tag{3.36}$$

Stiffness matrix (Eq.3.29) is two-dimensional integral. The product of $B^T DB \det J$ is quadratic in terms of ξ *and* η .So two point Gauss Quadrature yield an exact result. It is 8x8 matrix so it has 64 elements. Each term must be calculated by using Eq.3.36. If $f(\xi,\eta) = t_e\,(B^T DB \det J)_{ij}$ putting two for n in Eq.3.36 yields,

$$k_{ij} \approx w_1^{\,2} f(\xi_1,\eta_1) + w_1 w_2 f(\xi_1,\eta_2) + w_2 w_1 f(\xi_2,\eta_1) + w_2^{\,2} f(\xi_2,\eta_2) \tag{3.37}$$

where

$$\begin{gathered} w_1 = w_2 = 1 \\ \xi_1 = \eta_1 = -\frac{1}{\sqrt{3}} \quad and \quad \xi_2 = \eta_2 = \frac{1}{\sqrt{3}} \end{gathered} \tag{3.38}$$

After input of these weights and Gauss points into the Eq.3.37 *kij* can be found as,

$$k_{ij} \approx f(-\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}}) + f(-\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}) \tag{3.39}$$

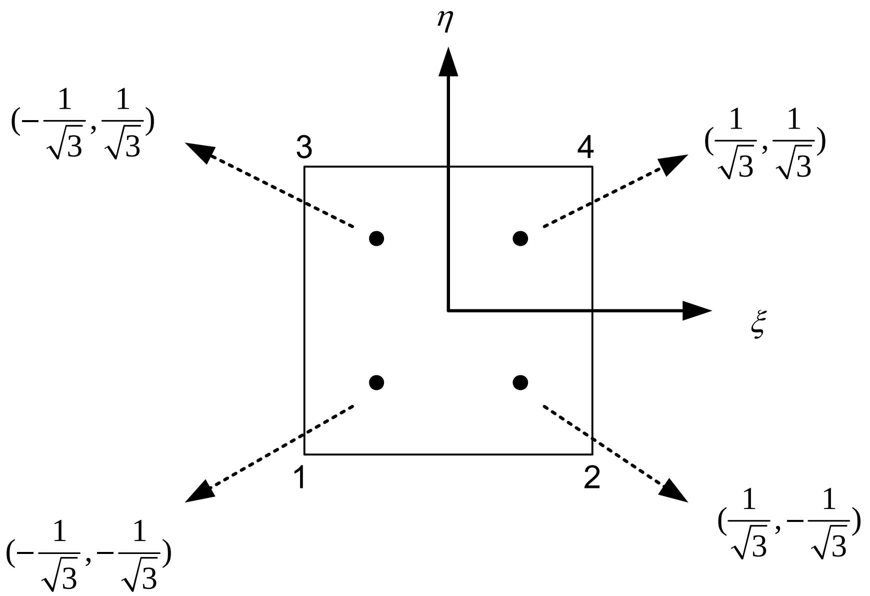

η
$(-\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}})$
3
4
$(\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}})$
ξ
1
2
$(\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}})$
$(-\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}})$

Figure 3.3 Two Points Gauss Quadrature using 2x2 rule.

3.4 The Global Stiffness Matrix and Matrix Banding

The element stiffness matrix is an 8x8 matrix and it has 64 elements. After calculating all stiffness matrices for all elements, the *global stiffness* matrix need to be formed from the element stiffness matrices. In Fig. 3.4 the plate is divided into *n* elements, this procedure is called *meshing*. For every element each of the stiffness terms $f(\xi,\eta) = t_e (B^T DB \det J)_{ij}$ is evaluated using Eq.3.29. While calculating the terms of global stiffness matrix, if a node is only used by one element, its stiffness terms should be placed in the global stiffness matrix directly (like Fig.3.4 node 1). If a node is used by two elements (like Fig.3.4 nodes 3,5..) , stiffness terms for this node from two elements must be added to form the stiffness terms of the global stiffness matrix for that node. Lastly if a node is used by four elements (like Fig.3.4 nodes 4,6…) stiffness terms for that node from four elements must be added to form the global stiffness matrix.

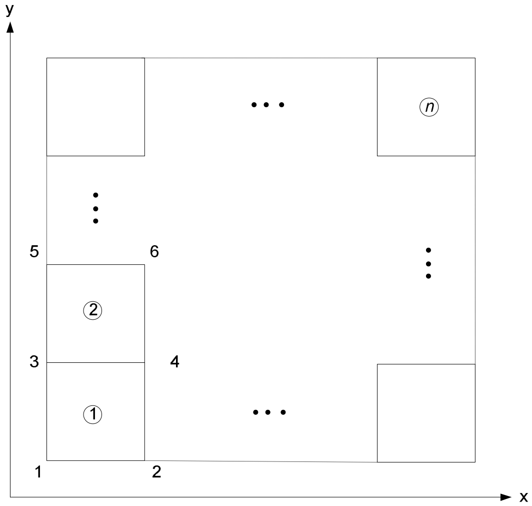


Figure 3.4 A Simple Model for Global Stiffness Matrix

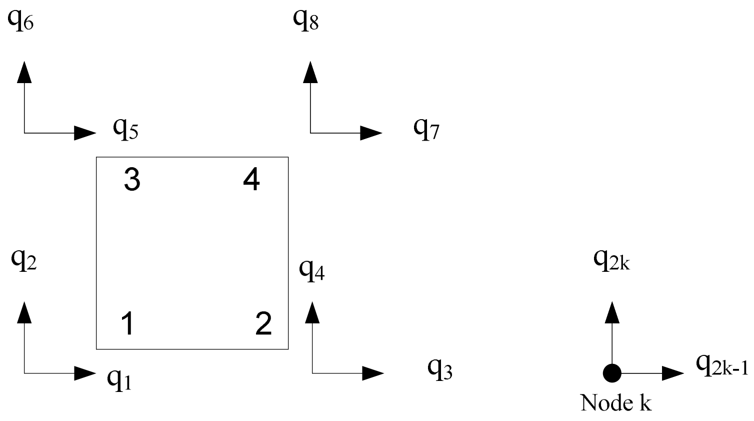


Figure 3.5 A Quadrilateral Element and Nodal Displacements

In Eq.3.40 and Eq.3.41 the stiffness matrices for element one and element two displayed in Fig.3.4 are shown

$$\begin{bmatrix} k^1{}_{11} & k^1{}_{12} & k^1{}_{13} & k^1{}_{14} & k^1{}_{15} & k^1{}_{16} & k^1{}_{17} & k^1{}_{18} \\ & k^1{}_{22} & k^1{}_{23} & k^1{}_{24} & k^1{}_{25} & k^1{}_{26} & k^1{}_{27} & k^1{}_{28} \\ & & k^1{}_{33} & k^1{}_{34} & k^1{}_{35} & k^1{}_{36} & k^1{}_{37} & k^1{}_{38} \\ & & & k^1{}_{44} & k^1{}_{45} & k^1{}_{46} & k^1{}_{47} & k^1{}_{48} \\ & & & & k^1{}_{55} & k^1{}_{56} & k^1{}_{57} & k^1{}_{58} \\ & sym. & & & & k^1{}_{66} & k^1{}_{67} & k^1{}_{68} \\ & & & & & & k^1{}_{77} & k^1{}_{78} \\ & & & & & & & k^1{}_{88} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \\ q_8 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \end{bmatrix} \qquad (3.40)$$

$$\begin{bmatrix} k^2_{11} & k^2_{12} & k^2_{13} & k^2_{14} & k^2_{15} & k^2_{16} & k^2_{17} & k^2_{18} \\ & k^2_{22} & k^2_{23} & k^2_{24} & k^2_{25} & k^2_{26} & k^2_{27} & k^2_{28} \\ & & k^2_{33} & k^2_{34} & k^2_{35} & k^2_{36} & k^2_{37} & k^2_{38} \\ & & & k^2_{44} & k^2_{45} & k^2_{46} & k^2_{47} & k^2_{48} \\ & & & & k^2_{55} & k^2_{56} & k^2_{57} & k^2_{58} \\ & sym. & & & & k^2_{66} & k^2_{67} & k^2_{68} \\ & & & & & & k^2_{77} & k^2_{78} \\ & & & & & & & k^2_{88} \end{bmatrix} \begin{bmatrix} q_5 \\ q_6 \\ q_7 \\ q_8 \\ q_9 \\ q_{10} \\ q_{11} \\ q_{12} \end{bmatrix} = \begin{bmatrix} F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \\ F_{10} \\ F_{11} \\ F_{12} \end{bmatrix} \qquad (3.41)$$

After global stiffness matrix is formed, all displacements are calculated by using *Gaussian elimination method*. After finding displacements, strains and stresses for all elements can be calculated. In the algorithm that is developed the user decides the mesh density. 1000 or more elements may be needed for long spans simulations. The resulting stiffness matrix will be large and much computational time will be required for solutions. Computational times can be large because a multistep solution will be required where loads are slowly increased and the [*D*] matrices updated after each load step. Thus the size of the stiffness matrix becomes important because the system of updated equations will be solved many times. The stiffness matrix in our problem is a *symmetric matrix*. Instead of using the whole stiffness matrix the *banded* form of the stiffness matrix can be employed and reduce the computation time.

To explain the form of the banded matrix, assume that the plate in Fig.3.4 is meshed with only two elements. In Eq.3.30 and 3.31 the stiffness matrixes of two elements were given. So, the global stiffness matrix for the plate will be:

$$\begin{bmatrix} k^{1}_{11} & k^{1}_{12} & k^{1}_{13} & k^{1}_{14} & k^{1}_{15} & k^{1}_{16} & k^{1}_{17} & k^{1}_{18} & 0 & 0 & 0 & 0 \\ & k^{1}_{22} & k^{1}_{23} & k^{1}_{24} & k^{1}_{25} & k^{1}_{26} & k^{1}_{27} & k^{1}_{28} & 0 & 0 & 0 & 0 \\ & & k^{1}_{33} & k^{1}_{34} & k^{1}_{35} & k^{1}_{36} & k^{1}_{37} & k^{1}_{38} & 0 & 0 & 0 & 0 \\ & & & k^{1}_{44} & k^{1}_{45} & k^{1}_{46} & k^{1}_{47} & k^{1}_{48} & 0 & 0 & 0 & 0 \\ & & & & k^{1}_{55}+k^{2}_{11} & k^{1}_{56}+k^{2}_{12} & k^{1}_{57}+k^{2}_{13} & k^{1}_{58}+k^{2}_{14} & k^{2}_{15} & k^{2}_{16} & k^{2}_{17} & k^{2}_{18} \\ & & & & & k^{1}_{66}+k^{2}_{22} & k^{1}_{67}+k^{2}_{23} & k^{1}_{68}+k^{2}_{24} & k^{2}_{25} & k^{2}_{26} & k^{2}_{27} & k^{2}_{28} \\ sym. & & & & & & k^{1}_{77}+k^{2}_{33} & k^{1}_{78}+k^{2}_{34} & k^{2}_{35} & k^{2}_{36} & k^{2}_{37} & k^{2}_{38} \\ & & & & & & & k^{1}_{88}+k^{2}_{44} & k^{2}_{45} & k^{2}_{46} & k^{2}_{47} & k^{2}_{48} \\ & & & & & & & & k^{2}_{55} & k^{2}_{56} & k^{2}_{57} & k^{2}_{58} \\ & & & & & & & & & k^{2}_{66} & k^{2}_{67} & k^{2}_{68} \\ & & & & & & & & & & k^{2}_{77} & k^{2}_{78} \\ & & & & & & & & & & & k^{2}_{88} \end{bmatrix} \begin{bmatrix} q1 \\ q2 \\ q3 \\ q4 \\ q5 \\ q6 \\ q7 \\ q8 \\ q9 \\ q10 \\ q11 \\ q12 \end{bmatrix} = \begin{bmatrix} F1 \\ F2 \\ F3 \\ F4 \\ F5 \\ F6 \\ F7 \\ F8 \\ F9 \\ F10 \\ F11 \\ F12 \end{bmatrix} \qquad (3.42)$$

The banded form of 3.42 is shown in 3.43. The bandwidth of the matrix has been reduced to 8.

$$\begin{bmatrix} k^{1}_{11} & k^{1}_{12} & k^{1}_{13} & k^{1}_{14} & k^{1}_{15} & k^{1}_{16} & k^{1}_{17} & k^{1}_{18} \\ k^{1}_{22} & k^{1}_{23} & k^{1}_{24} & k^{1}_{25} & k^{1}_{26} & k^{1}_{27} & k^{1}_{28} & 0 \\ k^{1}_{33} & k^{1}_{34} & k^{1}_{35} & k^{1}_{36} & k^{1}_{37} & k^{1}_{38} & 0 & 0 \\ k^{1}_{44} & k^{1}_{45} & k^{1}_{46} & k^{1}_{47} & k^{1}_{48} & 0 & 0 & 0 \\ k^{1}_{55}+k^{2}_{11} & k^{1}_{56}+k^{2}_{12} & k^{1}_{57}+k^{2}_{13} & k^{1}_{58}+k^{2}_{14} & k^{2}_{15} & k^{2}_{16} & k^{2}_{17} & k^{2}_{18} \\ k^{1}_{66}+k^{2}_{22} & k^{1}_{67}+k^{2}_{23} & k^{1}_{68}+k^{2}_{24} & k^{2}_{25} & k^{2}_{26} & k^{2}_{27} & k^{2}_{28} & 0 \\ k^{1}_{77}+k^{2}_{33} & k^{1}_{78}+k^{2}_{34} & k^{2}_{35} & k^{2}_{36} & k^{2}_{37} & k^{2}_{38} & 0 & 0 \\ k^{1}_{88}+k^{2}_{44} & k^{2}_{45} & k^{2}_{46} & k^{2}_{47} & k^{2}_{48} & 0 & 0 & 0 \\ k^{2}_{55} & k^{2}_{56} & k^{2}_{57} & k^{2}_{58} & 0 & 0 & 0 & 0 \\ k^{2}_{66} & k^{2}_{67} & k^{2}_{68} & 0 & 0 & 0 & 0 & 0 \\ k^{2}_{77} & k^{2}_{78} & 0 & 0 & 0 & 0 & 0 & 0 \\ k^{2}_{88} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.43)$$

Banded matrix equation solvers exist which helps to greatly reduce the computational time associated with solving set the set of equations during each load step.

To conclude, the element stiffness terms are calculated from Eq.3.29 where $f(\xi,\eta) = t_e\,(B^T DB \det J)_{ij}$. After developing element stiffness matrices, the global stiffness matrix is formed. After the total potential energy is formed, which is the sum of the strain energy terms from Eq.3.28 and the work potential of internal and external forces, the theory of minimum total potential can be used. The resulting systems of equations that must be solved have the form:

$$K\,q = F \tag{3.44}$$

This set of equations is arranged in the banded form discussed and then solved for the nodal deformations {q}. After finding displacements strains can be calculated from,

$$\varepsilon = B\,q \tag{3.45}$$

The stress and elastic strain components are related by a set of coefficients known as Generalized Hooke's Law. This law can be written as,

$$\sigma = D\,\varepsilon = D\;B\;q \tag{3.46}$$

Where *D* is equivalent elasticity matrix.

3.5 A [D] Matrix for Plane Stress-Isotropic Material and Partly Wrinkled Membrane Elements

The *D* matrix is used while forming element stiffness matrixes and finding element stresses. The algorithm that is developed is nonlinear since the membrane *D* matrix is dependent on principal strains. In our models, elastic elements are used for the web on the rollers and for the exiting free span. Partly wrinkled membrane elements that were developed by Miller and Hedgepeth [5] are used for the web in test free span. The *D* matrix can be explicitly stated for any material. For the plane stress state for an isotropic material, stress strain relations can be written as:

$$\begin{aligned} \varepsilon_x &= \frac{1}{E}(\sigma_x - v\sigma_y) \\ \varepsilon_y &= \frac{1}{E}(\sigma_y - v\sigma_x) \\ \gamma_{xy} &= \frac{2}{E}(1+v)\tau_{xy} \end{aligned} \tag{3.47}$$

By solving for the stresses in Eq.3.47 the *D* matrix can be found:

$$D = \frac{E}{1-v^2}\begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{(1-v)}{2} \end{bmatrix} \tag{3.48}$$

In taut membrane behavior, both principal stresses are greater than zero and the web cannot trough or buckles out of plane. In taut behavior, the two in-plane principal strains may both be greater than zero or ε_1 can be greater than zero and ε_2 less than zero as long as the ratio

($\varepsilon_2 / \varepsilon_1$) is greater than the negative value of Poisson's ratio $(-v)$. The [D] matrix given in Eq.3.48 is used to relate stress to strain in (1) membrane elements exhibiting taut behavior and (2) in linear elastic elements, used to model the web upon rollers. For the linear elastic elements there are no conditions placed upon the use of this [D] matrix, the principal stresses and strains can take on positive and negative values.

When unsupported membrane elements have a first principal strain ε_1 less than zero this infers that the second principal strain is also negative and less than ε_1 from the rules we use to order principal strains. In this case the membrane elements exhibit a slack or unstressed behavior between stress and strain and the [D] matrix is null.

In wrinkled membrane behavior the first principal stress σ_1 is greater than zero and the second principal stress σ_2 is zero. In terms of strain this behavior is represented with a first principal strain ε_1 greater than zero but now the second principal strain ε_2 is always negative. Not only ε_2 is negative, but the ratio of ($\varepsilon_2 / \varepsilon_1 < -v$). For a simple membrane in uniform tension with no lateral constraint we would expect a lateral contraction governed by the expression $\varepsilon_2 = -v\varepsilon_1$. We would expect this lateral contraction occur while the web remained planar, no buckling would be expected. However if ε_2 becomes more negative than $-v\varepsilon_1$ we would expect a compressive σ_2 stress to have developed but since membranes by definition can withstand no compressive σ_2 stress without buckling, we would assume this element has entered the wrinkled state. For the wrinkled membrane positive nonzero principal stress can be supposed to act along the wrinkle. In a wrinkled membrane element in longitudinal tension, if σ_1 is the nonzero positive principal stress, the longitudinal direction will be observed to be parallel to the wrinkles. By the property of Mohr's circle it is known that

$$\begin{aligned} R &= \frac{\varepsilon_1 - \varepsilon_2}{2} \\ O &= \frac{\varepsilon_x + \varepsilon_y}{2} \end{aligned} \tag{3.49}$$

where R is the radius of Mohr's circle for strains and O is the distance from the origin of the coordinate system to the center of the circle. Consider a thin flat membrane in a state of plane stress in an x-y coordinate system. Let the principal axes I and II be rotated through an angle α relative to the x-y axes (Figure 3.6).

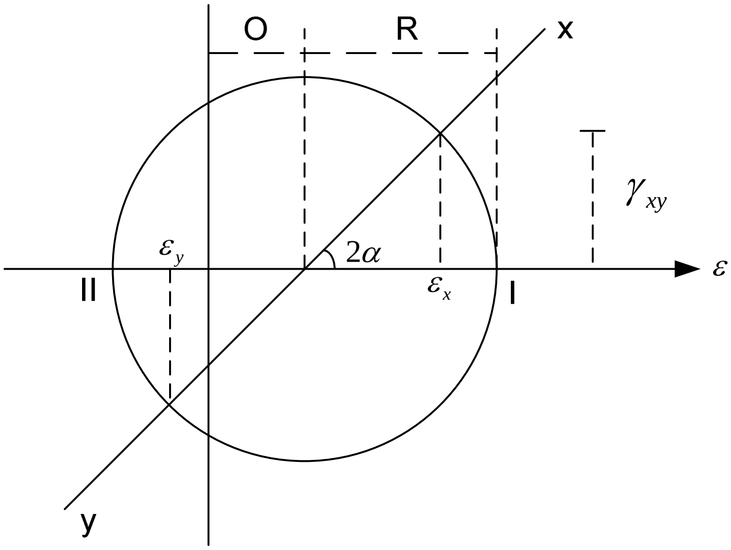


Figure 3.6 Mohr's Circle for Plane Strain

From the Fig.3.6

$$\begin{aligned}\varepsilon_x &= R\cos 2\alpha + \frac{\varepsilon_1+\varepsilon_2}{2} \\ \varepsilon_y &= \frac{\varepsilon_1+\varepsilon_2}{2} - R\cos 2\alpha\end{aligned} \tag{3.50}$$

Substituting the value of R (Eq.3.49) into Eq.3.50 yields

$$\begin{aligned}\varepsilon_x &= (\frac{\varepsilon_1-\varepsilon_2}{2})\cos 2\alpha + \frac{\varepsilon_1+\varepsilon_2}{2} \\ \varepsilon_y &= \frac{\varepsilon_1+\varepsilon_2}{2} - (\frac{\varepsilon_1-\varepsilon_2}{2})\cos 2\alpha\end{aligned} \tag{3.51}$$

Eq. 3.51 yields

$$\begin{aligned}\varepsilon_x &= \frac{\varepsilon_1}{2}(1+\cos 2\alpha) + \frac{\varepsilon_2}{2}(1-\cos 2\alpha) \\ \varepsilon_y &= \frac{\varepsilon_1}{2}(1-\cos 2\alpha) + \frac{\varepsilon_2}{2}(1+\cos 2\alpha)\end{aligned} \tag{3.52}$$

If $P = \cos 2\alpha$, then Eq.3.52 becomes

$$\begin{aligned}\varepsilon_x &= \frac{\varepsilon_1}{2}(1+P) + \frac{\varepsilon_2}{2}(1-P) \\ \varepsilon_y &= \frac{\varepsilon_1}{2}(1-P) + \frac{\varepsilon_2}{2}(1+P)\end{aligned} \tag{3.53}$$

Also γ_{xy} can be calculated from Fig.3.6.

$$\begin{aligned}\gamma_{xy} &= 2R\sin 2\alpha \\ \gamma_{xy} &= 2(\frac{\varepsilon_1-\varepsilon_2}{2})\sin 2\alpha = (\varepsilon_1-\varepsilon_2)\sin 2\alpha\end{aligned} \tag{3.54}$$

If $Q = \sin 2\alpha$ then Eq.3.54 becomes

$$\gamma_{xy} = (\varepsilon_1-\varepsilon_2)Q \tag{3.55}$$

As a result, the usual strain transformation equations yield,

$$\begin{aligned} \varepsilon_x &= \frac{\varepsilon_1}{2}(1+P)+\frac{\varepsilon_2}{2}(1-P) \\ \varepsilon_y &= \frac{\varepsilon_1}{2}(1-P)+\frac{\varepsilon_2}{2}(1+P) \\ \gamma_{xy} &= Q\varepsilon_1 - Q\varepsilon_2 \end{aligned} \tag{3.56}$$

where

$$\begin{aligned} P &= \cos 2\alpha = \frac{\varepsilon_x - \varepsilon_y}{\varepsilon_1 - \varepsilon_2} \\ Q &= \sin 2\alpha = \frac{\gamma_{xy}}{\varepsilon_1 - \varepsilon_2} \end{aligned} \tag{3.57}$$

Within a wrinkled region usual elasticity equations do not apply. Instead, the assumption of negligible bending stress in the membrane yields the stress state as:

$$\sigma_1 = E\varepsilon_1 \quad ; \quad \sigma_2 = 0 \tag{3.58}$$

Or

$$\begin{aligned} \sigma_x &= \frac{1}{2}(1+P)\sigma_1 & & \sigma_x = \frac{1}{2}(1+P)E\varepsilon_1 \\ \sigma_y &= \frac{\varepsilon_1}{2}(1-P)\sigma_1 & \Rightarrow \quad & \sigma_y = \frac{\varepsilon_1}{2}(1-P)E\varepsilon_1 \\ \tau_{xy} &= \frac{1}{2}Q\sigma_1 & & \tau_{xy} = \frac{1}{2}QE\varepsilon_1 \end{aligned} \tag{3.59}$$

Expressing stresses in terms of strains in the form of $\sigma = D\varepsilon$ is desirable for numerical analysis. D is 3×3 matrix, $\sigma = (\sigma_x, \sigma_y, \tau_{xy})^T \quad and \quad \varepsilon = (\varepsilon_x, \varepsilon_y, \gamma_{xy})^T$. Because the problem is statically determinant within the wrinkled region (for example $\sigma_2 = 0$ independent of the values ε_1 *and* ε_2) *D* is singular and many possible representations for *D* are possible. If λ plays role of Poisson's ratio then $\lambda = -\varepsilon_2 / \varepsilon_1$. In a wrinkled region λ is always greater Poisson's ratio [19]. In some points λ can take the value 1. So, choosing *D* matrix similar to Eq.3.48 is not a right choice because of the $1/(1-\lambda^2)$ term. Consider *D* matrix is like

$$\begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \tag{3.60}$$

If Eq.3.60 substitute into $\sigma = D\varepsilon$ and impose Eq.3.56 and Eq.3.59 yields

$$\begin{bmatrix} (1+P) & (1-P) & 2Q & 0 & 0 & 0 \\ (1-P) & (1+P) & -2Q & 0 & 0 & 0 \\ 0 & (1+P) & 0 & (1-P) & 2Q & 0 \\ 0 & (1-P) & 0 & (1+P) & -2Q & 0 \\ 0 & 0 & (1+P) & 0 & (1-P) & 2Q \\ 0 & 0 & (1-P) & 0 & (1+P) & -2Q \end{bmatrix} \begin{Bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{Bmatrix} = E \begin{Bmatrix} (1+P) \\ 0 \\ (1-P) \\ 0 \\ Q \\ 0 \end{Bmatrix} \tag{3.61}$$

Solutions for the elements of *D* matrix are not unique because the coefficient matrix in Eq.3.61 is not singular. If b=0 selected and replaced into Eq.3.61 it is found that

$$\begin{aligned} a &= \frac{E}{2}(1+P) \quad ; \quad d = \frac{E}{2}(1-P) \\ c = e &= \frac{E}{4}Q \quad ; \quad f = \frac{E}{4} \end{aligned} \tag{3.62}$$

The resulting *D* matrix is the matrix which is presented by Miller and Hedgepeth [5] for partly wrinkled membranes. *D* matrix is:

$$D = \frac{E}{4}\begin{bmatrix} 2(1+P) & 0 & Q \\ 0 & 2(1-P) & Q \\ 0 & Q & 1 \end{bmatrix} \tag{3.63}$$

where P and Q stated at Eq.3.57. As a result, in our models elastic elements and wrinkled membrane elements are used. For the elements in the web region on the rollers, the *D* matrix (Eq.3.48) is used to relate stress to strain. The elements in the web span have strain dependent *D* matrices as explained below. Although the slack behavior is possible in some applications it is not possible in a nonlinear formulation which employs an incremental force method. Once edge slackness begins in a web using a force method, convergence to an equilibrium solution is not possible. Convergence would be possible using an incremental displacement solution. Since this research is focused on applications where edge slackness does not occur, an incremental force solution was acceptable.

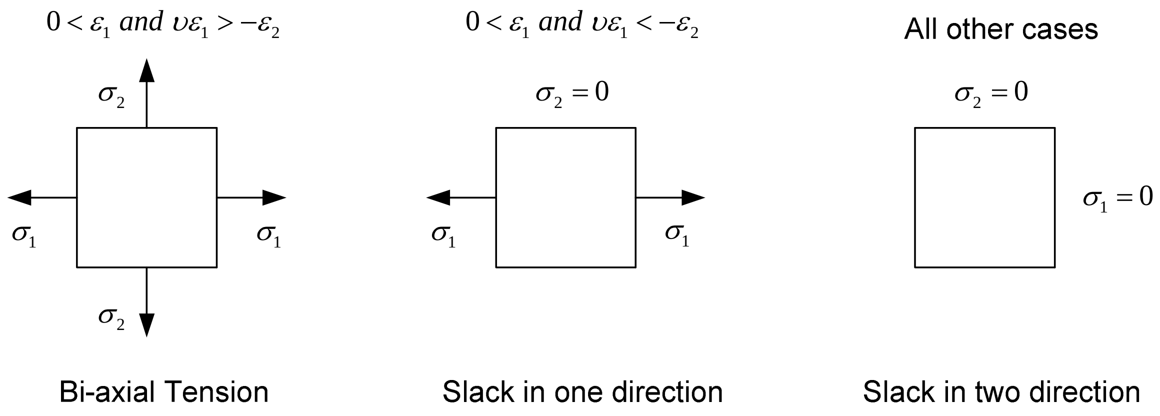


Figure 3.7 Behaviors of Wrinkling Membrane Elements [12]

D matrices are defined for all allowable behaviors (taut, wrinkled and slack). In Fig 3.7 behaviors of wrinkling membrane elements can be seen. A useful algorithm for choosing D matrix may be expressed as [5]:

$$\begin{aligned} &D = D_s \quad ; \varepsilon_1 < 0 \\ &D = D_w \quad ; \varepsilon_1 > 0 \quad and \quad \upsilon\varepsilon_1 < -\varepsilon_2 \\ &D = D_T \quad ; otherwise \end{aligned} \tag{3.64}$$

Where *D* matrices are defined as,

$$D_s = 0 \tag{3.65}$$

For slack behavior, as

$$D_T = \frac{E}{1-\upsilon^2}\begin{bmatrix} 1 & \upsilon & 0 \\ \upsilon & 1 & 0 \\ 0 & 0 & (1-\upsilon)/2 \end{bmatrix} \tag{3.66}$$

For taut behavior, and as

$$D_W = \frac{E}{4}\begin{bmatrix} 2(1+P) & 0 & Q \\ 0 & 2(1-P) & Q \\ Q & Q & 1 \end{bmatrix} \tag{3.67}$$

For wrinkled behavior where $P = \cos 2\alpha$,$Q = \sin 2\alpha$ and as stated in Eq.3.57. These algorithms will be used to establish the code that will be explained in the following chapters.

4. CHAPTER IV

4.1 MISALIGNED ROLLERS

In this chapter the work done of Beisel [12] will be briefly reviewed. The algorithm developed will be explained step by step. Also the code that implements the algorithm will be described. Then the measures taken to decrease CPU time and to automate the code will be discussed. Finally a new slack edge criterion for misaligned rollers will be developed.

4.2 Beisel’s Method for Modeling Wrinkles Due to the Misaligned Rollers

Beisel developed a method to model wrinkle formation due to a downstream misaligned roller. Like Webb he used commercial FE code COSMOS. He used wrinkling membrane elements in free spans to allow troughs to form. In Fig.4.1 Beisel’s wrinkling model for misaligned roller is shown.

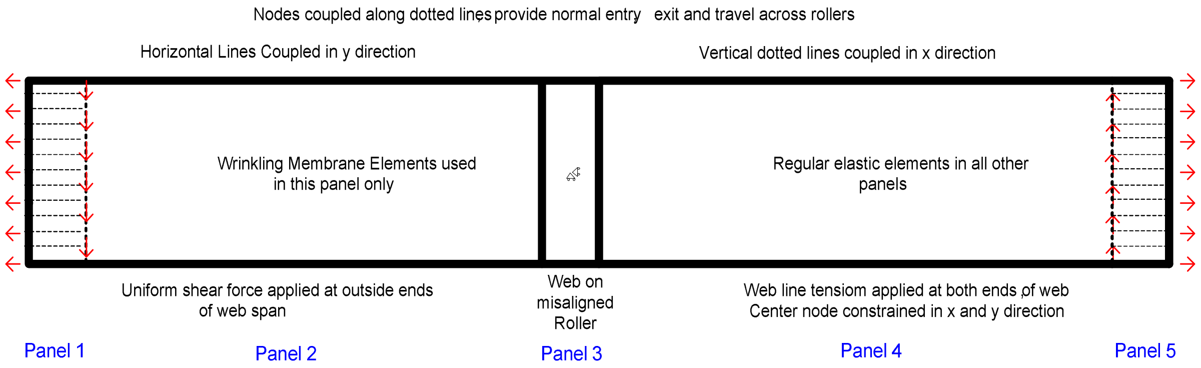

Nodes coupled along dotted lines provide normal entry exit and travel across rollers
Horizontal Lines Coupled in y direction
Vertical dotted lines coupled in x direction
Wrinkling Membrane Elements used in this panel only
Regular elastic elements in all other panels
Uniform shear force applied at outside ends of web span
Web on misaligned Roller
Web line tension applied at both ends of web
Center node constrained in x and y direction
Panel 1
Panel 2
Panel 3
Panel 4
Panel 5

Figure 4.1 Beisel's FE Wrinkle Model for Misaligned Roller

Contrary to Webb, he used five panels of elements rather than three panels. The first panel represents the web on the upstream roller. The second panel represents the free span where wrinkling membrane elements are used. The third panel represents the web on the downstream misaligned roller. He employed the fourth and fifth panels to enforce desired boundary conditions. First, he applied tension to the web when he reached the desired tension load he began to apply shear force to the model as shown in Fig. 4.1.

Beisel employed this five panel model for the following reasons:

A. The asymmetric shear forces allowed him to model the zero moment boundary condition at the misaligned roller found by Shelton [17].

B. The fourth panel was modeled using regular elastic elements. This was done because the fourth panel acts to increase the bending stiffness of the elastic elements in panel three. The failure sequence of events was:

1. Troughs form at a critical angle of misalignment given by Beisel's previous

work [2]. In the finite element code the troughs are modeled by elements that may assume the wrinkled membrane behavior described in Chapter III. This happens whenever $\sigma_2 < 0$. In a real web this does not occur until the σ_2 stress becomes more negative than the buckling stress in the free span σ_{ycr} ($\sigma_{ycr} = -\frac{\pi h}{a}\sqrt{\frac{E\sigma_x}{3(1-\nu^2)}}$; *h* is thickness, and *a* is the distance between two rollers [12]). The onset of troughs can be predicted with linear buckling theory with closed form expressions developed by Beisel [12]. To predict wrinkling of the web on rollers requires non-linear analysis since the entering free span has already buckled in the form of troughs.

2. After troughs form CMD compressive stresses begin to appear in the elastic elements in panel three that border panel two. As the shear forces and the associated misalignment is increased further, these CMD compressive stresses become more negative and finally approach the value in Eq. 2.1, at which point wrinkles are eminent. The elastic elements in the panel four restrict the bending in panel three due to the troughs that have formed in panel two.

Beisel increased the shear forces until the critical compressive stress given by Eq. 2.1 was induced in the linear elastic elements at the entry of the misaligned downstream roller. Then, he concluded that the rotation of the nodes at the entry of the downstream roller should be equal to the angle misalignment in the roller.

Beisel and Webb ran similar experiments to determine the onset of wrinkle formation due to the misaligned downstream roller. Beisel compared his results with these experimental results. He modeled a polyester web with a thickness of 0.00092 in (92 gages). The web parameters for this web were a Young' Modulus of 712000 psi, a Poisson's ratio of 0.3, a web width of 6'' and again the thickness was 0.00092''. The rollers had a radius of 1.45'' and σ_{ycr} from the Eq.2.1 was about -270 psi. He modeled 6''; 18'' and 30'' span lengths and compared his results with the experimental findings. In Fig.4.2-Fig.4.4 the comparison of model and experimental data is presented.

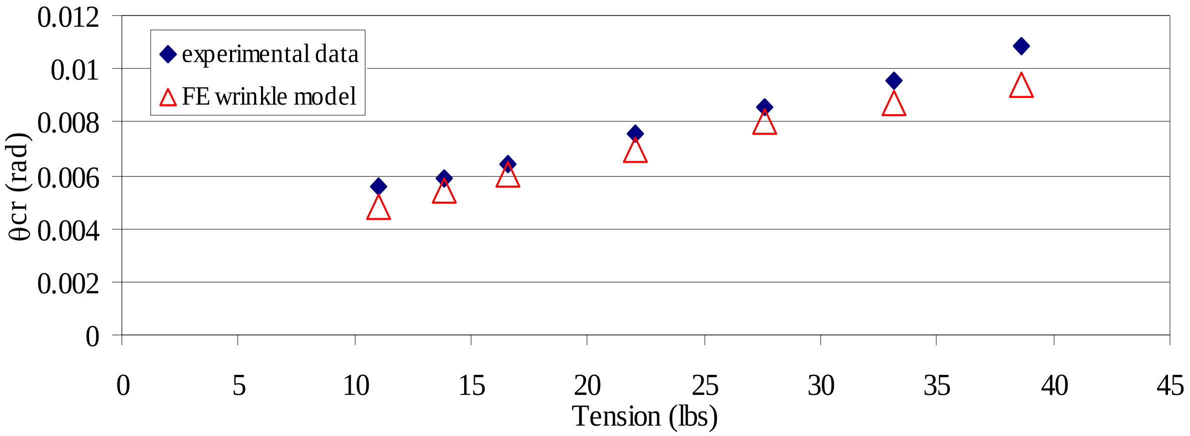


Figure 4.2 L = 6'' Span Results

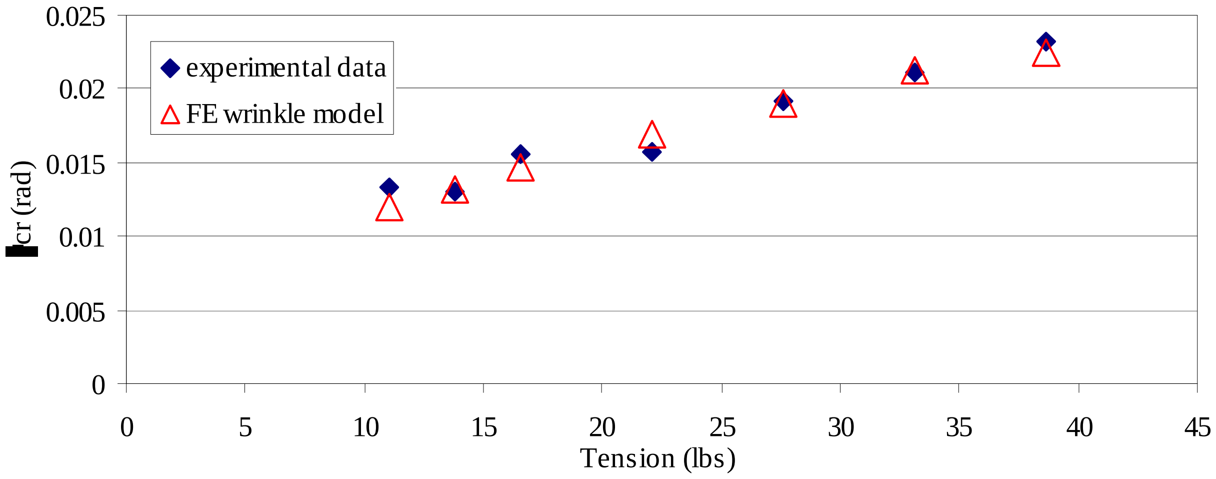


Figure 4.3 L = 18'' Span Results

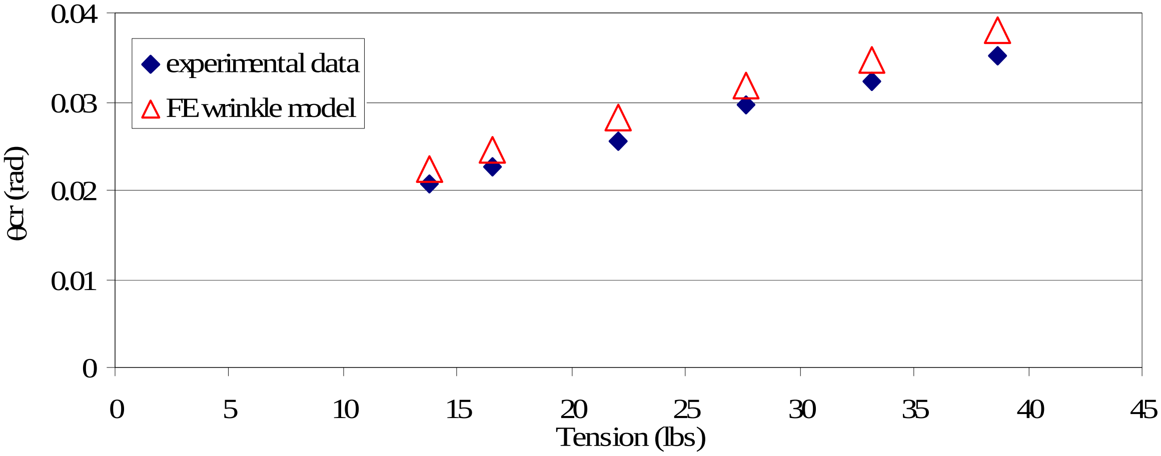


Figure 4.4 L = 30’’ Span Results

As seen from these three charts, the results from FE model and experimental results show good agreement. So Beisel’s model was successful in estimating wrinkle formation due to misaligned roller for a typical web span. In his study he compared his results with Webb’s results. He claims that his model yields better results than Webb’s model. He also claimed that the assumption that Webb proposed (θ_{cr} is the sum of $\theta_{cr,trough}$ and $\theta_{cr,wrinkle}$) is not true. Beisel achieved good agreement with experimental results without relying upon Webb’s assumption.

4.3 A New Algorithm for Predicting Wrinkles Due to the Misaligned Rollers

As mentioned in the introduction, there are three types of wrinkles: MD wrinkles, CMD wrinkles and shear wrinkles. Shear wrinkles can occur due to roller imperfections such as misaligned rollers or tapered rollers. The goal of this new algorithm is to codify and automate the analysis that Beisel perfected using a commercial finite element code (COSMOS). The term “shear wrinkle” resulted from the realization that these troughs and finally wrinkles were the result of shear forces in the web. Both the misaligned roller and the tapered roller induce shear in the web.

Beisel [12] was successful to develop a method for predicting web wrinkles on rollers by using membrane elements described by Miller and Hedgepeth [5]. He applied this method to the prediction of wrinkles due to misalignment in rollers, tapered rollers and crowned rollers and he confirmed his results with laboratory tests.

Beisel used commercial finite element code COSMOS to apply this method. In this method while the elements representing the web on rollers are modeled with elastic elements, the web in the free spans are modeled with wrinkle membrane elements. These elements cannot react compressive stresses and they can be in one of three states. These states include taut web, wrinkled web and slack web. In the taut web state, the elements can resist tensile stresses in both principal directions. In the wrinkled web state, membrane elements can withstand tensile principal stress in one direction and zero stress in the other principal direction. In the slack web state, the elements can carry no stresses in any direction.

In this algorithm, forces are increased in to the model step by step. In the first step, all elements are modeled with elastic elements. After the first step, the principal strains for each element in free spans are calculated and stored. By using principal strains, code will select which of D matrices given below for three states will be used for the next load step.

$$\begin{aligned} &D = D_s \quad ;\varepsilon_1 < 0 \\ &D = D_w \quad ;\varepsilon_1 > 0 \quad and \quad \upsilon\varepsilon_1 < -\varepsilon_2 \\ &D = D_T \quad ;otherwise \end{aligned} \tag{4.1}$$

where *D* matrices are defined at Eq.3.65-3.67. After convergence is obtained in each step, the compressive CMD stresses in the linear elastic elements are reviewed. If those stresses remain greater than the Timoshenko shell buckling stress (Eq.2.1) the shear force would be increased. If the CMD stresses in these elements became more negative than the Timoshenko shell buckling stress (Eq.2.1), the shear force would need to decrease and a bracketing method would be employed to determine what shear force would produce a negative a negative CMD stress essentially equal to the Timoshenko buckling stress. Once accomplished, the misalignment or taper that induced that level of shear force would be determined.

By using the method explained above, a finite element code will be developed in Excel VBA (Visual Basic Excel). This code can be executed in any PC with Excel installed without need of a commercial FE code license and it allows users to analyze the misaligned roller case using a simple Excel based interface. The advantage of this code will be that users will not need any linear or nonlinear finite element background to execute the code. They do not have to know the kinetic and kinematic boundary conditions for misaligned rollers. The inputs will include parameters such as web tension, web width, span length, roller diameter, Poisson’s ratio and elastic modulus. When executed the code will automatically form a finite element mesh based upon the inputs with elastic quadrilateral elements representing the web supported by rollers and with wrinkle membrane quadrilateral elements representing the web in the free span. The first code will implement boundary conditions for a web approaching a misaligned roller. Other boundary conditions will be studied later.

Beisel [2, 12] and Webb [9] studied the misaligned roller case. The boundary conditions that they used are first proposed by Shelton [14] and then Good et al. [4]. They considered the web span as a beam. A classic beam is one which the web span length would be ten times longer than the width. Shear effects become "important" when $L/w < 10$. Tension becomes "important" when the lateral deformations become large. "important" in this context means that these effects have sizable influence on the lateral deformations of the web.

The boundary conditions that are used by Beisel [2, 12], Webb [9], and others will be used in this model. The validity of using these boundary conditions was verified by comparison to experimental results by these authors.

The normal entry condition of a web approaching a roller was enforced using coupling equations which enforce multipoint constraints. Lines of adjacent nodes crossing a roller in panel one and panel five had their CMD displacements coupled. Each adjacent line of nodes was coupled separately and in this way Poisson contraction of the web could occur unimpeded. There was no coupling of nodes in the web in contact with the misaligned roller. Since the moment in the web in the vicinity of the misaligned roller is small or zero the deformations of nodes are nearly that associated with a rigid body rotation. This results in the normal entry condition being satisfied in the web at the misaligned roller without enforcing a multi-point constraint. The lines of nodes in the CMD at the exit of panel one and entry of panel five were each coupled in machine direction displacement. This procedure was done to ensure the maximum moment in the free web spans occurred at the border between span one and span two and the border between span four and span five.

The system that is modeled is shown in Fig .4.5.

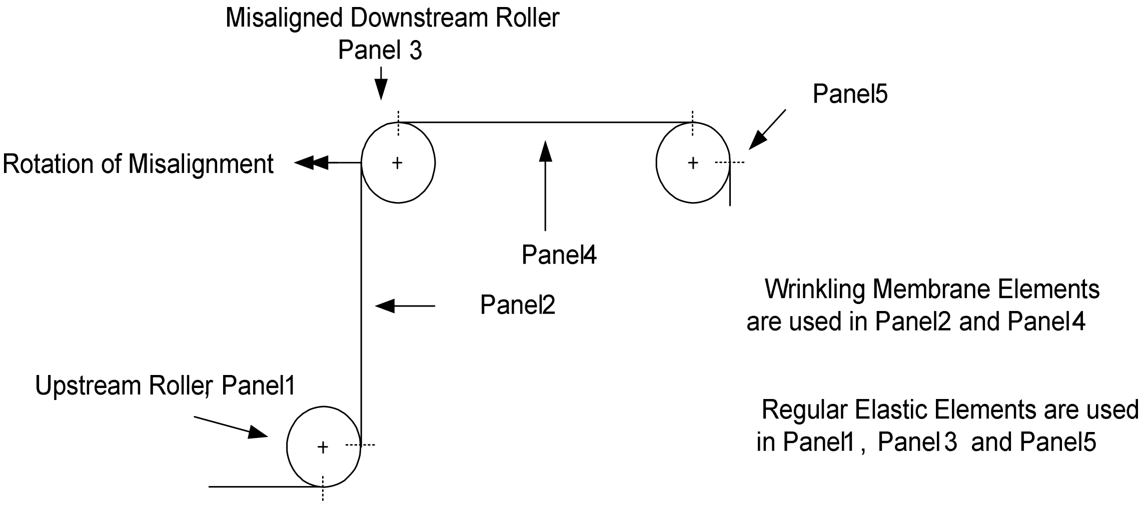

Misaligned Downstream Roller
Panel 3
Panel5
Rotation of Misalignment
Panel4
Wrinkling Membrane Elements
are used in Panel2 and Panel4
Panel2
Upstream Roller, Panel1
Regular Elastic Elements are used
in Panel1 , Panel 3 and Panel5

Figure 4.5 The System That is Modeled

The system of five panels is shown in Fig.4.5 modeled in Fig.4.6. The coupling discussed earlier is also shown.

