A GENERALIZED COHESIVE ZONE MODEL OF PEEL TEST FOR PRESSURE SENSITIVE ADHESIVES

A Thesis

by

LIANG ZHANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2008

Major Subject: Mechanical Engineering

A GENERALIZED COHESIVE ZONE MODEL OF PEEL TEST FOR PRESSURE SENSITIVE ADHESIVES

A Thesis

by

LIANG ZHANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Jyhwen Wang
Committee Members,	Ahmed Amine Benzerga
	Xin-Lin Gao
Head of Department,	Dennis O'Neal

May 2008

Major Subject: Mechanical Engineering

ABSTRACT

A Generalized Cohesive Zone Model of Peel Test for Pressure Sensitive Adhesives.

(May 2008)

Liang Zhang, B.S., University of Science and Technology of China

Chair of Advisory Committee: Dr.Jyhwen Wang

The peel test is a commonly used testing method for adhesive strength evaluation. The test involves peeling a pressure sensitive tape away from a substrate and measuring the peel force that is applied to rupture the adhesive bond. In the present study, the mechanics of the peel test is analyzed based on a cohesive zone model. Cohesive failure is assumed to prevail in the vicinity of the peel front, that is, the adhesive fails not by debonding from the adherends but by splitting of the adhesive itself. Generally, the failure of the adhesive is accompanied with a process of cavitation and fibrillation. Therefore, the cohesive zone is modeled as a continuous fibrillated region. A Maxwell model is employed to characterize the viscoelastic behavior of the adhesive. The governing equation and boundary conditions that describe the mechanics of the peel test are derived. Numerical results are obtained under steady state conditions. The model predicts the peel force in terms of the peel rate, the peel angle, the nature of the adhesive, and the properties of the backing and the substrate. The traction distribution on the substrate surface is found to depend on various test parameters. Finally, finite element analysis is performed using the commercial software package ABAQUS. The results

from FEA are compared with those from the mathematical method to evaluate the validity of the present model. The effective range of the present model is found to be related to the ratio of the critical fibril length to the extent of the cohesive zone. Given the nature of the adhesive as well as the properties of the backing and the substrate, the proposed model is able to predict the peel force and the traction distribution in terms of the peel rate and the peel angle, and thus provides a measure of the strength of the adhesive bond.

NOMENCLATURE

V	peel rate
ψ	peel angle
V^*	speed of the peel front propagation
l_0	adhesive thickness in the undeformed state
h	half backing thickness
E	Young's modulus of the backing
ν	Poisson's ratio of the backing
τ	yield stress in shear of the backing
F	peel force per unit width of the backing
E^*	$E/(1-\nu^2)$
s	arc length distance between points O and P
s^+	arc length distance between points O^+ and P^+
θ	inclination of the neutral axis at P measured from the x axis
κ	curvature of the neutral axis at P
I	moment of inertia per unit width of the backing
R	radius of curvature of the substrate surface
s^-	arc length distance between points O^- and P^-
φ	s^-/R
l	length of the fibril joining P^+ and P^-
α	inclination of the fibril measured from the x axis

t	time
κ^+	curvature of the lower surface at P^+
C_0	instantaneous compliance of a viscoelastic material
C_1	material constant of a viscoelastic material
m	creep exponent of a viscoelastic material
η	viscosity of the adhesive
σ	stress in the fibril
A_0	initial cross section areas of the fibril
A	current cross section areas of the fibril
σ^*	traction per unit area of the lower surface of the backing
s_b	location of the last intact fibril on the substrate surface
$\vec{i}$	unit vector in the x direction
$\vec{j}$	unit vector in the y direction
$\vec{r}_{PP^+}$	vector that points from P to P^+
θ_b	value of θ at $s = s_b$
L	length of the peel arm
f	horizontal force acting on the backing
v	vertical force acting on the backing
λ_b	critical stretch ratio of the fibril just before cohesive failure
β	$s_b^3/2I$