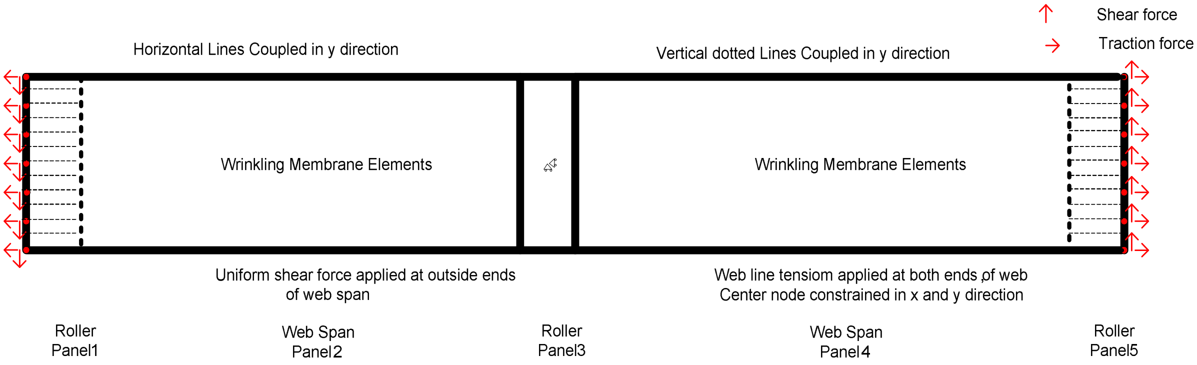
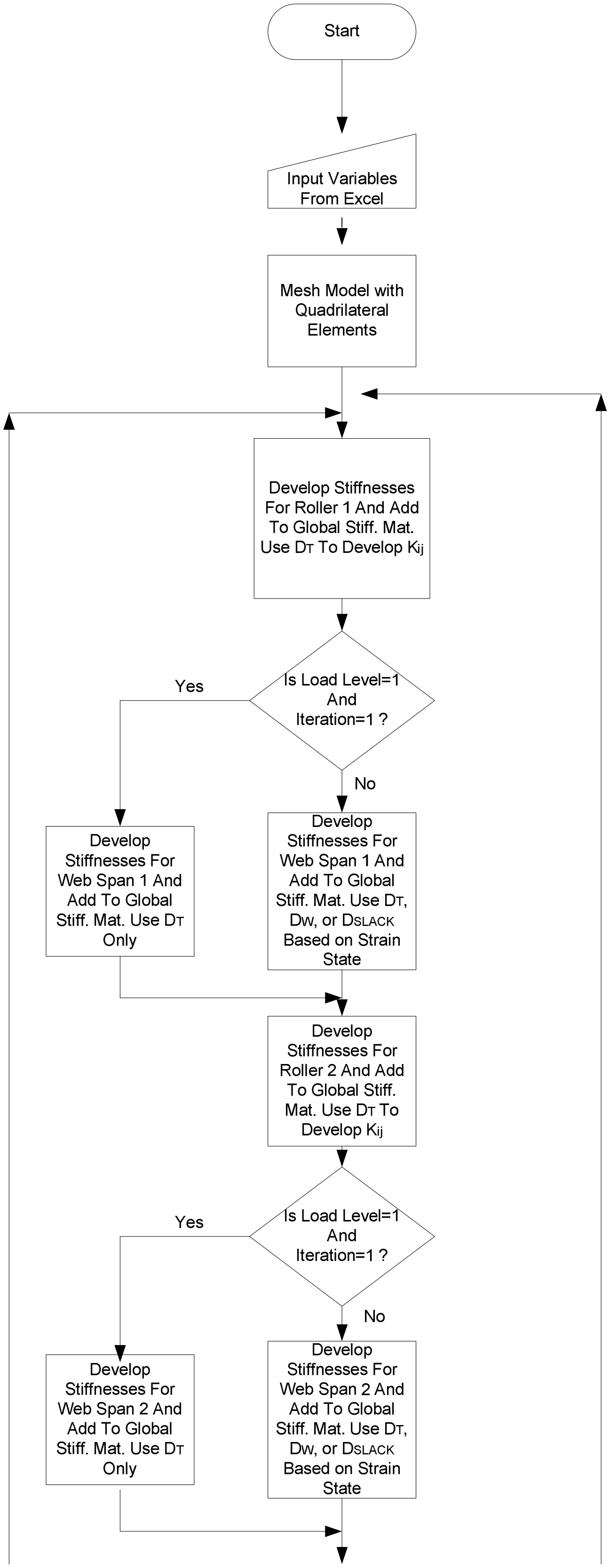

Shear force
Traction force
Horizontal Lines Coupled in y direction
Vertical dotted Lines Coupled in y direction
Wrinkling Membrane Elements
Wrinkling Membrane Elements
Uniform shear force applied at outside ends of web span
Web line tensiom applied at both ends ,of web
Center node constrained in x and y direction
Roller
Panel1
Web Span
Panel2
Roller
Panel3
Web Span
Panel4
Roller
Panel5

Figure 4.6 Misaligned Roller FE Wrinkle Model

The model is divided into five sections. The first panel represents the web on the upstream roller. The coupled nodes in this panel are used to enforce normal entry and exit on upstream roller. The second panel represents the entry web span to the misaligned roller. Here wrinkling membrane elements are used to simulate web behavior which allow troughs. Different from Beisel’s model at the first attempt the fourth panel is also modeled with wrinkling membrane elements. Shear forces are applied to the web on the upstream and downstream rollers to simulate the shear, moment, and lateral deformations of a web passing over a misaligned roller. The third panel represents the web on the downstream misaligned roller. A central node is fixed in the MD and CMD directions to prevent rigid body translations of the model. Rigid body rotation is prevented by the coupling of the CMD deformations of the lines of nodes crossing panel one and panel five. The fourth panel and the fifth panel elements and boundary conditions help ensure the zero moment boundary condition at the misaligned roller. The flow chart for the program is shown in Fig. 4.7.


Start
Input Variables From Excel
Mesh Model with Quadrilateral Elements
Develop Stiffnesses For Roller 1 And Add To Global Stiff. Mat. Use D_T To Develop K_{ij}
Is Load Level=1 And Iteration=1 ?
Yes
No
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use D_T Only
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use D_T, D_W, or D_{SLACK} Based on Strain State
Develop Stiffnesses For Roller 2 And Add To Global Stiff. Mat. Use D_T To Develop K_{ij}
Is Load Level=1 And Iteration=1 ?
Yes
No
Develop Stiffnesses For Web Span 2 And Add To Global Stiff. Mat. Use D_T Only
Develop Stiffnesses For Web Span 2 And Add To Global Stiff. Mat. Use D_T, D_W, or D_{SLACK} Based on Strain State

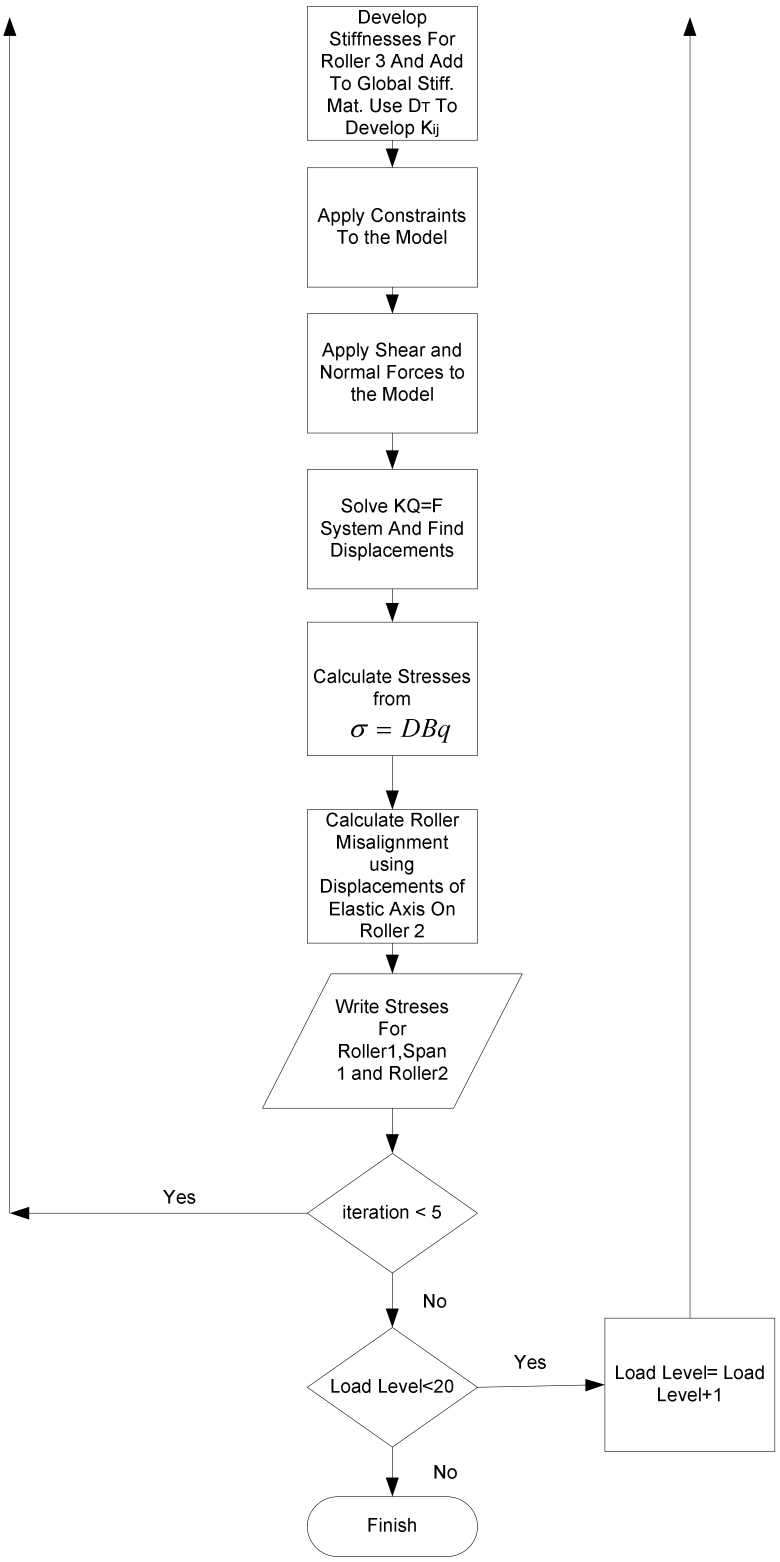

Develop Stiffnesses For Roller 3 And Add To Global Stiff. Mat. Use DT To Develop Kij
Apply Constraints To the Model
Apply Shear and Normal Forces to the Model
Solve KQ=F System And Find Displacements
Calculate Stresses from
σ = DBq
Calculate Roller Misalignment using Displacements of Elastic Axis On Roller 2
Write Streses For Roller1,Span 1 and Roller2
iteration < 5
Yes
No
Load Level<20
Yes
Load Level= Load Level+1
No
Finish

Figure 4.7 Flow Chart for New Algorithm

The stiffness terms for the elements on the upstream roller are calculated by using $D_{T.}$ Then these stiffness terms are assembled into the global stiffness matrix. The same procedure is followed for the other web regions on the rollers. For the web spans *D* matrix for the elements will be formed differently. At the first load level and first iteration D_T is used for all elements. Then, for all remaining load levels and for all iterations, the program selects one of the three *D* matrices from (Eq. *3.64)* D-taut, D-wrinkled or D-slack by evaluating the principal strains calculated in a previous step as explained in Eq. 3.65-67. The nonlinearity for this case is due to the variable D matrices for the span elements as the shear loads increase. After selecting *D* matrices, elemental stiffness matrices are formed and the stiffness terms will be assembled into the global stiffness matrix using the same procedure as the web on the rollers.

After the global stiffness matrix is formed, lines of nodes in the MD on the first and fifth panels are coupled in the *y* direction and the point at the center of the model is fixed in *x* and *y* directions as shown in figure 4.6.

Next, the shear and traction forces are applied to the system. From the set of equations *KQ=F* the displacements {*Q*} can now be calculated. Strains $\{\varepsilon\}$ and stresses $\{\sigma\}$ can then be determined using the displacements. The strains are calculated using $\varepsilon = BQ$, stresses are calculated with the aid of $\sigma = DBQ$ for all elements. After calculating strains, principal strains are also determined from the cartesian strains so that the proper *D* matrices can be selected for the next iteration or for the next load level.

The shear and tension forces are applied incrementally in 20 steps. Five iterations are performed for each load step to allow P and Q to converge. As mentioned before if the principal axes one and two are rotated through an angle α relative to the *x*-*y* axes, *P* is the cosine of that angle and *Q* is the sine of that angle.

For each new load step, the code first uses the principal strains calculated in the 5th iteration of the previous load step to determine the state of the wrinkling membrane for the 1st iteration. Then, the strains calculated in the first iteration for that load level are used to select *D* matrices for the second iteration. The procedure will continue in the same way. At each iteration, the strains calculated for the previous iteration are used to select *D* matrices till the maximum iteration number (5) is reached for that load step. The analysis proceeds in load steps with iterative analysis steps occurring within each load step. The iterative analysis steps are necessary to allow the values of P and Q to converge in the elements with the wrinkled behavior prior to moving to the next load step. In Fig.4.8 it shows how P and Q (Eq.3.57) behave with iteration for an element in Panel 2.

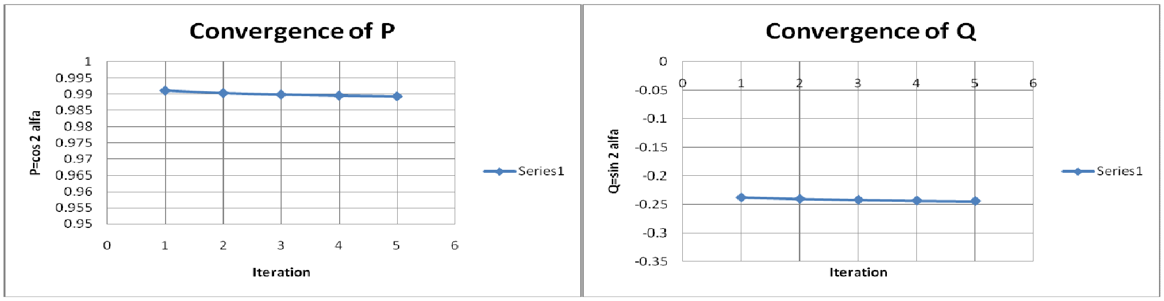
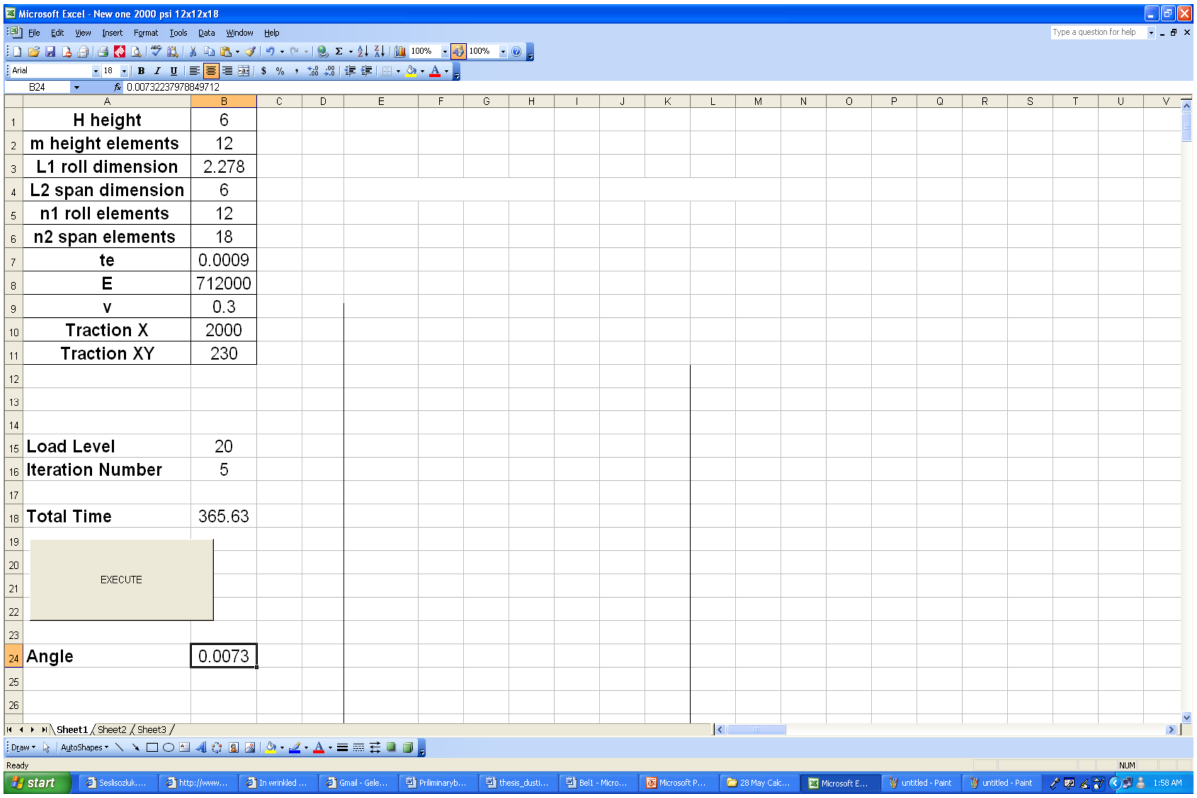


Figure 4.8 Convergences of P and Q



Microsoft Excel - New one 2000 psi 12x12x18

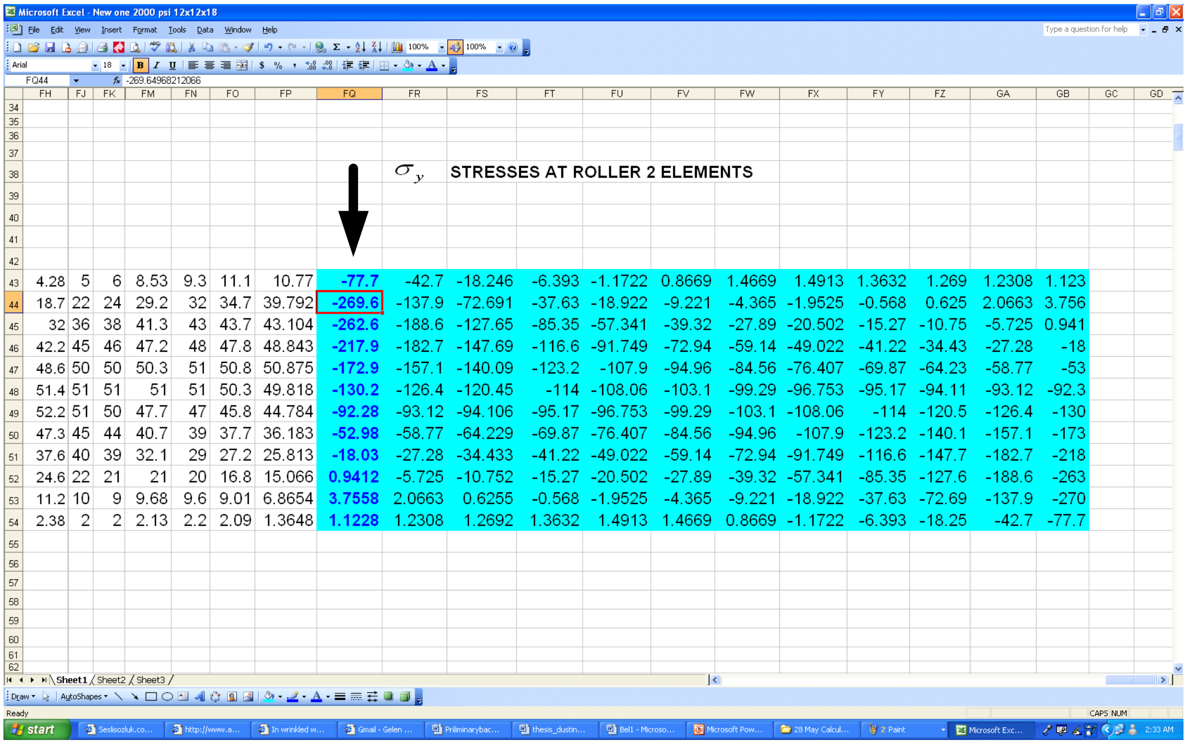
B24 0.00732237978849712

H height	6
m height elements	12
L1 roll dimension	2.278
L2 span dimension	6
n1 roll elements	12
n2 span elements	18
te	0.0009
E	712000
v	0.3
Traction X	2000
Traction XY	230
Load Level	20
Iteration Number	5
Total Time	365.63
Angle	0.0073

Figure 4.9 Excel Input for the Program

In Fig. 4.9, the screen where the web parameters are input to the code is shown. As seen from the Excel input screen, the user enters the parameters such as width of the web, a roll dimension (quarter circumference of the roller), a span length dimension, the thickness of the web, the elastic modulus of the web, Poisson's Ratio of the web, and the web tension in units of traction (stress) in the x direction. Also the shear in y direction in units of traction (stress) to the Excel sheet is input. In the "m height elements" cell, user enters the mesh density along *x* direction. In "n1 roll elements" cell, user inputs the mesh density of the rollers along *y* direction. Similarly in n2 span elements cell user decides about the mesh density of the span elements along *y* direction. This is not the final form of input. In the next chapters the efforts for the final form of input will be discussed.

The output of the code developed to predict wrinkling due to downstream misaligned roller are σ_x *and* σ_y stresses for all elements on upstream roller, the downstream roller and in the web span between the upstream and the downstream roller are shown in Figure 4.10.



Microsoft Excel - New one 2000 psi 12x12x18

FQ44 = -269.64968212066

σ_y STRESSES AT ROLLER 2 ELEMENTS

	FH	FJ	FK	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV	FW	FX	FY	FZ	GA	GB
43	4.28	5	6	8.53	9.3	11.1	10.77	-77.7	-42.7	-18.246	-6.393	-1.1722	0.8669	1.4669	1.4913	1.3632	1.269	1.2308	1.123
44	18.7	22	24	29.2	32	34.7	39.792	-269.6	-137.9	-72.691	-37.63	-18.922	-9.221	-4.365	-1.9525	-0.568	0.625	2.0663	3.756
45	32	36	38	41.3	43	43.7	43.104	-262.6	-188.6	-127.65	-85.35	-57.341	-39.32	-27.89	-20.502	-15.27	-10.75	-5.725	0.941
46	42.2	45	46	47.2	48	47.8	48.843	-217.9	-182.7	-147.69	-116.6	-91.749	-72.94	-59.14	-49.022	-41.22	-34.43	-27.28	-18
47	48.6	50	50	50.3	51	50.8	50.875	-172.9	-157.1	-140.09	-123.2	-107.9	-94.96	-84.56	-76.407	-69.87	-64.23	-58.77	-53
48	51.4	51	51	51	51	50.3	49.818	-130.2	-126.4	-120.45	-114	-108.06	-103.1	-99.29	-96.753	-95.17	-94.11	-93.12	-92.3
49	52.2	51	50	47.7	47	45.8	44.784	-92.28	-93.12	-94.106	-95.17	-96.753	-99.29	-103.1	-108.06	-114	-120.5	-126.4	-130
50	47.3	45	44	40.7	39	37.7	36.183	-52.98	-58.77	-64.229	-69.87	-76.407	-84.56	-94.96	-107.9	-123.2	-140.1	-157.1	-173
51	37.6	40	39	32.1	29	27.2	25.813	-18.03	-27.28	-34.433	-41.22	-49.022	-59.14	-72.94	-91.749	-116.6	-147.7	-182.7	-218
52	24.6	22	21	21	20	16.8	15.066	0.9412	-5.725	-10.752	-15.27	-20.502	-27.89	-39.32	-57.341	-85.35	-127.6	-188.6	-263
53	11.2	10	9	9.68	9.6	9.01	6.8654	3.7558	2.0663	0.6255	-0.568	-1.9525	-4.365	-9.221	-18.922	-37.63	-72.69	-137.9	-270
54	2.38	2	2	2.13	2.2	2.09	1.3648	1.1228	1.2308	1.2692	1.3632	1.4913	1.4669	0.8669	-1.1722	-6.393	-18.25	-42.7	-77.7

Figure 4.10 Excel Output for Stresses

For this case critical buckling stress by Eq.2.1 is about -270 psi. The marked row of stresses are the stresses of the elements at the entry of the downstream roller. The elements at the entry of the roller buckle first. After the output is displayed the minimum (most negative) stress in these elements should be compared with the critical shell buckling stress. If unequal, then the shear force should be increased or decreased until the minimum compressive stress in the elements at the entry of the downstream roller reaches the critical shell buckling stress. In Fig.4.10, the maximum compressive stress (-269.6) reached the Timoshenko shell buckling stress (-270). At this instant, the angle displayed as an output in Figure 4.9. is the critical angle of misalignment of the downstream misaligned roller for the onset of wrinkling (0.0073 rad.).

4.4 Comparison of Results with Previous Works

The code that was developed was executed for some cases defined by Beisel and the code results are compared with Beisel's experimental and FE results. As mentioned in the previous chapter he used a commercial FE code called COSMOS and modeled the misaligned roller case. He performed his tests using 92 gage polyester web to verify his model. For this web, Young's Modulus is 712000 psi, Poisson's ratio is 0.3, the width of web is 6'' and the thickness is 0.00092''. The roller has a radius of 1.45''. Eq.2.1 yields about -270 psi for the critical shell buckling stress for this case.

Beisel was comparing the stresses at the nodes at the entry of the roller with the critical compressive stress predicted by Timoshenko shell buckling criteria. In this study, compressive stresses in the elements at the entry of the roller are used in the comparison instead of nodal stresses. Stresses at four Gauss points of the elements are computed and their averages are found as elemental stresses.

The following graphs in Fig. 4.11-13 show the experimental critical angle of misalignment at the downstream roller, the angle predicted by Beisel's model and by Yurtcu's model for 6'', 18" and 30'' web span lengths.

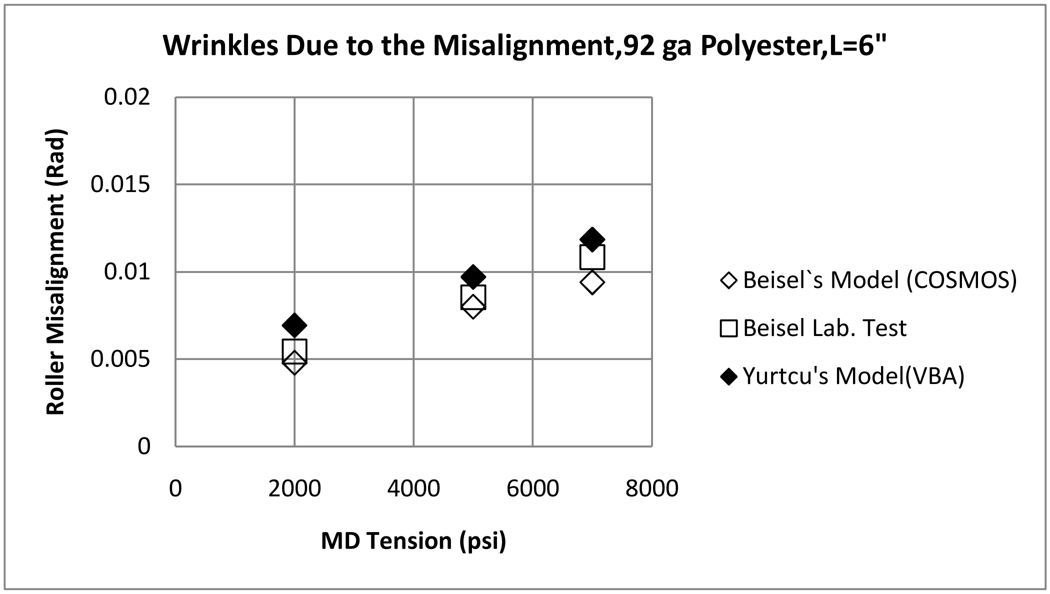


Figure 4.11 Comparison of 6'' Span Results

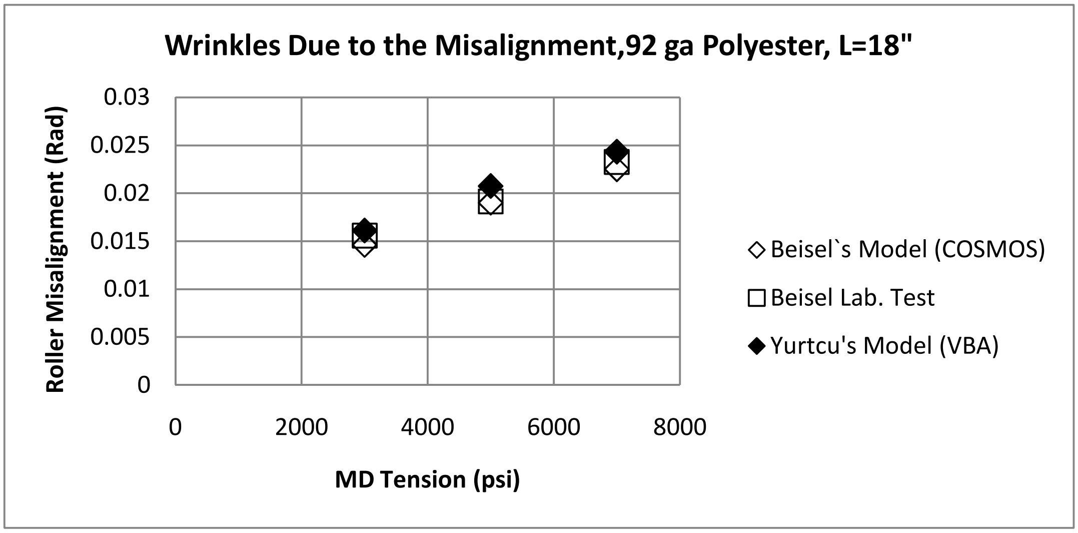


Figure 4.12 Comparison of 18'' Span Results

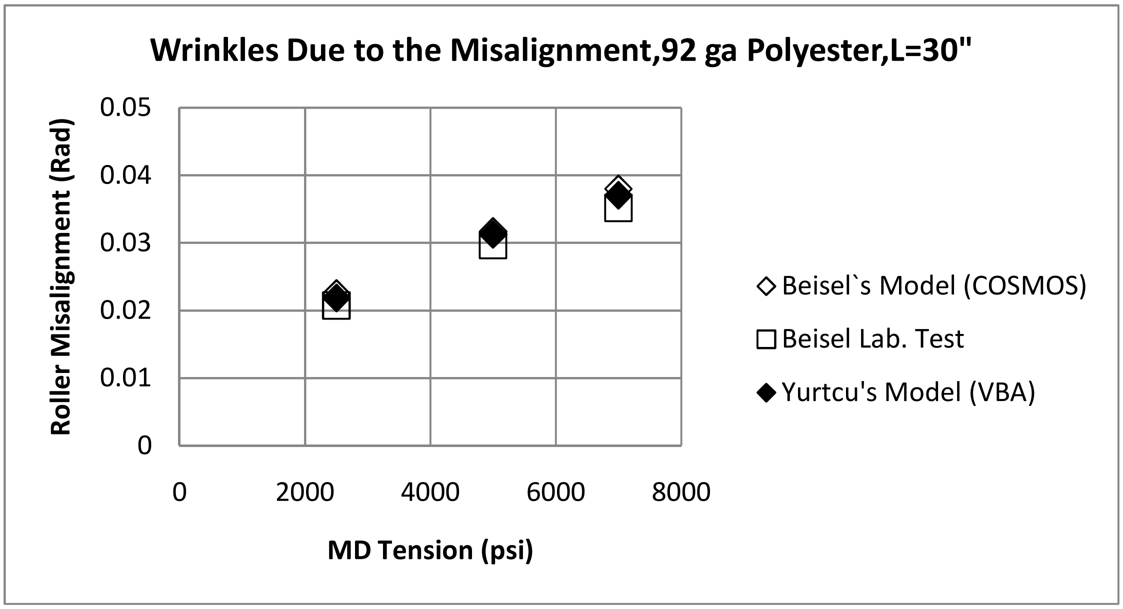


Figure 4.13 Comparison of 30’’ Span Results

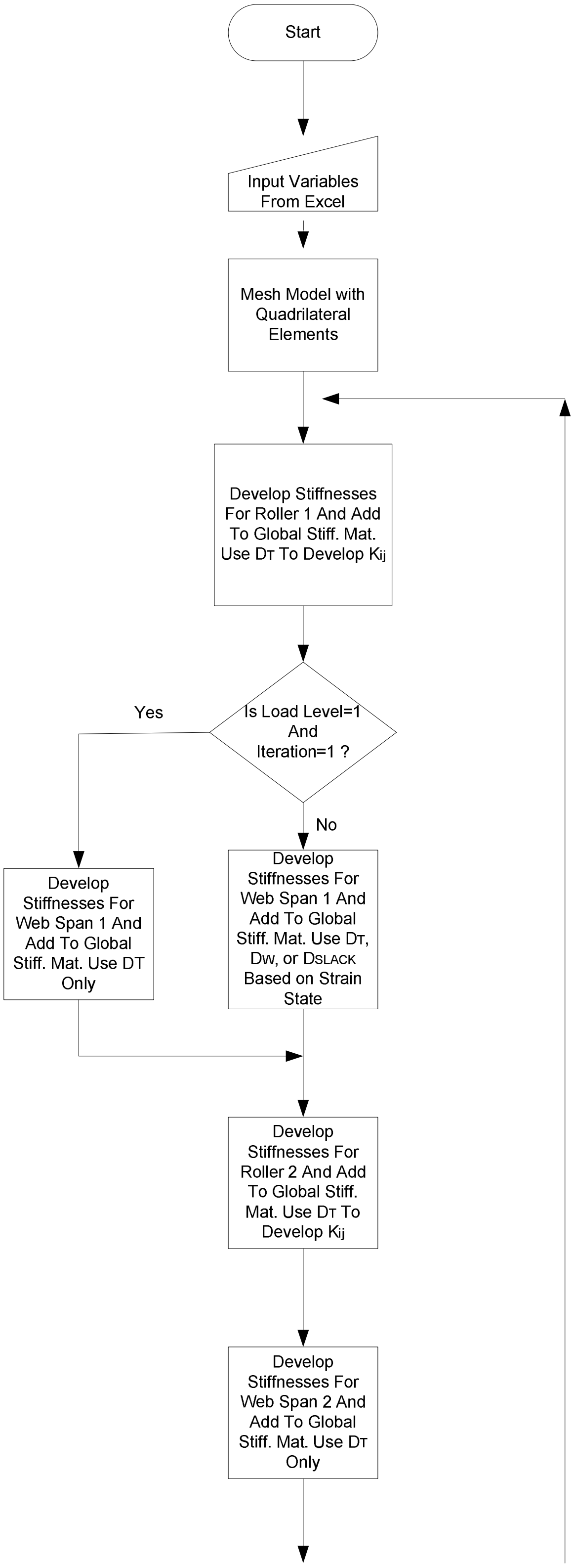
From the figures, it can be seen that the results of the code developed agree well with the lab test data and the results of the commercial code.

4.5 Improvement of Execution Time

It was found that execution the code for long spans that twenty load steps with five internal iterations in each load step required a large amount of CPU time. For example, 2000 elements may be employed in a long span case. For every element we have an (8*8) stiffness matrix. Forming the global stiffness matrix for 2000 elements and solving it 100 times (five iterations in 20 load steps) required extensive CPU time. Although we use banded matrix in our code long spans could require three hours to produce a result on a single core computer.

Due to these long execution times the next focus was decreasing the CPU time. First the number of iterations was decreased from five to two, then from two to one, and the result was not significantly changed. Then step by step the load step increments were decreased from twenty to four, and still reasonable results were achieved. However, it was found that decreasing the load step increments to less than four caused the results to err dramatically. This problem is appears to converge with four load increments and one iteration within each load increment.

It was mentioned in the previous section that Beisel modeled the second span with elastic elements, but that wrinkling membrane elements were used in this code. This means that we were calculating principal strains to select which D matrix would be used for every element in span two. Beisel ran taut elements in the second free span because the model was a 90^0 wrap case. For the misaligned roller case this subjects the upstream span to shear and the downstream span to twist. A web span will absorb a large amount of twist without forming negative σ_2 stresses. So Beisel assumed the D in all elements in the downstream span would remain taut. We forced D matrices to be D_T in the second free span. This also provided better agreement between our code and other results. Changing element types from wrinkling membrane elements to elastic elements also helped to decrease CPU time. Our code produced reasonable results, and did this within seconds. After making these changes, the flowchart of our program will be like Fig.4.14. The name "*WRINKLINGsystem*" will be applied to the part of the new chart that begins after "Mesh Model with Quadrilateral Elements" and continues to "Load Level<4". This name will be used while attempting to automate the code in Part 4.6.


Start
Input Variables From Excel
Mesh Model with Quadrilateral Elements
Develop Stiffnesses For Roller 1 And Add To Global Stiff. Mat. Use DT To Develop Kij
Is Load Level=1 And Iteration=1 ?
Yes
No
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT Only
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT, DW, or DSLACK Based on Strain State
Develop Stiffnesses For Roller 2 And Add To Global Stiff. Mat. Use DT To Develop Kij
Develop Stiffnesses For Web Span 2 And Add To Global Stiff. Mat. Use DT Only

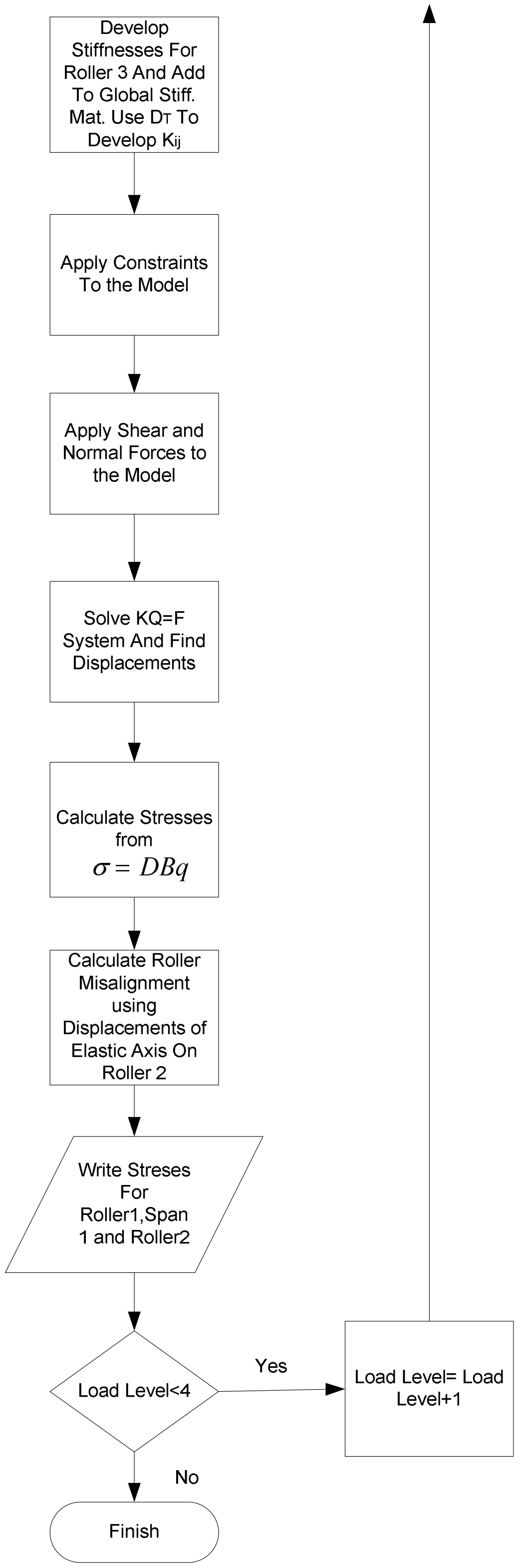

Develop Stiffnesses For Roller 3 And Add To Global Stiff. Mat. Use DT To Develop Kij
Apply Constraints To the Model
Apply Shear and Normal Forces to the Model
Solve KQ=F System And Find Displacements
Calculate Stresses from
σ = DBq
Calculate Roller Misalignment using Displacements of Elastic Axis On Roller 2
Write Streses For Roller1,Span 1 and Roller2
Load Level<4
Yes
Load Level= Load Level+1
No
Finish

Figure 4.14 Flow Chart of Computer Code for Improved Execution Speed

4.6 Automating the Code

From Fig 4.9 it can be seen that the user enters parameters such as web width, roller wrap arc length, span length, web thickness, web elastic modulus, Poisson's ratio, the number of elements across the web width, the number of elements down the span length, the number of elements across the roller, the web tension and shear. The research objective was to limit the inputs that could be comprehended by users with no knowledge of the finite element method. These would include only web width, roller wrap arc length, span length, web thickness, web elastic modulus, Poisson's ratio and web tension.

One of the inputs required by the code is web tension. Web tension is one of a very few parameters that are controllable in a web process machine. Thus it would be optimal for a chart to be produced for the user that shows how much misalignment is allowable as a function of web tension, rather than computing what misalignment in a roller is acceptable at one tension. The user can then decide to solve an instability problem by better aligning the rollers or by changing the web tension. Thus the user should enter only certain parameters and the code will determine the mesh parameters and shear force required to induce wrinkling.

To automate the mesh parameters the code was executed several times to explore what mesh density was required to produce a converged result. Very reasonable results were obtained by dividing the width and span length per piece per dimension and dividing the rollers with the integer part of six times of one-fourth of roller circumference. The mesh density is delimited for very long span lengths, very short span lengths, very long wide webs and very short wide webs. The meshing procedure and convergence check will be addressed more detailed at

Chapter VIII.

The next step was determining how to automate the search for the shear force that was required to induce the wrinkle instability. The user inputs the web tension. A linear interpolation scheme was used to determine the shear force. One level of shear will produce a certain level of compressive σ_y stress at the misaligned roller, which can be determined by the code. A second level of shear will produce another level of compressive σ_y stress. Interpolation can be used to estimate the level of shear that will produce the shell buckling stress. It is an estimate because this is nonlinear analysis. That estimate can then be input to the code to help refine the actual shear level that will induce wrinkling. A slack edge criteria was used as a starting point. If a slack edge forms during the computations the code will fail. This is because increased shear will not result in increased compressive stress in the web on the roller, it will only increase slackness. For more information about slack edge criteria earlier work done by Good [40] can be visited. Web spans can be modeled as beams [12]. From Euler-Bernoulli beam bending theory the stress in the cross-section is:

$$\sigma = \frac{F}{A} \pm \frac{My}{I} \tag{4.2}$$

If this stress is equal to zero, a slack edge occurs. If our web has a thickness of *t*, width of *w*, span length of *L* and the applied traction in *x* direction is T_x the shear stress (T_{xy}) for the web to be slack can be calculated from Eq.4.2 as:

$$\sigma = 0 = T_x - \frac{M\frac{w}{2}}{I} \tag{4.3}$$

Where moment M is calculated from:

$$M = T_{xy} \times w \times t \times L \tag{4.4}$$

If this value put into Eq.4.2 and do the calculations T_{xy} (T_{slack}) will be found as

$$T_{xy} = \frac{T_x \times w}{6L} \tag{4.5}$$

This was used as the starting value of *Traction XY* (shear) in the code (Fig.4.9).It found that for all cases computed the compressive stress in the first row of elements in the roller two was lower (less negative) than the value that was calculated from Timoshenko buckling criteria (Eq.2.1). Thus it found that if the shear value in Eq.4.5 was used as *traction xy* (T_{slack}) and half of it ($T_{slack}/2$) one could be sure that there were two data points which would produce compressive stresses less than that is needed to buckle the web (T$_{critical}$) (Figure 4.15).

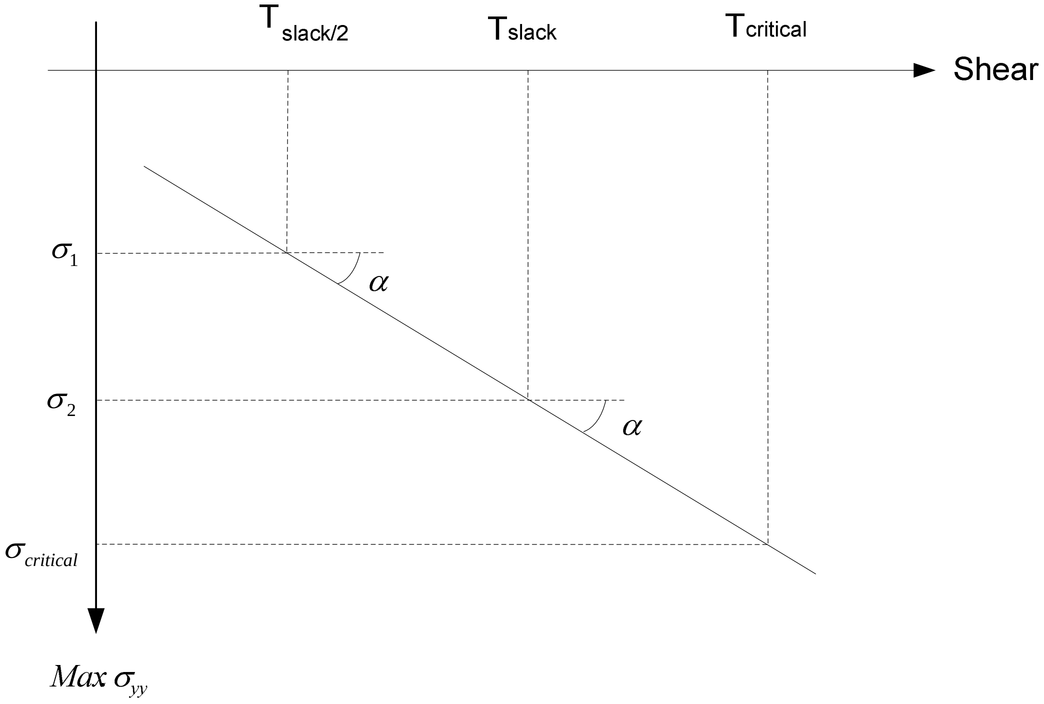


Figure 4.15 Linear Interpolation I

After finding two points we can estimate $T_{critical}$ from the tangent $\tan\alpha$:

$$\tan\alpha = \frac{\sigma_2 - \sigma_1}{T_{Slack} - (T_{Slack}/2)} = \frac{\sigma_{critical} - \sigma_2}{T_{Critical} - T_{Slack}} \tag{4.6}$$

from here $T_{critical}$ can be found as:

$$T_{Critical} = \frac{(\sigma_{critical} - \sigma_2) \times (T_{Slack} - (T_{Slack}/2))}{\sigma_2 - \sigma_1} + (T_{Slack}/2) \tag{4.7}$$

The algorithm for Linear Interpretation I can be seen in Figure 4.16. The code takes the value of web tension (traction *x*) from Excel input page and calculates *F1, F2* and *F3* values with the help of T_{slack} (Eq.4.5). Here *F1* is equal to T_{slack} /2, *F2* is equal to T_{slack} and *F3* is equal to 2* T_{slack} .Than code runs for *F1* and *F2* values within the *WRINKLINGsystem* and calculates *Sigma1* and *Sigma2* values. By using Eq.4.7 code calculates $T_{critical}$. The code checks whether $T_{critical}$ is larger than two times of T_{slack} . This was done because during the runs it was observed that if this was not done, the value of $T_{critical}$ increases dramatically because of the angle between two points. Taking this step helps control $T_{critical}$ value. Then the code runs *WRINKLINGsystem* for $T_{critical}$ value and finds *SigmaS.* The code continues this process until *SigmaS* is bigger than the Timoshenko buckling criteria (*Sigmacritical*). At the end of Linear Interpolation I, there is a *F1* value which is less than the traction that is needed to buckle the web, and there is a *F2* value which is more than the traction that is needed to buckle the web (Figure 4.17).