TABLE OF CONTENTS

Page

ABSTRACT iii

NOMENCLATURE v

TABLE OF CONTENTS vii

LIST OF FIGURES viii

1. INTRODUCTION 1

    1.1 Pressure Sensitive Adhesives 1
    1.2 Peel Test 2
    1.3 Factors Influencing Peel Test 4
    1.4 Research Objective 11

2. THEORETICAL CONSIDERATIONS 13

    2.1 Peel Geometry 13
    2.2 Constitutive Model 15
    2.3 Governing Equations 20
    2.4 Failure Criterion 25
    2.5 Dimensional Analysis 26

3. NUMERICAL METHOD 28

    3.1 Discretized Governing Equations 28
    3.2 Dimensionless Discretized Governing Equations 31

4. FINITE ELEMENT ANALYSIS 33

5. RESULTS AND DISCUSSION 37

6. CONCLUSION 52

REFERENCES 54

APPENDIX A 57

Page

APPENDIX B 59

APPENDIX C 78

VITA 82

LIST OF FIGURES

		Page
Figure 1	Schematics of the peel test	3
Figure 2	Tape profile and typical traction distribution predicted based on elementary beam bending theory by Kaelble	5
Figure 3	Experimental traction distribution obtained and profile proposed by Kaelble and Reylek	6
Figure 4	Schematics of the shrinkage in the transverse directions of the fibrils during the peel test	9
Figure 5	Schematics of the peeling device in the work of Kaelble	11
Figure 6	Schematics of (a) the cohesive zone and (b) the line elements PP^+ and P^+P^-	16
Figure 7	Moment balance in the cohesive zone	22
Figure 8	Transformation of the coordinate system	24
Figure 9	Free body diagram of two neighboring elements of the backing	28
Figure 10	Geometry of FEA model: (a) undeformed; (b) deformed	34
Figure 11	Dimensionless peel shapes from the mathematical method and FEA	38
Figure 12	Dimensionless traction distributions from the mathematical method and FEA	39
Figure 13	Dimensionless peel shapes from the mathematical method and FEA for high dimensionless critical fibril length	40
Figure 14	Dimensionless traction distributions for different critical stretch ratios	42
Figure 15	Dimensionless traction distributions for different critical stretch ratios	44

Page

Figure 16 Dimensionless traction distributions for different peel angles..... 45

Figure 17 Dimensionless peel rate versus dimensionless peel force for different peel angles..... 46

Figure 18 Dimensionless traction distributions for different viscosities of the adhesive and modulii of the backing..... 48

Figure 19 Dimensionless traction distributions for different backing thicknesses..... 49

Figure 20 Dimensionless traction distributions for different dimensionless radii of curvature of the substrate surface 51

1. INTRODUCTION

1.1 Pressure Sensitive Adhesives

The adhesion between soft and stiff materials comprises an important component of many technology applications. One of the most important adhesions is that of pressure sensitive adhesives (PSAs). PSAs are designed to ensure instantaneous adhesion upon application of a light pressure. The adhesion can be either permanent or removable. Permanent adhesives are used in applications such as safety labels for power equipment, foil tape for HVAC duct work, automotive interior trim assembly, and sound/vibration damping films. Most applications further require that PSAs can be easily removed from the adherends by applying a light pulling force. Example of removable applications include surface protection films, masking tapes, bookmark and note papers, price marking labels, and promotional graphics materials.

PSAs have been widely used since the late 19th century [1]. The application of PSAs started with medical tapes and dressing. The first U.S. patent regarding to PSAs, in which they was used for a soft, adhering bandage, was issued in 1846. Ninety years after that, Stanton Avery introduced the self-adhesive label. Two major industries –pressure sensitive tapes and labels –resulted from these innovations. Industrial tapes were

---

This thesis follows the style of International Journal of Adhesion and Adhesives.

developed in the 1920s and 1930s, while self-adhesive labels were introduced in 1935. Pressure sensitive protective films were manufactured about ten years later. Solvent-based PSAs using natural rubber were firstly developed in the 19th century. Hot-melt adhesives were developed in the 1940s. Recent developments of PSAs include pressure sensitive products without a coated PSA layer.