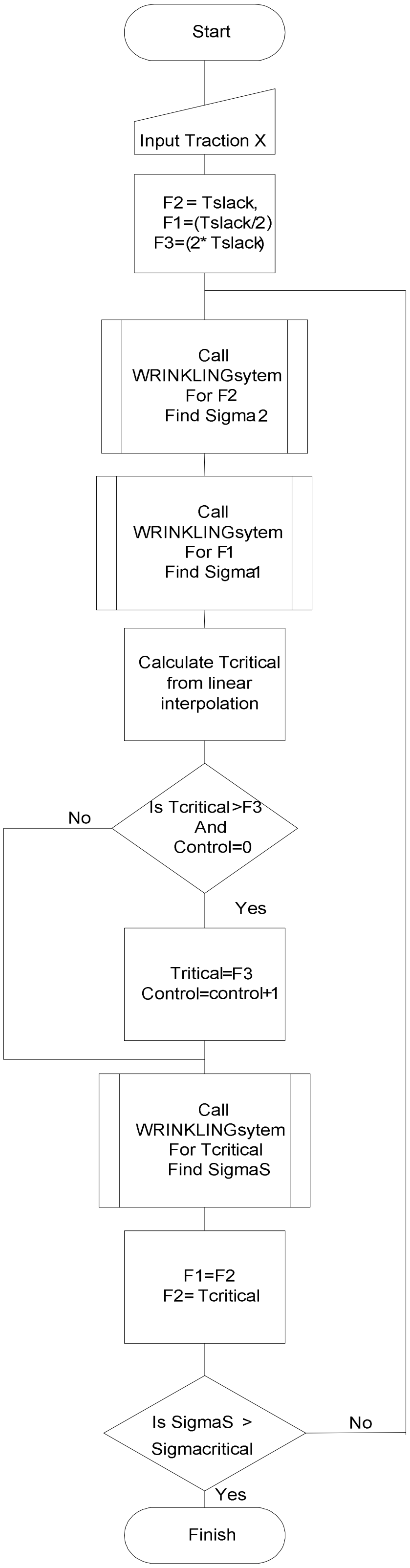


Figure 4.16 Flow Chart for Linear Interpolation I

The code for linear interpolation one will be:

```
TRACTIONXY = (TRACTIONX * H) / (6 * (L1 + L2 + L1 + L2 + L1))

F2 = TRACTIONXY

F3 = 2 * TRACTIONXY

Do Until sigmaS > syrc

   FORCEXY = F2 * H * te

   NFORCEXY = FORCEXY / m

   Call WRINKLINGsystem

   sigma2 = critical

   F1 = F2 / 2

   FORCEXY = F1 * H * te

   NFORCEXY = FORCEXY / m

   Call WRINKLINGsystem

   sigma1 = critical

   FS = ((sycr - sigma2) * (F2 - F1)) / (sigma2 - sigma1) + F2

      If FS > F3 And control = 0 Then

      FS = F3

      control = control + 1

      End If

   FORCEXY = FS * H * te

   NFORCEXY = FORCEXY / m

   Call WRINKLINGsystem

   sigmaS = critical

   F1 = F2

   F2 = FS

Loop
```

After establishing values F1 and F2 we can start Linear Interpolation II process. The new problem will look like Figure 4.17.

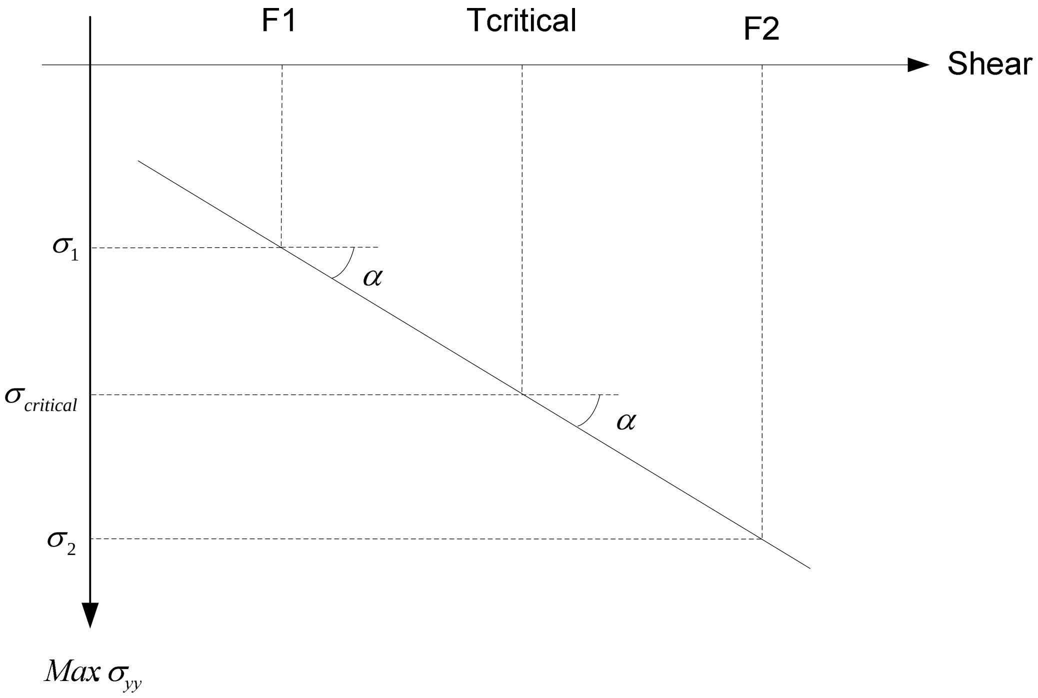


Figure 4.17 Linear Interpolation II

From Fig.4.17 $T_{critical}$ can be found as:

$$T_{critical} = \frac{(\sigma_{critical} - \sigma_1) \times (F2 - F1)}{\sigma_2 - \sigma_1} + F1 \qquad (4.8)$$

Here the aim is to approach $T_{critical}$ value by changing the values of *F1 and F2*. The flowchart for Linear Interpolation II can be seen at Figure 4.18.

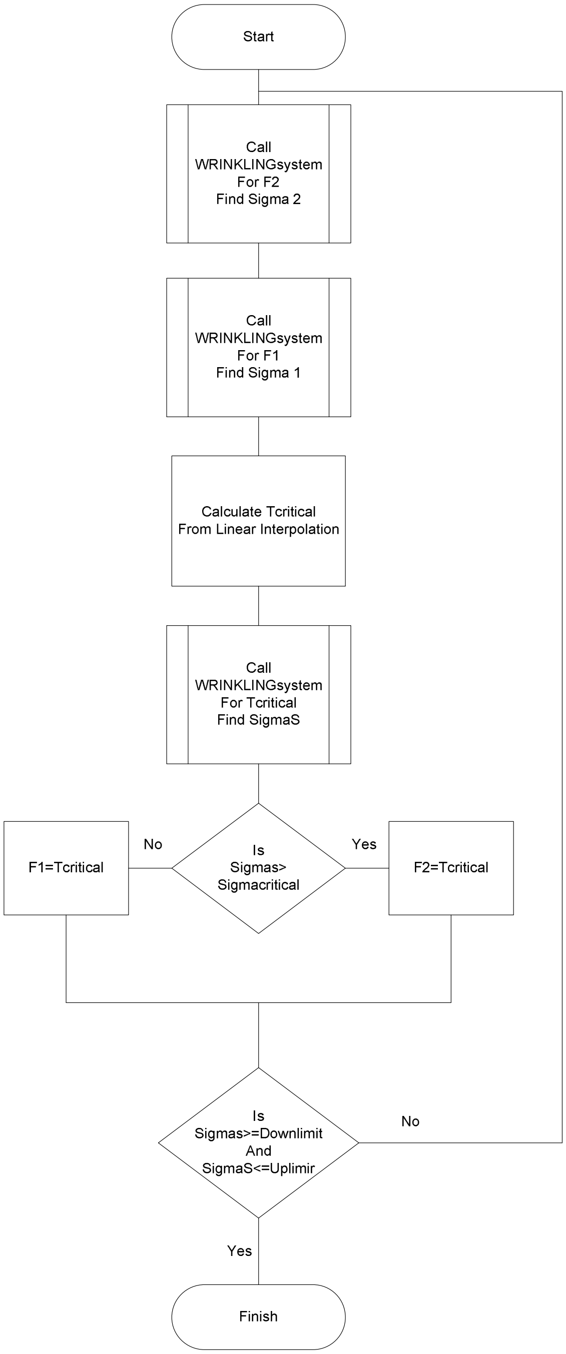


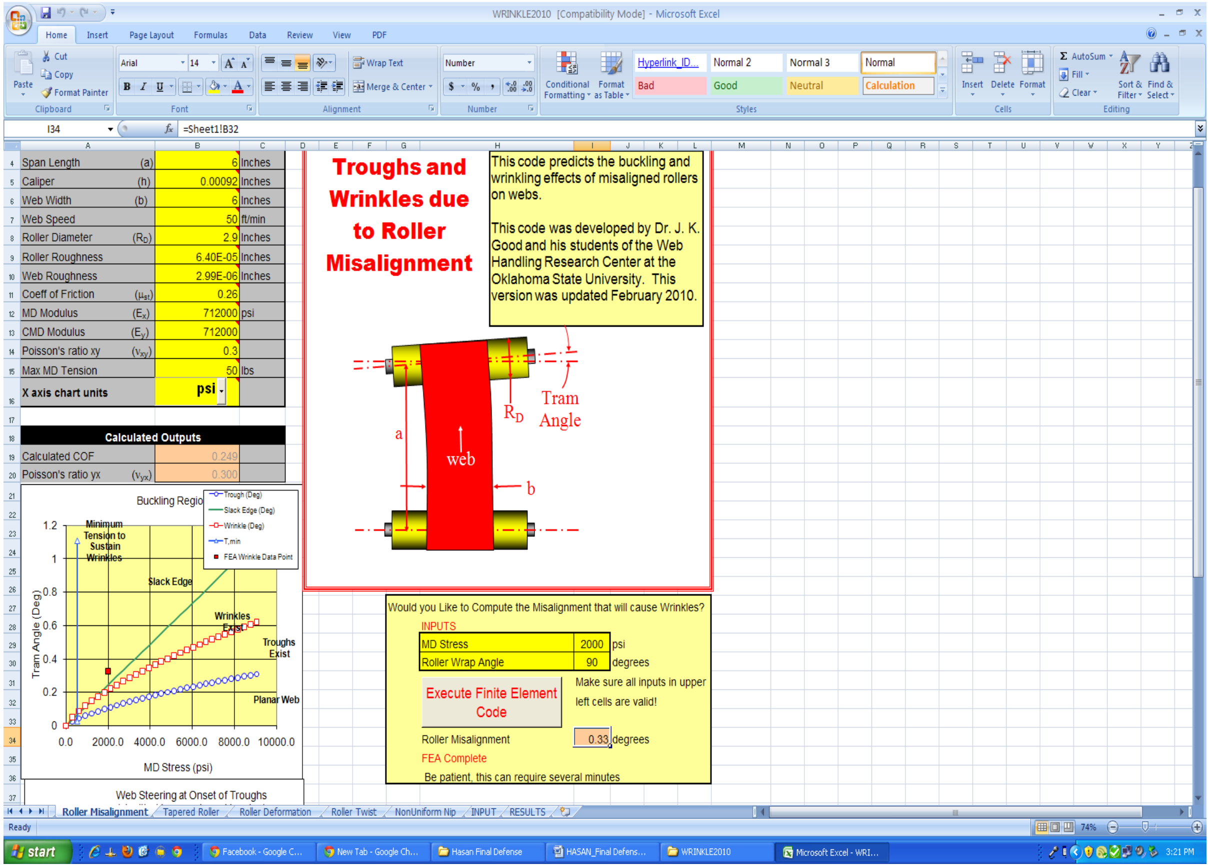
Figure 4.18 Flow Chart for Linear Interpolation II

From *Linear Interpolation I* the values F1 and F2 are known. By using Eq.4.8 $T_{critical}$ was calculated and SigmaS was found. If SigmaS is greater than Sigmacritical we replace $T_{critical}$ with F2. If not we replace $T_{critical}$ with F1. This process was continued until SigmaS is between the limits. In case of misaligned roller case the lower limit was set to 0.99*Sigmacritical and the upper limit was set to 1.01*Sigmacritical. The code for Linear Interpolation II will then be:

```
Downlimit = 0.99 * sycr
Uplimit = 1.01 * sycr
Do Until Downlimit <= sigmaS And sigmaS <= Uplimit
   FORCEXY = F2 * H * te
   NFORCEXY = FORCEXY / m
   Call WRINKLINGsystem
   sigma2 = critical
   FORCEXY = F1 * H * te
   NFORCEXY = FORCEXY / m
   Call WRINKLINGsystem
   sigma1 = critical
   FS = ((sycr - sigma1) * (F2 - F1)) / (sigma2 - sigma1) + F1
   FORCEXY = FS * H * te
   NFORCEXY = FORCEXY / m
   Call WRINKLINGsystem
   sigmaS = critical
     If sigmaS > sycr Then
               F2 = FS
      ElseIf sigmaS < sycr Then
                F1 = FS
      End If
Loop
```

Linear Interpolation I, Linear interpolation II and *WRINKLINGsystem* can be seen in the

Appendix. After automating, Excel input of the code will look like Figure 4.19.



I34 =Sheet1!B32

Span Length	(a)	6	Inches
Caliper	(h)	0.00092	Inches
Web Width	(b)	6	Inches
Web Speed		50	ft/min
Roller Diameter	(R_D)	2.9	Inches
Roller Roughness		6.40E-05	Inches
Web Roughness		2.99E-06	Inches
Coeff of Friction	(μ_{st})	0.26	
MD Modulus	(E_x)	712000	psi
CMD Modulus	(E_y)	712000	
Poisson's ratio xy	(ν_{xy})	0.3	
Max MD Tension		50	lbs
X axis chart units		**psi**	

Calculated Outputs		
Calculated COF		0.249
Poisson's ratio yx	(ν_{yx})	0.300

Troughs and Wrinkles due to Roller Misalignment

This code predicts the buckling and wrinkling effects of misaligned rollers on webs.

This code was developed by Dr. J. K. Good and his students of the Web Handling Research Center at the Oklahoma State University. This version was updated February 2010.

Would you Like to Compute the Misalignment that will cause Wrinkles?

INPUTS

MD Stress	2000	psi
Roller Wrap Angle	90	degrees

Execute Finite Element Code

Make sure all inputs in upper left cells are valid!

Roller Misalignment 0.33 degrees

FEA Complete

Be patient, this can require several minutes

Figure 4.19 Excel Input for the Automated Program

Here the user is supposed to enter web width, span length, thickness, elastic modulus, Poisson’s ratio, roller radius and the web tension. Other parameters are calculated automatically. Meshing elements are calculated with Excel equations and shear *traction xy* required to induce wrinkles is calculated within the code as mentioned above. After execution the code provides the following output: total time of execution (67 seconds), the maximum compressive stress (-266.8 psi) in the first row of elements and the roller misalignment angle (in degrees) that produced that compressive stress. For this case the sigma critical value calculated from Timoshenko buckling criteria (Eq.2.1) is -270 psi. So the interpolation scheme discussed has produced a compressive stress in the first row of elements that is very close to the critical value.

The results from the automated code are shown in Figures 4.20-22 and the results are compared with Beisel's commercial FE results and his test data.

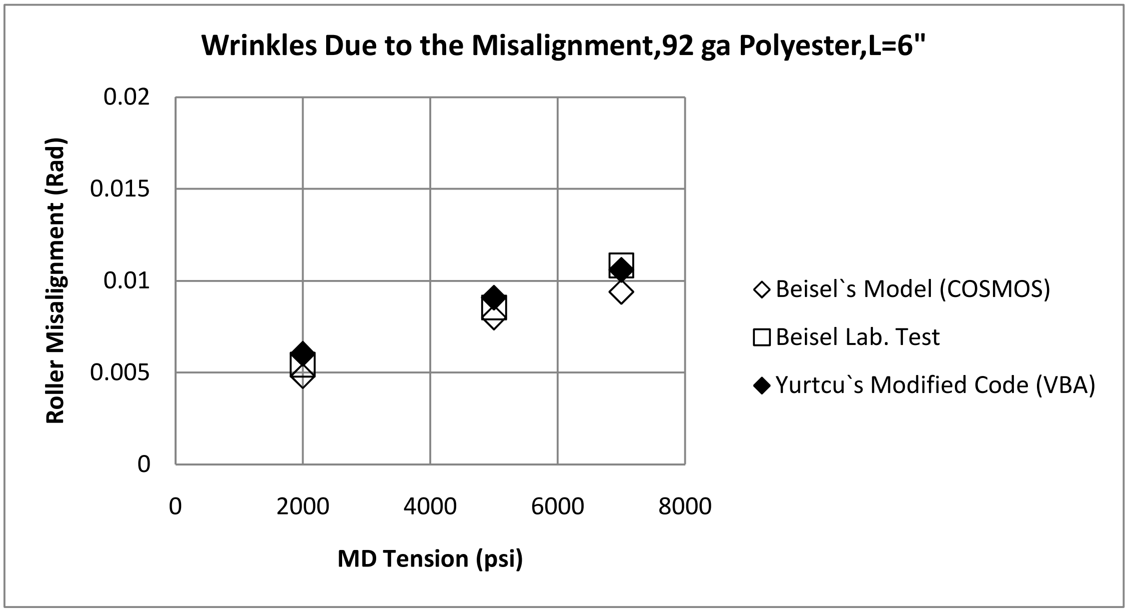


Figure 4.20 Comparison of 6'' Span Case with the Modified Code

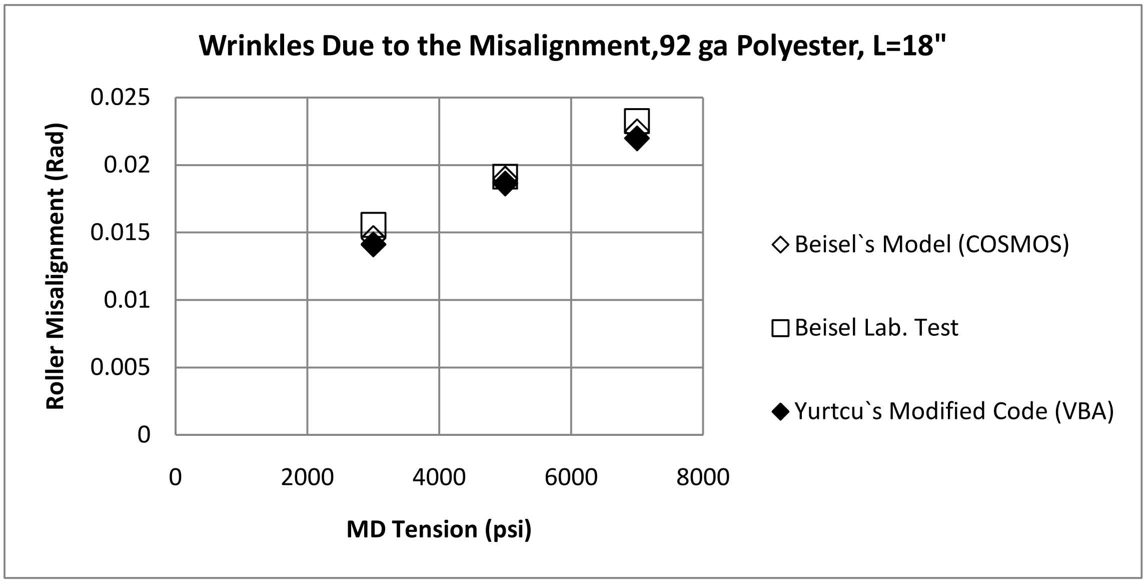


Figure 4.21 Comparison of 18'' Span Case with the Modified Code

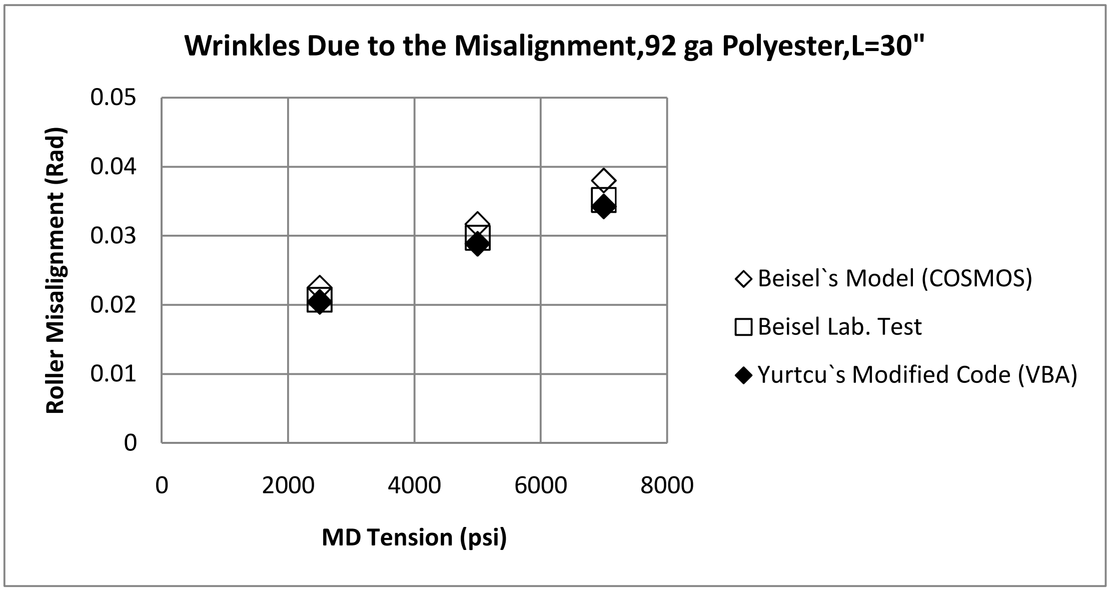


Figure 4.22 Comparison of 30’’ Span Case with the Modified Code

The execution time of the modified code is much less than the previous version. In the previous version execution time was around two to three hours and code was able to give one result for the specified shear force. For the modified code the execution time is around five to ten minutes and the modified code finds the right shear force that will buckle the web by itself. Over the parameter ranges of these examples the modified code appears to mesh the problem adequately and yields good results at all tension levels.

4.7 A New Slack Edge Criteria for a Misaligned Roller

In Figure 4.23, buckling region of a 92 gage polyester web can be seen. For this web Young’s Modulus is 725000 psi, Poisson’s ratio is 0.3, the width of web is 6’’, length of the web is 40” and the thickness is 0.00092’’. If 6000 psi MD stress applied to the system we are able to see wrinkles after the trough formation. If 2000 psi applied to the system after the formation of troughs slack edge occurs and we are not able to see wrinkles.

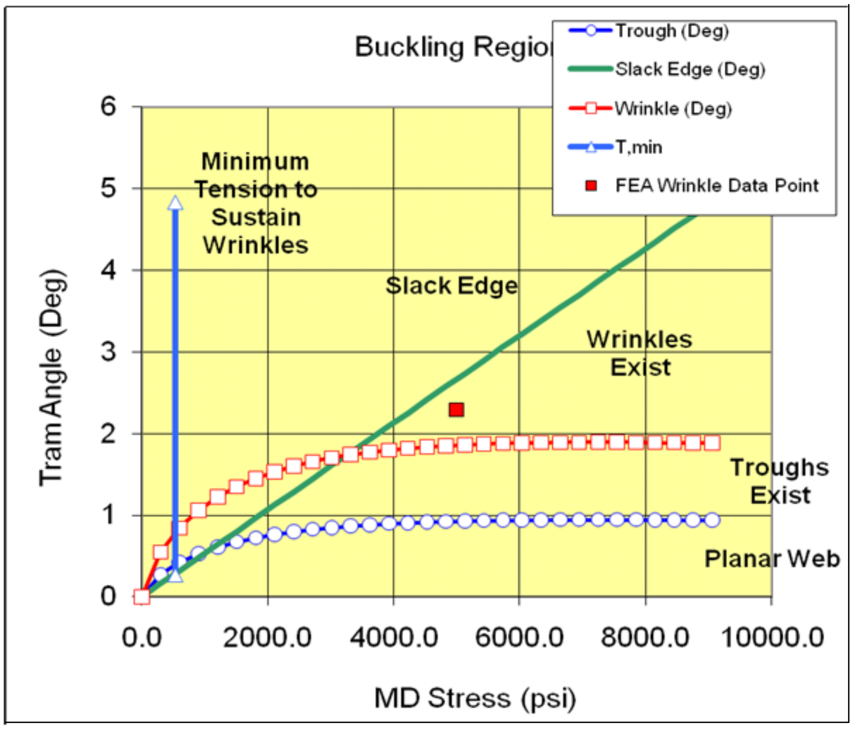


Figure 4.23 Slack Edge

The code was delivered to the Web Handling Research Center sponsors. One of the sponsors used the code with a low span (L/W) ratio. In that case W (width of the web) was five times larger than L (length of the web). With that specific web tension that was applied to the web a slack edge was supposed to occur. But Excel VBA code was came up with a result which means wrinkles was occurred after the trough formation instead of a slack edge. The case was also modeled with COSMOS and the result was very similar with Excel VBA code.

In the derivation of Eq.4.5 shear deformation was neglected. For L/W values of 0.2 it was obvious that shear deformation is important. Thus it was determined that a new slack edge criteria was needed that did account for shear deformation.

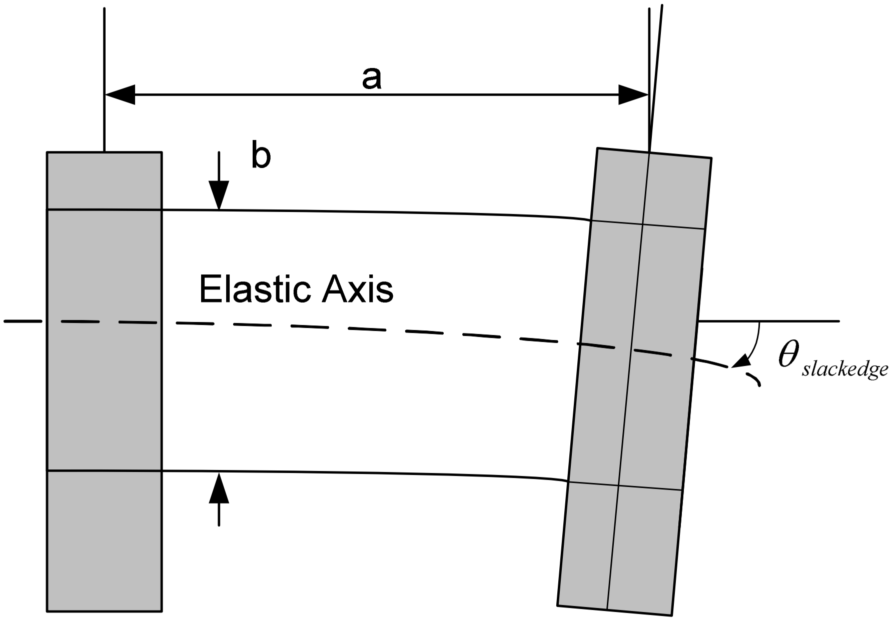


Figure 4.24 Slack Edge Criteria

The previous slack edge expression was derived based on Euler beam theory:

$$\theta_{SlackEdge} = \frac{T_x a}{b(Ehb)} \tag{4.9}$$

Here, a is length of the web, b is width of the web (Fig.4.23), h is thickness and E represents elastic modulus. To derive a new criteria the Timoshenko beam theory was used that incorporates both shear and tension stiffening.

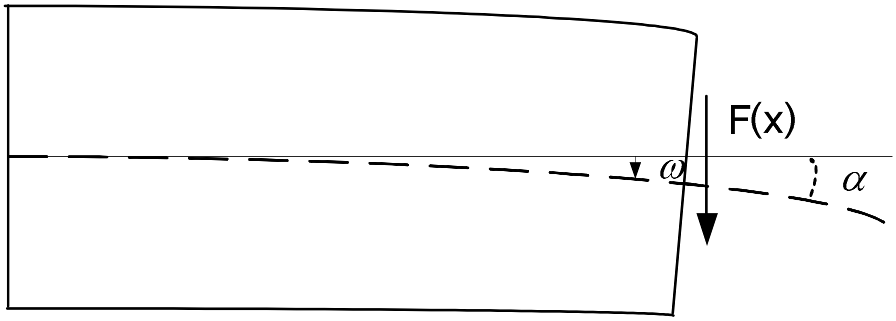


Figure 4.25 Beams with Shear

If w is deformation, the slope of the beam is α , and can be found from:

$$\alpha = w' = \frac{\partial w}{\partial x} = \theta + \gamma = \frac{\partial w_b}{\partial x} + \frac{\partial w_s}{\partial x} \tag{4.10}$$

Here θ is related to moment and γ is related to shear.

$$\frac{\partial \theta}{\partial x} = \frac{M(x)}{EI} \quad and \quad \gamma = \frac{F(x)}{GA_s} \tag{4.11}$$

The moment can be calculated from Eq.4.10 by using Eq.4.11

$$Moment = w'' = \theta,_x + \gamma,_x = \frac{M(x)}{EI} + \frac{1}{GA_s}.\frac{\partial F}{\partial x} \tag{4.12}$$

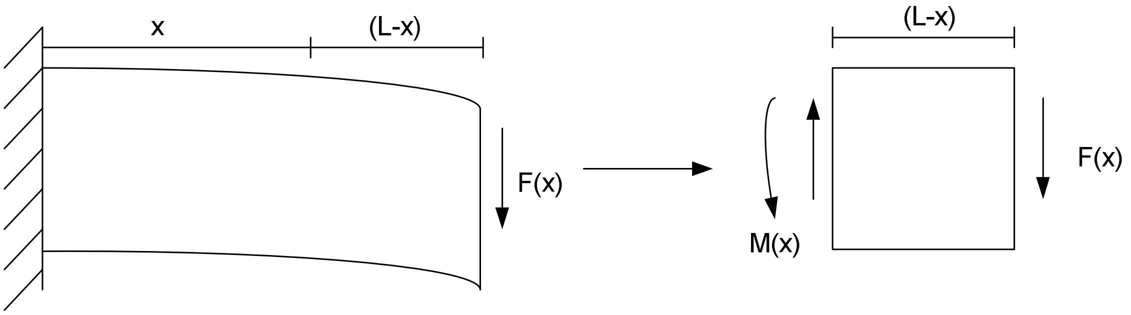


Figure 4.26 Moment in a Beam

At the boundary $x = 0$,

$$\left.\frac{\partial w}{\partial x}\right|_{x=0} = \left.\gamma\right|_{x=0} = \frac{F}{GA_S} \tag{4.13}$$

The moment $M(x) = F(L-x) \quad and \quad F(x) = F$, and substituting into Eq.4.12 yields

$$w'' = \frac{F(L-x)}{EI} + 0 \tag{4.14}$$

Integrating yields

$$w' = \frac{F}{EI}(Lx - \frac{x^2}{2}) + C_1 \tag{4.15}$$

From Eq.4.13 $\left.\gamma\right|_{x=0} = \frac{F}{GA_S} = C_1$ so Eq.4.15 becomes

$$w' = \frac{F}{EI}(Lx - \frac{x^2}{2}) + \frac{F}{GA_S} \tag{4.16}$$

Finally integration of Eq.4.16 yields,

$$w = \frac{F}{EI}(Lx^2 - \frac{x^3}{6}) + \frac{Fx}{GA_S} + C_2 \tag{4.17}$$

It is known that $w\big|_{x=0} = 0$. So C_2 becomes *0*. We seek $\theta_{critical}$ at *x=L* .From Eq.4.16,

$$w'\big|_{x=L} = \theta_{critical} = F(\frac{L^2}{2EI} + \frac{1}{GA_S}) \tag{4.18}$$

Solving for the steering force *F* yields:

$$F = \frac{\theta_{critical}}{(\frac{L^2}{2EI} + \frac{1}{GA_S})} \tag{4.19}$$

This allows us to calculate the moment at *x=0*,

$$M = F.L = \left(\frac{\theta_{critical}}{(\frac{L^2}{2EI} + \frac{1}{GA_S})} \right) L \tag{4.20}$$

The beam bending stress from moment can now be calculated as,

$$\sigma^B = \frac{M}{I} y_{\max} = \left(\frac{\theta_{critical}}{(\frac{L^2}{2EI} + \frac{1}{GA_S})} \right) L \frac{w}{2} \frac{1}{I} \tag{4.21}$$

The normal stress due to web tension can be calculated from,

$$\sigma^T = \frac{T}{wb} \tag{4.22}$$

In taut behavior $\sigma^T > \sigma^B$ if $\sigma^T = \sigma^B$ the slack case occurs. So if $\sigma^T = \sigma^B$,

$$\left(\frac{\theta_{critical}}{(\frac{L^2}{2EI}+\frac{1}{GA_S})} \right) \frac{L}{2}\frac{w}{I} = \frac{T}{wb} \tag{4.23}$$

From here $\theta_{critical}$ can be calculated as,

$$\theta_{critical} = \frac{T}{Lw^2b}(\frac{L^2}{E}+\frac{2I}{GA_S}) \tag{4.24}$$

For a rectangular cross section $I = \frac{bw^3}{12}, G = \frac{E}{2(1+v)}$ *and* $A_S = \frac{5}{6}bw$. Substituting into Eq.4.24 yields,

$$\theta_{critical} = \frac{TL}{w(Ebw)}(1+\frac{w^2}{L^2}\frac{2}{5}(1+v)) \tag{4.25}$$

In the calculations beginning from Eq4.10, L for span length, w for span width and t for thickness were used. If we replace them with a for length, b for width and h for thickness, it will be found,

$$\theta_{critical} = \frac{Ta}{b(Ehb)}(1+(\frac{b}{a})^2\frac{2}{5}(1+v)) \tag{4.26}$$

If this result is compared with Eq. 4.9 it will be found that

$$\frac{Ta}{b(Ehb)}((\frac{b}{a})^2\frac{2}{5}(1+v)) \tag{4.27}$$

is the effect of shear force with Timoshenko beam theory. If $v \approx 0.3$ then the term is

$\frac{2}{5}(1+v) \approx \frac{1}{2}$.

So Eq.4.26 can be simplified as,

$$\theta_{critical} = \frac{Ta}{b(Ehb)}(1+\frac{1}{2}(\frac{b}{a})^2) \tag{4.28}$$

If *W >L* (*b>a*) then shear effects are becoming significant. For instance if *W=5L* as it was in the case that we came up with earlier then,

$$\frac{\theta_{criticalTimoshenko}}{\theta_{criticalEuler}} \cong \frac{\frac{Ta}{b(Ehb)}(1+\frac{1}{2}25)}{\frac{Ta}{b(Ehb)}} \cong 13 \tag{4.29}$$

Eq. 4.26 supersedes Eq.4.9 and can be used for both long and short spans.

5. CHAPTER V

5.1 TAPERED ROLLERS

In this chapter, a brief definition of tapered rollers, previous work modeling with tapered roller cases with commercial finite element codes will be reviewed. Then a new model based on using Excel VBA will be given.

5.2 Description of Tapered Rollers and Beisel's Method for Modeling Wrinkles Due to the Tapered Rollers

A tapered roller is defined as a roller with a linearly varying radius across its width [12]. Tapered rollers are commonly seen in the web handling industry. The process of roller manufacture will almost certainly result in rollers with a slight taper. These tapered rollers are produced unintentionally; the use of tapered rollers in web lines may result in web damage. To solve this problem, machining techniques that involve feedback can be used, but this will be costly. Therefore, knowing the amount of taper that will not result in harm to the web process is beneficial for the industry.

In Figure 5.1 a web approaching the tapered roller is shown. In the figure, the taper is somewhat over-emphasized so that it can be seen.

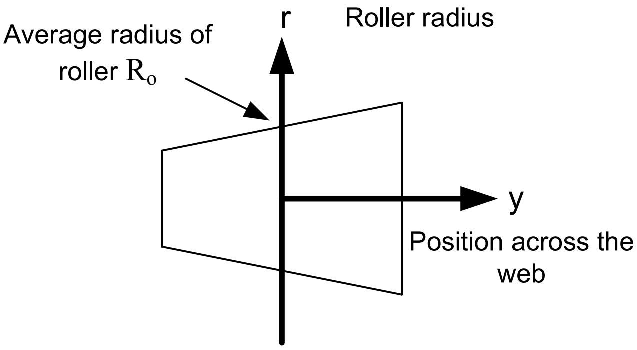


Figure 5.1 Tapered Roller Profile

The radius of the roller at any point across the width is:

$$r(y) = my + R_o \tag{5.1}$$

Here *m* is the slope of the roller. The velocity across the roller width is:

$$V(y) = r(y)\,\omega = (my + R_o)\omega \tag{5.2}$$

Here ω is the angular velocity of the roller. The average web velocity can be found by using average roller radius as:

$$V_{avg} = R_o \omega \tag{5.3}$$

Variation in the velocity across the web width will cause a stress and strain upon the web:

$$\varepsilon_{md}(y) = \frac{V(y) - V_{avg}}{V_{avg}} = \frac{my}{R_o} \quad and \quad \sigma(y) = E\varepsilon_{md}(y) = \frac{Emy}{R_o} \tag{5.4}$$

These equations assume the web and roller achieve the same velocity at contact. The variation in stress across the web width causes a steering moment on the web:

$$M_j = \int_{\frac{-b}{2}}^{\frac{b}{2}} -\sigma(y) hy dy = \int_{\frac{-b}{2}}^{\frac{b}{2}} \frac{-Emhy^2}{R_o} dy = \frac{-mEhb^3}{12R_o} \tag{5.5}$$

where *m* is roller taper, R_0 is roller radius, *E* is Young's Modulus, *h* is thickness and *b* is web width.

Beisel [12] determined that wrinkle formation due to a downstream tapered roller is similar to the wrinkle formation due to a misaligned roller. As in the misaligned roller case, he used Timoshenko buckling criteria (Eq.2.1) to decide whether the web on the tapered roller will wrinkle or not. His wrinkle model for the tapered roller is shown in the Figure 5.2.

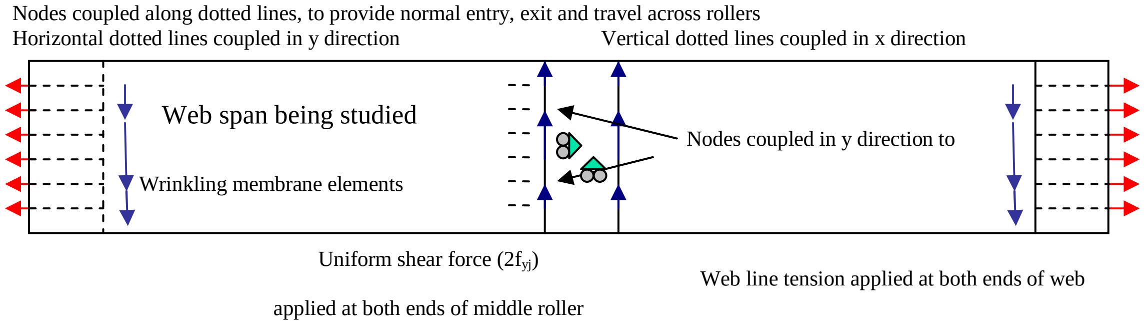

Nodes coupled along dotted lines, to provide normal entry, exit and travel across rollers
Horizontal dotted lines coupled in y direction
Vertical dotted lines coupled in x direction
Web span being studied
Wrinkling membrane elements
Nodes coupled in y direction to
Uniform shear force (2f$_{yj}$)
applied at both ends of middle roller
Web line tension applied at both ends of web

Figure 5.2 Beisel Tapered Roller Model [12]

As in the misaligned case, he divided the model into five panels. The first panel is the web on the upstream roller. The second panel represents the web span being studied. The third panel represents the tapered roller. The fourth panel represents the web between the tapered roller and third roller, and the fifth panel represents the third roller. He locked the nodes in the first panel along horizontal lines. Each node on the line must move with the same y displacement as the rest of the nodes on the same line. This allows for Poisson contraction due to web line tension but requires a zero slope at the beginning of the troughed web span. He also locked the nodes at the exit of the upstream roller in the x direction along a vertical line to simulate the roller gripping the web in a no-slip condition. He used wrinkling membrane elements in the second panel. Along the right edge of the second panel he locked the nodes along horizontal lines in the *y* direction for a very short distance. He did this to ensure the normal entry of the web to the tapered roller. He modeled the right hand side of the model to enforce the boundary conditions.

He executed the model by first applying web line tension and then increasing shear force until the compressive stress across the first row of elastic nodes on the elastic wrinkling membrane element boundary reached the critical value predicted by Eq. 2.1. Then he calculated the moment associated with the last row of span elements by using Eq. 5.5. He calculated the critical taper that would induce wrinkles.

He obtained the experimental results for the onset of wrinkles due to a downstream tapered roller. He compared his experimental results and model results for two materials. In the following chapters, we will compare our model results with his experimental results and with his model results.

5.3 A New Algorithm for Predicting Wrinkles Due to the Tapered Rollers

As mentioned earlier, shear wrinkles can occur due to tapered rollers. Beisel [12] successfully modeled the tapered roller case by using the commercial finite element code COSMOS. As in the misaligned case he used wrinkling membrane elements while modeling the elements between the upstream roller and the tapered roller. The goal of our new algorithm related to the tapered roller is to codify and automate the analysis that Beisel did with the commercial finite element code COSMOS. Similar to the misaligned roller case a finite element code that calculates critical taper will be developed in Excel VBA by using wrinkling membrane elements. The boundary conditions will be similar to Beisel's model boundary conditions.

The problem is modeled material on the upstream roller, the upstream span, the web on the tapered roller, the downstream span, and finally material on the downstream roller. The system that is modeled is shown in Fig.5.3.

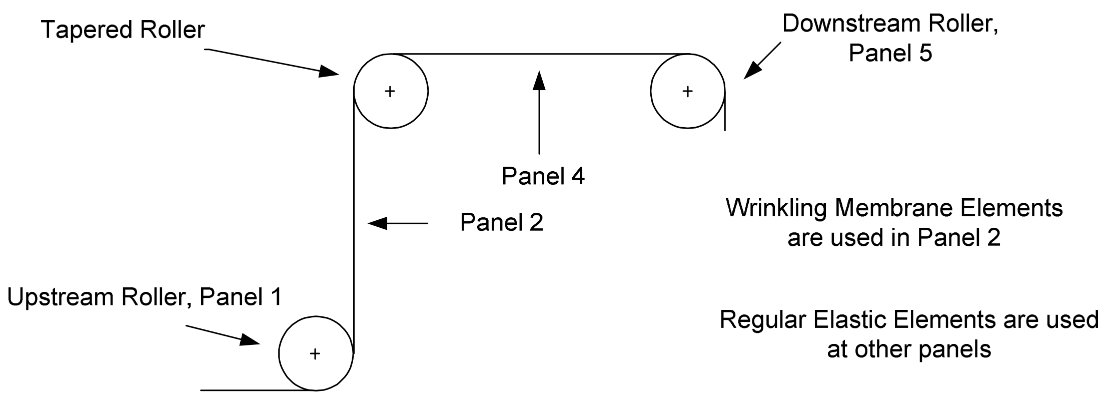


Figure 5.3 The Figure for Tapered Roller

The system with a tapered roller shown in Figure 5.3 is modeled like the Figure 5.4 shown below.

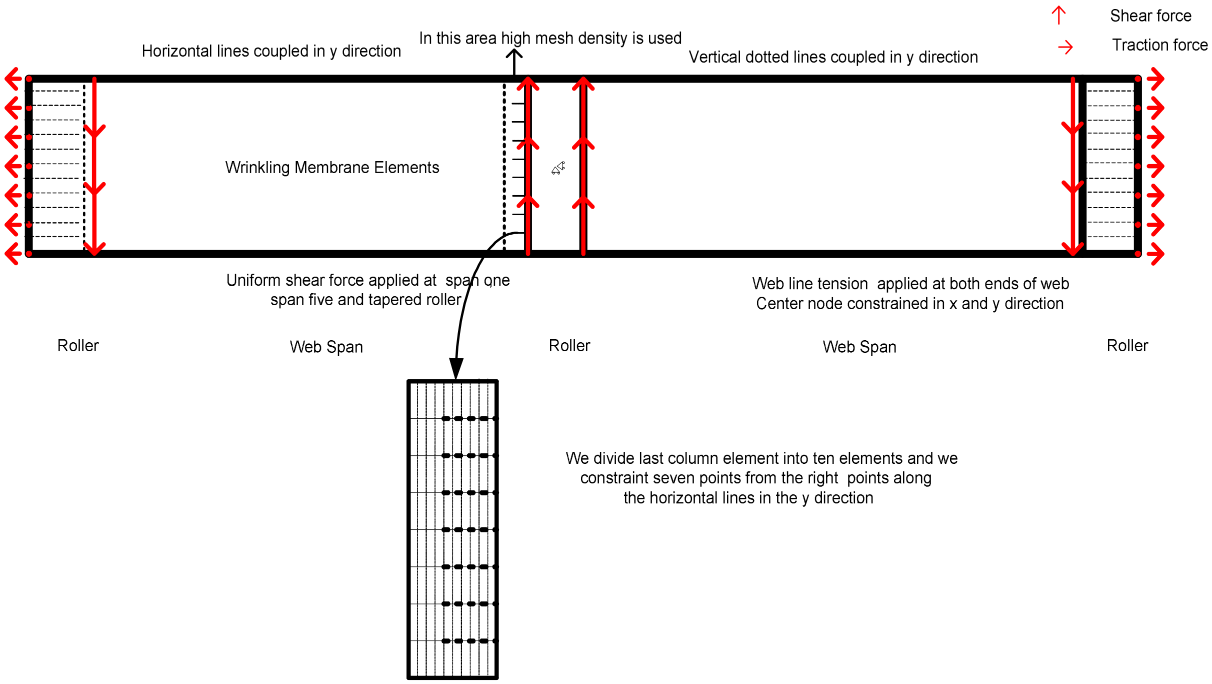
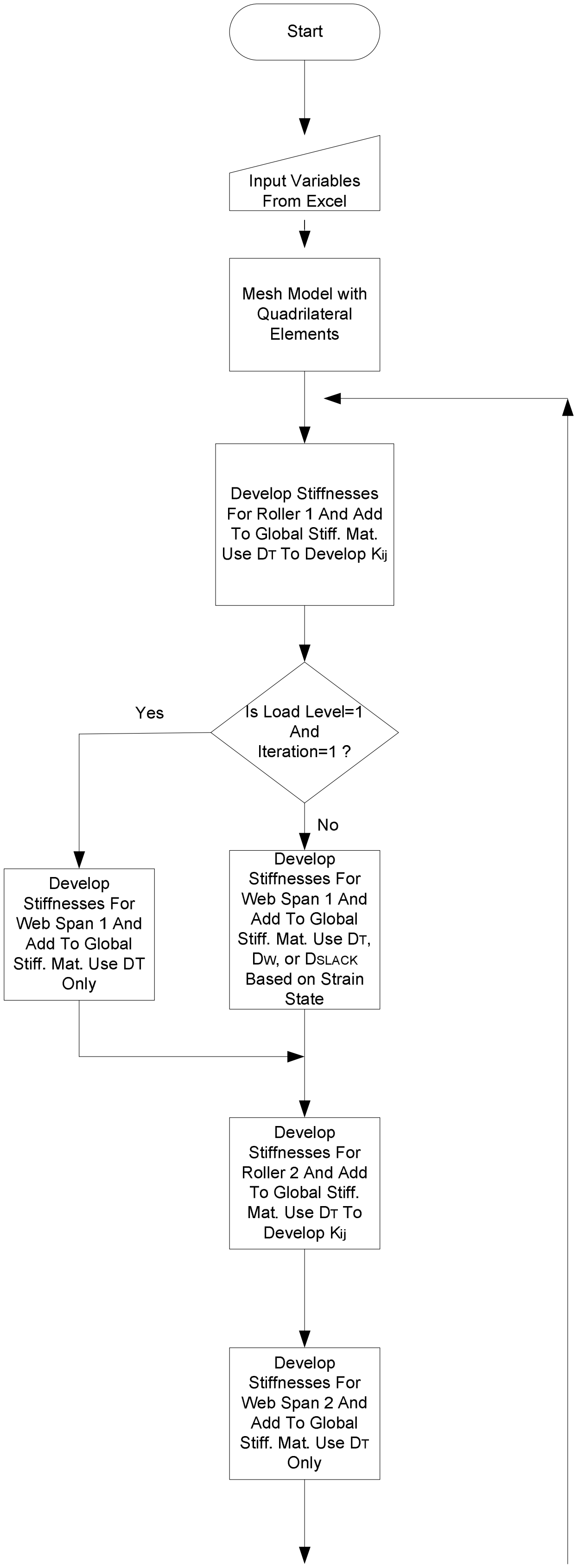

↑ Shear force
→ Traction force
Horizontal lines coupled in y direction
In this area high mesh density is used
Vertical dotted lines coupled in y direction
Wrinkling Membrane Elements
Uniform shear force applied at span one
span five and tapered roller
Web line tension applied at both ends of web
Center node constrained in x and y direction
Roller
Web Span
Roller
Web Span
Roller
We divide last column element into ten elements and we
constraint seven points from the right points along
the horizontal lines in the y direction

Figure 5.4 Tapered Roller FE Wrinkle Model

The new model is similar to the misaligned roller case. The model was divided into five panels. The first panel was the panel on the upstream roller. The second panel was representing the web between upstream roller and tapered roller. Here wrinkling membrane elements were used; in the rest of the model elastic elements were used. The third panel represents the tapered roller. The fourth and fifth panels help to achieve the no moment boundary condition at the tapered roller. Boundary conditions and loads were enforced which Beisel found to be appropriate for the tapered roller. The center is pinned to prevent rigid body motions. Multi-point constraints were applied on each row of nodes on the entering and exiting rollers. Along the right edge of the second panel (span one), Beisel locked the nodes along horizontal lines in the y direction for a very short distance to ensure normal entry. In contrast to his method, high mesh was used at the last row of the span one element (ten elements were used in the last row elements) and six points along the horizontal lines were locked in the y direction. This was decided after trying many ways to achieve normal entry to the tapered roller. The flowchart for the program is shown in Fig.5.5. It is similar to the misaligned roller flowchart so it will not explain it here in detail.


Start
Input Variables From Excel
Mesh Model with Quadrilateral Elements
Develop Stiffnesses For Roller 1 And Add To Global Stiff. Mat. Use DT To Develop Kij
Is Load Level=1 And Iteration=1 ?
Yes
No
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT Only
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT, DW, or DSLACK Based on Strain State
Develop Stiffnesses For Roller 2 And Add To Global Stiff. Mat. Use DT To Develop Kij
Develop Stiffnesses For Web Span 2 And Add To Global Stiff. Mat. Use DT Only

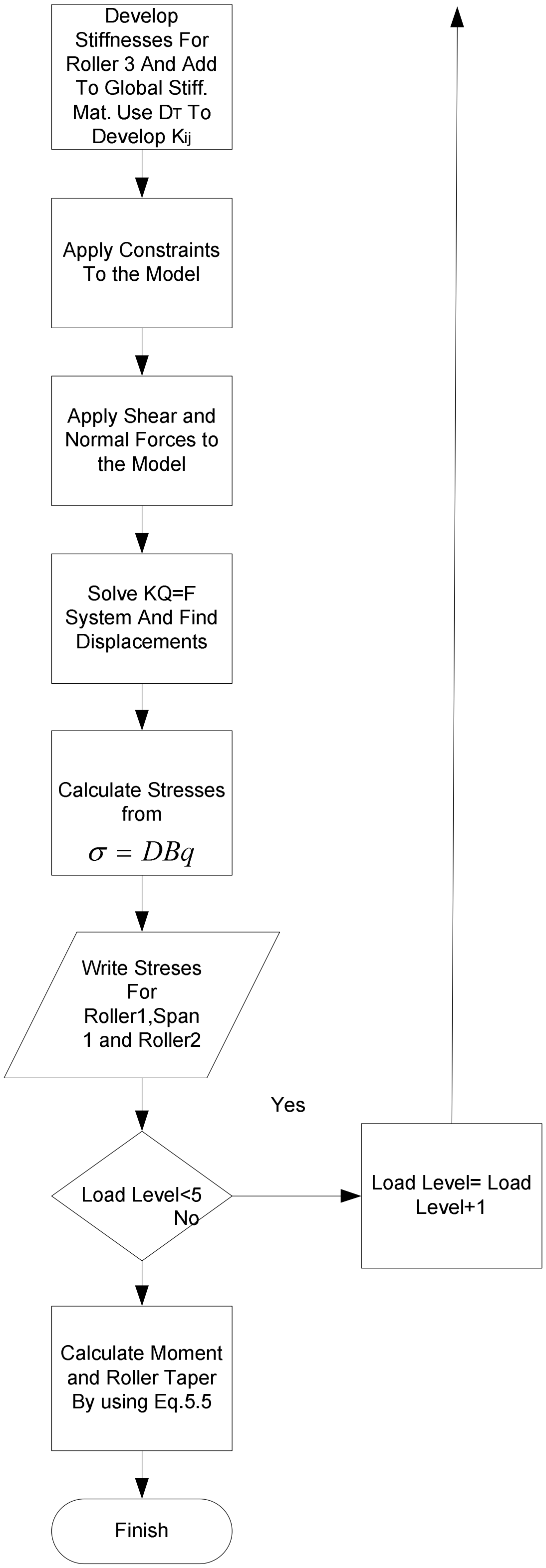

Develop Stiffnesses For Roller 3 And Add To Global Stiff. Mat. Use D_T To Develop K_{ij}
Apply Constraints To the Model
Apply Shear and Normal Forces to the Model
Solve KQ=F System And Find Displacements
Calculate Stresses from $\sigma = DBq$
Write Streses For Roller1,Span 1 and Roller2
Yes
Load Level<5
No
Load Level= Load Level+1
Calculate Moment and Roller Taper By using Eq.5.5
Finish

Figure 5.5 The Flow Chart for Tapered Roller

In the tapered roller flowchart the forces are applied in five time steps. After five steps from the first row of elements the moment is calculated and by using Eq. 5.5 critical taper is also calculated. Assume that the stresses of the first row of elements at tapered roller look like Fig.5.6. Here, one half of the width is demonstrated.

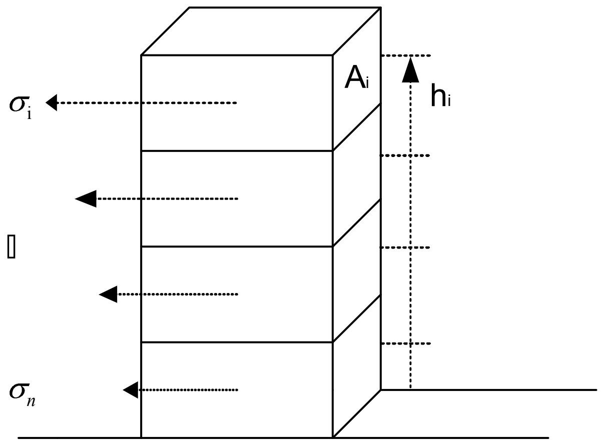


Figure 5.6 Calculating Moment and Critical Taper

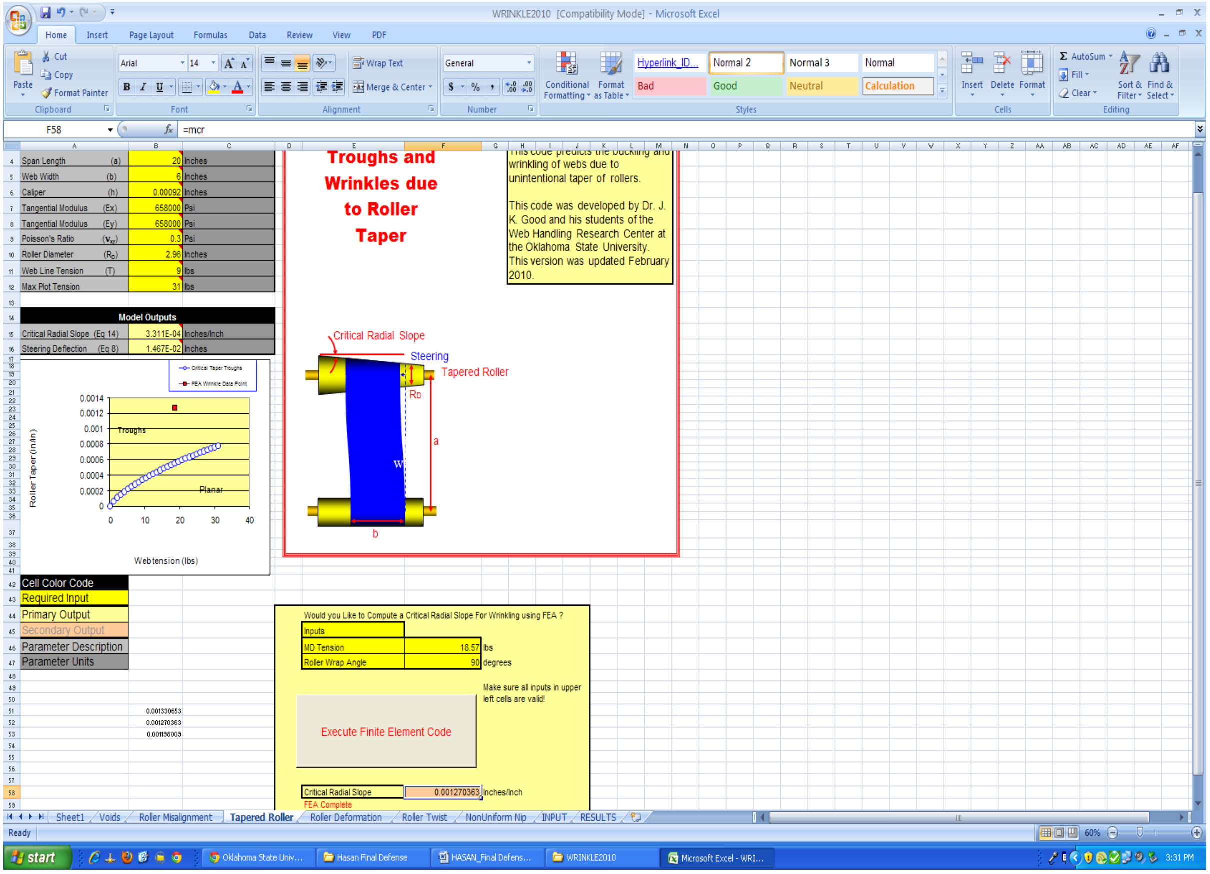
Moment is $M = F\,h$ and $\sigma_x = F / A$. From these two equations the moment of an element can be calculated from $M = \sum_{i=1}^{n} \sigma_i A_i\, h_i$. If all moments are added the total moment of the first row of the elements of the tapered roller is found. If this value put into Eq .5.5 the critical taper for that specific case is found. At the end of the code, the code calculates total moment and taper for that case.

The name "*SystemWrinkler*" will be applied to the part of the flow chart (Fig.5.5) that begins after "Mesh Model with Quadrilateral Elements" and continues to "Load Level <5" .This term will be used in the following chapter.

5.4 Automating the Code

Similar to the misaligned roller case, it was aimed for the user to enter only certain parameters such as web width, roller wrap dimension, span length, web thickness, elastic modulus, Poisson's ratio, and web tension, and run the code. The attempts to automate the code began with trying to automate the mesh. After running many cases, it was determined that reasonable results were achieved by using three elements per dimension for the web width, one element for every dimension of the web spans and the integer part of four times of the roll dimension for rollers. Excel equations were used to set these values. The meshing procedure and convergence check will be addressed more detailed at Chapter VIII.

The flowcharts for Linear Interpolations I and II are used for automating the tapered roller case. Similar methods to the misaligned case were applied. T_{slack} and $T_{slack}/2$ (Eq.4.5) were used as a starting point for Linear Interpolation I. In Fig.4.16 and Fig.4.18, if we use the term *SystemWrinkler* instead of *WRINKLINGsystem*, the way the new flow chart works can be explained. After automating our tapered roller code the Excel input of the code will look like Figure 5.7.



WRINKLE2010 [Compatibility Mode] - Microsoft Excel

F58 =mcr

	A		B	C
4	Span Length	(a)	20	Inches
5	Web Width	(b)	6	Inches
6	Caliper	(h)	0.00092	Inches
7	Tangential Modulus	(Ex)	658000	Psi
8	Tangential Modulus	(Ey)	658000	Psi
9	Poisson's Ratio	(v_{xy})	0.3	Psi
10	Roller Diameter	(R_D)	2.96	Inches
11	Web Line Tension	(T)	9	lbs
12	Max Plot Tension		31	lbs
13				
14	Model Outputs			
15	Critical Radial Slope	(Eq 14)	3.311E-04	Inches/Inch
16	Steering Deflection	(Eq 8)	1.467E-02	Inches

42	Cell Color Code
43	Required Input
44	Primary Output
45	Secondary Output
46	Parameter Description
47	Parameter Units

51	0.001330653
52	0.001270363
53	0.001198009

Troughs and Wrinkles due to Roller Taper

This code predicts the buckling and wrinkling of webs due to unintentional taper of rollers.

This code was developed by Dr. J. K. Good and his students of the Web Handling Research Center at the Oklahoma State University. This version was updated February 2010.

Would you Like to Compute a Critical Radial Slope For Wrinkling using FEA ?

Inputs		
MD Tension	18.57	lbs
Roller Wrap Angle	90	degrees

Make sure all inputs in upper left cells are valid!

Execute Finite Element Code

Critical Radial Slope	0.001270363	Inches/Inch

FEA Complete

Figure 5.7 Excel Input of Tapered Roller

Here the user is supposed to enter web width, span dimension, thickness, elastic modulus, Poisson’s ratio, roller radius and the web tension. After entering these parameters with the help of the Excel equations the mesh parameters, one fourth of the circumferences of the rollers (L1 roll dimension in Fig.5.7) and the stress calculated from Timoshenko’s buckling criteria (Sigma Critical in Fig.5.7 that is found from Eq. 2.1) can be calculated. After clicking the *EXECUTE* button, the code runs Linear Interpolation I and Linear Interpolation II. As a result of using the code, the maximum stress at the first row of elements (MAX sigma y), total moment at the first row of elements (Moment), and the critical taper (mcr) that will result in a wrinkle for that specific element for that specific case are given as an output.

Figure 5.7 shows that for that element the stress calculated from Timoshenko's buckling criteria is -247 psi. After linear interpolations, the code finds a maximum stress of -245 psi, which is very close to the Timoshenko buckling criteria. The critical taper for the case shown in Fig. 5.7 is found to be 0.00127 (in/in). The code calculated this case within six minutes.

5.5 Comparison of Results with Previous Works for the Tapered Roller

Beisel [12] performed experiments and obtained data and compared his experimental results with his model. The results from the Excel VBA code will be compared with his FE model results and his experimental results.

The first web he tested was a 92 gage (0.00092”) opaque polyester with a Young's Modulus of E_x = 712000 psi, a Poisson's Ratio of ν = 0.3, and a width of W = 6”. The nominal radius of his tapered roller was R_o = 1.49”. The compressive stress required by Eq.2.1 was approximately -265 psi for this case. Comparison of our results with his FE model and his experimental results are given below.

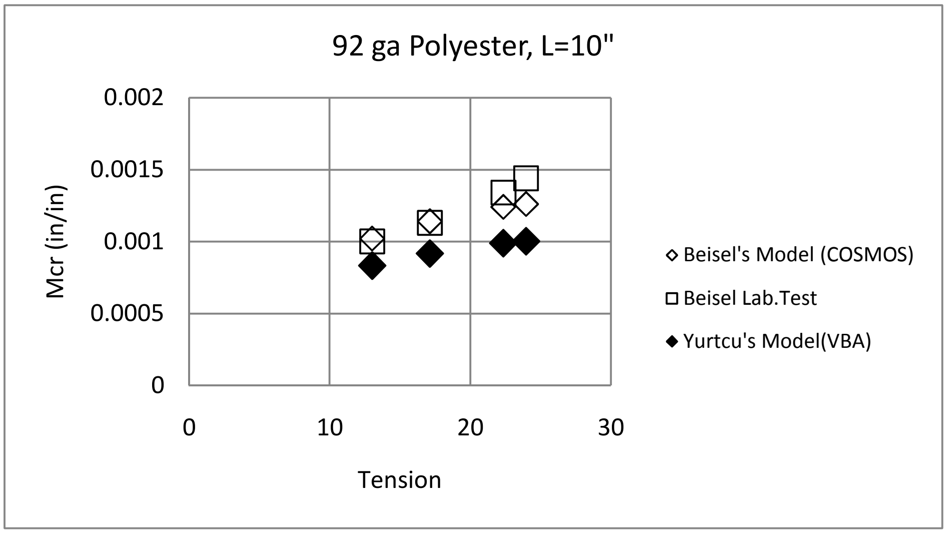


Figure 5.8 Wrinkles Due to Taper, 92 ga Polyester, L=10”

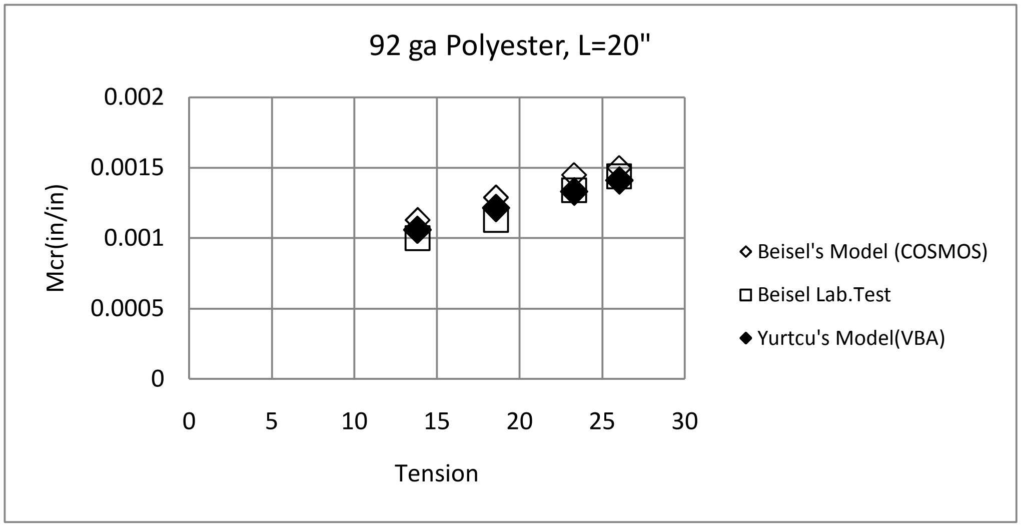


Figure 5.9 Wrinkles Due to Taper, 92 ga Polyester, L=20”

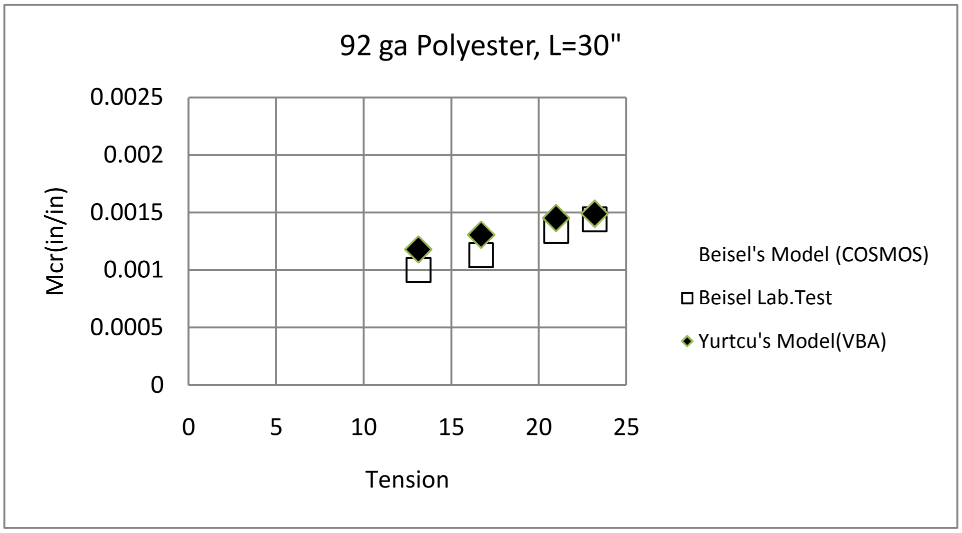


Figure 5.10 Wrinkles Due to Taper, 92 ga Polyester, L=30”

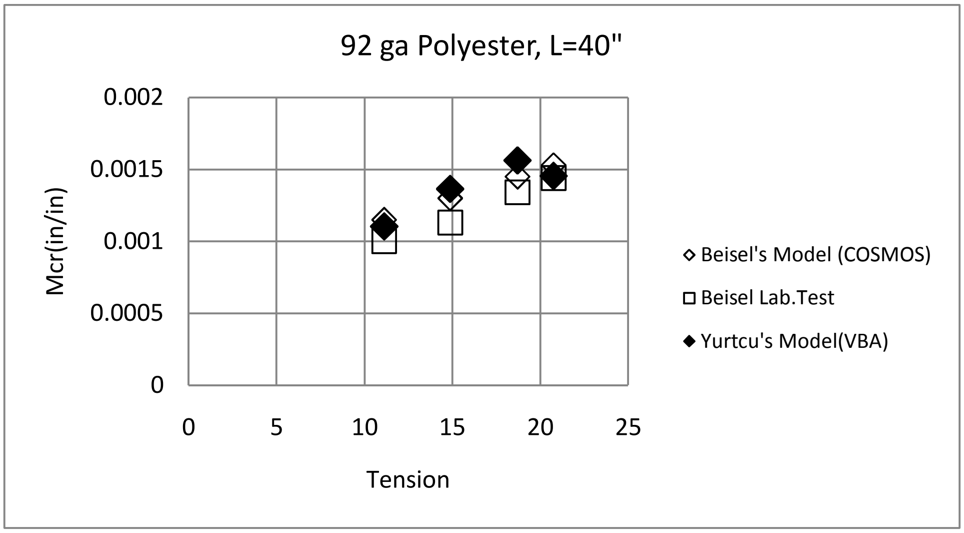


Figure 5.11 Wrinkles Due to Taper, 92 ga Polyester, L=40”

Figure 5.8-11 shows a good agreement between our model results and his model results and his experimental results. For the 10” case some drift from the data can be observed. For longer spans (20”, 30” and 40”) our model tracks correctly with changes in span length.

He also compared his model results with other test results. This was a relatively thin web, a 56 ga Polyester, with a Young’s Modulus of E_x = 658000 psi, an assumed Poisson’s ratio of ν = 0.3, and a width of W = 6”. The critical stress calculated from Eq.2.1 is -150 psi. The results from the Excel VBA code will be compared with both his model results and his experimental results below.

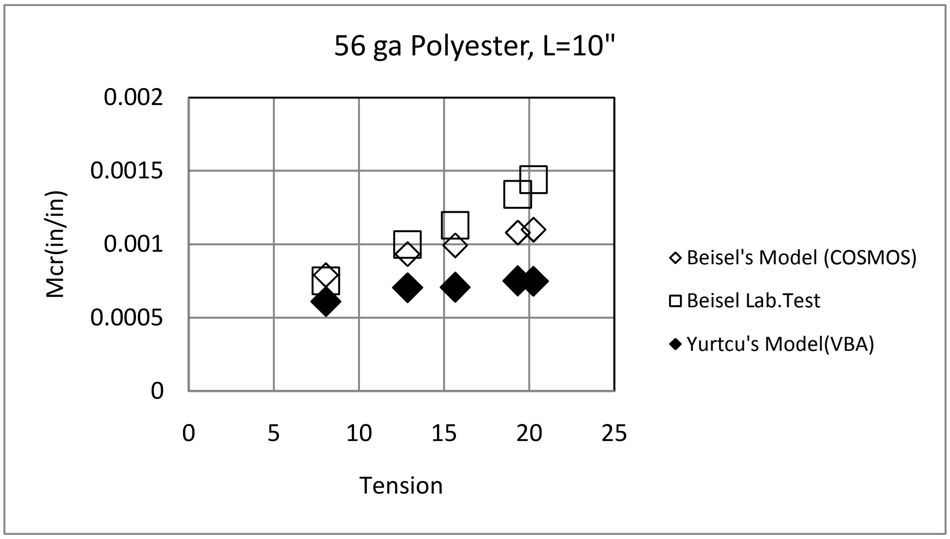


Figure 5.12 Wrinkles Due to Taper, 56 ga Polyester, L=10”

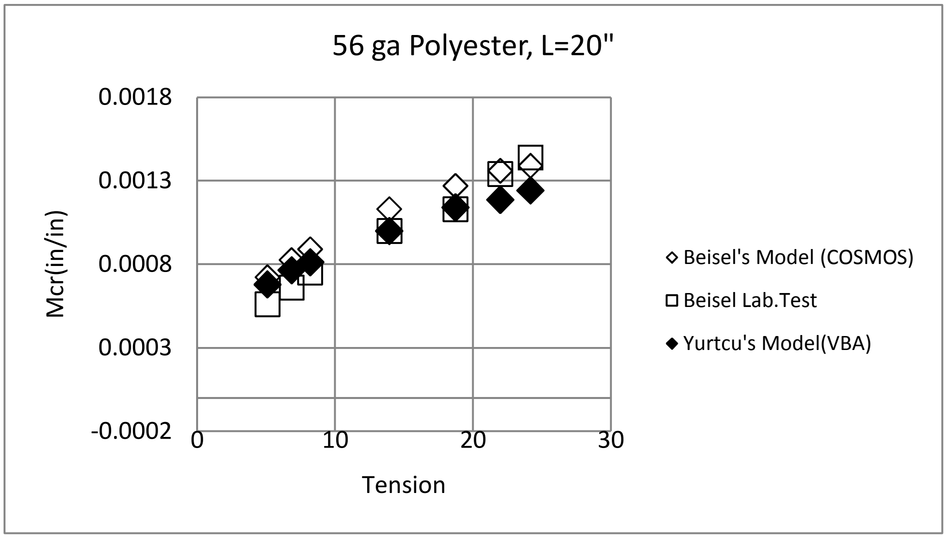


Figure 5.13 Wrinkles Due to Taper, 56 ga Polyester, L=20”

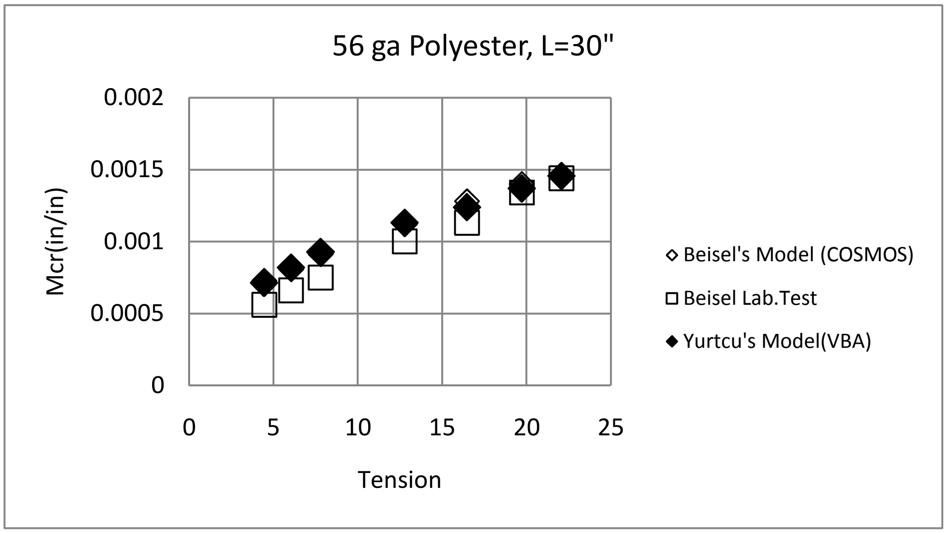


Figure 5.14 Wrinkles Due to Taper, 56 ga Polyester, L=30”

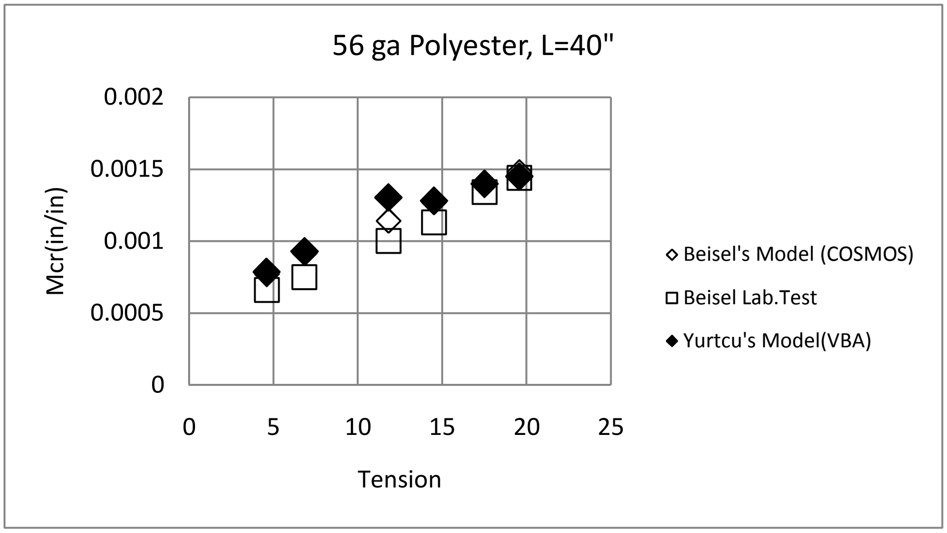


Figure 5.15 Wrinkles Due to Taper, 56 ga Polyester, L=40”

Analysis of Fig.5.8 and Fig.5.15 may suggest the limitations of the model for use in short web spans with high web tensions. In longer spans tested (20”, 30” and 40”) our model follows the experimental values. It can be said that for the ranges tested our model provides acceptable results in all cases.

6. CHAPTER VI

6.1 HOLES IN WEBS

In this chapter, the previous work done by Mallya [15] concerning holes in webs and his modeling using the commercial finite element code COSMOS will be reviewed. Then a new model based on using Excel VBA will be described.

6.2 Mallya's Method for Modeling Wrinkles Due to Circular Discontinuity

Converting processes often require that shapes be cut in webs. These webs must be handled in process machines without wrinkles. Some webs may have holes of various shapes intentionally cut into them as a part of the manufacturing or converting processes. Mallya [15] studied how these holes affect the web's propensity to wrinkle. In his experiments, he bored a hole in roll of 79 gage thick polyester film 12 inches wide with a ½ inch diameter bit in the center of a roll. He unwound the roll, and transported the web through a test section in a span of 28 inch long over rollers with a radius of 2 inches. He observed that troughs formed around the hole region and that wrinkles formed on the downstream roller. He also observed that the level of web tension affected the distance between the hole and the downstream roller when the wrinkles first appeared on the downstream roller. Troughs and wrinkles due to a circular hole can be seen in Figure 6.1. As seen in the figure, two wrinkles would form at both sides of the hole.

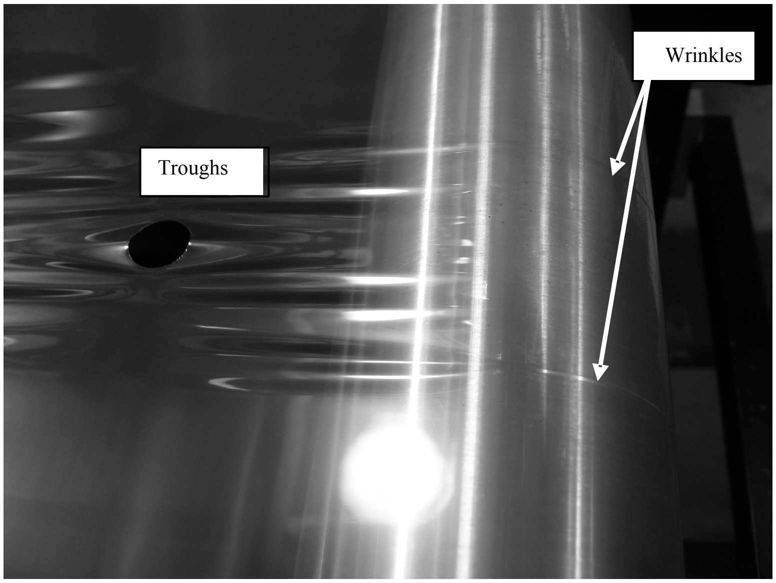


Figure 6.1 Troughs and wrinkles due to a circular hole in the web [41]

The hole would travel very close to the downstream roller before wrinkles were seen at low tension. If tension was low enough, wrinkles were not visible. At high tension values the distance between the downstream roller and the hole was much larger before wrinkles were seen. He recorded the distance between the hole and the tangent line at fixed values of web tension. This distance is *L* is defined in Figure 6.2.

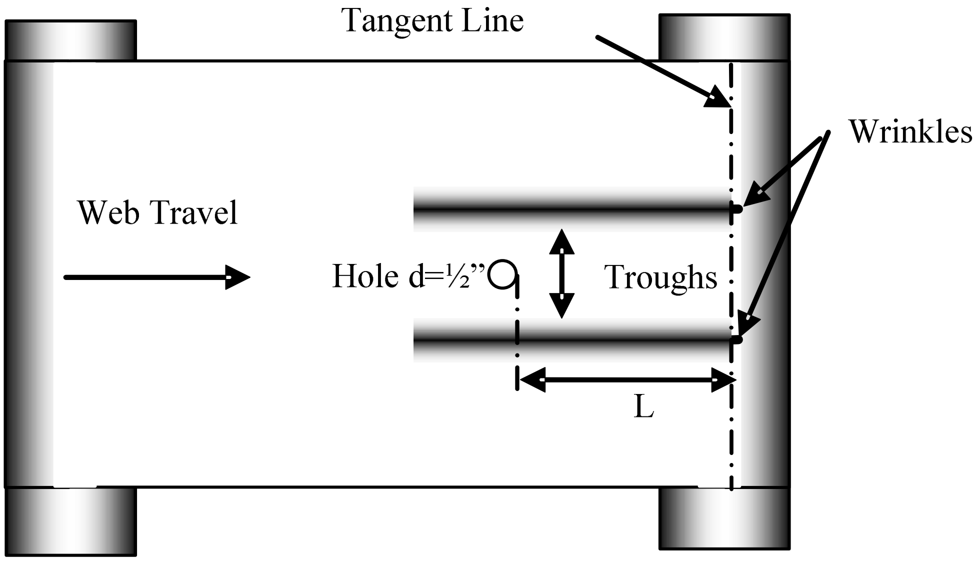


Figure 6.2 Distance Between the Hole and the Tangent Line

Like Beisel he modeled the hole case using the commercial finite element code COSMOS. He also modeled the problem by using five panels: upstream roller, test free span, downstream roller, exiting free span and exiting roller. In Figure 6.3 a schematic of the Mallya’s finite element model is shown.

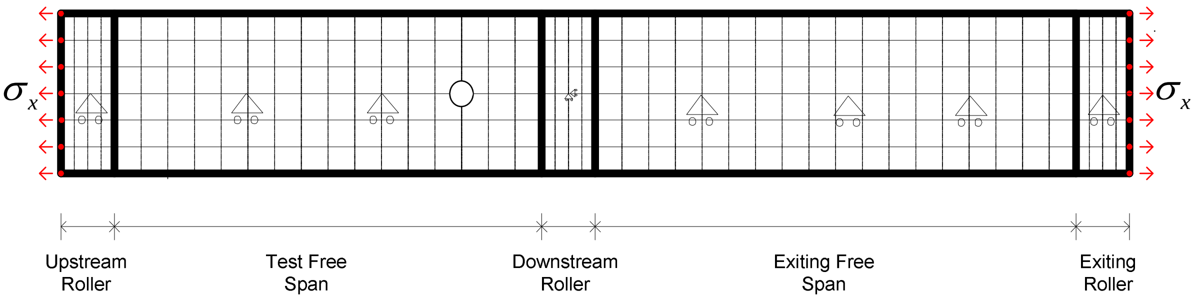


Figure 6.3 Mallya's Model

In the test free span he used wrinkling-membrane quadrilateral eight node elements; in other parts he used linear elastic quadrilateral eight node elements. He used eight node elements because that facilitated modeling the hole. He made the hole in the web by deleting a few elements in the region where the hole was located, and modified the position of the neighboring nodes to create a circular hole of 0.5 inch diameter. He constrained one node in the web span through the *x* direction to constrain the web from moving in *x* direction. He constrained the centerline of the web to prevent web to deform in the cross machine direction. He coupled the cross machine direction deformations of rows of nodes on the rollers to lock them together near the rollers. He followed this procedure to enforce normal entry of the web into the rollers. He increased the σ_x machine direction stress linearly until negative cross machine direction stresses of the Timoshenko buckling criteria (Eq.2.1) was attained.

He compared his experimental results and model results. In the following chapters we will compare our model results with his experimental results and with his model results.

6.3 A New Algorithm for Predicting Wrinkles Due to a Circular Discontinuity

The circular hole case was modeled by using Excel VBA. The model is similar to the misaligned roller case model and tapered roller case model. The material on the upstream roller, the upstream span where the hole is traveling, the downstream roller, the downstream span, and finally material on the exiting roller were modeled. In the upstream span wrinkling membrane elements were used. Elastic membrane elements were used at other parts of the system. The system that is modeled is shown in Fig.6.4.

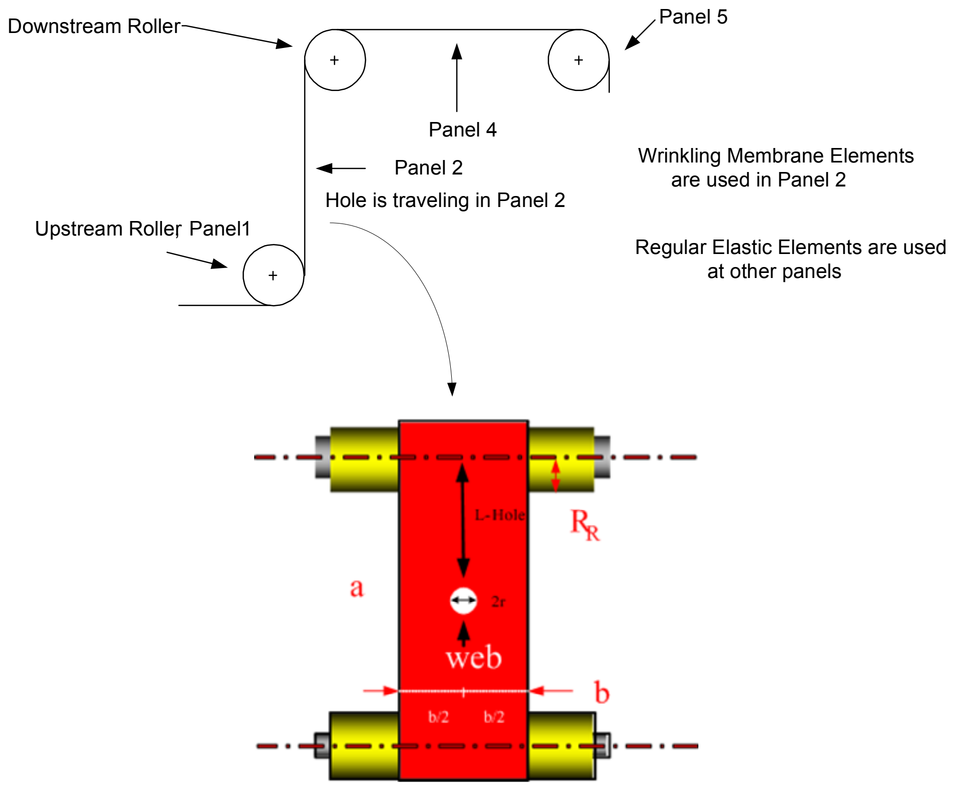

Downstream Roller
Panel 5
Panel 4
Panel 2
Hole is traveling in Panel 2
Wrinkling Membrane Elements are used in Panel 2
Upstream Roller Panel1
Regular Elastic Elements are used at other panels
L-Hole
R_R
a
2r
web
b
b/2
b/2

Figure 6.4 The Figure for the Hole Model Traveling Between Rollers

Because of the symmetry of the problem half of the problem was modeled to take the advantage of the computational time. In Figure 6.5 the method of modeling the problem is shown.

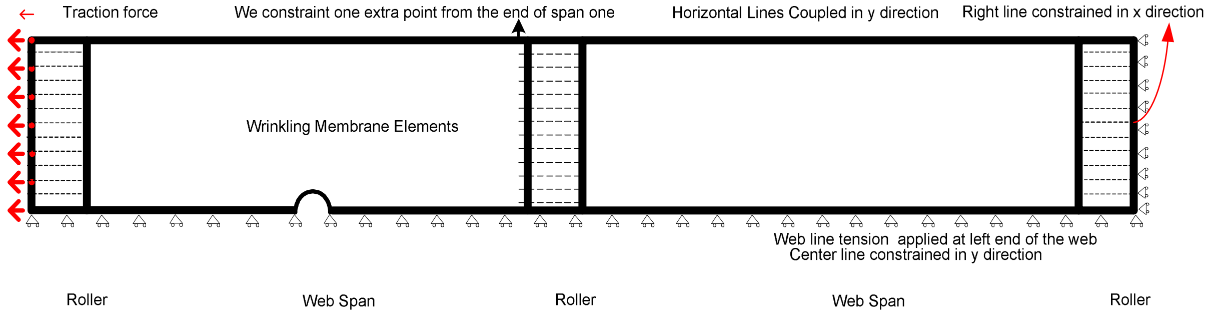

Traction force
We constraint one extra point from the end of span one
Horizontal Lines Coupled in y direction
Right line constrained in x direction
Wrinkling Membrane Elements
Web line tension applied at left end of the web
Center line constrained in y direction
Roller
Web Span
Roller
Web Span
Roller

Figure 6.5 The Web Hole FE Wrinkle Model

The center line was constrained in the y direction to prevent the web from deforming in the cross machine direction. The right line of the third roller in the *x* direction was constrained and traction force was applied only from the upstream roller which is different from the tapered roller and misaligned roller cases. This condition helped to prevent web rigid body motion in the *x* direction. The cross machine direction deformations of rows of nodes on the rollers are coupled to lock them together near the rollers so as to enforce normal entry of the web to the roller. In the second roller as seen in Fig.6.5 one extra point from the first span was constrained with the points of roller two to achieve normal entry to the second roller. The hole region was meshed as seen in Fig.6.6.

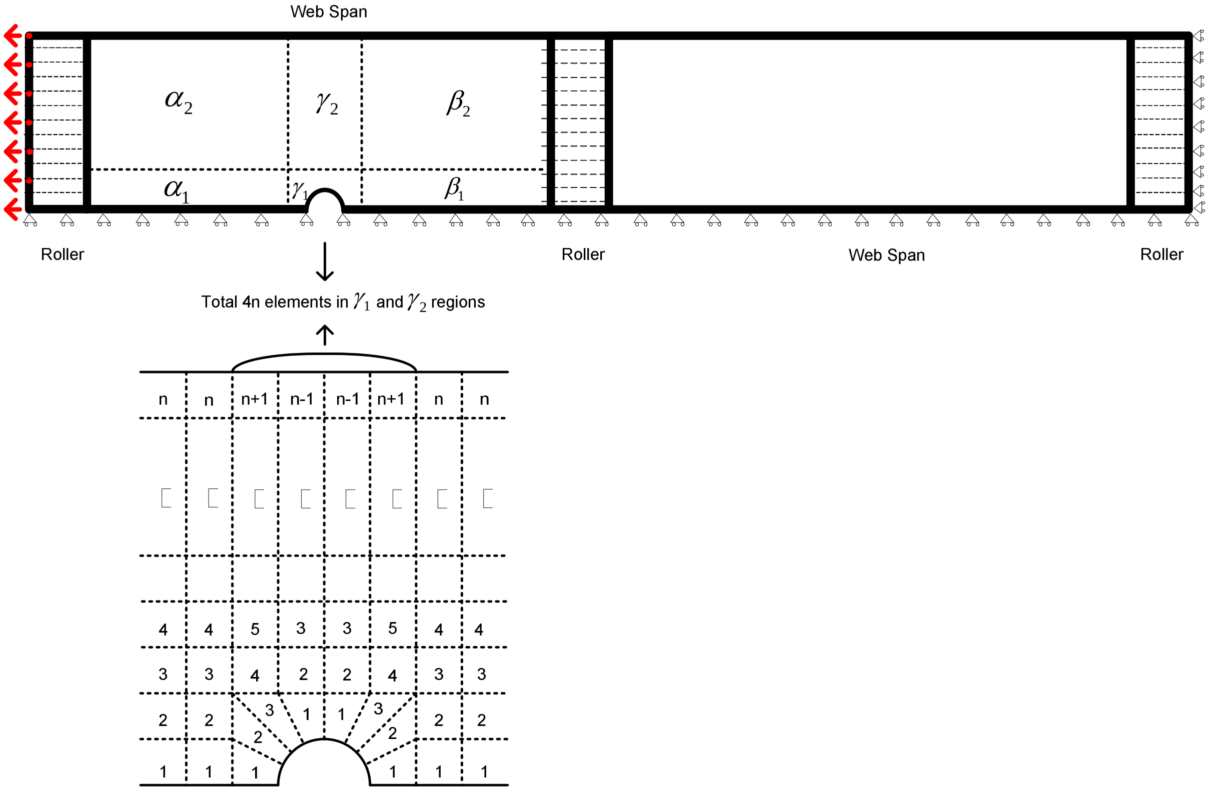


Figure 6.6 Meshing the Hole Region

The span two region was divided into six parts $\alpha_1, \alpha_2, \gamma_1, \gamma_2, \beta_1$ *and* β_1. The γ_2 region represents the hole region. One more element was used at the first and fourth rows in γ region, and one less element was used in the second and third rows in the γ region. This helped to have total of 4n elements that will allow using previous codes sub modules. The Excel output for the hole region (γ_2) can be seen at Figure 6.7.

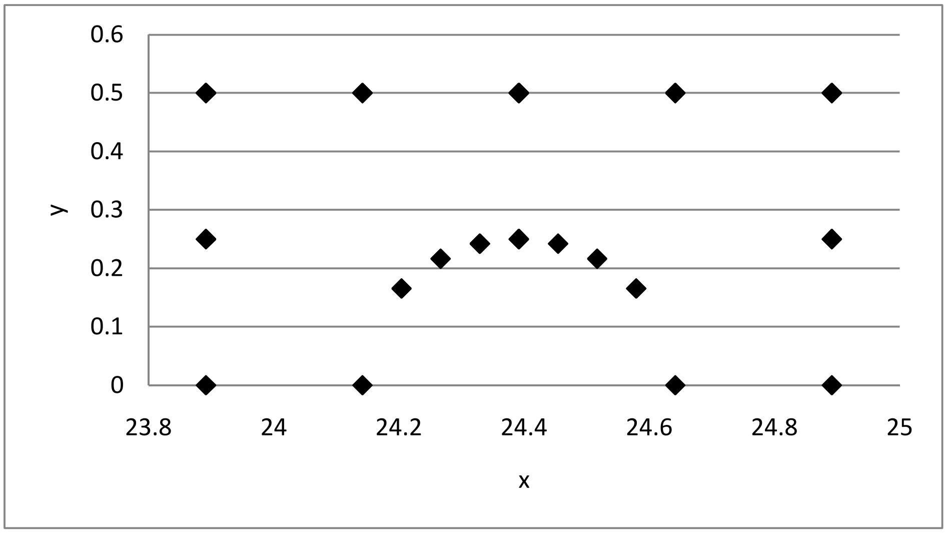
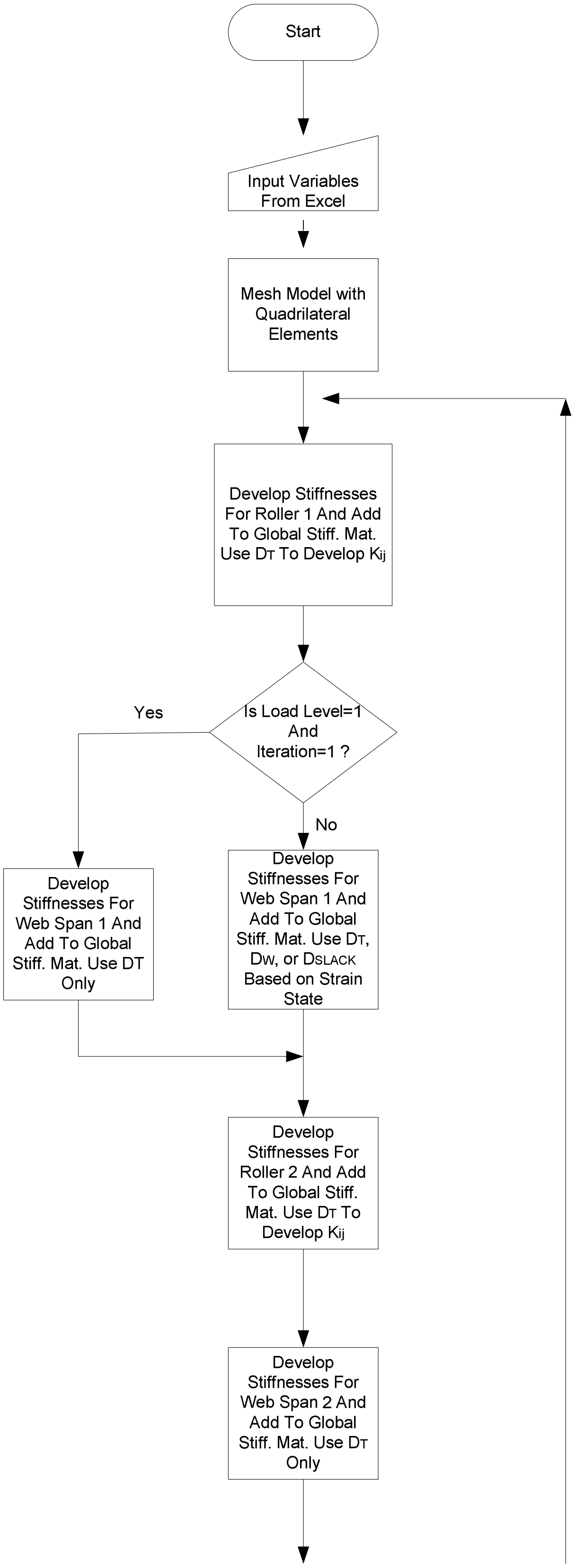


Figure 6.7 Excel Output of Hole Region

The traction forces were applied in four steps. The flowchart for the program is shown in Fig.6.8. It is similar to the misaligned roller and tapered roller flow chart so it will not be explained here in detail. The name “*VOIDsystem*” will be applied to the part of the flow chart (Fig.6.8) that begins after “Mesh Model with Quadrilateral Elements” and continuous to “Load Level < 4”. This term will be used while explaining the automation of the code.