Pressure sensitive tapes are designed that, by applying a light pressure, PSAs form a continuous layer to bond the tape to the adherend. The layer has to be soft enough to adhere to the adherend, whereas it has to be hard enough to offer a proper bond resistance. This special behavior requires PSAs exhibiting a viscoelastic character [1]. The viscoelastic behavior of PSAs results from their macromolecular nature. Generally, PSAs in a bond behave like fluids or solids. Fluids are characterized by their viscosity, whereas solids are characterized by their elastic modulus. The mechanical properties of PSAs are time dependent. Creton and Leibler [2] studied the tack of PSAs on rigid substrates. The viscoelastic behavior was found to be accounted for a time dependent elastic modulus. Hui et al. [3] extended their work to model the tack of PSAs on a rough surface with a Gaussian distribution of asperity heights.

1.2 Peel Test

The strength of the adhesive bond can be quantified with a peel test. A 120° peel test is illustrated schematically in Figure 1, in which a thin flexible backing that has been bonded to a rigid surface is peeled away from the substrate. The peel test provides a

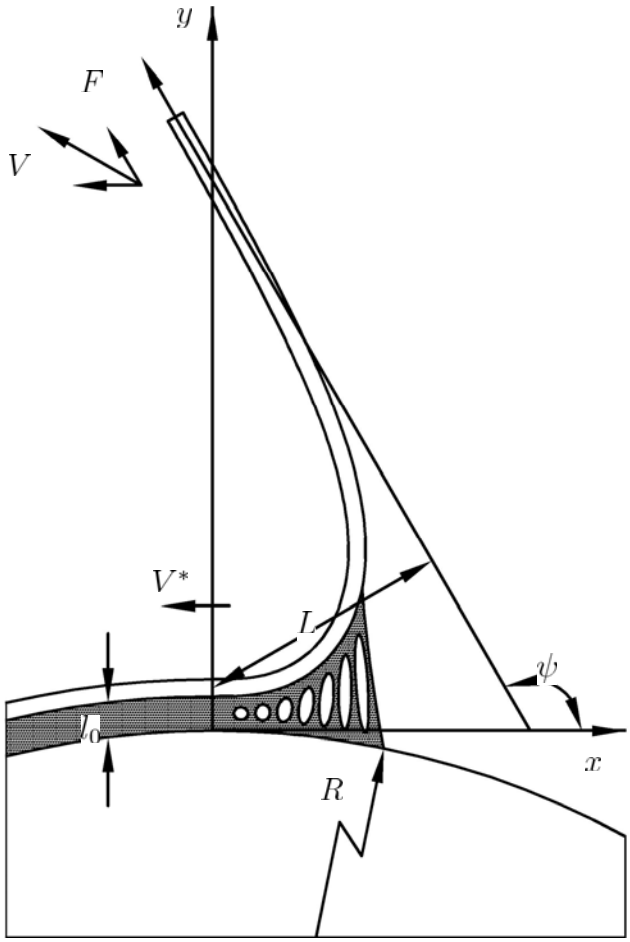


Figure 1. Schematics of the peel test.

measure of the strength of the adhesive bond. The strength can be either cohesive or adhesive, depending on the mode of failure. The test is commonly conducted by pulling the backing at a specific rate and angle, and measuring the peel force that is applied to rupture the adhesive bond. The peel force depends on such factors as the rate at which the backing is detached, the angle at which the detachment occurs, the nature of the adhesive, the mechanical and physical properties of the backing and the substrate, the

temperature and humidity of the environment, and the conditioning process [4]. However, the interaction of these factors still remains unclear.