Start
Input Variables From Excel
Mesh Model with Quadrilateral Elements
Develop Stiffnesses For Roller 1 And Add To Global Stiff. Mat. Use DT To Develop Kij
Is Load Level=1 And Iteration=1 ?
Yes
No
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT Only
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT, DW, or DSLACK Based on Strain State
Develop Stiffnesses For Roller 2 And Add To Global Stiff. Mat. Use DT To Develop Kij
Develop Stiffnesses For Web Span 2 And Add To Global Stiff. Mat. Use DT Only

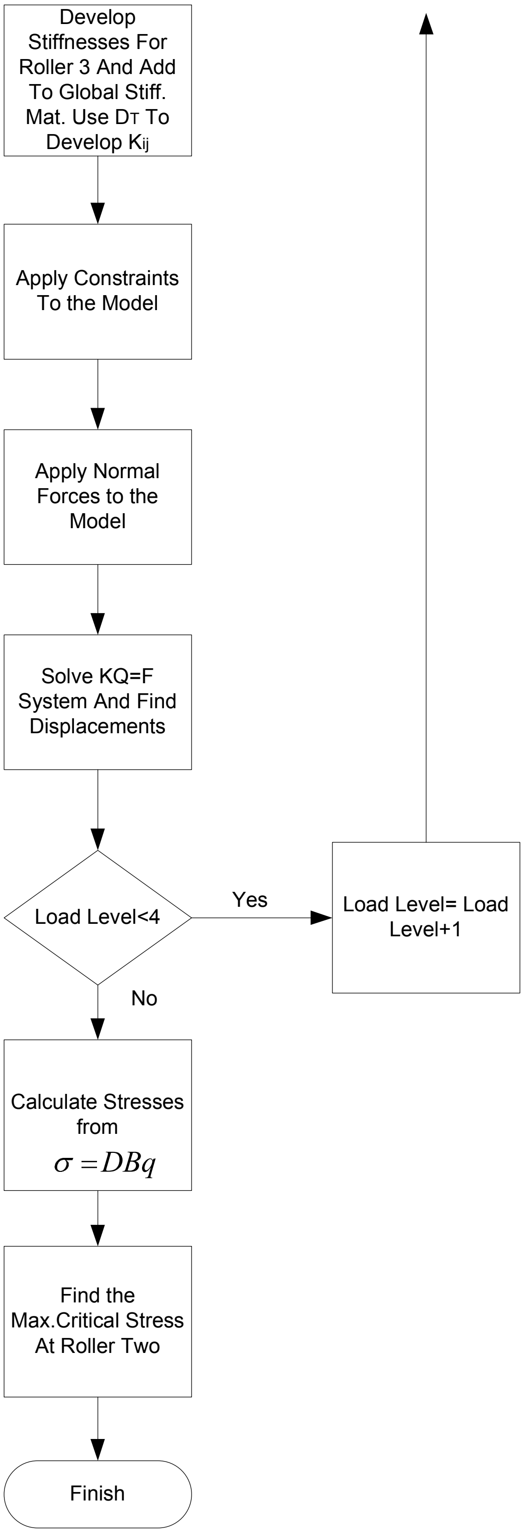

Develop Stiffnesses For Roller 3 And Add To Global Stiff. Mat. Use DT To Develop Kij
Apply Constraints To the Model
Apply Normal Forces to the Model
Solve KQ=F System And Find Displacements
Load Level<4
Yes
Load Level= Load Level+1
No
Calculate Stresses from σ =DBq
Find the Max.Critical Stress At Roller Two
Finish

Figure 6.8 The Flowchart for Circular Discontinuity

As an output the code calculates the maximum compressive stress at the first row of elements from roller two.

6.4 Automating the Code

Here, in contrast to other cases, it aimed for the user the user to enter only certain parameters such as web width, roller radius, span length, web thickness, elastic modulus, Poisson's ratio and then the code to run automatically. The user will not enter web tension or shear force in this case. First step was automating the mesh. After running many cases it was found that dividing the web width to the roller radius and dividing the web width to the integer even part of that value, one element for every dimension for the web spans, and the integer part of four times the roll dimension for rollers gave reasonable results. The Excel equations were used to set these values. The meshing procedure and convergence check will be addressed more detailed at Chapter VIII.

In the hole case, the code attempts to find the web tension that will generate a wrinkle for a given hole distance. The goal was automating finding that web tension. For that purpose a starting point was needed. It was found that the minimum tension required to sustain a wrinkle over the surface of a roller could be used as a starting point. If it is assumed that the web tension is uniform across the web width (Figure 6.9).

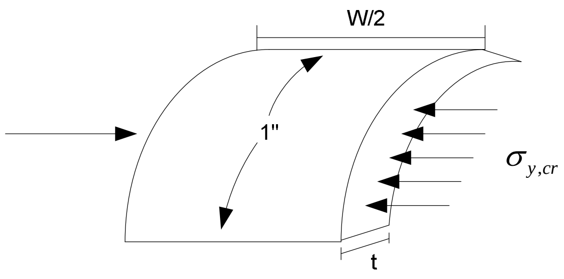


Figure 6.9 The Section of Web over the roller

Here from the Fig.6.9 [42]:

$$F_{friction} = \frac{T_w}{r}\frac{w}{2}1\mu \quad ; \quad F_{wrinkle} = \sigma_{y,cr} t\, 1 = \frac{t^2}{r}\frac{E}{\sqrt{3(1-v^2)}} \quad and \quad F_{CMD} = 0 \tag{6.1}$$

By performing the ASTM standard test, D1894 [43]:

$$F_{friction} = F_{wrinkle} \Rightarrow \frac{T_w}{r}\frac{w}{2}\mu = \frac{t^2}{r}\frac{E}{\sqrt{3(1-v^2)}} \tag{6.2}$$

If some calculations were done from Eq.6.2 it found that:

$$T_w = \sigma_{y,cr}\,\frac{2r}{\mu w} \tag{6.3}$$

From our experience *r* is much smaller than *w* and μ is around 0.3. As a result it can be said that the value of T_w is very close to $\sigma_{y,cr}$,and $\sigma_{y,cr}$ is much smaller than the tension that is needed to buckle the web. So T_w and 2 T_w were used as a starting point for *Linear Interpolation I* (Fig.4.16). By taking the user inputs from Excel, meshing the problem, and using these values in *Linear Interpolation I* and *Linear InterpolationII* it is possible to find the critical tension that buckles the web.

6.5 Comparison of Results with Previous Works for Wrinkles Due to Holes

Mallya compared his experiment results with his finite element model results. The results from the Excel VBA code will be compared with his experimental results and test results.

The web he tested was 79 gage thick polyester film with a width of 12 in. The span length was 28 in and the roller radius was 2 in. This polyester has a Young's modulus of 712000 psi and an assumed Poisson's ratio of 0.3. For these properties the Timoshenko buckling stress was found to be around -170 psi by using Eq. 2.1. The comparison can be seen in Figure 6.10.

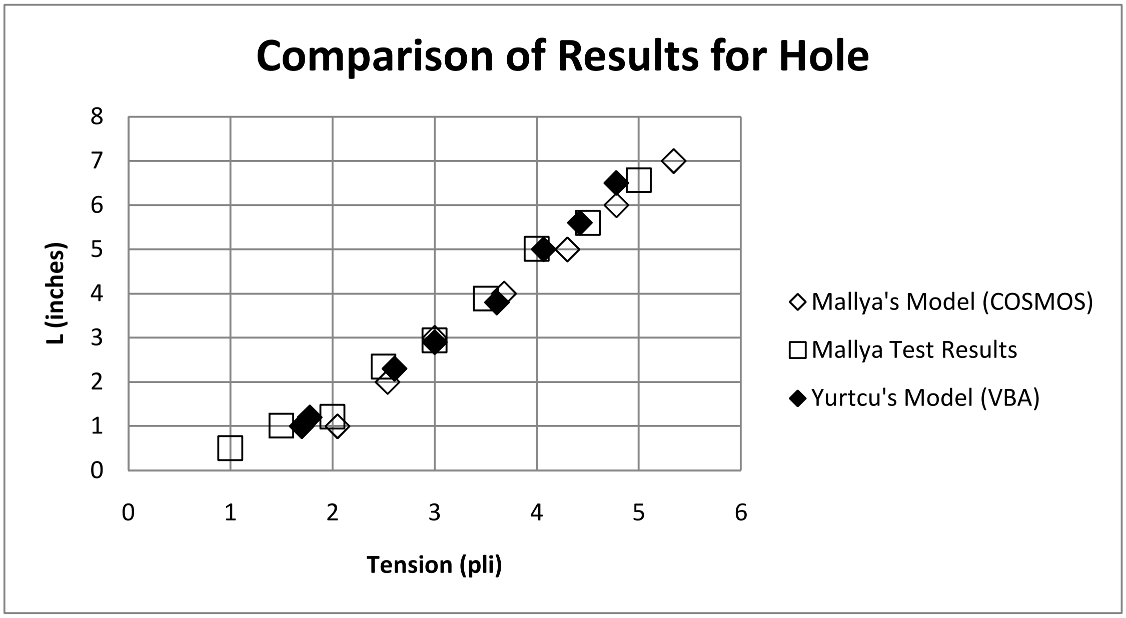


Figure 6.10 Comparisons of Results for Hole

By changing the mesh of the problem we are able to compare a circle, a square and an equilateral quadrangle hole shapes. In Figure 6.11 the Excel output of the square region and in Figure 6.12 Excel output of equilateral quadrangle region can be seen. Here diameters of the holes have the same length (2r in Fig.6.4, 6.11 and 6.12).

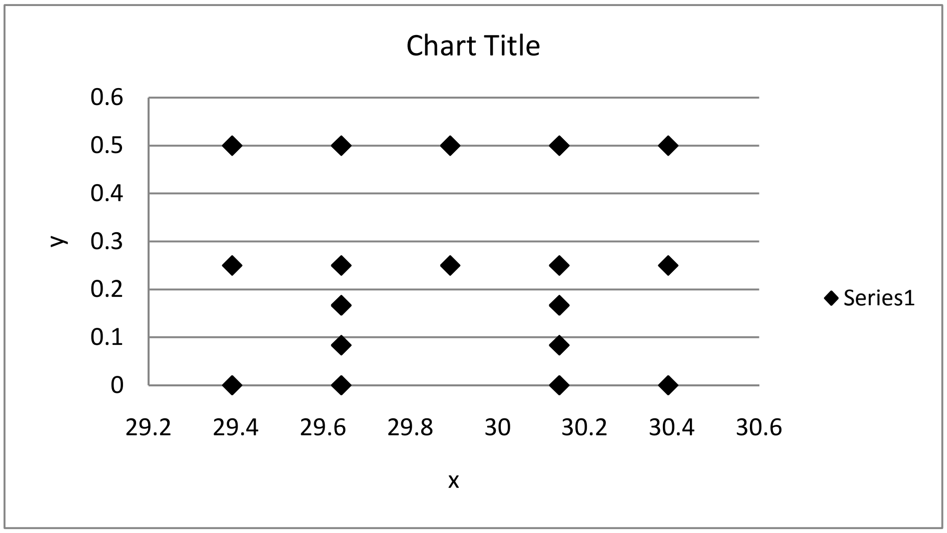


Figure 6.11 Excel output of square

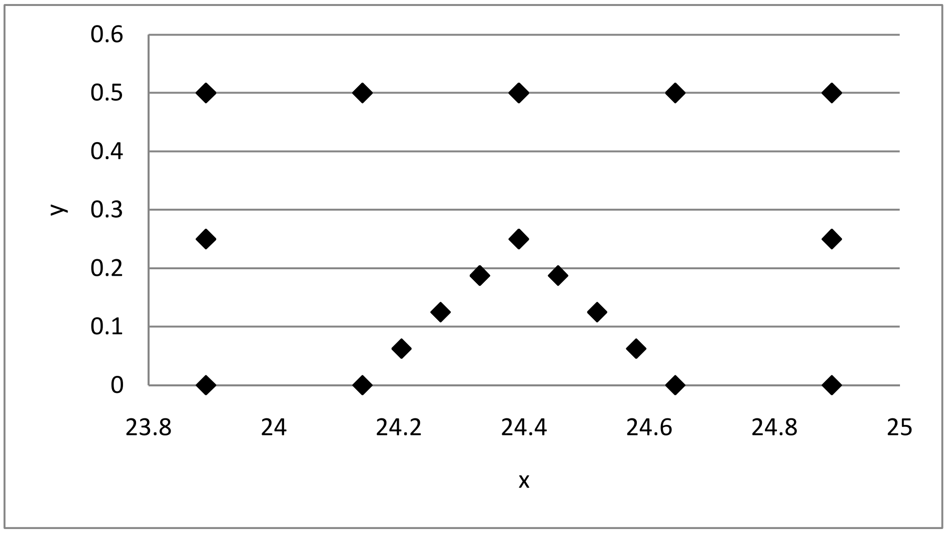


Figure 6.12 Excel output of equilateral quadrangle

In Figure 6.13 comparison of these three shapes can be seen.

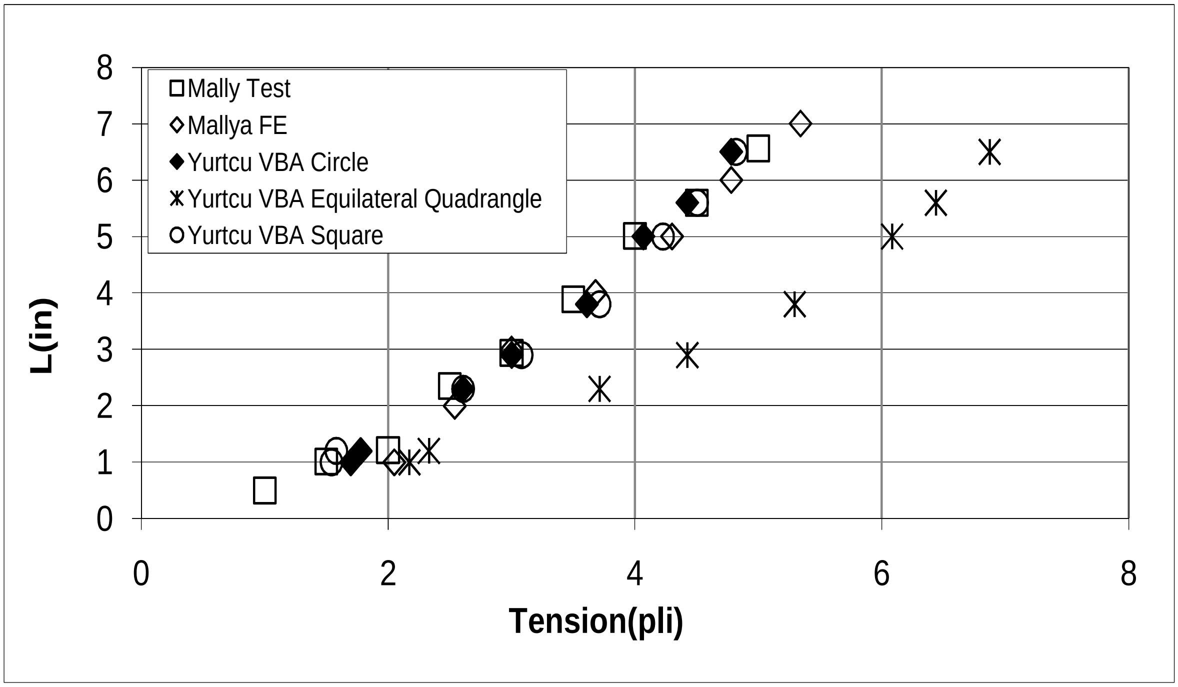


Figure 6.13 Comparison of Circle, Equilateral Quadrangle and Square Shaped Holes

As seen from Fig.6.13 it can be said that among a circular hole, a square hole and an equilateral quadrangle hole, the equilateral quadrangle shape requires more tension to wrinkle. This conclusion can be explained by the element stiffness matrix. In an element stiffness matrix the area term is constant. The area of equilateral quadrangle hole is less than the areas of the other holes. Therefore, removing an equilateral quadrangle hole from the web has less effect on the web stiffness. As a result, if a hole is needed in a web for any reason, we can suggest that it be an equilateral quadrangle hole, because by using an equilateral quadrangle hole, more tension can be applied to the web.

It can be seen from Fig.6.10 that up to a certain level, the linear behavior of the problem is obvious. This encouraged us to update the code to run with only certain parameters such as web width, roller wrap length, span length, web thickness, elastic modulus and Poisson's ratio. One fourth of the span length and half of the span length were used as the location for the hole. Since the behavior always appears linear, straight line behavior is assumed for the tension levels at intermediate location.

The problem was meshed and the tension was calculated when the circular hole was at a distance of ¼ of the span length by using *Linear Interpolation I* and *Linear Interpolation II.* Then the tension was calculated when the circular hole is at a distance of ½ of the span length by using *Linear Interpolation I* and *Linear Interpolation II.* The flowchart in Figure 6.14 shows how new algorithm works.

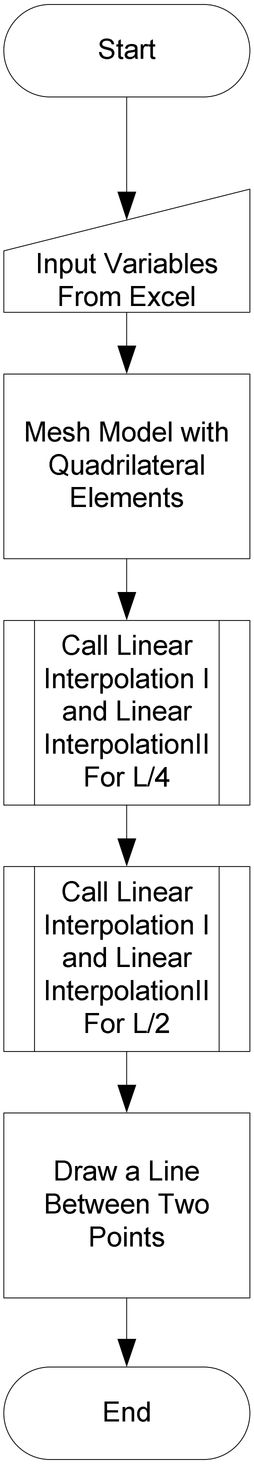
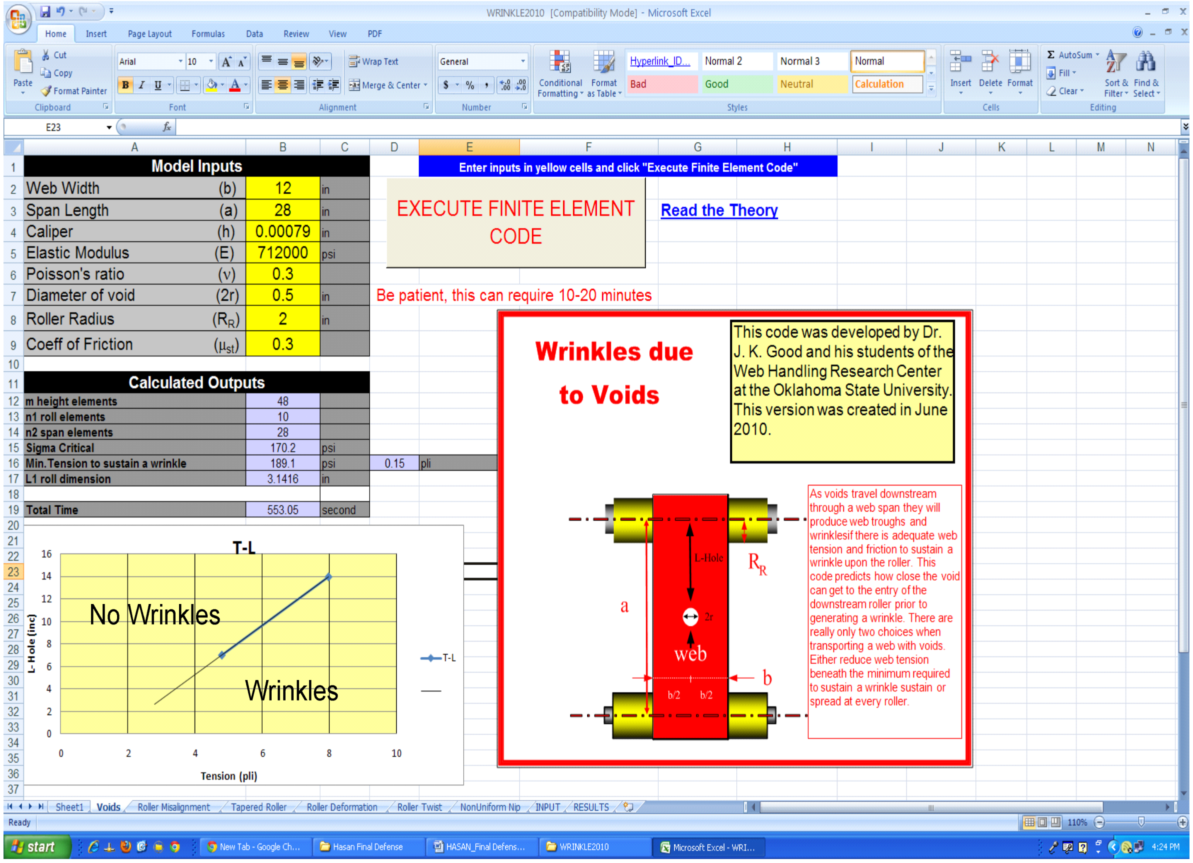

Start
Input Variables From Excel
Mesh Model with Quadrilateral Elements
Call Linear Interpolation I and Linear InterpolationII For L/4
Call Linear Interpolation I and Linear InterpolationII For L/2
Draw a Line Between Two Points
End

Figure 6.14 Flow Chart for Automating Hole Code

After automating the code, the Excel interface of the code will look like Figure 6.15. As seen from the figure, the user inputs the web width, span length, web thickness, elastic modulus, Poisson's ratio, diameter of void, roller radius, and the coefficient of friction. By using the Excel equations the code calculates mesh parameters, Timoshenko buckling criteria (Sigma Critical), the minimum tension to sustain a wrinkle (Eq.6.3), and the L1 roll dimension as an output. When the user clicks the *EXECUTE* button the code calculates the tension that will buckle the web when the hole is ¼ of the span length away from the circumference line (here 28/4=7"). The code then calculates the tension that will buckle the web when the hole is 1/2 of the span length away from the circumference line (here 28/2=14"). Finally, by using these points with the help of Excel, the relation between the tension and the distance of the hole from the downstream roller is plotted. For this material by the help of this plot it can be suggested to run this material at less than 1.8 pli in order to avoid wrinkles.



Model Inputs			
Web Width	(b)	12	in
Span Length	(a)	28	in
Caliper	(h)	0.00079	in
Elastic Modulus	(E)	712000	psi
Poisson's ratio	(v)	0.3	
Diameter of void	(2r)	0.5	in
Roller Radius	(R_R)	2	in
Coeff of Friction	(μ_{st})	0.3	

Calculated Outputs				
m height elements	48			
n1 roll elements	10			
n2 span elements	28			
Sigma Critical	170.2	psi		
Min.Tension to sustain a wrinkle	189.1	psi	0.15	pli
L1 roll dimension	3.1416	in		
Total Time	553.05	second		

Enter inputs in yellow cells and click "Execute Finite Element Code"

EXECUTE FINITE ELEMENT CODE

Read the Theory

Be patient, this can require 10-20 minutes

Wrinkles due to Voids

This code was developed by Dr. J. K. Good and his students of the Web Handling Research Center at the Oklahoma State University. This version was created in June 2010.

As voids travel downstream through a web span they will produce web troughs and wrinklesif there is adequate web tension and friction to sustain a wrinkle upon the roller. This code predicts how close the void can get to the entry of the downstream roller prior to generating a wrinkle. There are really only two choices when transporting a web with voids. Either reduce web tension beneath the minimum required to sustain a wrinkle sustain or spread at every roller.

Figure 6.15 Excel Interface of Circular Void Excel VBA Code

Now it can be said that an Excel VBA code that is capable of finding the relation between the hole distance and tension automatically have been written.

7. CHAPTER VII

7.1 NON UNIFORMITIES IN WEBS

In this chapter, non uniformities in webs will be studied. Modeling the case with commercial finite element code COSMOS and modeling the case with Excel VBA will be described.

Mallya [15] proved that the modeling method developed by Beisel for modeling wrinkling of webs encountering misaligned rollers and tapered rollers was also viable for studying how holes in webs cause wrinkles. The research done by the author has shown this same modeling is possible with user friendly Excel VBA codes.

The sponsors of the Web Handling Research Center were excited by the results of this research, because some of their worst problems are caused by web non uniformity. A hole is an example of the worst non uniformity possible. Many of these sponsors handle webs that are made by processes that are far from perfect. As a result, webs often have length, thickness, and density non uniformities.

Thus the research on the effects of voids on wrinkles excited them because they saw potential application of the same modeling methods to study the effect of web non uniformity on wrinkling. So, a new investigation has begun where a region in the web differs in some way from the surrounding web. Perhaps it is thinner or thicker. Perhaps the non uniformity occupies a large portion of the web width or perhaps it is relatively small. The end result will be a development tool that can be used to determine the size and degree of a non uniformity that is permissible without the development of web wrinkles.

7.2 Modeling Non uniformities with Commercial Finite Element Code COSMOS

Devising controlled experiments with non uniform webs is very difficult. It was decided to model these problems with COSMOS and study the behavior. The data from COSMOS was compared with the data from a developed Excel VBA code and helped us to verify the code.

Instead of a circular void a circular region with reduced thickness was modeled and the remainder of the web was left uniform. The case modeled with five sections, the upstream roller, the upstream span, the downstream roller, the downstream span, and finally material on the last roller (Fig.7.1).

In the free span wrinkling membrane eight node elements were used. In the other regions linear elastic quadrilateral eight node elements were used. One node in the web span was constrained in the *X* direction to prevent web from moving in the *X* direction. The centerline of the web was constrained to prevent web deformation in the cross machine direction.

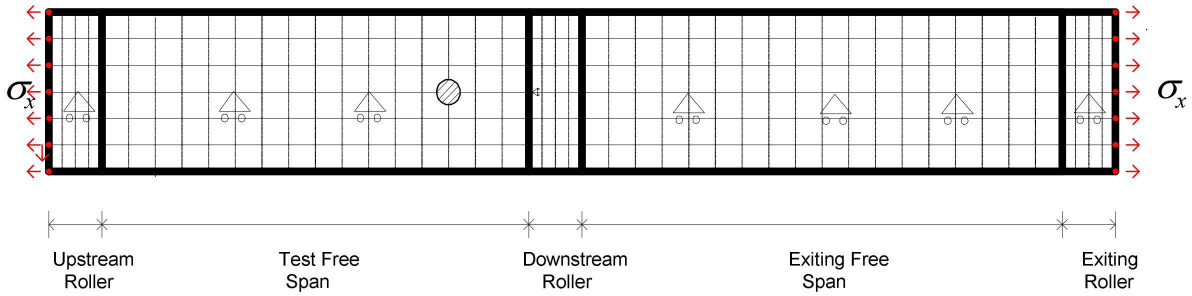


Figure 7. 1 Non uniform Web Models (COSMOS)

All nodes were constrained in *Z* direction so that the web remains planar. The thickness of the non uniformity was varied. Tension was applied by using a pressure curve on the edges of the webs along the *X* direction as shown in Figure 7.1. The non uniform area was modeled in the web with its center on the axis of symmetry. The tension was increased linearly until negative cross machine direction stresses of the Timoshenko buckling stress (Eq. 2.1) was attained.

The web modeled was a 0.001 inch thick film and a roller radius of 1.45 inch was chosen. In the first case the web width was 6 inch and the span length was 18 inch. In the second case the web width was 50 inch and the span length was 50 inch. The web's Young's modulus was 712000 psi and Poisson's ratio was assumed to be 0.3. For these properties the Timoshenko buckling stress was found to be around -297 psi (Eq.2.1).

In Fig.7.2 an example of the model is shown. Here the web width was 6 inch; the span length was 18 inch. The thickness of the non uniform area was 0.0008 inch, the non uniform area was 2 inch in diameter and the non uniform area was 3 inch away from the second roller.

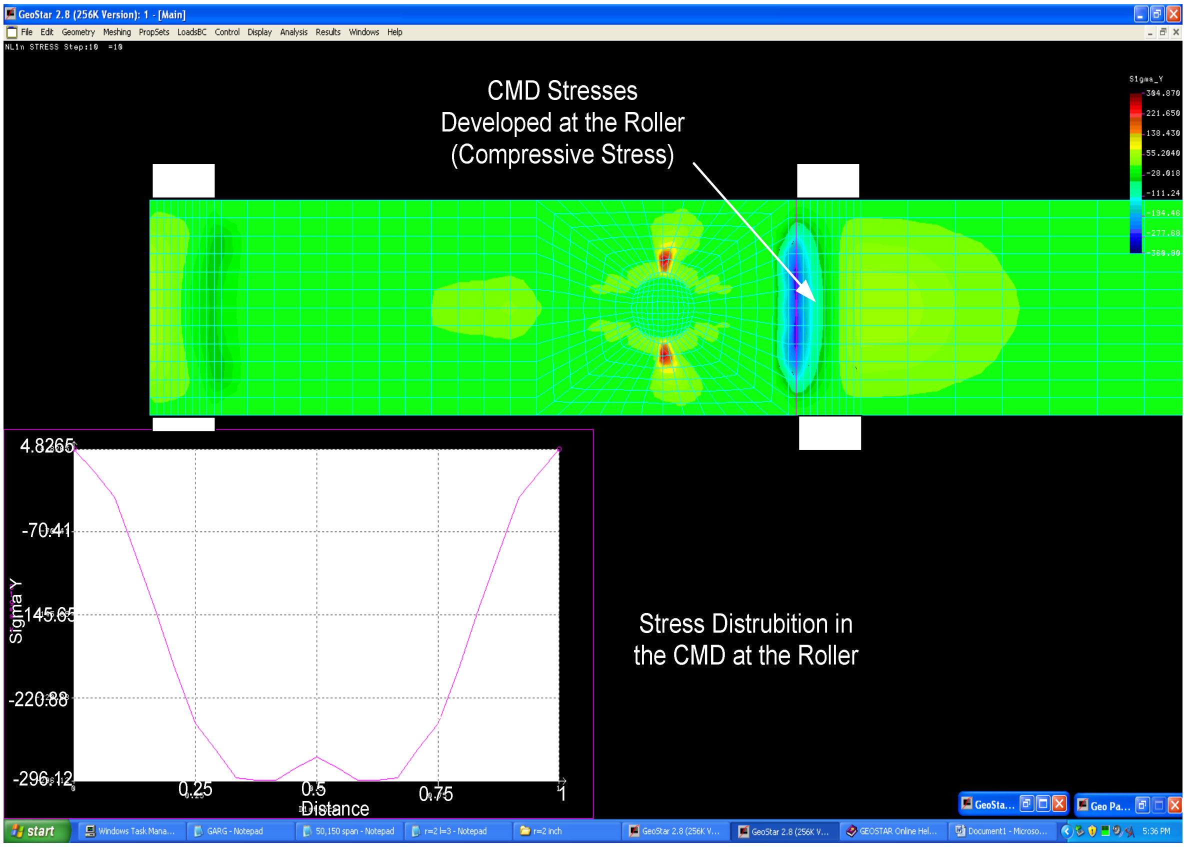

GeoStar 2.8 (256K Version): 1 - [Main]
File Edit Geometry Meshing PropSets LoadsBC Control Display Analysis Results Windows Help
NL1n STRESS Step:10 =10
CMD Stresses Developed at the Roller (Compressive Stress)
Sigma_Y
304.870
221.650
138.430
55.2040
-28.018
-111.24
-194.46
-277.68
-360.90
4.8265
-70.41
-145.65
-220.88
-296.12
Sigma Y
0.25
0.5
0.75
1
Distance
Stress Distrubition in the CMD at the Roller
start

Figure 7. 2 Critical CMD Stresses Developed When the Non uniform Area is 3” Away From the Roller

For a six inch wide web the finite element analysis was conducted with the non uniformities at different distances from the roller. The fifty inch wide and fifty inch long web span was modeled with the same web material properties. In the fifty inch wide case the non uniform region is three inches away from the downstream roller and diameter versus tension required to induce wrinkles was investigated.

7.3 A New Algorithm for Predicting Wrinkles Due to the Circular Non uniformities

An Excel VBA model was developed whose purpose was to determine when web non uniformity would cause web wrinkles. The non uniformity was assumed to be circular in shape and centrally located widthwise in the web. The thickness of the non uniformity was constant but different from that of the surrounding web. The model is similar to the misaligned roller case, tapered roller case and void case. The material on the upstream roller, the upstream span where the non uniform region is traveling, the downstream roller, the downstream span, and finally material on the exiting roller were modeled. In the upstream span wrinkling membrane elements were used and at the other parts of the system elastic membrane elements were used. The system modeled is shown in Fig.7.3.

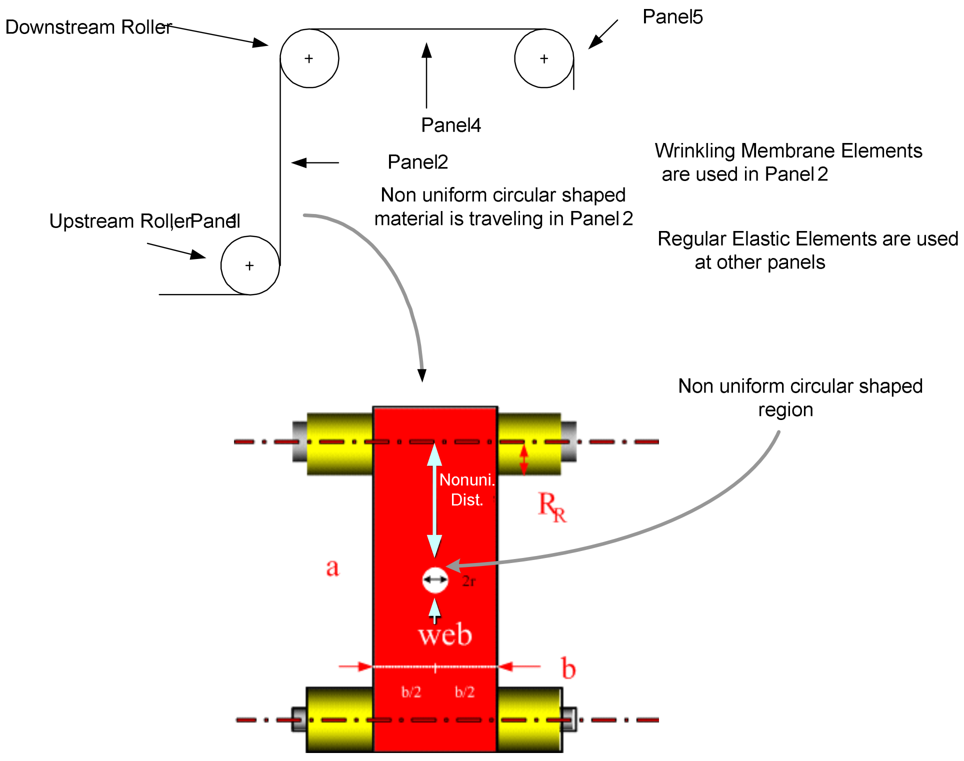

Downstream Roller
Panel5
Panel4
Panel2
Wrinkling Membrane Elements are used in Panel 2
Non uniform circular shaped material is traveling in Panel 2
Upstream Roller
Panel1
Regular Elastic Elements are used at other panels
Non uniform circular shaped region
Nonuni. Dist.
R_R
a
2r
web
b
b/2
b/2

Figure 7. 3 A Web with a Non uniform Circular Shaped Region Travelling Between Rollers

This case is similar to the void case because of the symmetry and half of the problem was modeled to take the advantage of the computational time. In the figure below (Fig.7.4) the method of modeling the problem is shown.

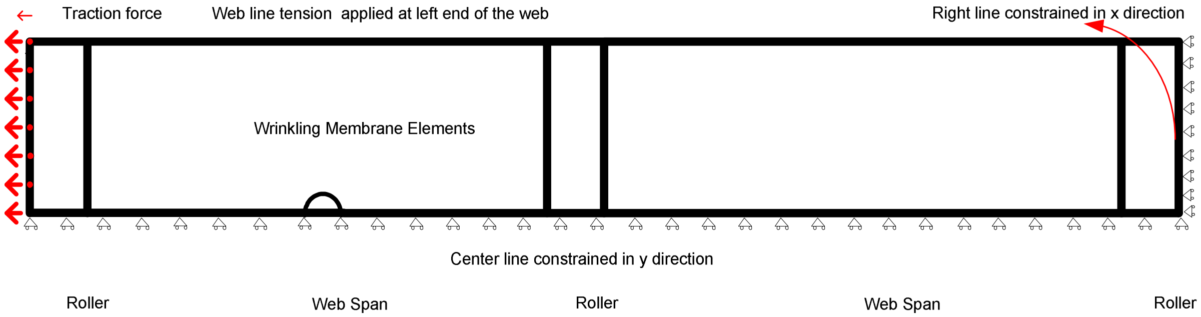

←
Traction force
Web line tension applied at left end of the web
Right line constrained in x direction
Wrinkling Membrane Elements
Center line constrained in y direction
Roller
Web Span
Roller
Web Span
Roller

Figure 7. 4 The Model for the Non uniform Hole Shaped Material

Similar to the void case the center line was constrained in the *y* direction to prevent the web from deforming in the CMD direction. Traction force was applied from the upstream roller and the downstream end of the web at the exit of the third roller was constrained in the *x* direction. This helped to prevent web rigid body motion in the *x* direction. Other than that no other boundary conditions were applied to the model. The non uniform area was meshed as seen in Fig.7.5.

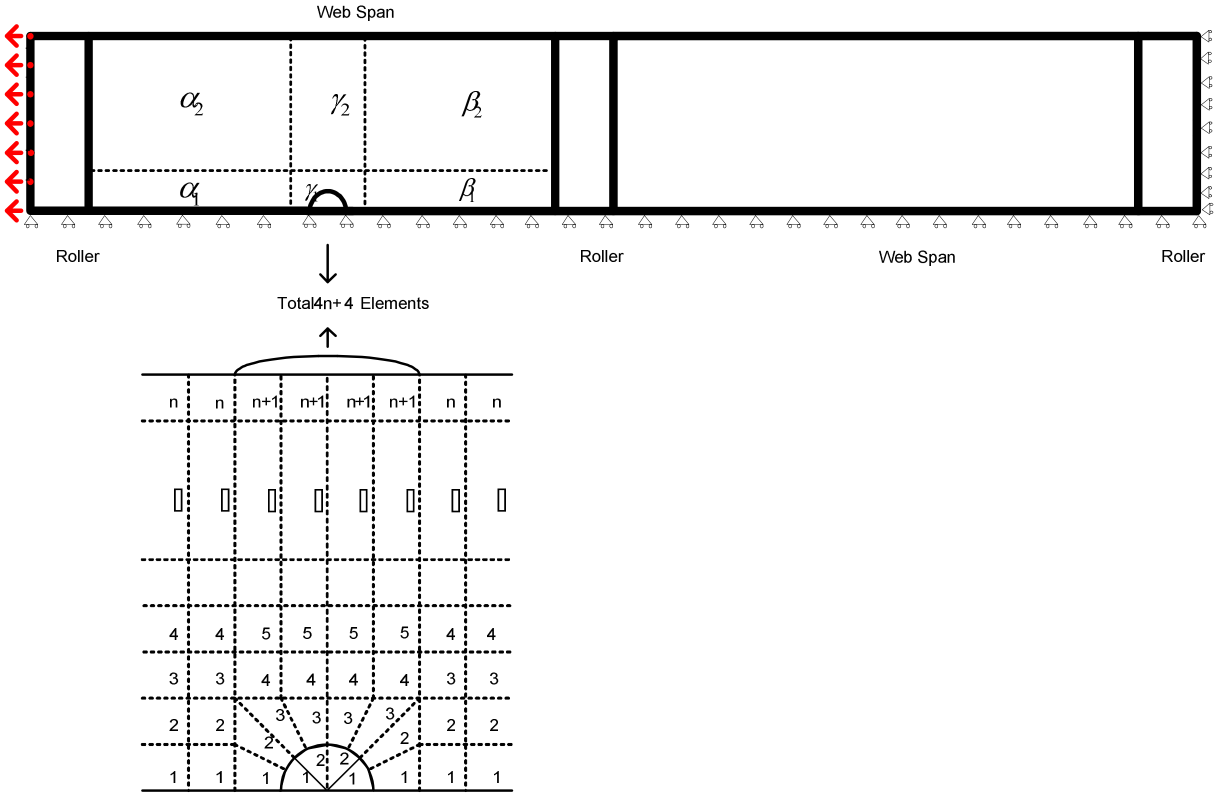


Figure 7.5 Meshing the Non uniform Region

The model is very similar to the void case. The span two region was meshed in six parts. The γ_2 region represents the non uniform area and the surrounding area. This situation differs from the void case because new elements were added in the void area to represent the web non uniformity which could be less or greater than the surrounding web. While calculating stiffness terms and stresses of the elements the code calculates the non uniform area stiffness terms with non uniform thickness value. The mesh for the non uniform region (γ_2) can be seen at Fig.7.6.

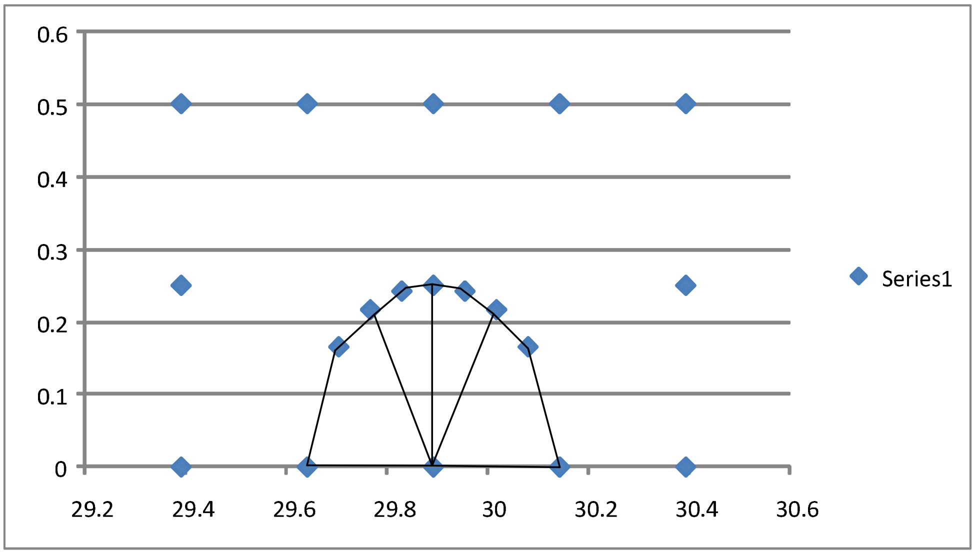
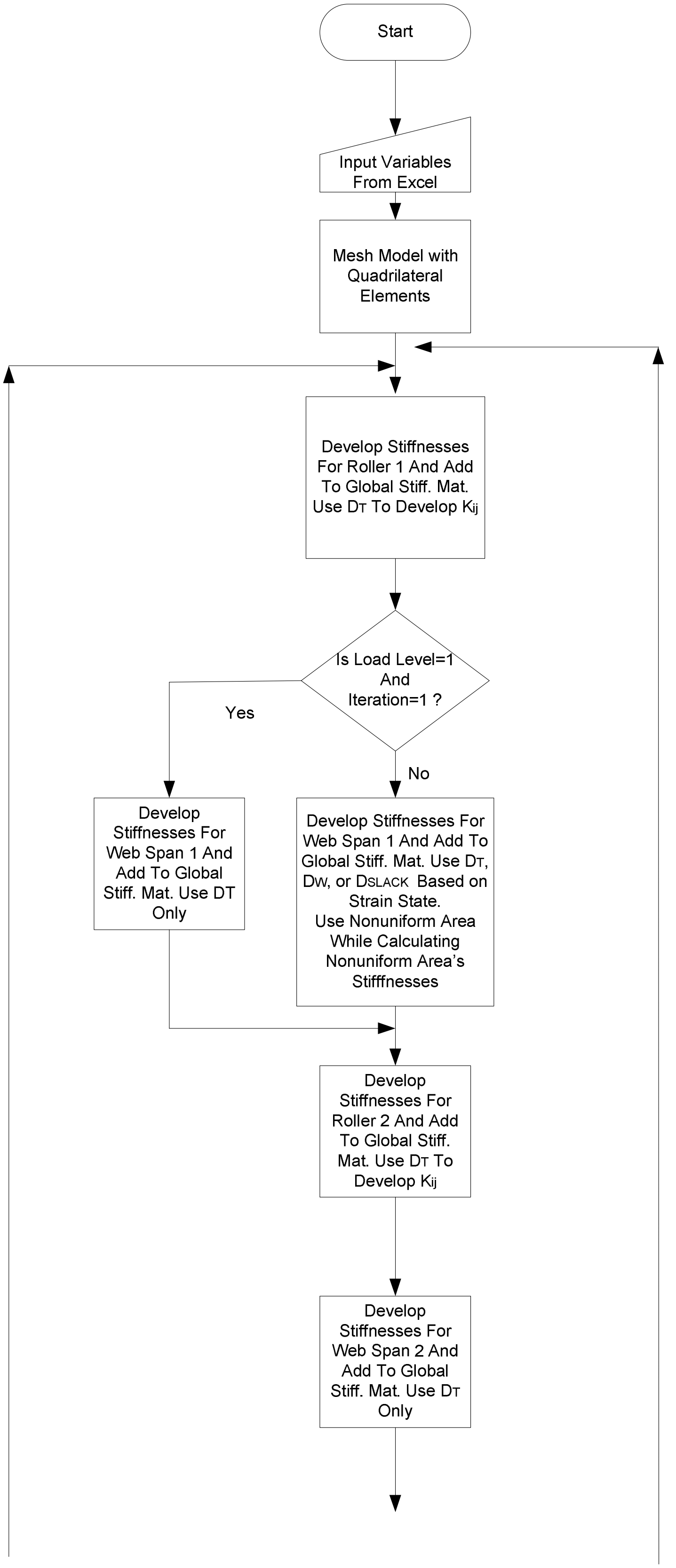


Figure 7. 6 Troughs and Wrinkles

The traction forces associated with web tension were applied in four load steps and two iterations were made within every load step to achieve convergence. The flowchart for the Excel VBA code is shown in Fig.7.7. The flowchart is similar to the previous flowcharts. The name “*NONUNIFORMsystem*” will be applied to the part of the flow chart (Fig.7.7) that begins after “Mesh Model with Quadrilateral Elements” and continuous to “Load Level < 4”. This term will be used while explaining the automation of the code. As an output the code calculates the maximum compressive stress at first row of elements from roller two.


Start
Input Variables From Excel
Mesh Model with Quadrilateral Elements
Develop Stiffnesses For Roller 1 And Add To Global Stiff. Mat. Use D_T To Develop K_{ij}
Is Load Level=1 And Iteration=1 ?
Yes
No
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use DT Only
Develop Stiffnesses For Web Span 1 And Add To Global Stiff. Mat. Use D_T, D_W, or D_{SLACK} Based on Strain State.
Use Nonuniform Area While Calculating Nonuniform Area's Stifffnesses
Develop Stiffnesses For Roller 2 And Add To Global Stiff. Mat. Use D_T To Develop K_{ij}
Develop Stiffnesses For Web Span 2 And Add To Global Stiff. Mat. Use D_T Only

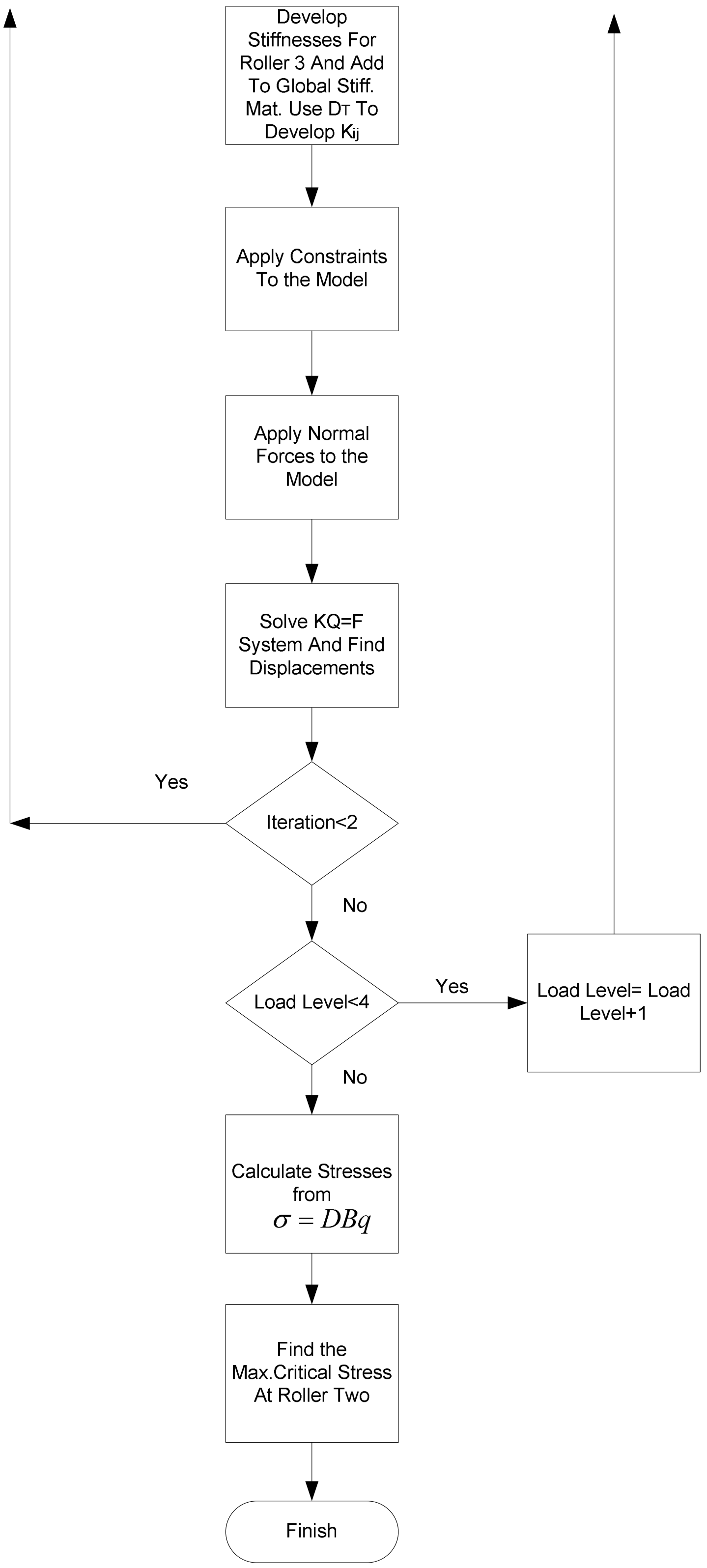

Develop Stiffnesses For Roller 3 And Add To Global Stiff. Mat. Use D_T To Develop K_{ij}
Apply Constraints To the Model
Apply Normal Forces to the Model
Solve KQ=F System And Find Displacements
Yes
Iteration<2
No
Load Level<4
Yes
Load Level= Load Level+1
No
Calculate Stresses from $\sigma = DBq$
Find the Max.Critical Stress At Roller Two
Finish

Figure 7. 7 The Flowchart for Non uniform Webs

7.4 Comparison of COSMOS and Excel VBA Results

As mentioned before doing controlled experiments with non uniform webs is difficult. So, data taken from commercial finite element code COSMOS were compared with Excel VBA code to verify Excel VBA code.

The web modeled was 0.001 inch thick. The roller radius was 1.45 inch. The web's Young's modulus was 712000 psi and Poisson's ratio was assumed to be 0.3. The web width was 6 inch and the span length was 18 inch. For these properties Timoshenko buckling stress was found to be -297 psi from Eq.2.1. Two different non uniform area properties were modeled for comparison.

Fig.7.8 and Fig.7.9 show the results for the web that is 6 inch wide and 18 inch long. The non uniform area has a thickness of 0.0008 inch. The diameter of the non uniform area is 3 inches in Fig.7.8 and 2 inches in Fig.7.9. The web has a thickness of 0.001 inch. In the charts L (distance from the roller) versus the tension required to induce wrinkles is shown using both COSMOS and results from the developed VBA code.

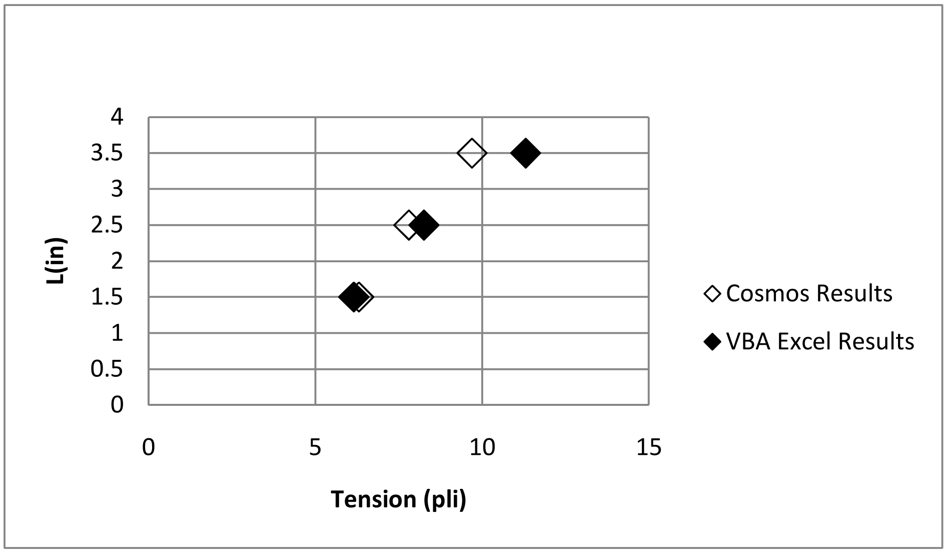


Figure 7. 8 Wrinkles Due to Non uniformities, 2r = 3”, t’ = 0.0008”

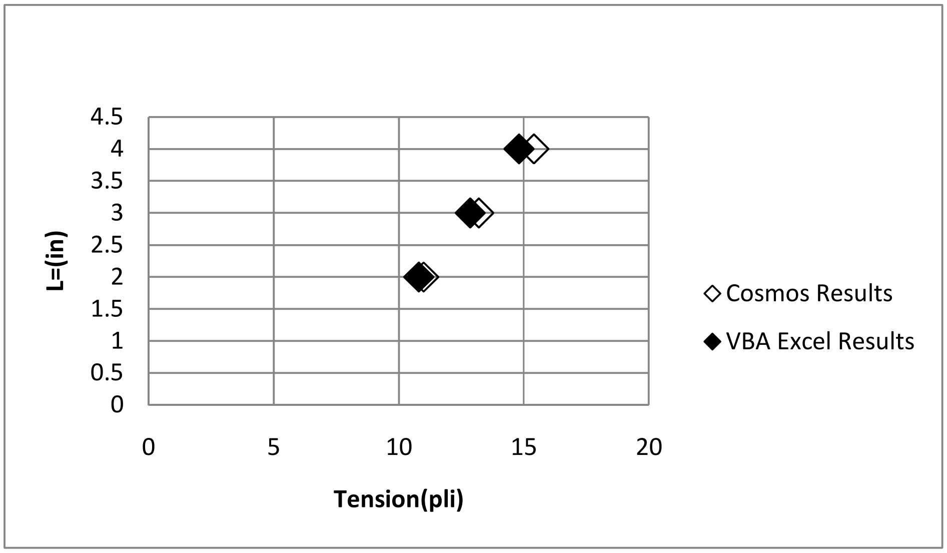


Figure 7. 9 Wrinkles Due to Non uniformities, 2r = 2”, t’ = 0.0008”

The web material properties were kept the same and the thickness of the non uniform area was decreased to 0.0001 inch. Fig.7.10 and Fig.7.11 show results for these cases.

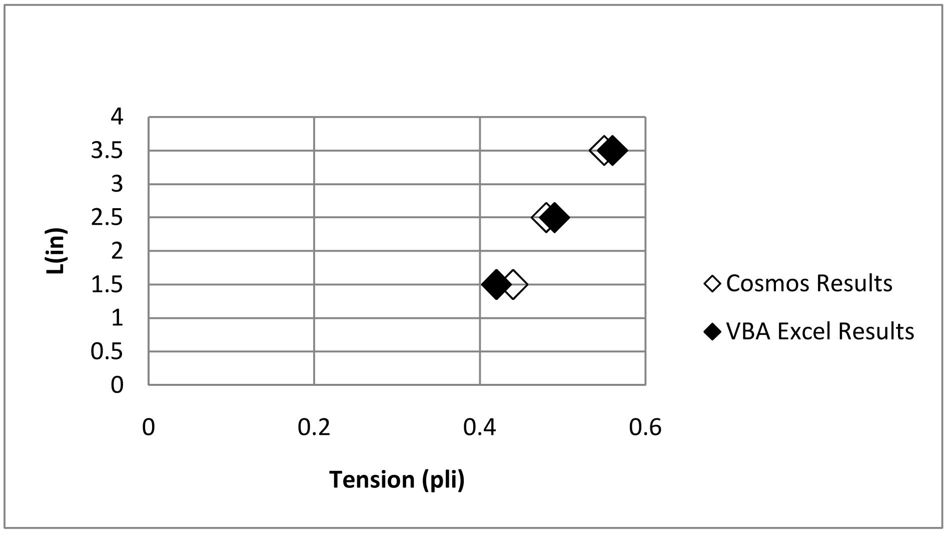


Figure 7. 10 Wrinkles Due to Non uniformities, 2r = 3”, t’ = 0.0001”

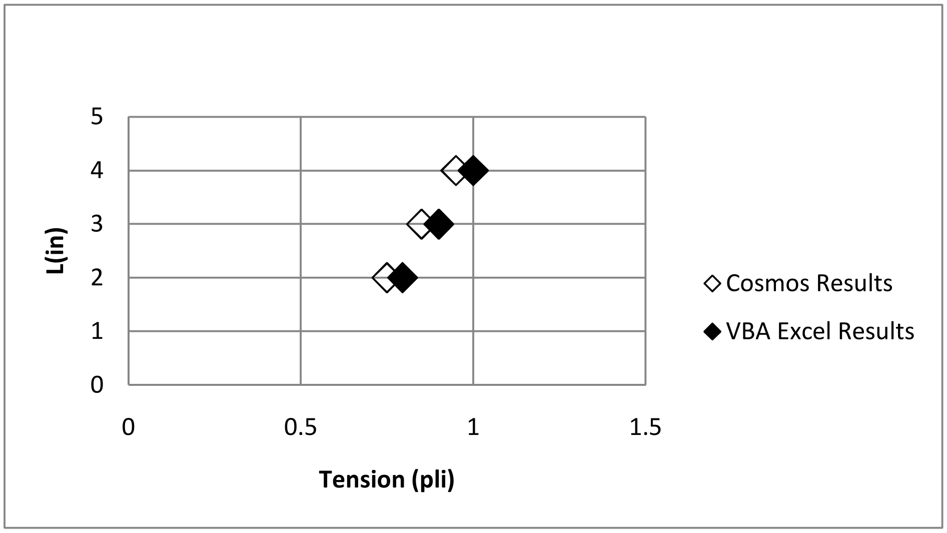


Figure 7. 11 Wrinkles Due to Non uniformities, 2r = 2”, t’ = 0.0001”

Gargeyi Baipa, a master student at Web Handling Research Center performed a sensitivity analysis of how the different aspects of the non uniformities affect wrinkling by using COSMOS. She modeled 50 inch wide and 50 inch long wide web. The web modeled was

0.001 inch thick. The roller radius was 1.45 inch. The web's Young's modulus was 712000 psi and the Poisson's ratio was assumed to be 0.3. These results will be used to further validify the results produced by the VBA code developed.

For the 50 inch wide web Baipa modeled non uniform area 3 inch away from the second roller and she modeled different diameters for non uniform area. In Fig.7.12 comparison of Baipa's COSMOS model and Yurtcu's Excel VBA model is shown. In the chart diameter of void versus the tension required to induce wrinkling is shown.

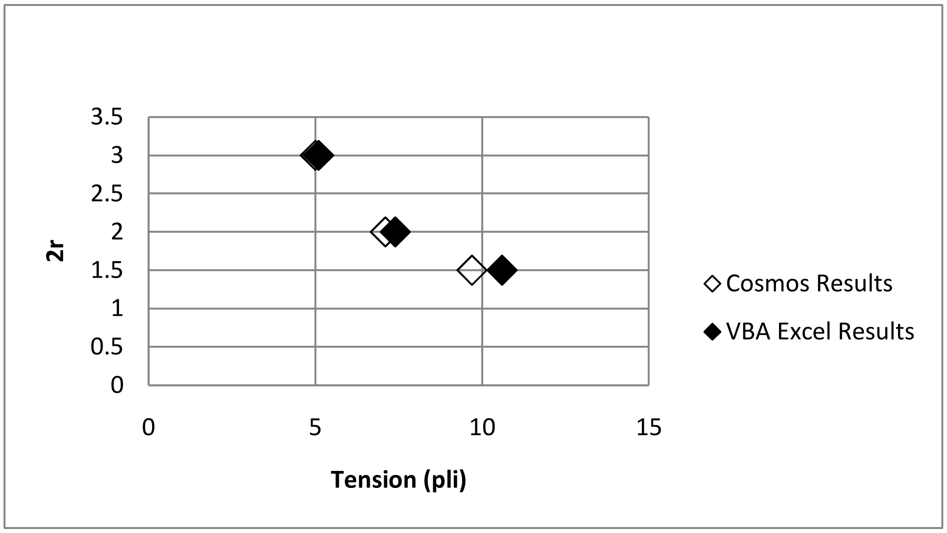


Figure 7. 12 Wrinkles Due to Non uniformities, L = 3”, t’ = 0.0005”

The results from the two analyses shown in Fig.7.8- Fig.7.11 and Fig.7.12 agree well with each other. Thus, it appears the Excel VBA code developed produces similar results to the commercial finite element code COSMOS.

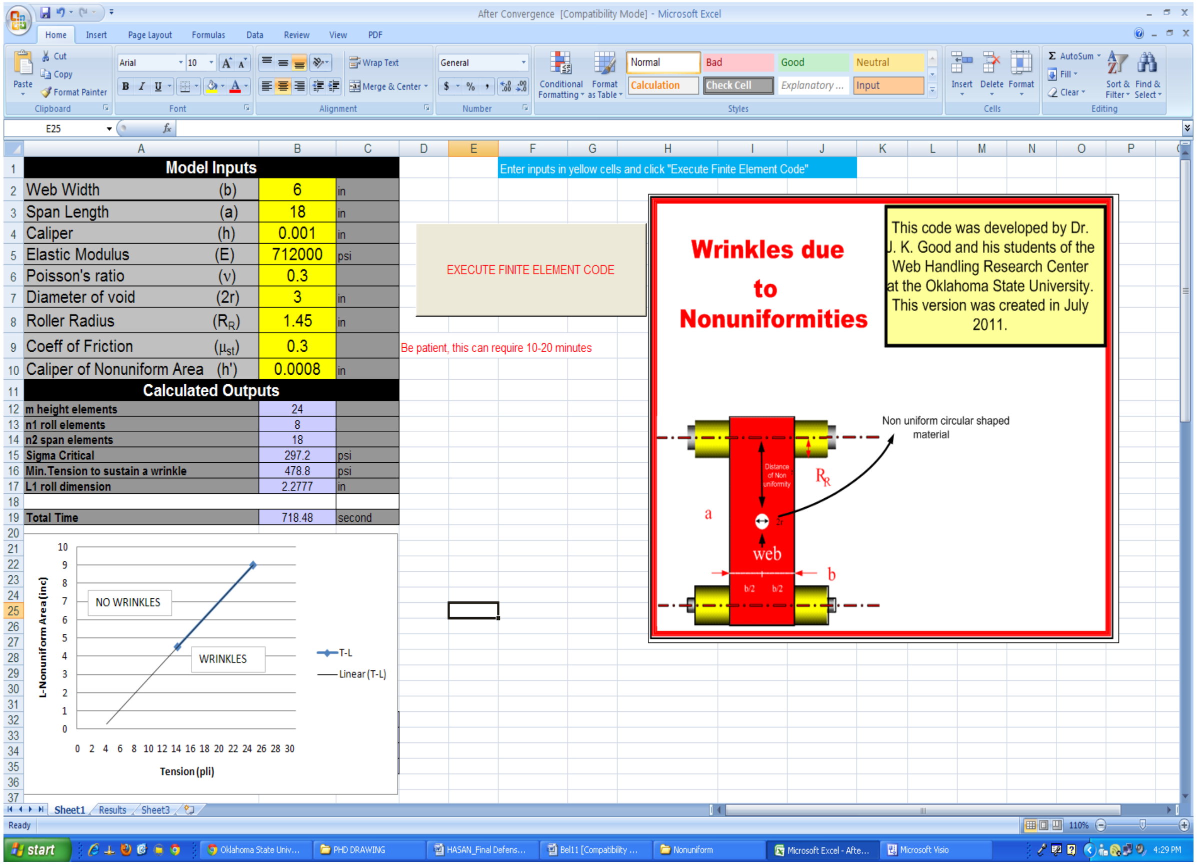
7.5 Automating the Excel VBA Code

The automation process was very similar to the void case in Chapter 6. The user is supposed to enter only basic parameters such as web width, roller radius, span length, web thickness, elastic modulus, Poisson's ratio, non uniformity size and thickness of the non uniform area. Firstly mesh parameters were decided. If the dimension of the web width was *x*, from the Excel interface the value *x* was multiplied with four and the even value of the result was attained to *m* and the web width was divided into *m* elements. If the dimension of the web length was *y*, from the Excel interface even value of *y* was taken as *n2* and the web length was divided into *n2* elements. If the value of one fourth of the roller circumference was *z*, the even integer value of three times of *z* was given to the *n1*. The roller area was divided into *n2* elements. The Excel equations were used to set these values. For very short, for very long and for extreme cases mesh parameters are limited. The meshing procedure and convergence check will be addressed more detailed at Chapter VIII.

To automate determining the minimum tension required to sustain a wrinkle over the surface of a roller a method similar to that used for the void case was employed (Section 6.4). Like the void case T_w (min.tension required to sustain a wrinkle) and 2 T_w were used as a starting point for *Linear Interpolation I* . After *Linear Interpolation I* and *Linear Interpolation II* the critical tension that buckles the web was found for the non uniform area for the specific distance from the second roller.

By using the linear behavior of the problem ¼ of the span length and ½ of the span length were used. The problem was meshed and the tension was calculated when the non uniform area was at a distance of ¼ of the span length by using *Linear Interpolation I* and *Linear Interpolation II*. Then, the tension was calculated when the non uniform area was at a distance of ½ of the span length by using *Linear Interpolation I* and *Linear Interpolation II.* Since the behavior always appears linear like the void case, straight line behavior is assumed for the tension levels at intermediate location. The flowchart is similar to the void case (Fig.6.14).

After automating the code the Excel interface of the code will look like Figure 7.13.



Enter inputs in yellow cells and click "Execute Finite Element Code"

Be patient, this can require 10-20 minutes

This code was developed by Dr. J. K. Good and his students of the Web Handling Research Center at the Oklahoma State University. This version was created in July 2011.

Model Inputs			
Web Width	(b)	6	in
Span Length	(a)	18	in
Caliper	(h)	0.001	in
Elastic Modulus	(E)	712000	psi
Poisson's ratio	(ν)	0.3	
Diameter of void	(2r)	3	in
Roller Radius	(R_R)	1.45	in
Coeff of Friction	(μ_{st})	0.3	
Caliper of Nonuniform Area	(h')	0.0008	in
Calculated Outputs			
m height elements		24	
n1 roll elements		8	
n2 span elements		18	
Sigma Critical		297.2	psi
Min.Tension to sustain a wrinkle		478.8	psi
L1 roll dimension		2.2777	in
Total Time		718.48	second

Figure 7. 13 The Excel Interface of Non uniform Excel VBA Code

The user is supposed to input the web width, span length, web thickness, elastic modulus, Poisson’s ratio, diameter of void, roller radius, the coefficient of friction, and different from the void code, the thickness of the non uniform area. With the help of Excel equations the code calculates mesh parameters, Timoshenko bucking criteria (from Eq.2.1), the minimum tension to sustain a wrinkle (from Eq.6.3), and one fourth of the circumference of the roller (L1 roll dimension).

When user clicks the *EXECUTE* button the code calculates the tension that will buckle the web when the non uniform area is ¼ of the span length away from the circumference line (here 18/4 = 4.5”). Then the code calculates the tension that will buckle the web when the non uniform area is 1/2 of the span length away from the circumference line (here 18/2=9”). By using these two points the relation between the tension and the distance of the non uniform area is plotted.

It can be said that an Excel VBA code that can find the relation between the non uniform area distance from the second roller versus the tension have been written successfully.

8. CHAPTER VIII

8.1 CONVERGENCE CHECK

All finite element analysis requires convergence checking. In this development quadrilateral elements have been employed. They have been employed to model the onset of wrinkling for a uniform web encountering a misaligned roller and a tapered roller. These elements were also used to nonuniform webs approaching aligned cylindrical rollers. The non uniformity could take the form of a centrally located circular void. It could also take the form of a centrally located circular region whose thickness was either less or greater than the surrounding web.

Each case presents different distributions of MD and CMD stresses in the web and each case is meshed differently. Algorithms will be developed for each case to ensure convergence.

8.2 The Misaligned Roller Case

In Chapter IV a VBA algorithm was presented to predict the onset of wrinkling for a web approaching a misaligned roller. The rules used to generate meshes were sufficient to produce a threshold misalignment angle, associated with the onset of wrinkling, which matched Beisel's test results.

The user is prompted to input the real span width *x* in units of inches. The Excel interface uses the Even Function to convert the real variable x to an even integer, m (i.e. $4 < x \leq 6 \rightarrow m = 6$). The web width was then divided into *m* elements. If the value of *m* was less than six, the value six was assigned to the value of *m*. If the value of *m* was larger than thirty, the value thirty was assigned to the value of *m*. Thus narrow webs would be assigned no less than six elements across the web width and wide webs would be assigned no more than thirty elements across the web width.

The user is also prompted to input the dimension of the web span length (*y*). The Excel interface would then convert the real number *y* to an even integer *n2*. The span length was divided into *n2* elements. If the value of *n2* input was larger than fifty, the value fifty was assigned to the value of *n2*. If the value of *n2* input was less than six, the value six was assigned to the value of *n2*. Thus short web spans could be assigned no less than six elements and long web spans could be assigned no more than fifty elements.

The user is also prompted to enter the roller diameter. Only ninety degree wrap angles were considered for the misaligned roller. The length of web wrapping the roller was one-fourth of the roller circumference. The Excel interface converted six times the wrap length to an even integer which was assigned to the variable *n1*. Thus the wrap length was divided into *n1* elements. The value of *n1* was limited to not exceed thirty and not be less than six. The mesh density across the web width in this region was set equal to *m* elements across the web width to ensure compatibility with the entering free span, previously discussed. A generic mesh for the model is shown below.

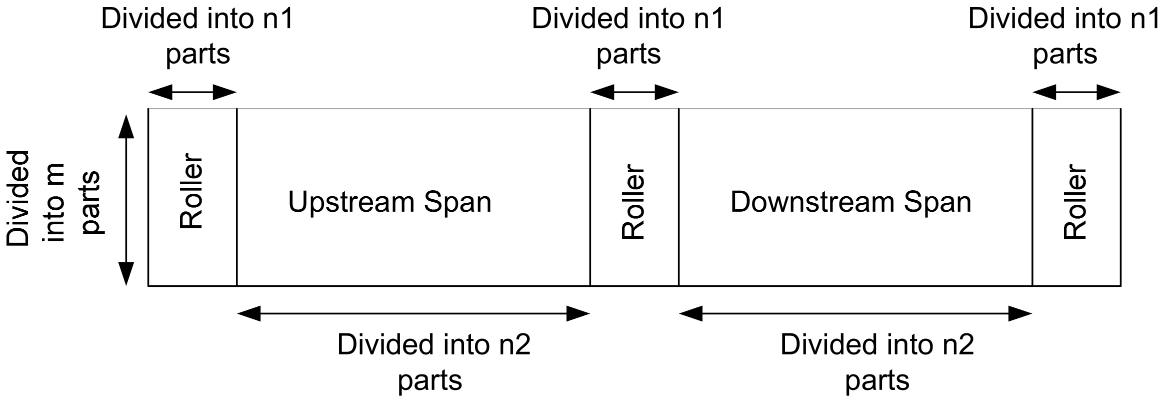


Figure 8.1 Meshing the Model

In Chapter IV, the mesh parameters that were described were shown to yield reasonable results compared with the experimental results. A convergence check must be done to ensure that the mesh parameters that were described would be accurate for cases other than those that had been tested. For this purpose the code was modified to check whether the results that were output were converged or not.

The code was updated to run two meshes in addition to the original mesh. After running the two additional meshes the results were compared with the results from the original mesh. If the results from the second mesh were within 5% of the results output for the first mesh the code would stop and the results output for the second mesh were provided as output. If the results from the first and second meshes were not within 5% the code was run again with yet a higher mesh density.

In this convergence check, the second mesh was the mesh which has already been described. The density of the first mesh was less than that of the second mesh. The third mesh density was greater than that of the second mesh. For the first and third meshes the mesh parameters *m* and *n2* values were either increased or decreased.

The flowchart for the code that incorporates convergence checks is shown in Fig.8.2.

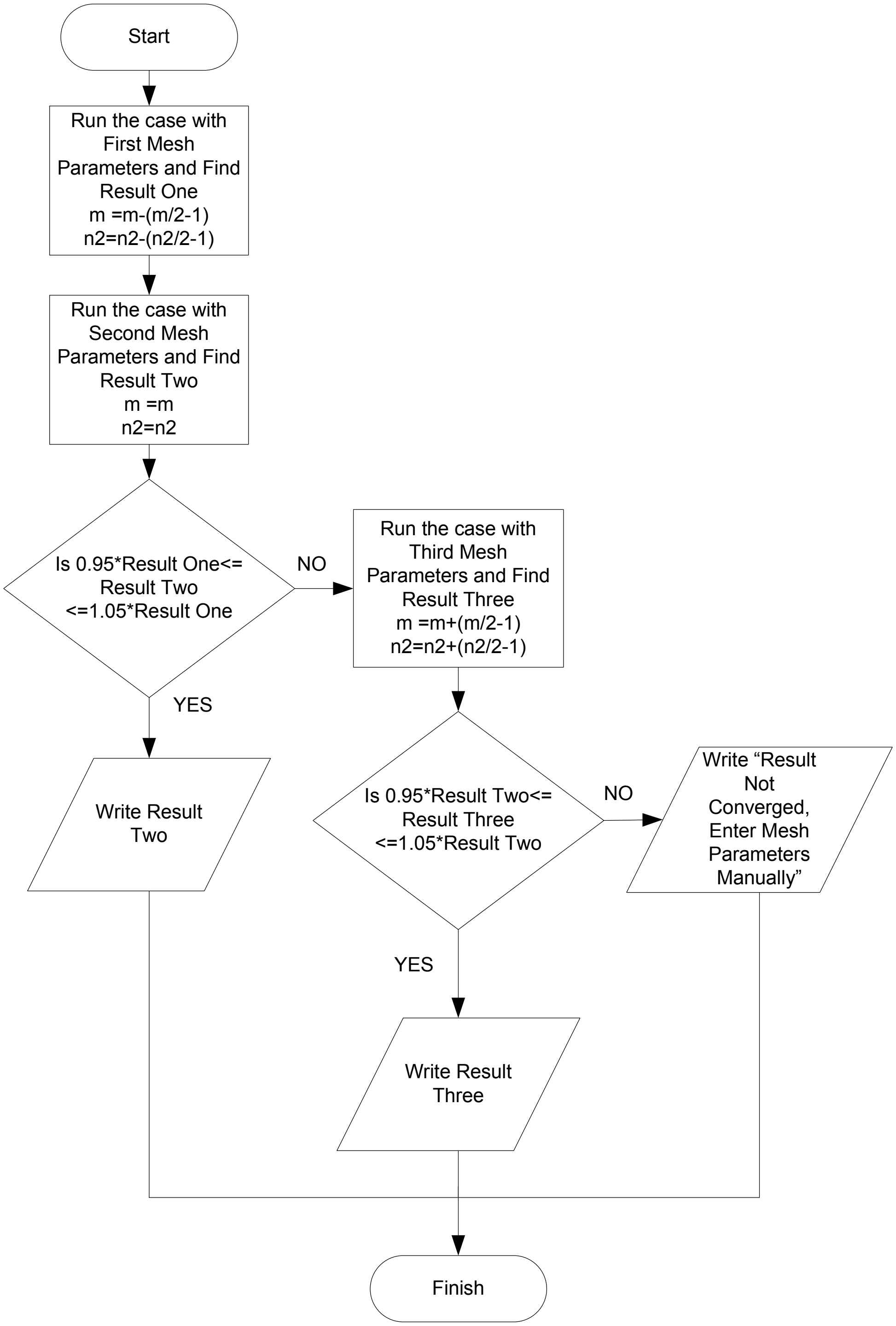

Start
Run the case with First Mesh Parameters and Find Result One
m =m-(m/2-1)
n2=n2-(n2/2-1)
Run the case with Second Mesh Parameters and Find Result Two
m =m
n2=n2
Is 0.95*Result One<= Result Two <=1.05*Result One
NO
Run the case with Third Mesh Parameters and Find Result Three
m =m+(m/2-1)
n2=n2+(n2/2-1)
YES
Write Result Two
Is 0.95*Result Two<= Result Three <=1.05*Result Two
NO
Write "Result Not Converged, Enter Mesh Parameters Manually"
YES
Write Result Three
Finish

Figure 8. 2 The Flowchart for the Convergence Check

After incorporating these convergence criteria into the code, the code was used to again analyze Beisel's test cases. The original results were shown in Figures 4.20-4.22. The new results are shown in Figure 8.3-8.5. Whichever mesh produced the converged result is shown in the charts. These results compare nicely with Beisel's test data and the results of his COSMOS models.

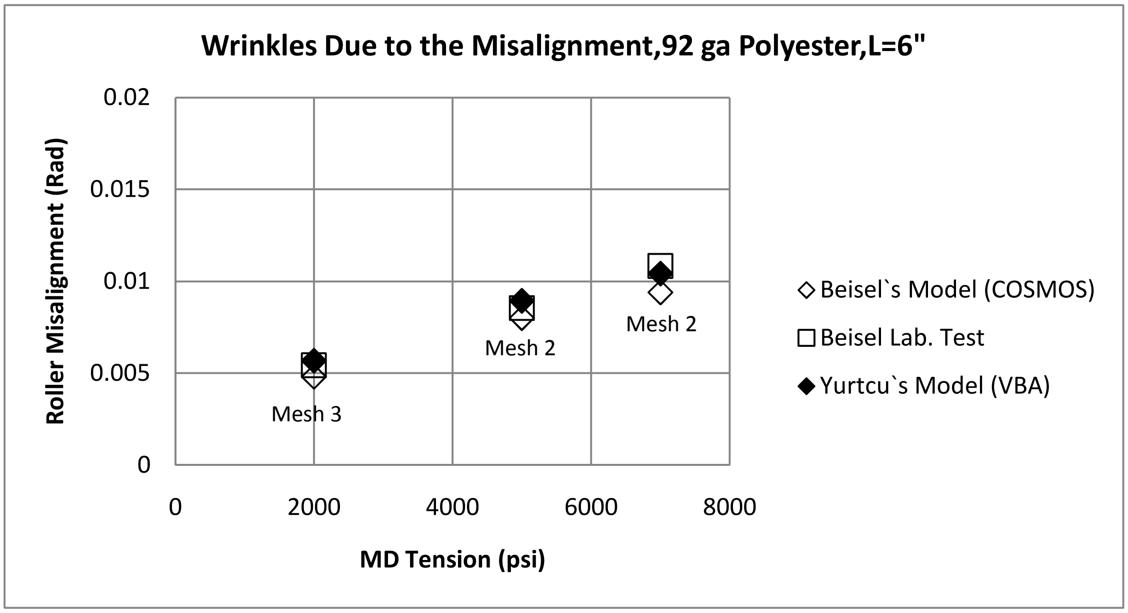


Figure 8.3 Comparison of 6” Span Case after the Convergence Check

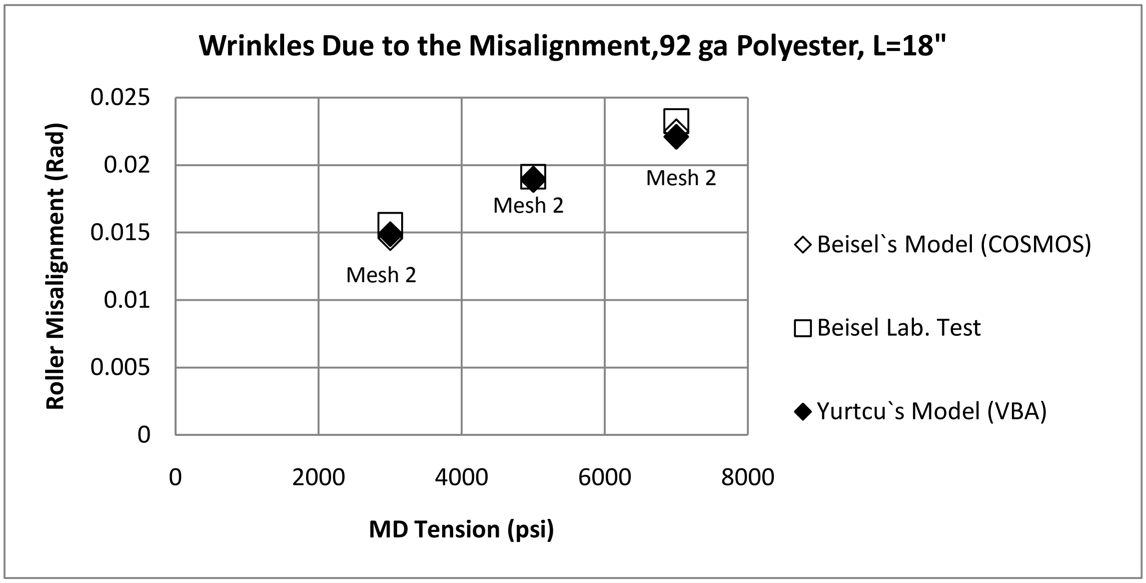


Figure 8.4 Comparison of 18” Span Case after the Convergence Check

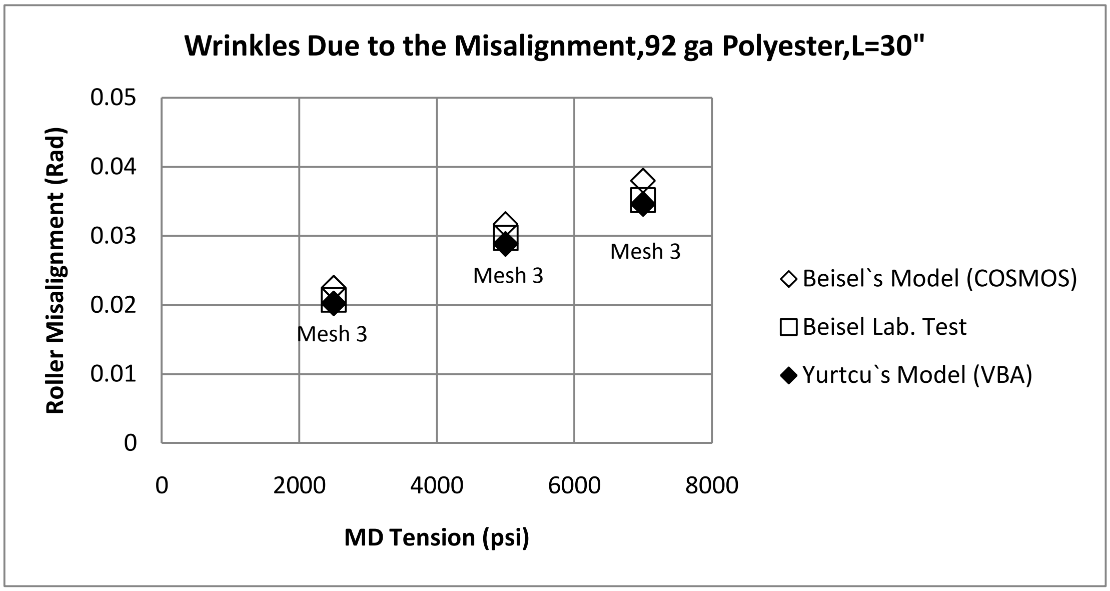


Figure 8.5 Comparison of 30” Span Case after the Convergence Check

In the Fig.8.6 the results for a specific case is shown as an example. Here span length is 30" and the tension is 7000 psi. The convergence criteria appear to work well for the test cases.

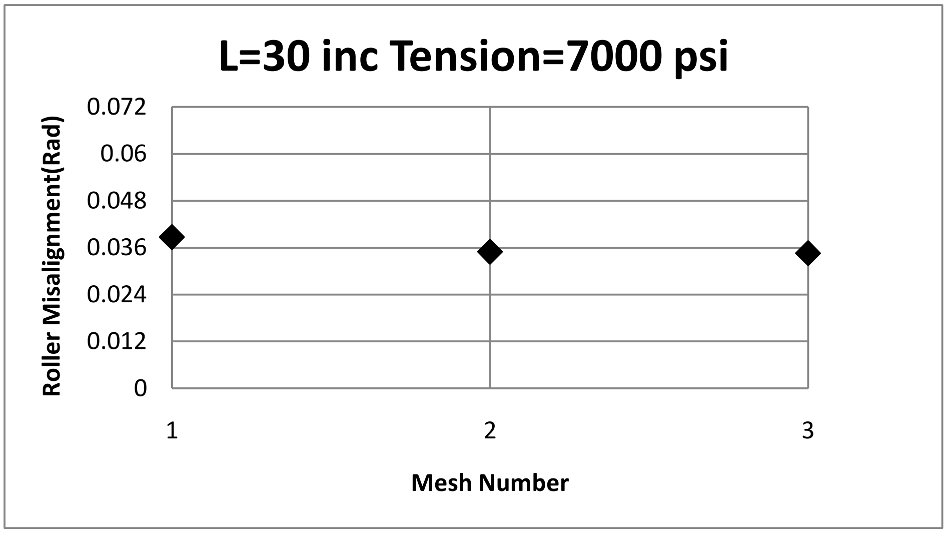


Figure 8.6 Converging Result for a Specific Case for Misaligned Roller Case

In the chart results from different meshes are plotted. As seen from the figure, after running the case with three different meshes result is converging.

It has been shown that the code converges nicely for cases from Chapter IV. Some other cases were picked to see whether the convergence routine works well. First case was a long narrow web. The free span length was 100 inches long and the web width was 6 inches wide. Other material properties were kept the same like the Beisel's test material properties. In the Fig.8.7 the result for this case is shown.

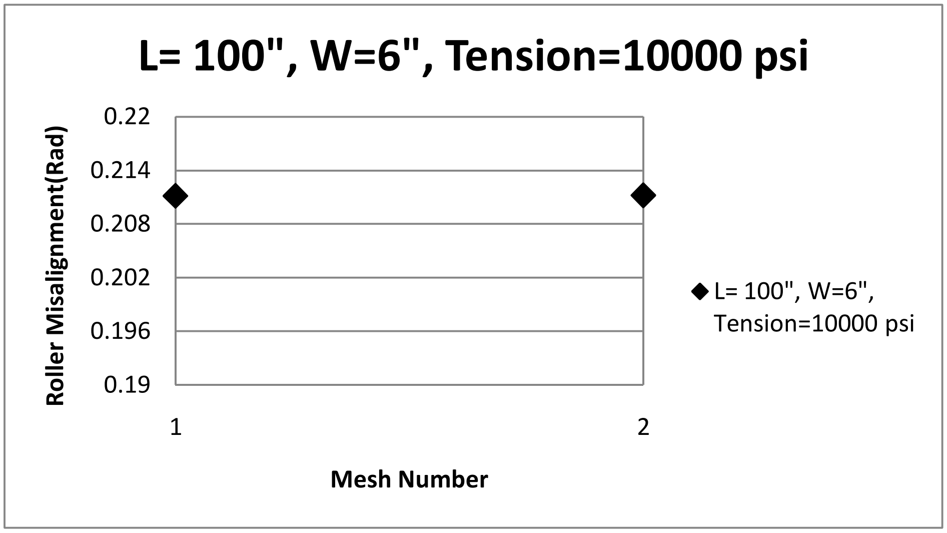


Figure 8. 7 Converging Result for a Long Narrow Web

In the second case a wide web was selected. The web was 50 inches wide and 30 inches long. In the Fig.8.8 result for this case is shown.

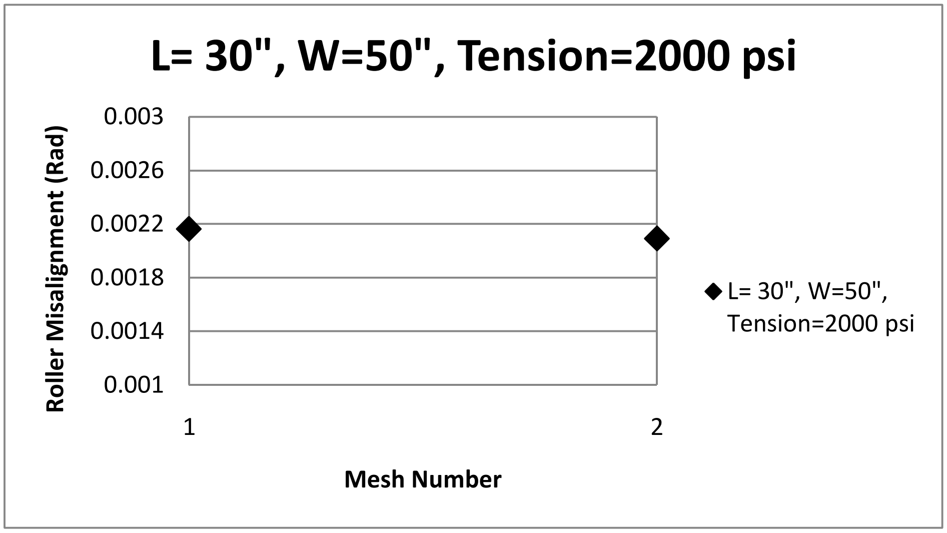


Figure 8. 8 Converging Result for a Wide Web

At the Fig.8.15 and Fig.8.16 results from different meshes are plotted. As seen from the figures the cases were converged after running two meshes.

As a last example a different material was selected. The web parameters for this web were a Young' Modulus of 725000 psi, a Poisson's ratio of 0.3, a web width of 50'' and the thickness was 0.002''. The rollers had a radius of 1.45'' and σ_{ycr} from the Eq .2.1 was about -605 psi. The case was modeled with COSMOS. The critical angle of misalignment of the downstream misaligned roller for the onset of wrinkling was calculated around 0.005 radians with COSMOS. In Fig.8.9 result from this case was shown.

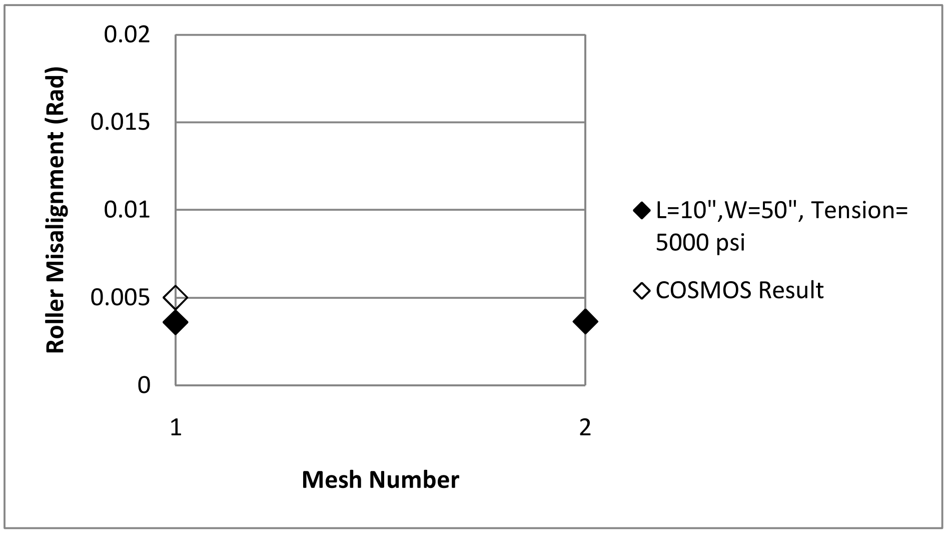


Figure 8. 9 Converging Result for 0.002" Thick 50" Wide Web

As seen from the figure after running the case with two different meshes result is converging.

8.3 The Tapered Roller Case

The development of the mesh parameters and convergence check for web wrinkling due to tapered rollers was similar to the development for the misaligned roller case.

If the dimension of the web width was *x*, from the Excel interface the value *x* was multiplied with three and the Even Function value of the result was assigned to *m*. Different from the misaligned roller case *x* was multiplied with three because during the calculation of moment more elements were needed for a sensitive calculation. If the value of *m* was less than six, the value six was given to the value of *m*. If the value of *m* was bigger than thirty, the value thirty was given to the value of *m*.

If the dimension of the web length was *y (inches)*, the Excel interface used the Even value of *y* to assign *n2*, the number of elements in the MD in the free span. The variable *n2* was limited to not exceed fifty and it was limited to be not less than six. The assignment of *n2* was identical to the misaligned roller case.

The even integer value of four times of the one fourth of the roller circumference was given to the mesh parameter *n1*. The *n1* value was limited to not exceed from thirty and limited to not less than six.

After calculating the values of *m, n1 and n2* the problem was meshed as shown in Figure 8.1. The values of *m, n1 and n2* were decided after running many cases. As shown in Chapter V the results calculated from the mesh parameters described herein were provided good agreement with the experimental results.

Similar to the misaligned roller case a convergence check must be done to verify that the mesh parameters that were suggested are valid for non test cases. For that purpose the tapered roller case code was updated to check convergence for different cases.

The code was updated to run for two meshes other than that which was just described. The first mesh was less dense than those which were described. The second mesh resulted from the mesh parameters that were described. After running the first and second mesh the code checked whether the result from second case was within 5% of the result that was output from the first case. If the results were within 5% the results that were calculated from the second case were output. If the result from the first and second mesh were not within 5%, the code would then compute results using the third mesh. The algorithm for that routine is like Fig.8.10.

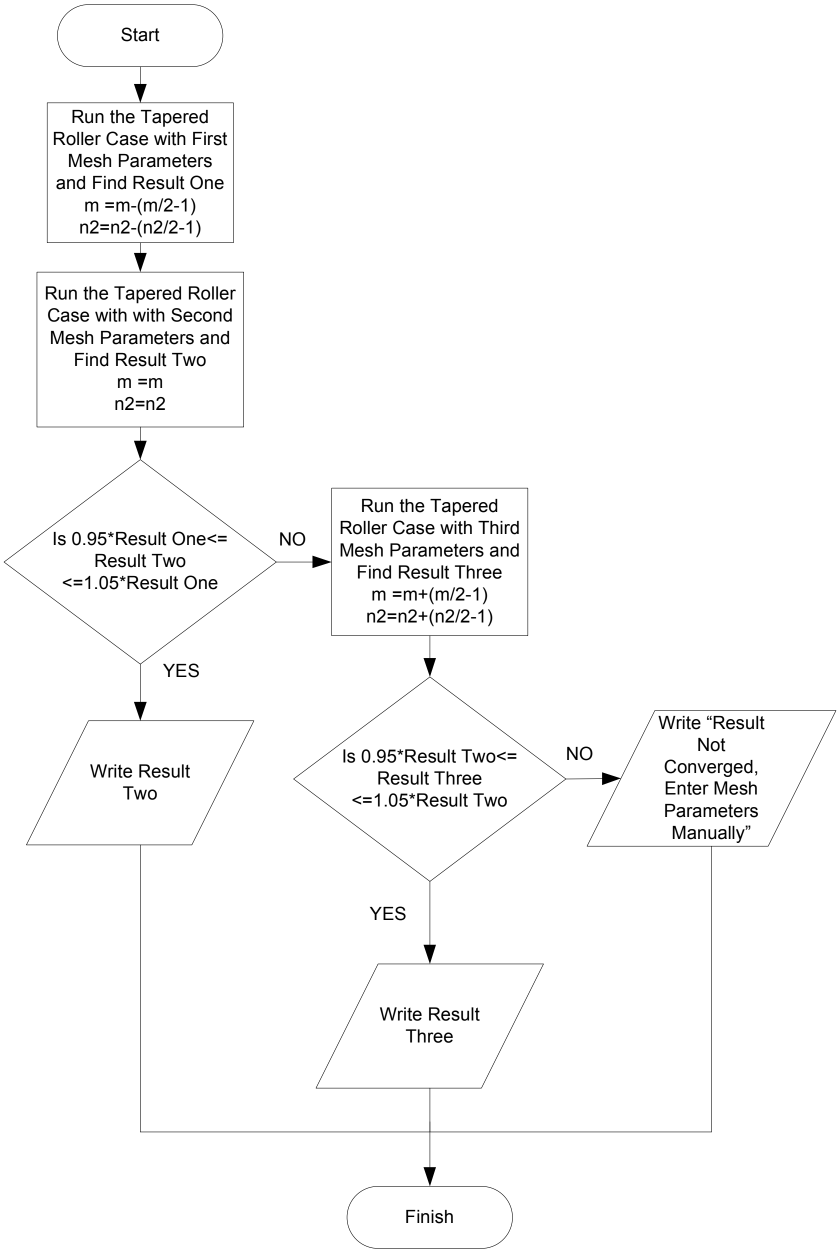

Start
Run the Tapered Roller Case with First Mesh Parameters and Find Result One m =m-(m/2-1) n2=n2-(n2/2-1)
Run the Tapered Roller Case with with Second Mesh Parameters and Find Result Two m =m n2=n2
Is 0.95*Result One<= Result Two <=1.05*Result One
NO
Run the Tapered Roller Case with Third Mesh Parameters and Find Result Three m =m+(m/2-1) n2=n2+(n2/2-1)
YES
Write Result Two
Is 0.95*Result Two<= Result Three <=1.05*Result Two
NO
Write “Result Not Converged, Enter Mesh Parameters Manually”
YES
Write Result Three
Finish

Figure 8. 10 The Flowchart for the Tapered Roller

After doing the chances the results shown previously in Fig.5.9-11were updated as shown Fig.8.11-13.

Whichever mesh produced the converged result is shown in the charts. These results compare nicely with Beisel's test data and the results of his COSMOS models.

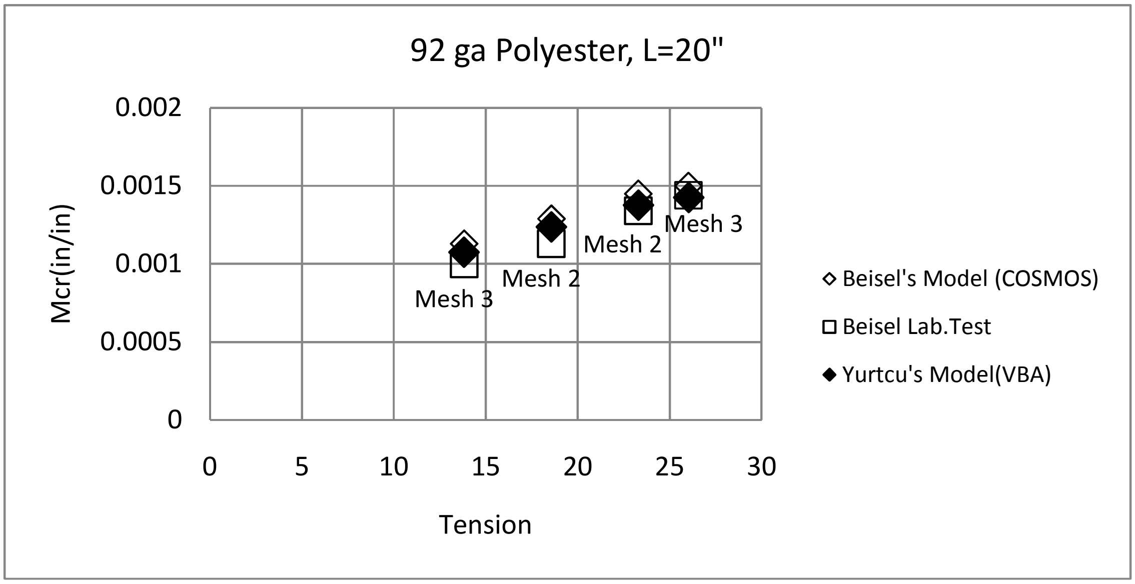


Figure 8.11 Converging Results for 20" Case

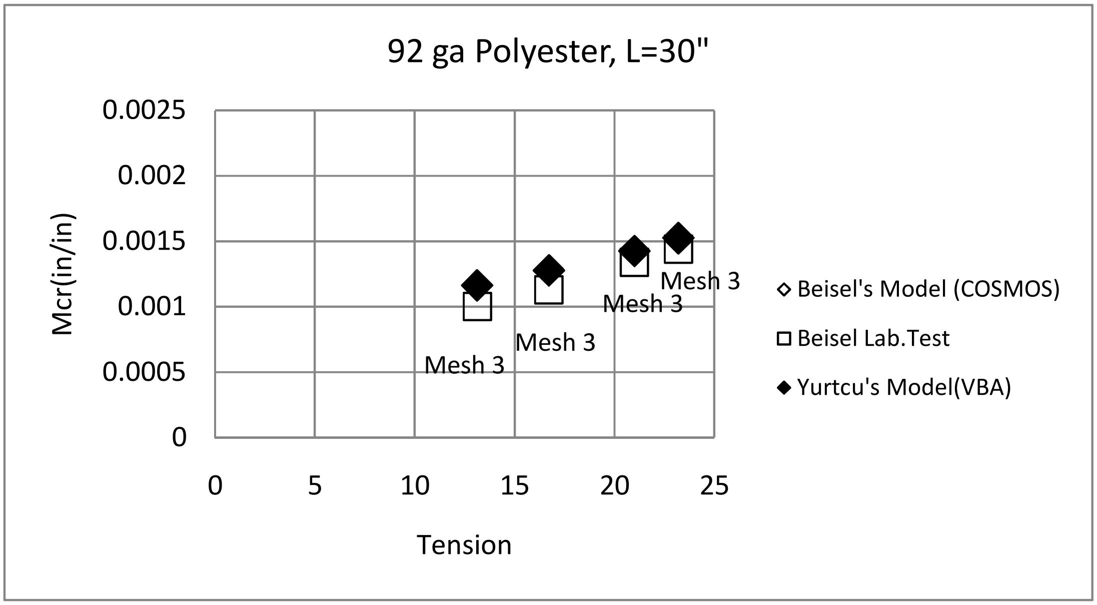


Figure 8.11 Converging Results for 30” Case

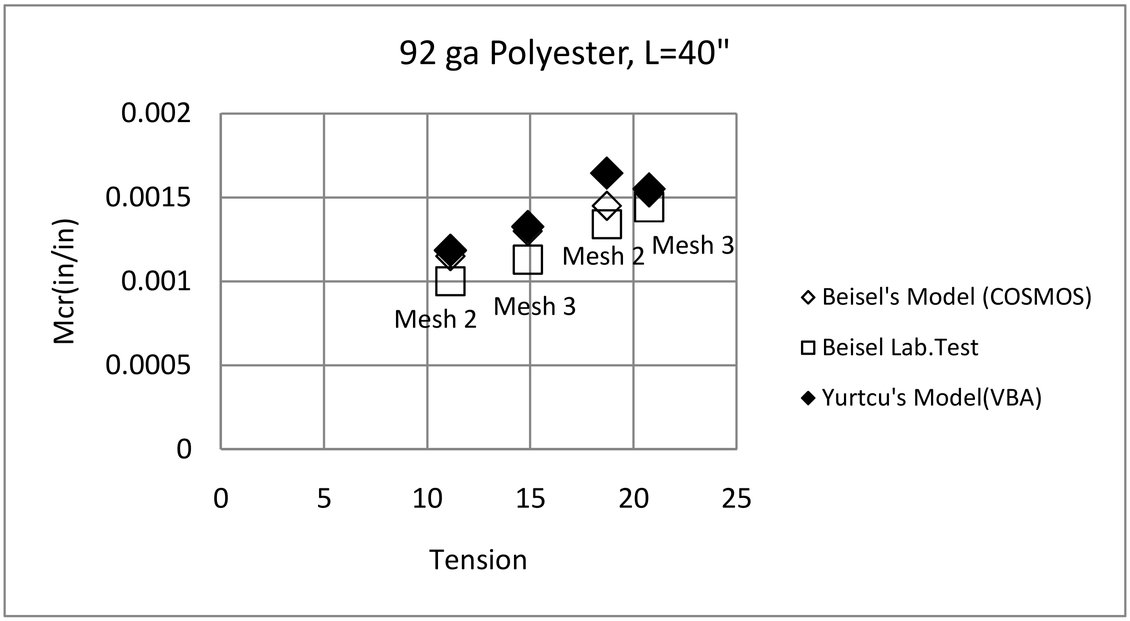


Figure 8.11 Converging Results for 40” Case

As an example of how the results converged after three runs, a specific case is shown in Fig.8.14. Here the span length is 30” and the web tension is 16.72 lbs.

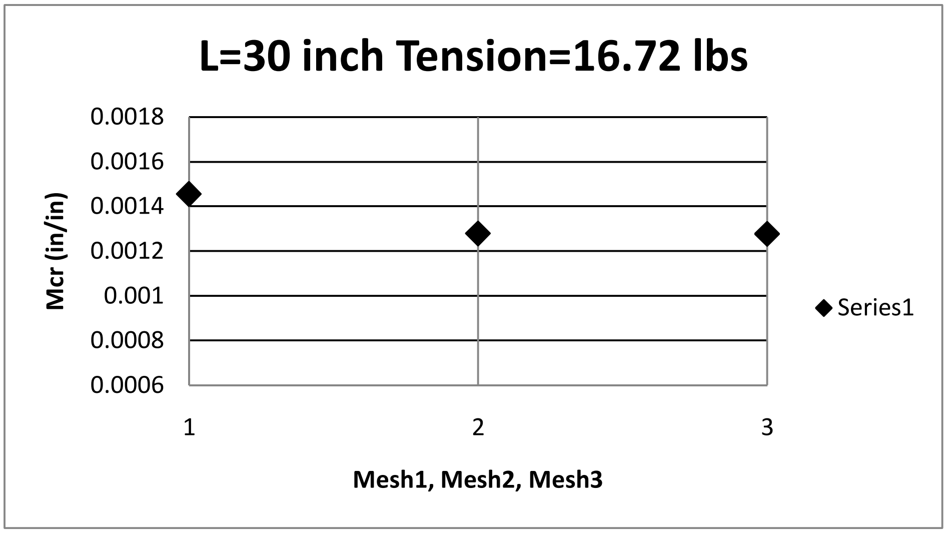


Figure 8. 14 Converging Result for a Specific Case for Tapered Roller Problem

It has been shown that the code converges nicely for cases from Chapter V. Some other cases were picked to see whether the convergence routine works well.

In the first case free span length was 60 inches long and the web width was 24 inches wide. Other material properties were kept the same like the Beisel’s test material properties. In the Fig.8.15 the result for this case is shown.

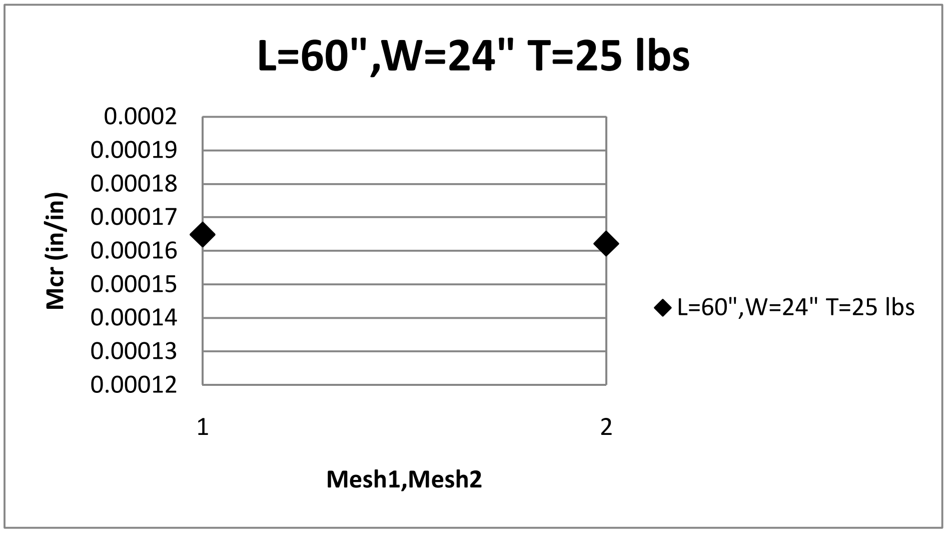


Figure 8.15 Converging Result for a 60” Long Web

In the second case, free span length was 80 inches long and the web width was 24 inches wide. In the Fig.8.16 the result for this case is shown.

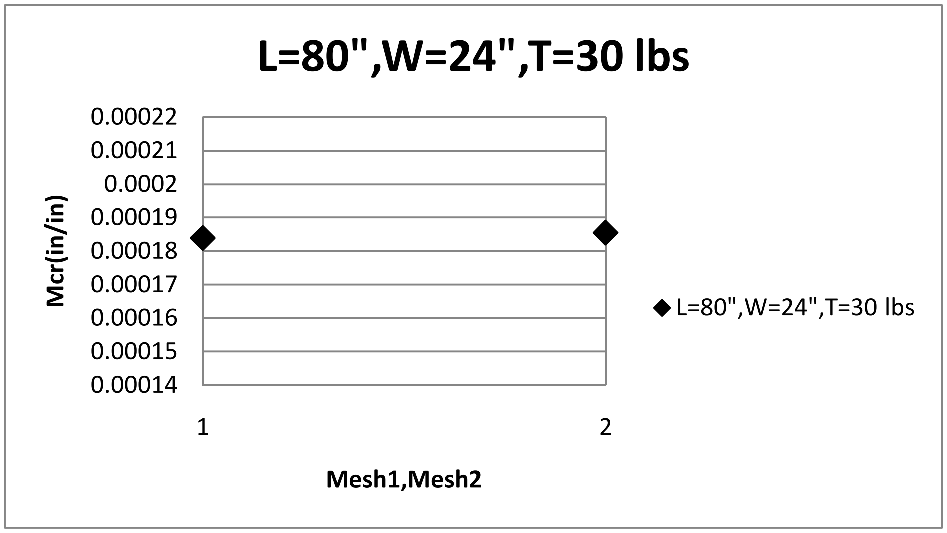


Figure 8.16 Converging Result for a 80” Long Web

At the Fig.8.15 and Fig.8.16 results from different meshes are plotted. As seen from the figures the cases were converged after running two meshes.

8.4 The Central Circular Void Case

For the circular void case the code was modified to check for convergence and to run for three different mesh parameters if needed.

If the user chooses the dimension of the web width as *x*, and the radius of the central void as *r*, the even integer value of (*x/r*) was assigned to the variable *m* which is the number of elements across the web width. The purpose of selecting *m* like that was to divide the web width homogeneously. The value of *m* was limited to not exceed forty eight and it was limited to be not less than twenty.

If the dimension of the web span length was *y (inches)*, the Excel interface used the Even value of *y* to assign *n2*, the number of elements in the *MD* in the free span. The variable *n2* was limited to not exceed fifty and it was limited to be not less than six. This parameter was set identically for the misaligned roller and the tapered roller case.

The web was assumed to wrap the roller 90 degrees. If the value of one fourth of the roller circumference was *z*, the even integer value of three times of *z* was given to the *n1*, the number of elements in the *MD* crossing the roller. The value of *n1* was limited to not exceed thirty and not be less than six.

These mesh parameters were the suggested mesh parameters and their validity was shown at chapter six to model cases that were tested in the laboratory. The values of *m* and *n2* for the first mesh used in the convergence code are shown below:

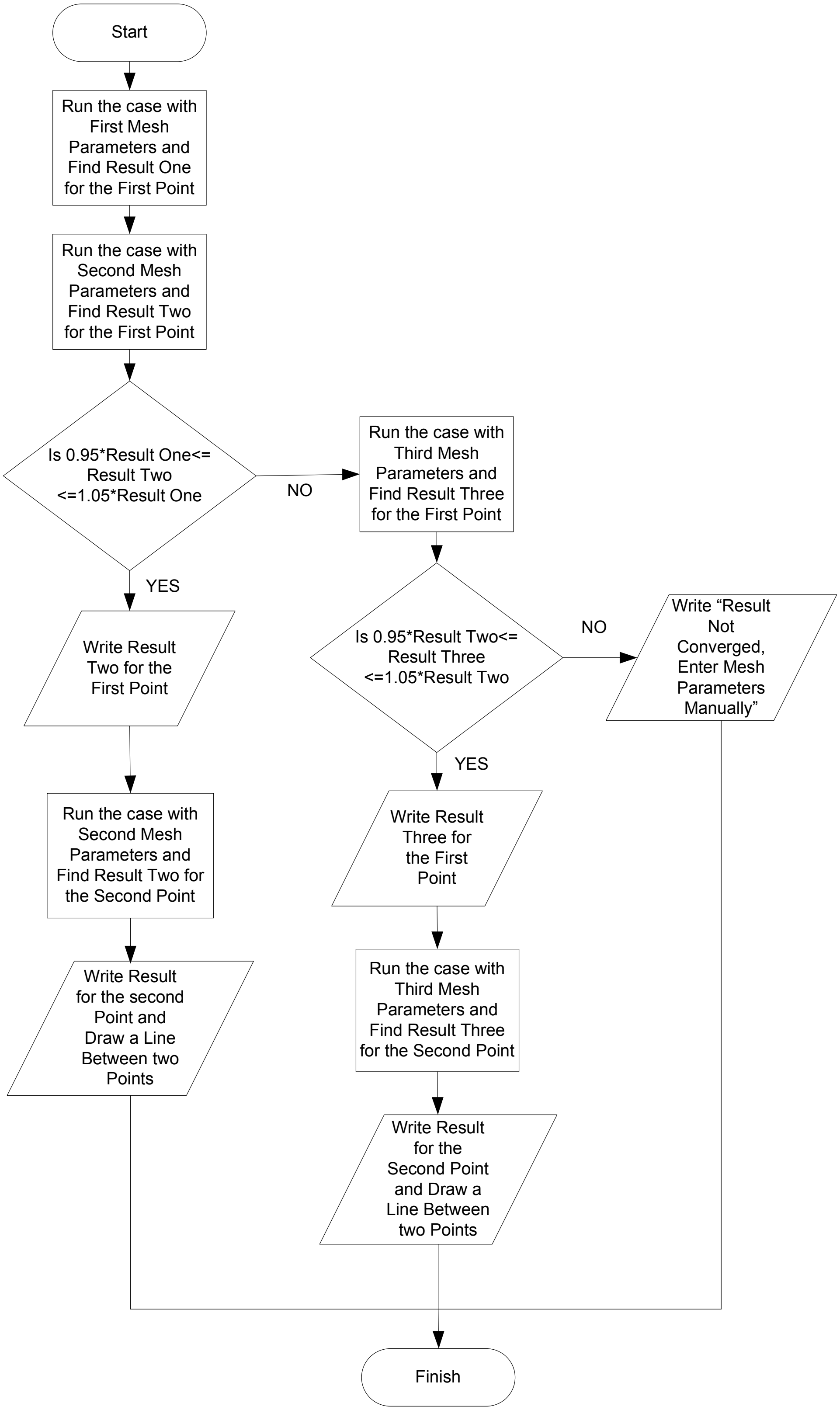
$$m = m - 8,\ n2 = n2 - 2 \tag{8.1}$$

For the second mesh the suggested values were used. Finally the third mesh the values of *m* and *n2* are shown below:

$$m = m + 8,\ n2 = n2 + 2 \tag{8.2}$$

In the void case the code was executed twice to study the impact of the hole location in the free span on the tension level required to induce wrinkles. The first run was executed when the circular hole was at a distance of ¼ of the span length from the downstream roller and the second run was executed when the circular hole was at a distance of ½ of the span length from the downstream roller. The code was modified to run for the first and second meshes for the first point and check whether the result from the second mesh was within 5% of the result that was calculated from the first mesh. If 5% relation was not valid the code was run with the third mesh. For the second point the choice of mesh was made based upon which mesh resulted in convergence for the first point. If the second mesh produced convergence for the first point the second point was also calculated with the second mesh. If the second mesh was not produced convergence for the first point, the second point was calculated using the third mesh.

The flowchart for the convergence check for the void case is shown in Fig.8.17.



Start
Run the case with First Mesh Parameters and Find Result One for the First Point
Run the case with Second Mesh Parameters and Find Result Two for the First Point
Is 0.95*Result One<= Result Two <=1.05*Result One
NO
Run the case with Third Mesh Parameters and Find Result Three for the First Point
YES
Write Result Two for the First Point
Is 0.95*Result Two<= Result Three <=1.05*Result Two
NO
Write "Result Not Converged, Enter Mesh Parameters Manually"
YES
Run the case with Second Mesh Parameters and Find Result Two for the Second Point
Write Result Three for the First Point
Write Result for the second Point and Draw a Line Between two Points
Run the case with Third Mesh Parameters and Find Result Three for the Second Point
Write Result for the Second Point and Draw a Line Between two Points
Finish

Figure 8. 17 Convergence Check for the Void Case

After updating the code the case seen at Fig.6.15 was run again. The result is shown at Fig.8.12.

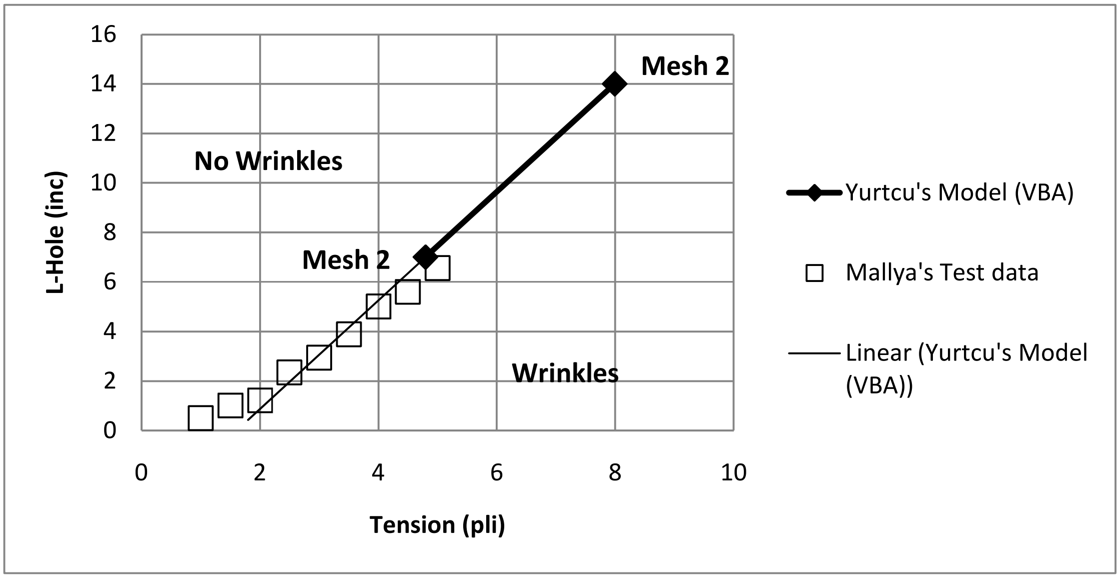


Figure 8. 18 Result of Modified Code for the Case at Figure 6.15

To show the convergence routine works well, two very different cases from those that produced the test data were executed. The first case was a wide web. The material properties and other dimensions were kept the same and the width of the web was increased to 50 inches. In the Fig.8.19 the result for this case is shown.

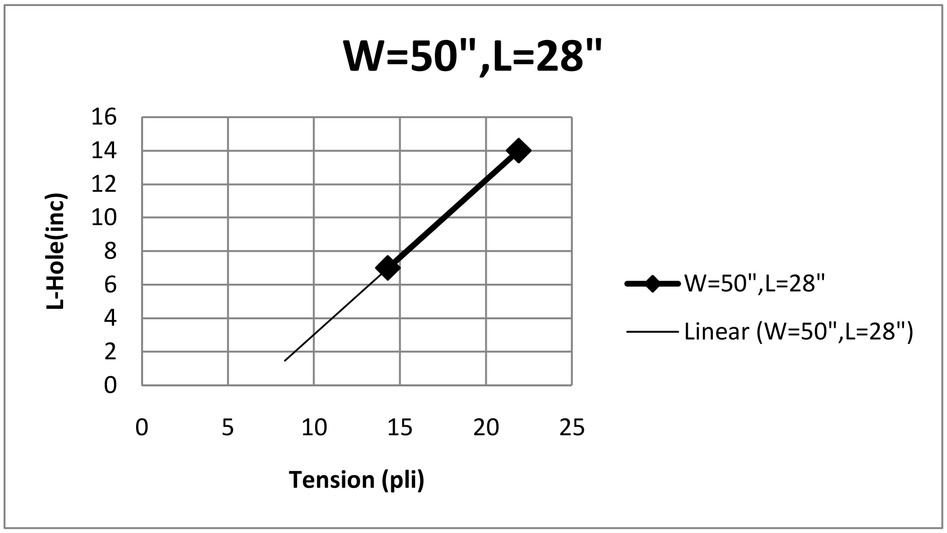


Figure 8.19 Convergence Result for a Wide Web for Circular Void Code

The second case was a long web. The material properties and other dimensions were kept the same and the length of the web was increased to 100 inches. In the Fig.8.20 the result for this case is shown.

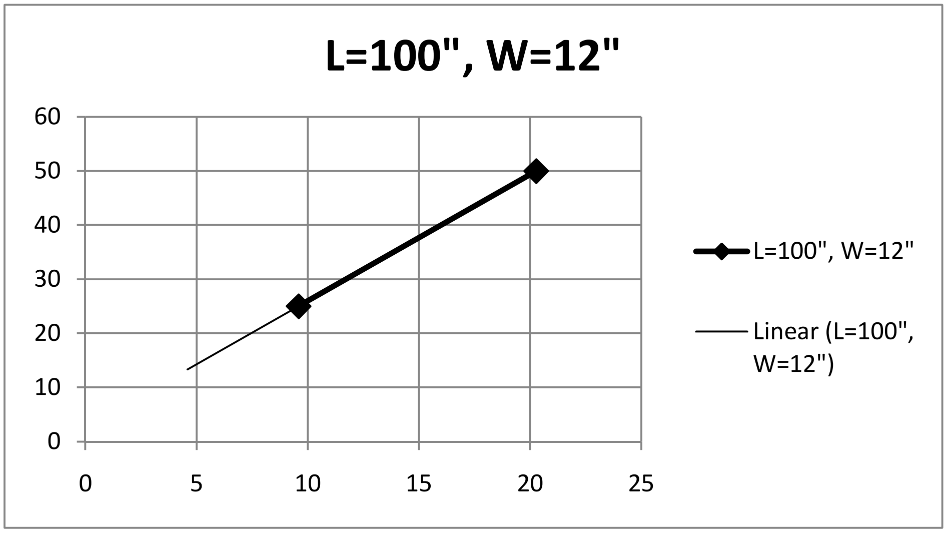


Figure 8.20 Convergence Result for a Long Web for Circular Void Code

8.5 The Central Circular Non Uniform Web Case

For the non uniformities case the code was modified to check convergence very similar to the void case.

If the dimension the user input for web width was *x (inches)*, the Excel interface multiplied the value *x (in)* by four and the Even function value of the result was assigned to variable *m*, the number of elements across the web width. If the value of *m* was less than twenty two, the value twenty two was given to the value of *m*. If the value of *m* was larger than fifty, the value fifty was assigned to the value of *m*. This is different from the void case because the central circular non uniform region may occupy a larger area than the central circular void.

If the dimension the user input for web span length was *y (in)*, the Excel interface evaluated the Even Function value of *y* and assigned it to the variable *n2*. The variable *n2* was limited to not exceed fifty and it was limited to be not less than ten.

The web was assumed to wrap the roller 90 degrees. If the value of one fourth of the roller circumference was *z*, the even integer value of three times of *z* was given to the *n1*, the number of elements in the *MD* crossing the roller. The value of *n1* was limited to not exceed thirty and not be less than six.

Similar to the void case in the non uniformities case was run twice to determine the impact of the non uniformity location in the web span on the web tension required to induce a wrinkle. For the first point the code was run for mesh one and mesh two. For the first mesh the values of *m* and *n2* were taken from Eq.8.1. The second mesh parameters were those described in the previous paragraphs. The code was executed the results from the second mesh were reviewed to determine if they were within 5% of the result from the first mesh. If so, the output result that was calculated from the second mesh was output as the converged values. If the results from meshes were not within 5%, the code then executed the third mesh. If the result from mesh three were within 5% of those from mesh two, the mesh three results were output as the converged values. For the third mesh the values of *m and n2* were taken from Eq.8.2. The flowchart for the non uniform case is similar to the flowchart that is shown in Fig.8.17. After modifying the code the case seen at Fig.7.13 was run again. The results are shown in Fig.8.21.

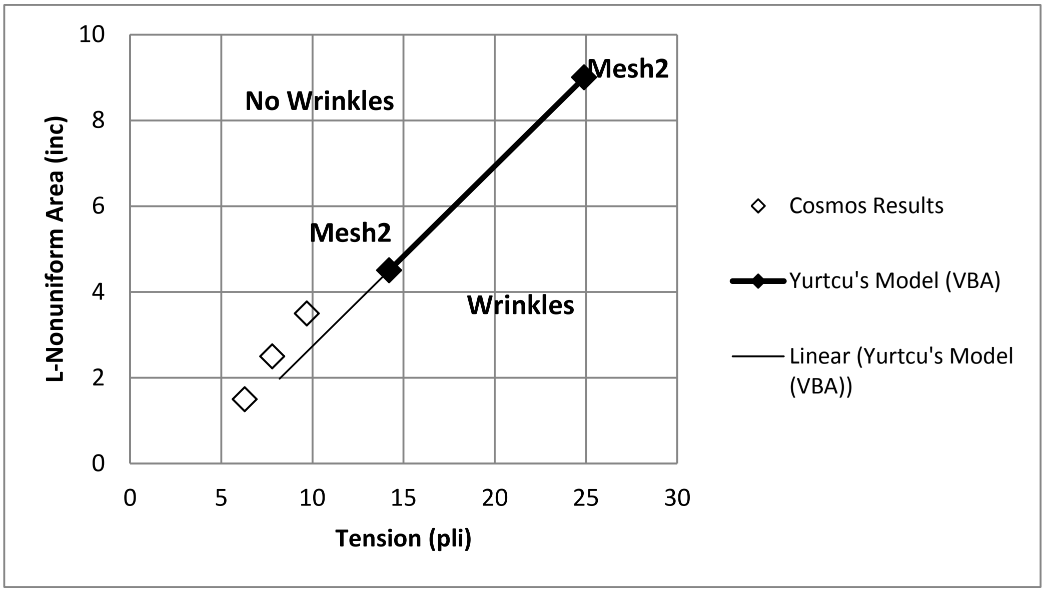


Figure 8. 21 Result of Modified Code for the Case at Figure 7.15

To check whether the routine produces convergence or not a case from Fig.7.12 was selected. It was a 50 inch wide and 50 inch long wide web. The web modeled was 0.001 inch thick. The roller radius was 1.45 inch. The web's Young's modulus was 712000 psi and the

Poisson’s ratio was assumed to be 0.3. The non uniform area was 3 inch away from the second roller and the diameters of non uniform area was 3 inches. In Fig.8.22 the results from COSMOS and Yurtcu’s modified Excel VBA are shown.

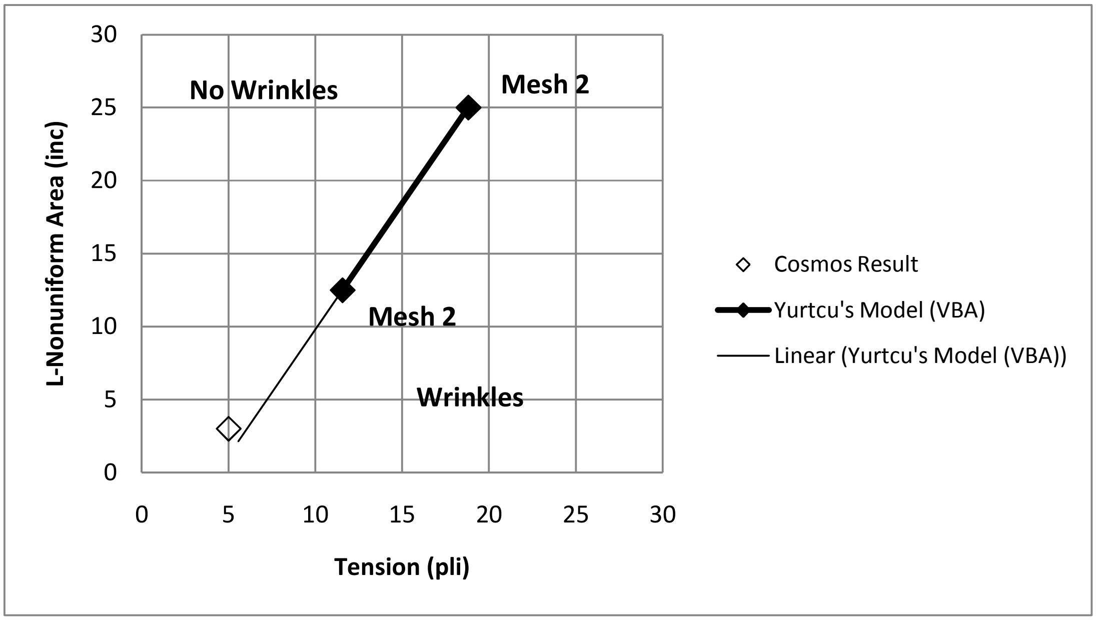


Figure 8. 22 Result of Modified Code for a Wide, Long Web

9. CHAPTER IV

9.1 SUMMARY

The objective of the proposed research was to determine whether user friendly finite element codes could be developed that would solve nonlinear instability problems associated with strain state dependent material properties and boundary conditions of moving webs.

The results of the work show that for the case of a web encountering a misaligned roller, a web encountering a tapered roller, for the case of central circular discontinuity, and for the case of central circular non uniformities this is in fact possible. The codes developed rely on inputs of simple geometry and material parameters which should be apparent to the user. From there on the code executes automatically in the meshing of the problem, the generation of the elastic and wrinkle membrane elements, and the solution of the nonlinear instability problem. The results from this codes have been shown to compare favorably with test results and results from commercial FE codes.

Interpolation methods were used successfully while automating the code. During the first interpolation process two tension levels that were too low to wrinkle the web were chosen. At the end of the first interpolation process these two starting tension levels were increased in such a way that one was greater than the tension that will buckle the web and the other was less than the tension that will buckle the web. After the second interpolation process the code was found the tension that would buckle the web.

While working on misaligned roller case a new slack edge criterion that could be used for both long spans and short spans was established.

While modeling the hole case a circle hole, a square hole and a equilateral quadrangle hole were compared and it is found that among these three hole shapes the quadrangle shaped hole could withstand more tension prior to wrinkling. Thus if cutting voids in webs is necessary, this hole shape is more robust than others in resisting wrinkles.

9.2 RECOMMENDATIONS FOR THE FUTURE STUDY

In this study, problems are modeled in five sections: the upstream roller, the upstream span, the downstream roller, the downstream span and the existing roller. In the future these problems can be modeled in seven or more sections and the behavior of the problem can be observed in more detail.

For the tapered roller case, the model that was suggested had limitations for short web spans. In the future, if boundary conditions that would cover both long web spans and short web spans the code that calculates the taper for rollers could be modified to give reasonable results for short web spans too.

For the misaligned roller case, the tapered roller case and the centrally located circular void case, the results of the Excel VBA code compared well with experimental results. For the non uniform web case the results of the Excel VBA code were not compared with the experimental results, since none existed. In the future, if successful experiments are completed, the results of the code could be compared with the experimental results. Since the VBA code results and COSMOS results compare well, there is at least numerical validation.

REFERENCES

1. Roisum, D.R., The Mechanics of Wrinkling, in Finishing and Converting Conference,1996

2. Beisel, J.A, Prediction of Single Span Web Buckling and Experimental Verification, M.S.Thesis, Oklahoma State Unv. ,July 2000

3. Lorig, E.T., Automatic Self Centering Rolls and Pulleys, in AISE Convention.1950, Cleveland, Ohio.

4. Good, J.K., D.M.Kedl and J.J.Shelton, Shear Wrinkling in Isolated Spans, Proceedings of the Fourth International Conference on Web Handling, Stillwater, OK,WHRC, Oklahoma State University, June 1997

5. Miller R.K.,Hedgepeth J.M.,An Algorithm for Finite Element Analsis of Partly Wrinkled Membranes, AIAA Journal,December,1982, pp.1761-1763

6. Miller R.K.,Hedgepeth J.M., Wiengarten V.I., Das P.,Kahyai S.,Finite Element Analsisis of Parly Wrinkled Membranes, Computers & Structures Vol 20, No1-3, pp 631-639,1985

7. Papandreadis, A., The Development of Finite Element Modelling Techniques of Webs and the Analysis of Web Wrinkle Formation, M.S. Thesis, Oklahoma State Unv.,July 1986.

8. Benson, R.C.,Chiu H.C.,LaFleche J., Stack K.D., Diehl T., Simulation of Wrinkling Patterns in Webs due to Non uniform Transport Conditions, Proceedings of the Second International Conference in Web Handling, Stillwater, OK, WHRC, Oklahoma State Unv., June 1993

9. Webb, D.K., Prediction of Web Wrinkles Due to Misalignment of a Downstream Roll in a Web Span, M.S.Thesis, Oklahoma State Unv.,Dec.2004

10. Good,J.K.and Beisel J.A.,Buckling of Orthotropic Webs in Process Machinary, Proceedings of the Seventh International Conference on Web Handling, Stillwater.OK,WHRC, Oklahoma Stete Unv.,June 2003

11. S.P Timoshenko, J.M.Gere, Theory of Elastic_Stability, 1st ed., McGraw-Hill, New York, 1936

12. Beisel A.B.,Single Span Web Buckling Due to Roller Imperfections in Web Process Machinery,Ph.D.Dissertation , Oklahoma State Unv.,October 2006

13. Allen, H.G. and Bulson, P.S., Background to Buckling, McGraw-Hill, Maidenhead, 1980.

14. Shelton, J.J., “Buckling of Webs from Lateral Compressive Forces”, Proceedings of the Second International Conference on Web Handling, WHRC, Oklahoma State University, June 1993.

15. Mallya, S.,Investigation of the Effects of Voids on the Stability of Webs, M.S.Thesis, Oklahoma State Unv.,Dec.2007

16. Kara I., Wrinkle Formation Due To Non-Uniform Web Length, M.S.Thesis, Oklahoma State Unv., July 2008.

17. Shelton,J.J., Lateral Dynamics of a Moving Web, PhD Thesis, Oklahoma State Unv.,1968

18. H.Wagner, Flat Sheet Metal Girders with Very Thin Metal web. Z.Flugtechn.Motorluftschiffahrtt 20, 8-12 (Translation into English, NACA TM 604-606)(1929)

19. M.Stein and J.M.Hedgepeth, Analysis of Partly Wrinkled Mebranes, NASA TN D-813(1961)

20. M.M.Mikulas, Jr., Behavior of a Flat Stretched Membrane Wrinkled by the Rotation of an Attached Hub. NASA TN D-2456(1964)

21. M.M.Mikulas, Jr., Behaviour of Doubly Curved Partly Wrinkled Membrane Structures Formed From an Initially Falt Membrane, Ph.D.Thesis, Virginia Polytechn.Ins.,Blacksburg, Virginia,1970

22. R.K.Miller, J.M.Hedgepeth, V.I.Weingarten, P.Das and S.Kahyai, Finite Element Analysis of partly Wrinkled Membranes.Rep.USC-CE-8305.Dep.of Civil Eng., Univ..of Southern California, Los Angeles,California , 1983

23. Reissner, E., On tension Field Theory, Proc.5th Congress of Applied Mechanics, p.88-92,1938

24. Wu,C.H., Nonlinear Wrinkling of Nonlinear Membranes of Revolution, ASME J.Applied Mechanics, 45,p.533-538,1978

25. Wu,C.H., Large Finite Strain Membrane Problems, Q.Applied Mathematics, p.347-359,1979

26. Wu,C.H. and Canfield, T.R., Wrinkling in Finite Plane-Stress Theory, Q.Applied Mathematics,p.179-199,1981

27. J.Walkenbach, Excel VBA Programming for Dummies, Wiley Publishing,2004

28. D. R.Roisum, The Mechanics of Wrinkling, Finishing and Converting Conference/1,1996

29. D. R.Roisum, Thin Materials:Pushing Web Handling to the Limit, Converting Magazine, 17,5,May 1999

30. D. R.Roisum, Wrinkling of Thin Webs, Socety of Vacuum Coaters, 505/856-7188,41 st Annual Technical Conference Proceedings,1998

31. D. R.Roisum, Thinking Thin, Association of Industrial Metallizers,Coaters and Laminators, January 2003

32. L.S Gehlbach, J.K.Good, D.M.Kedl, Prediction of Shear Wrinkles in Web Spans, TAPPI Journal, Vol.72, No:8,129-134,1989

33. H.Gopal and D.M.Kedl, Using Finite Element Method to Define How Wrinkles Form in a Single Web Span Without Moment Transfer, Proceedings of the First International Conference on Web Handling, Stillwater, OK,WHRC, Oklahoma State

Unv.,1991

34. H.Hashimoto, Prediction Model of Pape-Web Wrinkling and Some Numerical Calculation Examples with Experimental Verifications, Proceedings of the Eighth International Conference on Web Handling, Stillwater, OK,WHRC, Oklahoma State Unv., 2005

35. D.P.Jones and M.J.McCann, Wrinkling of Webs on Rollers and Drums, Proceedings of the Eighth International Conference on Web Handling, Stillwater, OK,WHRC, Oklahoma State Unv., 2005.

36. Chandrupatla T.R. and Belegundu A.D., Itroduction to Finite Elements in Engineering, 3^{rd} Ed., Prentice Hall, 2002.

37. O.C.Zienkiewicz and Morgan, Finite Elements and Approximation Dover Books on Engineering,1983.

38. Good,,J.K., MAE 5563 Finite Element Analysis Class Notes,2007

39. Swift, H.W., Camber For Belt Pulleys, Proceedings-Institute of Mechanical Engineers, June 1932.

40. J.K. Good. Shear in multi span web systems. Proceedings of the Fourth International Conference on Web Handling, Web Handling Research Center, Stillwater, Oklahoma. June 1- 4, 1997.

41. Beisel J.A, Yurtcu H.H. and Good, J.K.The Instability of Webs in Transport. Proceedings of the Tenth International Conference on Web Handling, Web Handling Research Center, Stillwater, Oklahoma. June 7-10, 2009.

42. Good J.K. and Beisel J.A., Calculations Relating to Web Buckling Resulting from Roller Misalignment, TAPPI, 2006.15(12):p9-16

43. ASTM D1894-06 Standard Test Method for Static and Kinetic Coefficients of Friction of Plastic Film and Sheeting. Annual book of ASTM standarts.Vol.08.01

44. Shelton, J.J., "Derivation of Equations of Lateral Behavior", IAB Review, Stillwater, OK, WHRC, Oklahoma State University, June 1998.

45. Good, J.K. and J.A. Beisel, "Buckling of Orthotropic Webs in Process Machinery", Proceedings of the Seventh International Conference on Web Handling, Stillwater, OK, WHRC, Oklahoma State University, June 2003.

46. Brown, J.L., "A New Method for Analyzing the Deformation and Lateral Translation of a Moving Web", Proceedings of the Eighth International Conference on Web Handling, Stillwater, OK, WHRC, Oklahoma State University, June 2005.

47. Shimizu S., Yoshida S. and Enomoto N., “Buckling of Plates with a Hole under Tension”, Thin Walled Structures 12 (1991) 35-49.

48. Shimizu S., “Tension buckling of plate having a hole”, Thin Walled Structures 45 (2007) 827-833.

49. El-Sawy K.M and Martini M.I, “ Elastic stability of bi-axially loaded rectangular plates with a single circular hole”, Thin Walled Structures 45 (2007)122-133

APPENDIX EXCEL VBA CODE FOR MISALIGNED ROLLERS

```
Option Explicit
Dim H, L1, L2, W, W1 As Double 'dimensional parameters of domain
Dim n1, n2, k1, m, en As Integer ' mesh density parameters
Dim z1, z2, m1, m2, m3, m4, q1, q2 As Integer 'loop counters
Dim roll1P(), roll2P(), roll3P(), span1P(), span2P() As Double
Dim roll1en, roll2en, roll3en, span1en, span2en As Integer
Dim KG(), ke(), u(), FG(), ue(8) As Double
Dim te As Double 'element thickness
Dim p(8) As Integer 'local to global pointer
Dim matcons, E, v, D(3, 3) As Double
Dim x1, x2, x3, x4, y1, y2, y3, y4 As Double
Dim sf1t, sf1s, sf2t, sf2s, sf3t, sf3s, sf4t, sf4s As Double
Dim detJ, J11, J12, J21, J22 As Double
Dim BL(3, 8), t, s, gp(2) As Double
Dim STRAINP(2), STRAINXY(), STRESSXY(), PP, QQ As Double
Dim pfixing(), pbinding(), pubinding(), pload(), PC, Control As Double
Dim sdummy, dimdummy, sbw As Integer
Dim para, critical, Sigma2, Sigma1, SigmaS, sycr As Double
Dim PSTRAIN(), PQSTATE(), ELEMENTSTATE(), ESA(), STRAINPAVE() As Double
Dim iterload, iterloadT, itercon As Integer
Dim TRACTIONX, TRACTIONXY, FORCEX, FORCEXY, NFORCEX, NFORCEXY, F1, F2, F3, FS As Double
Dim D1, D2, Tangent, Angle, Downlimit, Uplimit As Double
Dim clock As Double
Dim TempAngle1, TempAngle2, TempAngle3, TempSigmaS1, TempSigmaS2, TempSigmaS3, TempFS1,
TempFS2, TempFS3 As Double
Dim Meshparameter As Integer


Sub WRINKLINGmain()
clock = Timer

Meshparameter = 1
Control = 0
SigmaS = 0
Range("message").Clear
Range("arad").Clear
Range("message2").Clear

Call WRINKLINGmesh

Call WRINKLINGmain2
Range("ARAD") = Angle
TempAngle1 = Angle

Range("MAXSY") = SigmaS
TempSigmaS1 = SigmaS

Range("TXY") = FS
TempFS1 = FS

'increasing mesh density

Meshparameter = 2
Control = 0
SigmaS = 0
```

```
Range("message").Clear
Range("arad").Clear

Call WRINKLINGmesh

Call WRINKLINGmain2
Range("ARAD") = Angle
TempAngle2 = Angle

Range("MAXSY") = SigmaS
TempSigmaS2 = SigmaS

Range("TXY") = FS
TempFS2 = FS


  If (0.95 * TempAngle1) <= TempAngle2 Or (1.05 * TempAngle1) >= TempAngle2 Then

               Range("ARAD") = TempAngle2

               Range("MAXSY") = TempSigmaS2

               Range("TXY") = TempFS1
               Range("message2") = "Result Converged"

               Else:

                     Meshparameter = 3
                     Control = 0
                     SigmaS = 0
                     Range("message").Clear
                     Range("arad").Clear

                     Call WRINKLINGmesh

                     Call WRINKLINGmain2
                     Range("ARAD") = Angle
                     TempAngle3 = Angle

                     Range("MAXSY") = SigmaS
                     TempSigmaS3 = SigmaS

                     Range("TXY") = FS
                     TempFS3 = FS

                     If (0.95 * TempAngle2) <= TempAngle3 Or (1.05 * TempAngle2) >=
TempAngle3 Then
                     Range("message2") = "Result Converged"
                     Else: Range("message2") = "Result Not Converged,Please Enter Mesh
Parameters Manually"
                     End If

  End If


Range(Cells(40, 2), Cells(40, 2)) = TempAngle1
Range(Cells(40, 3), Cells(40, 3)) = TempAngle2
Range(Cells(40, 4), Cells(40, 4)) = TempAngle3


Range("TT") = Timer - clock
Range("message") = "FEA Complete"
End Sub

Sub WRINKLINGmain2()


'Worksheets("Sheet1").Activate

'Range(Cells(17, 1), Cells(17, 1)).Clear
```

```
'traction components input

sycr = Range("SYCR")

TRACTIONX = Range("TX")
FORCEX = TRACTIONX * H * te
NFORCEX = FORCEX / m



TRACTIONXY = (TRACTIONX * H) / (6 * (W1 + L2 + L1 + L2 + W1))
F2 = TRACTIONXY
F3 = 2 * TRACTIONXY

Do Until SigmaS > sycr


    FORCEXY = F2 * H * te
    NFORCEXY = FORCEXY / m

    Call WRINKLINGsystem

    Sigma2 = critical


    F1 = F2 / 2

    FORCEXY = F1 * H * te
    NFORCEXY = FORCEXY / m

    Call WRINKLINGsystem

    Sigma1 = critical

    FS = ((sycr - Sigma2) * (F2 - F1)) / (Sigma2 - Sigma1) + F2

    If FS > F3 And Control = 0 Then
    FS = F3
    Control = Control + 1
    End If

    FORCEXY = FS * H * te
    NFORCEXY = FORCEXY / m

    Call WRINKLINGsystem

    SigmaS = critical

    F1 = F2
    F2 = FS



Loop



Downlimit = 0.95 * sycr
Uplimit = 1.05 * sycr

Do Until Downlimit <= SigmaS And SigmaS <= Uplimit

    FORCEXY = F2 * H * te
    NFORCEXY = FORCEXY / m

    Call WRINKLINGsystem

    Sigma2 = critical
    FORCEXY = F1 * H * te
    NFORCEXY = FORCEXY / m

    Call WRINKLINGsystem

    Sigma1 = critical



    FS = ((sycr - Sigma1) * (F2 - F1)) / (Sigma2 - Sigma1) + F1

    FORCEXY = FS * H * te
    NFORCEXY = FORCEXY / m

    Call WRINKLINGsystem

    SigmaS = critical

       If SigmaS > sycr Then
                    F2 = FS


        ElseIf SigmaS < sycr Then
                     F1 = FS

        End If


Loop


D1 = u(2 * ((k1 + n2 + n1) * (m + 1) + m / 2 + 1))
D2 = u(2 * ((k1 + n2) * (m + 1) + m / 2 + 1))

Tangent = (D1) / (L1 / 2)
Angle = Atn(Tangent)


'Worksheets("StressX").Activate
'Worksheets("StressX").Cells.ClearContents
'For q1 = 1 To k1 + n2 + n1
'For q2 = 1 To m
'en = (q1 - 1) * m + q2

''Range(Cells(25 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(25 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 1)
''Range(Cells(55 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(55 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 2)

'Range(Cells(25 - q2, q1 + 6), Cells(25 - q2, q1 + 6)) = ESA(en, 1)
''Range(Cells(55 - q2, q1 + 6), Cells(55 - q2, q1 + 6)) = ESA(en, 2)
''Range(Cells(85 - q2, q1 + 6), Cells(85 - q2, q1 + 6)) = ESA(en, 3)
'Next
'Next


'For q1 = 1 To k1
'Range(Cells(24 - m, q1 + 6), Cells(24 - m, q1 + 6)) = q1
''Range(Cells(54 - m, q1 + 6), Cells(54 - m, q1 + 6)) = q1
'Next

'For q1 = 1 To n2
'Range(Cells(24 - m, q1 + k1 + 6), Cells(24 - m, q1 + k1 + 6)) = q1
''Range(Cells(54 - m, q1 + k1 + 6), Cells(54 - m, q1 + k1 + 6)) = q1
'Next

'For q1 = 1 To n1
'Range(Cells(24 - m, q1 + k1 + n2 + 6), Cells(24 - m, q1 + k1 + n2 + 6)) = q1
''Range(Cells(54 - m, q1 + k1 + n2 + 6), Cells(54 - m, q1 + k1 + n2 + 6)) = q1
'Next
```

```
'For q1 = 1 To m
'Range(Cells(25 - q1, 6), Cells(25 - q1, 6)) = q1
''Range(Cells(55 - q1, 6), Cells(55 - q1, 6)) = q1
'Next

'Worksheets("StressY").Activate
'Worksheets("StressY").Cells.ClearContents
'For q1 = 1 To k1 + n2 + n1
'For q2 = 1 To m
'en = (q1 - 1) * m + q2

''Range(Cells(25 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(25 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 1)
''Range(Cells(55 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(55 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 2)

'Range(Cells(25 - q2, q1 + 6), Cells(25 - q2, q1 + 6)) = ESA(en, 2)
''Range(Cells(55 - q2, q1 + 6), Cells(55 - q2, q1 + 6)) = ESA(en, 2)
''Range(Cells(85 - q2, q1 + 6), Cells(85 - q2, q1 + 6)) = ESA(en, 3)
'Next
'Next


'For q1 = 1 To k1
'Range(Cells(24 - m, q1 + 6), Cells(24 - m, q1 + 6)) = q1
''Range(Cells(54 - m, q1 + 6), Cells(54 - m, q1 + 6)) = q1
'Next

'For q1 = 1 To n2
'Range(Cells(24 - m, q1 + k1 + 6), Cells(24 - m, q1 + k1 + 6)) = q1
''Range(Cells(54 - m, q1 + k1 + 6), Cells(54 - m, q1 + k1 + 6)) = q1
'Next

'For q1 = 1 To n1
'Range(Cells(24 - m, q1 + k1 + n2 + 6), Cells(24 - m, q1 + k1 + n2 + 6)) = q1
''Range(Cells(54 - m, q1 + k1 + n2 + 6), Cells(54 - m, q1 + k1 + n2 + 6)) = q1
'Next

'For q1 = 1 To m
'Range(Cells(25 - q1, 6), Cells(25 - q1, 6)) = q1
''Range(Cells(55 - q1, 6), Cells(55 - q1, 6)) = q1
'Next

'Worksheets("StressXY").Activate
'Worksheets("StressXY").Cells.ClearContents
'For q1 = 1 To k1 + n2 + n1
'For q2 = 1 To m
'en = (q1 - 1) * m + q2

''Range(Cells(25 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(25 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 1)
''Range(Cells(55 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(55 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 2)

'Range(Cells(25 - q2, q1 + 6), Cells(25 - q2, q1 + 6)) = ESA(en, 3)
''Range(Cells(55 - q2, q1 + 6), Cells(55 - q2, q1 + 6)) = ESA(en, 2)
''Range(Cells(85 - q2, q1 + 6), Cells(85 - q2, q1 + 6)) = ESA(en, 3)
'Next
'Next


'For q1 = 1 To k1
'Range(Cells(24 - m, q1 + 6), Cells(24 - m, q1 + 6)) = q1
''Range(Cells(54 - m, q1 + 6), Cells(54 - m, q1 + 6)) = q1
'Next

'For q1 = 1 To n2
'Range(Cells(24 - m, q1 + k1 + 6), Cells(24 - m, q1 + k1 + 6)) = q1
''Range(Cells(54 - m, q1 + k1 + 6), Cells(54 - m, q1 + k1 + 6)) = q1
'Next
```

```
'For q1 = 1 To n1
'Range(Cells(24 - m, q1 + k1 + n2 + 6), Cells(24 - m, q1 + k1 + n2 + 6)) = q1
''Range(Cells(54 - m, q1 + k1 + n2 + 6), Cells(54 - m, q1 + k1 + n2 + 6)) = q1
'Next

'For q1 = 1 To m
'Range(Cells(25 - q1, 6), Cells(25 - q1, 6)) = q1
''Range(Cells(55 - q1, 6), Cells(55 - q1, 6)) = q1
'Next

'Worksheets("Roller Misalignment").Activate
End Sub
Sub WRINKLINGsystem()



For iterload = 1 To 4

'For itercon = 1 To 1


Range(Cells(24, 2), Cells(24, 2)) = iterload
'Range(Cells(25, 2), Cells(25, 2)) = itercon





'ReDim KG(2 * (k1 + n2 + n1 + n2 + k1 + 1) * (m + 1), 2 * ((k1) * (m + 1) + 2)), FG(2 * (k1 +
n2 + n1 + n2 + k1 + 1) * (m + 1))
ReDim KG(2 * (k1 + n2 + n1 + n2 + k1 + 1) * (m + 1), 2 * (m + 3)), FG(2 * (k1 + n2 + n1 + n2
+ k1 + 1) * (m + 1))

'form roll1 region's elements

Call ROLLmaterial

For q1 = 1 To k1
For q2 = 1 To m

en = (q1 - 1) * m + q2

x1 = roll1P(en, 1)
x2 = roll1P(en, 2)
x3 = roll1P(en, 3)
x4 = roll1P(en, 4)
y1 = roll1P(en, 5)
y2 = roll1P(en, 6)
y3 = roll1P(en, 7)
y4 = roll1P(en, 8)


Call WRINKLINGelement


'pointers

p(1) = 2 * ((q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((q1) * (m + 1) + q2) - 1
p(6) = 2 * ((q1) * (m + 1) + q2)

p(7) = 2 * ((q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((q1) * (m + 1) + q2 + 1)


For m1 = 1 To 8
For m2 = m1 To 8
KG(p(m1), p(m2) - p(m1) + 1) = KG(p(m1), p(m2) - p(m1) + 1) + ke(m1, m2)
Next
Next


Next
Next

'form span1 region's elements
For q1 = 1 To n2
For q2 = 1 To m

en = (q1 - 1) * m + q2



x1 = span1P(en, 1)
x2 = span1P(en, 2)
x3 = span1P(en, 3)
x4 = span1P(en, 4)
y1 = span1P(en, 5)
y2 = span1P(en, 6)
y3 = span1P(en, 7)
y4 = span1P(en, 8)

'pointers

p(1) = 2 * ((k1 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((k1 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((k1 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((k1 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((k1 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((k1 + q1) * (m + 1) + q2)

p(7) = 2 * ((k1 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((k1 + q1) * (m + 1) + q2 + 1)

For m1 = 1 To 8
ue(m1) = u(p(m1))
Next


Call WRINKLINGstrain  'calculate ex,ey,exy,e1,e2


Call WRINKLINGcriteria 'check for the state Taut?Slack?Wrinkled?


Call WRINKLINGelement 'form element stiffness matrix


For m1 = 1 To 8
For m2 = m1 To 8
KG(p(m1), p(m2) - p(m1) + 1) = KG(p(m1), p(m2) - p(m1) + 1) + ke(m1, m2)
Next
Next


Next
Next

'form roll2 region's elements
Call ROLLmaterial

For q1 = 1 To n1
For q2 = 1 To m

en = (q1 - 1) * m + q2
```

```
x1 = roll2P(en, 1)
x2 = roll2P(en, 2)
x3 = roll2P(en, 3)
x4 = roll2P(en, 4)
y1 = roll2P(en, 5)
y2 = roll2P(en, 6)
y3 = roll2P(en, 7)
y4 = roll2P(en, 8)

Call WRINKLINGelement

'pointers

p(1) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((k1 + n2 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((k1 + n2 + q1) * (m + 1) + q2)

p(7) = 2 * ((k1 + n2 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((k1 + n2 + q1) * (m + 1) + q2 + 1)


For m1 = 1 To 8
For m2 = m1 To 8
KG(p(m1), p(m2) - p(m1) + 1) = KG(p(m1), p(m2) - p(m1) + 1) + ke(m1, m2)
Next
Next


Next
Next

'form span2 region's elements
For q1 = 1 To n2
For q2 = 1 To m

en = (q1 - 1) * m + q2

x1 = span2P(en, 1)
x2 = span2P(en, 2)
x3 = span2P(en, 3)
x4 = span2P(en, 4)
y1 = span2P(en, 5)
y2 = span2P(en, 6)
y3 = span2P(en, 7)
y4 = span2P(en, 8)


'pointers

p(1) = 2 * ((k1 + n2 + n1 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((k1 + n2 + n1 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((k1 + n2 + n1 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((k1 + n2 + n1 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((k1 + n2 + n1 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((k1 + n2 + n1 + q1) * (m + 1) + q2)

p(7) = 2 * ((k1 + n2 + n1 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((k1 + n2 + n1 + q1) * (m + 1) + q2 + 1)

For m1 = 1 To 8
ue(m1) = u(p(m1))
Next
```

```
'Call WRINKLINGstrain  'calculate ex,ey,exy,e1,e2

'Call WRINKLINGcriteria 'check for the state Taut?Slack?Wrinkled?

Call ROLLmaterial

Call WRINKLINGelement 'form element stiffness matrix



For m1 = 1 To 8
For m2 = m1 To 8
KG(p(m1), p(m2) - p(m1) + 1) = KG(p(m1), p(m2) - p(m1) + 1) + ke(m1, m2)
Next
Next


Next
Next

'form roll3 region's elements

Call ROLLmaterial

For q1 = 1 To k1
For q2 = 1 To m

en = (q1 - 1) * m + q2

x1 = roll3P(en, 1)
x2 = roll3P(en, 2)
x3 = roll3P(en, 3)
x4 = roll3P(en, 4)
y1 = roll3P(en, 5)
y2 = roll3P(en, 6)
y3 = roll3P(en, 7)
y4 = roll3P(en, 8)

Call WRINKLINGelement

'pointers

p(1) = 2 * ((k1 + n2 + n1 + n2 + q1 - 1) * (m + 1) + q2) - 1 'k1 + n2 + n1+n2
p(2) = 2 * ((k1 + n2 + n1 + n2 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((k1 + n2 + n1 + n2 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((k1 + n2 + n1 + n2 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((k1 + n2 + n1 + n2 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((k1 + n2 + n1 + n2 + q1) * (m + 1) + q2)

p(7) = 2 * ((k1 + n2 + n1 + n2 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((k1 + n2 + n1 + n2 + q1) * (m + 1) + q2 + 1)


For m1 = 1 To 8
For m2 = m1 To 8
KG(p(m1), p(m2) - p(m1) + 1) = KG(p(m1), p(m2) - p(m1) + 1) + ke(m1, m2)
Next
Next


Next
Next

'determine penalty number
PC = 0
For m1 = 1 To 2 * (m + 1) * (k1 + n2 + n1 + n2 + k1 + 1)

If KG(m1, 1) > PC Then PC = KG(m1, 1)
```

```
Next

PC = PC * 10000

'apply constraints

Call WRINKLINGconstraints

'apply force

Call WRINKLINGforceshear

Call WRINKLINGforcenormal


'solve system

Call WRINKLINGsolver

'check state of the elements

'Call WRINKLINGcheck

'print state of the elements

'For q1 = 1 To n2
'For q2 = 1 To m

'en = (q1 - 1) * m + q2

'Range(Cells(5 + en, 4), Cells(5 + en, 4)) = en
'Range(Cells(5 + en, 4 + itercon), Cells(5 + en, 4 + itercon)) = PQSTATE(en, 1)
'Range(Cells(5 + en, 16 + itercon), Cells(5 + en, 16 + itercon)) = PQSTATE(en, 2)

'Next
'Next

'For q1 = 1 To n2
'For q2 = 1 To m

'en = (q1 - 1) * m + q2

'Range(Cells(5 + en, 9 + iter), Cells(5 + en, 9 + iter)) = PQSTATE(en + n2 * m, 1)

'Next
'Next

'Next 'itercon
Next 'iterload



Call WRINKLINGstresscalc

'For q1 = 1 To k1 + n2 + n1
'For q2 = 1 To m
'en = (q1 - 1) * m + q2

'Range(Cells(25 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(25 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 1)
'Range(Cells(55 - q2, q1 + 10 + (k1 + n2 + n1 + 2) * (iterload - 1)), Cells(55 - q2, q1 + 10
+ (k1 + n2 + n1 + 2) * (iterload - 1))) = ESA(en, 2)

'Range(Cells(25 - q2, q1 + 6), Cells(25 - q2, q1 + 6)) = ESA(en, 1)
'Range(Cells(55 - q2, q1 + 6), Cells(55 - q2, q1 + 6)) = ESA(en, 2)
'Range(Cells(85 - q2, q1 + 6), Cells(85 - q2, q1 + 6)) = ESA(en, 3)
'Next
'Next

critical = 0
```

```
    For q1 = 1 To m
    en = (k1 + n2) * m + q1
    If critical < Abs(ESA(en, 2)) Then critical = Abs(ESA(en, 2))

    Next

'For q1 = 1 To k1
'Range(Cells(24 - m, q1 + 6), Cells(24 - m, q1 + 6)) = q1
'Range(Cells(54 - m, q1 + 6), Cells(54 - m, q1 + 6)) = q1
'Next

'For q1 = 1 To n2
'Range(Cells(24 - m, q1 + k1 + 6), Cells(24 - m, q1 + k1 + 6)) = q1
'Range(Cells(54 - m, q1 + k1 + 6), Cells(54 - m, q1 + k1 + 6)) = q1
'Next

'For q1 = 1 To n1
'Range(Cells(24 - m, q1 + k1 + n2 + 6), Cells(24 - m, q1 + k1 + n2 + 6)) = q1
'Range(Cells(54 - m, q1 + k1 + n2 + 6), Cells(54 - m, q1 + k1 + n2 + 6)) = q1
'Next

'For q1 = 1 To m
'Range(Cells(25 - q1, 6), Cells(25 - q1, 6)) = q1
'Range(Cells(55 - q1, 6), Cells(55 - q1, 6)) = q1
'Next

'Call WRINKLINGstresscalc


'print u
'For m1 = 1 To (m + 1)
'For m2 = 1 To n1 + 1
'Range(Cells(5 + m1, 4 + m2), Cells(5 + m1, 4 + m2)) = u(2 * ((m2 - 1) * (m + 1) + m1))
'Next
'Next

'For m1 = 1 To (m + 1)

'Range(Cells(6 + m + m1, 5), Cells(6 + m + m1, 5)) = u(2 * ((3 * n1 + 2 * n2) * (m + 1) +
m1))

'Next

End Sub
Sub WRINKLINGmesh()


'input variables

H = Range("WW")


m = Range("mw")

If m > 30 Then m = 30
If m < 6 Then m = 6

If Meshparameter = 1 Then
   m = m - (m / 2 - 1)

   ElseIf Meshparameter = 2 Then
   m = m

   ElseIf Meshparameter = 3 Then
   m = m + (m / 2 - 1)
End If

If (m Mod 2) <> 0 Then m = m + 1
```

```
L1 = Range("rs")


L2 = Range("fs")

n1 = Range("rsn")
 If n1 > 30 Then n1 = 30
 If n1 < 6 Then n1 = 6


n2 = Range("fsn")
 If n2 > 50 Then n2 = 50
 If n2 < 6 Then n2 = 6

 If Meshparameter = 1 Then
   n2 = n2 - (n2 / 2 - 1)
 ElseIf Meshparameter = 2 Then
   n2 = n2
 ElseIf Meshparameter = 3 Then
   n2 = n2 + (n2 / 2 - 1)
 End If

If (n2 Mod 2) <> 0 Then n2 = n2 + 1


te = Range("TE")
E = Range("Ex")
v = Range("Prxy")

'k1 = 10
'W = H / m
'W1 = k1 * W
k1 = n1
W1 = L1


'gauss points
gp(1) = 0.57735
gp(2) = -0.57735

'penalty constraint

PC = 10 ^ 12

'ReDim KG(2 * (k1 + n2 + n1 + n2 + k1) * (m + 1), 2 * ((k1) * (m + 1) + 1)), u(2 * (k1 + n2 +
n1 + n2 + k1 + 1) * (m + 1))
ReDim KG(2 * (k1 + n2 + n1 + n2 + k1) * (m + 1), 2 * (m + 3)), u(2 * (k1 + n2 + n1 + n2 + k1
+ 1) * (m + 1))

ReDim FG(2 * (k1 + n2 + n1 + n2 + k1 + 1) * (m + 1))

ReDim pbinding(k1 + 1)
ReDim pload(m + 1)
ReDim pfixing(2 * (m + 1))

ReDim PSTRAIN(2 * m * n2, 2), PQSTATE(2 * m * n2, 2), ELEMENTSTATE(2 * m * n2)
ReDim ESA((k1 + n1 + n2) * m, 3)

'form up roll1 region position array

roll1en = k1 * m

ReDim roll1P(roll1en, 8)

For q1 = 1 To k1
For q2 = 1 To m
en = (q1 - 1) * m + q2

roll1P(en, 1) = (q1 - 1) * W1 / k1
roll1P(en, 2) = roll1P(en, 1)
roll1P(en, 3) = (q1) * W1 / k1
```

```
roll1P(en, 4) = roll1P(en, 3)

roll1P(en, 5) = (q2 - 1) * H / m
roll1P(en, 6) = (q2) * H / m
roll1P(en, 7) = roll1P(en, 5)
roll1P(en, 8) = roll1P(en, 6)


Next
Next


'form up span1 region position array

span1en = n2 * m

ReDim span1P(span1en, 8)

For q1 = 1 To n2
For q2 = 1 To m
en = (q1 - 1) * m + q2



span1P(en, 1) = W1 + (q1 - 1) * L2 / n2
span1P(en, 2) = span1P(en, 1)
span1P(en, 3) = W1 + (q1) * L2 / n2
span1P(en, 4) = span1P(en, 3)

span1P(en, 5) = (q2 - 1) * H / m
span1P(en, 6) = (q2) * H / m
span1P(en, 7) = span1P(en, 5)
span1P(en, 8) = span1P(en, 6)


Next
Next


'form up roll2 region position array
roll2en = n1 * m

ReDim roll2P(roll2en, 8)

For q1 = 1 To n1
For q2 = 1 To m
en = (q1 - 1) * m + q2

roll2P(en, 1) = W1 + L2 + (q1 - 1) * L1 / n1
roll2P(en, 2) = roll2P(en, 1)
roll2P(en, 3) = W1 + L2 + (q1) * L1 / n1
roll2P(en, 4) = roll2P(en, 3)

roll2P(en, 5) = (q2 - 1) * H / m
roll2P(en, 6) = (q2) * H / m
roll2P(en, 7) = roll2P(en, 5)
roll2P(en, 8) = roll2P(en, 6)


Next
Next

'form up span2 region position array

span2en = n2 * m

ReDim span2P(span1en, 8)

For q1 = 1 To n2
For q2 = 1 To m
en = (q1 - 1) * m + q2
```

```
span2P(en, 1) = W1 + L1 + L2 + (q1 - 1) * L2 / n2
span2P(en, 2) = span2P(en, 1)
span2P(en, 3) = W1 + L1 + L2 + (q1) * L2 / n2
span2P(en, 4) = span2P(en, 3)

span2P(en, 5) = (q2 - 1) * H / m
span2P(en, 6) = (q2) * H / m
span2P(en, 7) = span2P(en, 5)
span2P(en, 8) = span2P(en, 6)


Next
Next

'form up roll3 region position array
roll3en = n1 * m

ReDim roll3P(roll3en, 8)

For q1 = 1 To k1
For q2 = 1 To m
en = (q1 - 1) * m + q2

roll3P(en, 1) = W1 + L2 + L1 + L2 + (q1 - 1) * W1 / k1
roll3P(en, 2) = roll3P(en, 1)
roll3P(en, 3) = W1 + L2 + L1 + L2 + (q1) * W1 / k1
roll3P(en, 4) = roll3P(en, 3)

roll3P(en, 5) = (q2 - 1) * H / m
roll3P(en, 6) = (q2) * H / m
roll3P(en, 7) = roll3P(en, 5)
roll3P(en, 8) = roll3P(en, 6)


Next
Next


End Sub


Sub WRINKLINGelement()

ReDim ke(8, 8)

For z1 = 1 To 2
For z2 = 1 To 2

t = gp(z1)
s = gp(z2)

'derivatives of shape functions
sf1t = (-1 + s) / 4
sf1s = (-1 + t) / 4

sf2t = (-1 - s) / 4
sf2s = (1 - t) / 4

sf3t = (1 - s) / 4
sf3s = (-1 - t) / 4

sf4t = (1 + s) / 4
sf4s = (1 + t) / 4

'calculate jakobien

J11 = x1 * sf1t + x2 * sf2t + x3 * sf3t + x4 * sf4t
J12 = y1 * sf1t + y2 * sf2t + y3 * sf3t + y4 * sf4t
J21 = x1 * sf1s + x2 * sf2s + x3 * sf3s + x4 * sf4s
J22 = y1 * sf1s + y2 * sf2s + y3 * sf3s + y4 * sf4s
```

```
detJ = J11 * J22 - J12 * J21

'linear calculate strain gradient matrix

BL(1, 1) = (-J12 * sf1s + J22 * sf1t) / detJ
BL(3, 1) = (J11 * sf1s - J21 * sf1t) / detJ
BL(2, 2) = BL(3, 1)
BL(3, 2) = BL(1, 1)

BL(1, 3) = (-J12 * sf2s + J22 * sf2t) / detJ
BL(3, 3) = (J11 * sf2s - J21 * sf2t) / detJ
BL(2, 4) = BL(3, 3)
BL(3, 4) = BL(1, 3)

BL(1, 5) = (-J12 * sf3s + J22 * sf3t) / detJ
BL(3, 5) = (J11 * sf3s - J21 * sf3t) / detJ
BL(2, 6) = BL(3, 5)
BL(3, 6) = BL(1, 5)

BL(1, 7) = (-J12 * sf4s + J22 * sf4t) / detJ
BL(3, 7) = (J11 * sf4s - J21 * sf4t) / detJ
BL(2, 8) = BL(3, 7)
BL(3, 8) = BL(1, 7)


For m1 = 1 To 8
For m2 = m1 To 8
For m3 = 1 To 3
For m4 = 1 To 3

ke(m1, m2) = ke(m1, m2) + BL(m3, m1) * D(m3, m4) * BL(m4, m2) * te * detJ
Next
Next
Next
Next



Next
Next



End Sub



Sub ROLLmaterial()
Erase D
matcons = E / (1 - v ^ 2)
D(1, 1) = matcons
D(1, 2) = v * matcons
D(2, 1) = D(1, 2)
D(2, 2) = D(1, 1)
D(3, 3) = matcons * (1 - v) / 2

End Sub


Sub WRINKLINGstrain()

ReDim STRAINXY(3)
For z1 = 1 To 2
For z2 = 1 To 2


t = gp(z1)
s = gp(z2)

'derivatives of shape functions
sf1t = (-1 + s) / 4
```

```
sf1s = (-1 + t) / 4

sf2t = (-1 - s) / 4
sf2s = (1 - t) / 4

sf3t = (1 - s) / 4
sf3s = (-1 - t) / 4

sf4t = (1 + s) / 4
sf4s = (1 + t) / 4

'calculate jakobien

J11 = x1 * sf1t + x2 * sf2t + x3 * sf3t + x4 * sf4t
J12 = y1 * sf1t + y2 * sf2t + y3 * sf3t + y4 * sf4t
J21 = x1 * sf1s + x2 * sf2s + x3 * sf3s + x4 * sf4s
J22 = y1 * sf1s + y2 * sf2s + y3 * sf3s + y4 * sf4s
detJ = J11 * J22 - J12 * J21

'linear calculate strain gradient matrix

BL(1, 1) = (-J12 * sf1s + J22 * sf1t) / detJ
BL(3, 1) = (J11 * sf1s - J21 * sf1t) / detJ
BL(2, 2) = BL(3, 1)
BL(3, 2) = BL(1, 1)

BL(1, 3) = (-J12 * sf2s + J22 * sf2t) / detJ
BL(3, 3) = (J11 * sf2s - J21 * sf2t) / detJ
BL(2, 4) = BL(3, 3)
BL(3, 4) = BL(1, 3)

BL(1, 5) = (-J12 * sf3s + J22 * sf3t) / detJ
BL(3, 5) = (J11 * sf3s - J21 * sf3t) / detJ
BL(2, 6) = BL(3, 5)
BL(3, 6) = BL(1, 5)

BL(1, 7) = (-J12 * sf4s + J22 * sf4t) / detJ
BL(3, 7) = (J11 * sf4s - J21 * sf4t) / detJ
BL(2, 8) = BL(3, 7)
BL(3, 8) = BL(1, 7)

For m1 = 1 To 3
For m2 = 1 To 8
STRAINXY(m1) = STRAINXY(m1) + BL(m1, m2) * ue(m2)
Next
Next

Next
Next

STRAINP(1) = (STRAINXY(1) / 4 + STRAINXY(2) / 4) / 2 + ((STRAINXY(1) / 4 - STRAINXY(2) / 4) ^
2 + (STRAINXY(3) / 4) ^ 2) ^ 0.5 / 2
STRAINP(2) = (STRAINXY(1) / 4 + STRAINXY(2) / 4) / 2 - ((STRAINXY(1) / 4 - STRAINXY(2) / 4) ^
2 + (STRAINXY(3) / 4) ^ 2) ^ 0.5 / 2



End Sub

Sub WRINKLINGcriteria()



If STRAINP(1) < 0 Then
Erase D
Range("message") = "MD Stress input is too small, a slack edge occurs prior to wrinkling"
Worksheets("Roller Misalignment").Activate
End


ElseIf STRAINP(1) > 0 And v * STRAINP(1) < -STRAINP(2) Then
```

```
Erase D
PP = (STRAINXY(1) / 4 - STRAINXY(2) / 4) / (STRAINP(1) - STRAINP(2))
QQ = (STRAINXY(3) / 4) / ((STRAINP(1) - STRAINP(2)))
D(1, 1) = (1 + PP) * E / 2
D(2, 2) = (1 - PP) * E / 2
D(3, 3) = E / 4
D(1, 3) = QQ * E / 4
D(2, 3) = D(1, 3)
D(3, 1) = D(1, 3)
D(3, 2) = D(1, 3)
D(1, 2) = 0
D(2, 1) = 0
Else
Erase D
matcons = E / (1 - v ^ 2)
D(1, 1) = matcons
D(1, 2) = v * matcons
D(2, 1) = D(1, 2)
D(2, 2) = D(1, 1)
D(3, 3) = matcons * (1 - v) / 2


End If

End Sub


Sub WRINKLINGconstraints()

'fixing of middle line

For m1 = 1 To m + 1
pfixing(m1) = (2 * k1 + n1 + 2 * n2) * (m + 1) / 2 + m1
Next

For m1 = m / 2 + 1 To m / 2 + 1
'For m1 = 1 To m + 1

KG(2 * pfixing(m1) - 1, 1) = KG(2 * pfixing(m1) - 1, 1) + PC
Next


For m1 = m / 2 + 1 To m / 2 + 1
'For m1 = 1 To m + 1
KG(2 * pfixing(m1), 1) = KG(2 * pfixing(m1), 1) + PC
Next


'binding roll1 region's points in v direction

For m1 = 1 To m + 1

For m2 = 1 To k1 + 1
pbinding(m2) = 2 * (m1 + (m + 1) * (m2 - 1))
Next

For m2 = 1 To k1
KG(pbinding(m2), 1) = KG(pbinding(m2), 1) + PC
KG(pbinding(m2 + 1), 1) = KG(pbinding(m2 + 1), 1) + PC
KG(pbinding(m2), pbinding(m2 + 1) - pbinding(m2) + 1) = KG(pbinding(m2), pbinding(m2 + 1) -
pbinding(m2) + 1) - PC
Next

'****
'KG(pbinding(1), 1) = KG(pbinding(1), 1) + PC
'KG(pbinding(k1 + 1), 1) = KG(pbinding(k1 + 1), 1) + PC
'****

'For q1 = 1 To n1
'For q2 = q1 + 1 To n1 + 1
```

```
'KG(pbinding(q1), pbinding(q2) - pbinding(q1) + 1) = KG(pbinding(q1), pbinding(q2) -
pbinding(q1) + 1) - PC

'Next
'Next

'For q1 = 1 To n1

'KG(pbinding(q1), pbinding(q1 + 1) - pbinding(q1) + 1) = KG(pbinding(q1), pbinding(q1 + 1) -
pbinding(q1) + 1) - PC

'Next

'*****
'KG(pbinding(1), pbinding(k1 + 1) - pbinding(1) + 1) = KG(pbinding(1), pbinding(k1 + 1) -
pbinding(1) + 1) - PC
'*****

Next

'binding roll1 region's first line points in u direction

ReDim pubinding(m + 1)

For m1 = 1 To m + 1
pubinding(m1) = 2 * m1 - 1
Next

For m1 = 1 To m
KG(pubinding(m1), 1) = KG(pubinding(m1), 1) + PC
KG(pubinding(m1 + 1), 1) = KG(pubinding(m1 + 1), 1) + PC
KG(pubinding(m1), pubinding(m1 + 1) - pubinding(m1) + 1) = KG(pubinding(m1), pubinding(m1 +
1) - pubinding(m1) + 1) - PC
Next



'binding roll3 region's points in v direction

For m1 = 1 To m + 1

For m2 = 1 To k1 + 1
pbinding(m2) = 2 * (m1 + (m + 1) * (m2 - 1) + (k1 + n2 + n1 + n2) * (m + 1))
Next


For m2 = 1 To k1
KG(pbinding(m2), 1) = KG(pbinding(m2), 1) + PC
KG(pbinding(m2 + 1), 1) = KG(pbinding(m2 + 1), 1) + PC
KG(pbinding(m2), pbinding(m2 + 1) - pbinding(m2) + 1) = KG(pbinding(m2), pbinding(m2 + 1) -
pbinding(m2) + 1) - PC
Next


'For q1 = 1 To n1 + 1
'KG(pbinding(q1), 1) = KG(pbinding(q1), 1) + PC
'Next

'*******
'KG(pbinding(1), 1) = KG(pbinding(1), 1) + PC
'KG(pbinding(k1 + 1), 1) = KG(pbinding(k1 + 1), 1) + PC
'*******

'For q1 = 1 To n1
'For q2 = q1 + 1 To n1 + 1

'KG(pbinding(q1), pbinding(q2) - pbinding(q1) + 1) = KG(pbinding(q1), pbinding(q2) -
pbinding(q1) + 1) - PC

'Next
```

```
'Next

'For q1 = 1 To n1

'KG(pbinding(q1), pbinding(q1 + 1) - pbinding(q1) + 1) = KG(pbinding(q1), pbinding(q1 + 1) -
pbinding(q1) + 1) - PC

'Next

'******
'KG(pbinding(1), pbinding(k1 + 1) - pbinding(1) + 1) = KG(pbinding(1), pbinding(k1 + 1) -
pbinding(1) + 1) - PC
'******

Next

'binding roll3 region's first line points in u direction

ReDim pubinding(m + 1)

For m1 = 1 To m + 1
pubinding(m1) = 2 * ((2 * k1 + 2 * n2 + n1) * (m + 1) + m1) - 1
Next

For m1 = 1 To m
KG(pubinding(m1), 1) = KG(pubinding(m1), 1) + PC
KG(pubinding(m1 + 1), 1) = KG(pubinding(m1 + 1), 1) + PC
KG(pubinding(m1), pubinding(m1 + 1) - pubinding(m1) + 1) = KG(pubinding(m1), pubinding(m1 +
1) - pubinding(m1) + 1) - PC
Next




End Sub
Sub WRINKLINGforcenormal()

'ReDim FG(2 * (3 * n1 + 2 * n2 + 1) * (m + 1))

'iterloadT = iterload

'If iterload > 10 Then
'iterloadT = 10
'End If

For m1 = 2 To m
pload(m1) = 2 * m1 - 1
FG(pload(m1)) = -NFORCEX * iterload / 4
Next

For m1 = 2 To m
pload(m1) = 2 * (m1 + (k1 + n2 + n1 + n2 + k1) * (m + 1)) - 1
FG(pload(m1)) = NFORCEX * iterload / 4
Next

'edges
FG(1) = -(NFORCEX / 2) * iterload / 4
FG(2 * (m + 1) - 1) = -(NFORCEX / 2) * iterload / 4

FG(2 * (m + 1) * (k1 + n2 + n1 + n2 + k1 + 1) - 1) = (NFORCEX / 2) * iterload / 4
FG(2 * ((m + 1) * (k1 + n2 + n1 + n2 + k1) + 1) - 1) = (NFORCEX / 2) * iterload / 4


End Sub

Sub WRINKLINGforceshear()
'ReDim FG(2 * (3 * n1 + 2 * n2 + 1) * (m + 1))

For m1 = 2 To m
pload(m1) = 2 * m1
FG(pload(m1)) = -NFORCEXY * iterload / 4
```

```
Next

For m1 = 2 To m
pload(m1) = 2 * (m1 + (k1 + n2 + n1 + n2 + k1) * (m + 1))
FG(pload(m1)) = NFORCEXY * iterload / 4
Next

'edges
FG(2) = -(NFORCEXY / 2) * iterload / 4
FG(2 * (m + 1)) = -(NFORCEXY / 2) * iterload / 4

FG(2 * (m + 1) * (k1 + n2 + n1 + n2 + k1 + 1)) = (NFORCEXY / 2) * iterload / 4
FG(2 * ((m + 1) * (k1 + n2 + n1 + n2 + k1) + 1)) = (NFORCEXY / 2) * iterload / 4
End Sub


Sub WRINKLINGsolver()

'required dummies
'dimdummy

dimdummy = 2 * (k1 + n2 + n1 + n2 + k1 + 1) * (m + 1)
ReDim u(dimdummy)

'band reduction
'sbw = 2 * ((k1) * (m + 1) + 1)  'semi band width

sbw = 2 * (m + 3)

For m1 = 1 To dimdummy - 1
 If m1 <= dimdummy - sbw + 1 Then
 sdummy = m1 + sbw - 1
 Else
 sdummy = dimdummy
 End If
 For m2 = m1 + 1 To sdummy
 para = KG(m1, m2 - m1 + 1) / KG(m1, 1)
  For m3 = m2 To sdummy
   KG(m2, m3 - m2 + 1) = KG(m2, m3 - m2 + 1) - para * KG(m1, m3 - m1 + 1)

    Next
  FG(m2) = FG(m2) - FG(m1) * para

 Next

Next

'back substituion

For m1 = dimdummy To 1 Step -1
 If m1 > dimdummy - sbw + 1 Then
 sdummy = dimdummy
 Else
 sdummy = m1 + sbw - 1
 End If
 For m2 = m1 + 1 To sdummy
  FG(m1) = FG(m1) - u(m2) * KG(m1, m2 - m1 + 1)
 Next

 u(m1) = FG(m1) / KG(m1, 1)

Next

End Sub
Sub WRINKLINGcheck()

'span1

For q1 = 1 To n2
For q2 = 1 To m
```

```
en = (q1 - 1) * m + q2



x1 = span1P(en, 1)
x2 = span1P(en, 2)
x3 = span1P(en, 3)
x4 = span1P(en, 4)
y1 = span1P(en, 5)
y2 = span1P(en, 6)
y3 = span1P(en, 7)
y4 = span1P(en, 8)

'pointers

p(1) = 2 * ((n1 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((n1 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((n1 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((n1 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((n1 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((n1 + q1) * (m + 1) + q2)

p(7) = 2 * ((n1 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((n1 + q1) * (m + 1) + q2 + 1)

For m1 = 1 To 8
ue(m1) = u(p(m1))
Next

ReDim STRAINPAVE(2)

Call WRINKLINGstrain

STRAINPAVE(1) = STRAINP(1)
STRAINPAVE(2) = STRAINP(2)

For m1 = 1 To 2
PSTRAIN(en, m1) = STRAINPAVE(m1)
Next

PP = (STRAINXY(1) / 4 - STRAINXY(2) / 4) / (STRAINPAVE(1) - STRAINPAVE(2))
QQ = (STRAINXY(3) / 4) / ((STRAINPAVE(1) - STRAINPAVE(2)))

PQSTATE(en, 1) = PP
PQSTATE(en, 2) = QQ

If STRAINPAVE(1) < 0 Then
ELEMENTSTATE(en) = 1
ElseIf STRAINPAVE(1) > 0 And v * STRAINPAVE(1) < -STRAINPAVE(2) Then
ELEMENTSTATE(en) = 2
Else
ELEMENTSTATE(en) = 3
End If

Next
Next

'span2

For q1 = 1 To n2
For q2 = 1 To m

en = (q1 - 1) * m + q2

x1 = span2P(en, 1)
x2 = span2P(en, 2)
x3 = span2P(en, 3)
x4 = span2P(en, 4)
y1 = span2P(en, 5)
```

```
y2 = span2P(en, 6)
y3 = span2P(en, 7)
y4 = span2P(en, 8)


'pointers

p(1) = 2 * ((2 * n1 + n2 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((2 * n1 + n2 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((2 * n1 + n2 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((2 * n1 + n2 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((2 * n1 + n2 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((2 * n1 + n2 + q1) * (m + 1) + q2)

p(7) = 2 * ((2 * n1 + n2 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((2 * n1 + n2 + q1) * (m + 1) + q2 + 1)


For m1 = 1 To 8
ue(m1) = u(p(m1))
Next

ReDim STRAINPAVE(2)

Call WRINKLINGstrain
STRAINPAVE(1) = STRAINP(1)
STRAINPAVE(2) = STRAINP(2)

For m1 = 1 To 2
PSTRAIN(en + n2 * m, m1) = STRAINPAVE(m1)
Next

PP = (STRAINXY(1) / 4 - STRAINXY(2) / 4) / (STRAINPAVE(1) - STRAINPAVE(2))
QQ = (STRAINXY(3) / 4) / ((STRAINPAVE(1) - STRAINPAVE(2)))

PQSTATE(en + n2 * m, 1) = PP
PQSTATE(en + n2 * m, 2) = QQ

If STRAINPAVE(1) < 0 Then
ELEMENTSTATE(en + n2 * m) = 1
ElseIf STRAINPAVE(1) > 0 And v * STRAINPAVE(1) < -STRAINPAVE(2) Then
ELEMENTSTATE(en + n2 * m) = 2
Else
ELEMENTSTATE(en + n2 * m) = 3
End If


Next
Next


End Sub
Sub WRINKLINGstresscalc()

'calculate stresses for roll1 region

Call ROLLmaterial

For q1 = 1 To k1
For q2 = 1 To m

en = (q1 - 1) * m + q2
ReDim STRESSXY(3)

x1 = roll1P(en, 1)
x2 = roll1P(en, 2)
x3 = roll1P(en, 3)
x4 = roll1P(en, 4)
y1 = roll1P(en, 5)
```

```
y2 = roll1P(en, 6)
y3 = roll1P(en, 7)
y4 = roll1P(en, 8)

'pointers

p(1) = 2 * ((q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((q1) * (m + 1) + q2) - 1
p(6) = 2 * ((q1) * (m + 1) + q2)

p(7) = 2 * ((q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((q1) * (m + 1) + q2 + 1)


For m1 = 1 To 8
ue(m1) = u(p(m1))
Next

ReDim STRESSXY(3)

Call WRINKLINGstrain

STRESSXY(1) = D(1, 1) * STRAINXY(1) / 4 + D(1, 2) * STRAINXY(2) / 4
STRESSXY(2) = D(1, 2) * STRAINXY(1) / 4 + D(2, 2) * STRAINXY(2) / 4
STRESSXY(3) = D(3, 3) * STRAINXY(3) / 4

For m1 = 1 To 3
ESA(en, m1) = STRESSXY(m1)
Next

Next
Next

'calculate stresses for span1 region
For q1 = 1 To n2
For q2 = 1 To m

en = (q1 - 1) * m + q2
ReDim STRESSXY(3)

x1 = span1P(en, 1)
x2 = span1P(en, 2)
x3 = span1P(en, 3)
x4 = span1P(en, 4)
y1 = span1P(en, 5)
y2 = span1P(en, 6)
y3 = span1P(en, 7)
y4 = span1P(en, 8)

'pointers

p(1) = 2 * ((k1 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((k1 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((k1 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((k1 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((k1 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((k1 + q1) * (m + 1) + q2)

p(7) = 2 * ((k1 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((k1 + q1) * (m + 1) + q2 + 1)

For m1 = 1 To 8
ue(m1) = u(p(m1))
Next
```

```
Call WRINKLINGstrain

Call WRINKLINGcriteria

STRESSXY(1) = D(1, 1) * STRAINXY(1) / 4 + D(1, 2) * STRAINXY(2) / 4 + D(1, 3) * STRAINXY(3) /
4
STRESSXY(2) = D(2, 1) * STRAINXY(1) / 4 + D(2, 2) * STRAINXY(2) / 4 + D(2, 3) * STRAINXY(3) /
4
STRESSXY(3) = D(3, 1) * STRAINXY(1) / 4 + D(3, 2) * STRAINXY(2) / 4 + D(3, 3) * STRAINXY(3) /
4

For m1 = 1 To 3
ESA(en + k1 * m, m1) = STRESSXY(m1)
Next

Next
Next

'calculate stresses for roll2 region
Call ROLLmaterial

For q1 = 1 To n1
For q2 = 1 To m

en = (q1 - 1) * m + q2
ReDim STRESSXY(3)

x1 = roll2P(en, 1)
x2 = roll2P(en, 2)
x3 = roll2P(en, 3)
x4 = roll2P(en, 4)
y1 = roll2P(en, 5)
y2 = roll2P(en, 6)
y3 = roll2P(en, 7)
y4 = roll2P(en, 8)
'pointers

p(1) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2) - 1
p(2) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2)

p(3) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2 + 1) - 1
p(4) = 2 * ((k1 + n2 + q1 - 1) * (m + 1) + q2 + 1)

p(5) = 2 * ((k1 + n2 + q1) * (m + 1) + q2) - 1
p(6) = 2 * ((k1 + n2 + q1) * (m + 1) + q2)

p(7) = 2 * ((k1 + n2 + q1) * (m + 1) + q2 + 1) - 1
p(8) = 2 * ((k1 + n2 + q1) * (m + 1) + q2 + 1)


For m1 = 1 To 8
ue(m1) = u(p(m1))
Next

ReDim STRESSXY(3)
Call WRINKLINGstrain
STRESSXY(1) = D(1, 1) * STRAINXY(1) / 4 + D(1, 2) * STRAINXY(2) / 4
STRESSXY(2) = D(1, 2) * STRAINXY(1) / 4 + D(2, 2) * STRAINXY(2) / 4
STRESSXY(3) = D(3, 3) * STRAINXY(3) / 4

For m1 = 1 To 3
ESA(en + (k1 + n2) * m, m1) = STRESSXY(m1)
Next

Next
Next


End Sub
```

VITA

HASAN H.YURTCU

Candidate for the Degree of

Doctor of Philosophy

Thesis: DEVELOPMENT OF NONLINEAR FINITE ELEMENT CODES FOR STABILITY STUDIES OF WEB MATERIAL WITH STRAIN STATE DEPENDENT PROPERTIES

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Afyon,Turkiye, On November 2, 1979, the son of Mustafa and Samiye Yurtcu.

Education: Graduated from Ertugrulgazi High School, Eskisehir, Turkiye in June 1998; received Bachelor of Science in Mathematical Engineering, from Yildiz Technical University Istanbul, Turkiye, in March 2003; received Master of Science in Mathematical Engineering from Yildiz Technical University Istanbul, Turkiye, in December 2006. Completed the requirements for the Doctor of Philosophy degree with a major in Mechanical Engineering at Oklahoma State University, Stillwater, Oklahoma in December 20011.

Experience:
Research Asistant at Yildiz Technical Unv.
Employed by Department of Mechanical Engineering at Oklahoma State University, Stillwater as a graduate teaching and research assistant, 2007 to 2011.

Name: HASAN H.YURTCU

Date of Degree: December, 2011

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: DEVELOPMENT OF NONLINEAR FINITE ELEMENT CODES FOR STABILITY STUDIES OF WEB MATERIAL WITH STRAIN STATE DEPENDENT PROPERTIES

Pages in Study: 201

Candidate for the Degree of Doctor of Philosophy

Major Field: Mechanical & Aerospace Engineering

Scope and Method of Study: The objective of the proposed research is to develop user friendly finite element codes that will solve nonlinear instability problems associated with strain state dependent material properties and boundary conditions of moving webs. This code will be unique and will have economic value by helping minimize web material losses.

Findings and Conclusions: The results of the work are user friendly finite element codes applicable to the following four cases: for a web encountering a misaligned roller, a web encountering a tapered roller, a web with a central circular discontinuity, and a web with a central circular non uniformity. The codes developed rely on inputs of simple geometry and material parameters which should be apparent to the user. From there on the codes execute automatically in the meshing of the problem, the generation of the elastic and wrinkle membrane elements, and the solution of the nonlinear instability problem. The results from this codes have been shown to compare favorably with test results and results from commercial FE codes.

ADVISER’S APPROVAL: Dr. J.K. Good