1.3 Factors Influencing Peel Test

The fundamental theories of the peel test result primarily form the work based on elementary beam bending theory by Kaelble [5]. The theory assumes the adhesive to be linearly elastic. A sharp bond boundary is assumed at the line of detachment (see Fgiure 2). This assumption eliminates the process of cavitation and fibrillation of the adhesive. As a result, the predicted traction distribution shows a sudden transition from maximum to zero stress in the vicinity of the peel front.

Kaelble and Reylek [6] succeeded in determining experimentally the traction distribution in the vicinity the peel front. They demonstrated that the peel force is significantly affected by cavitation and fibrillation of the adhesive. This process influences the peel force even under conditions in which the fibrils are visually nearly undetectable. The corresponding traction distribution differs from that by Kaelble [5]. The traction attains a peak value in the vicinity of the peel front and then decreases moderately across the fibrillated zone (see Figure 3).

Niesiolowski and Aubrey [7] observed the pattern of traction distribution during the peeling of pressure sensitive tapes. They also used scale-up rules derived peeling theory to predict the pattern of traction distribution theoretically. They suggested that linear scale-up of the backing thickness, the adhesive thickness and the peel rate would

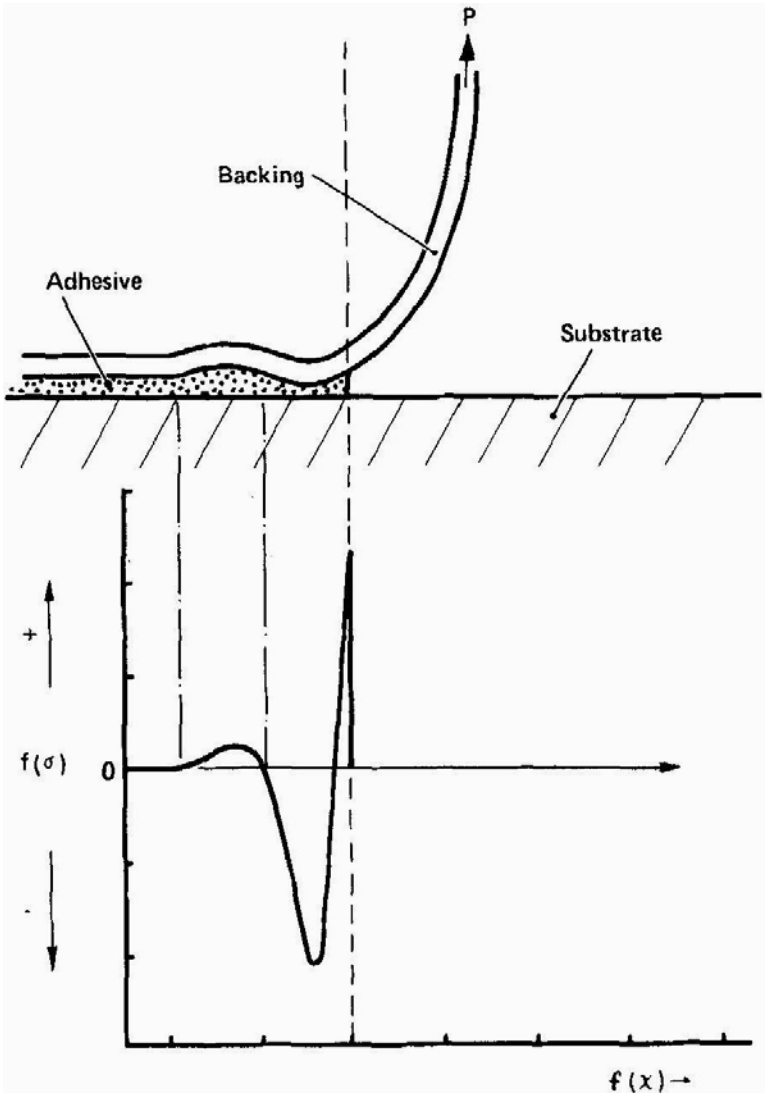


Figure 2. Tape profile and typical traction distribution predicted based on elementary beam bending theory by Kaelble [5].

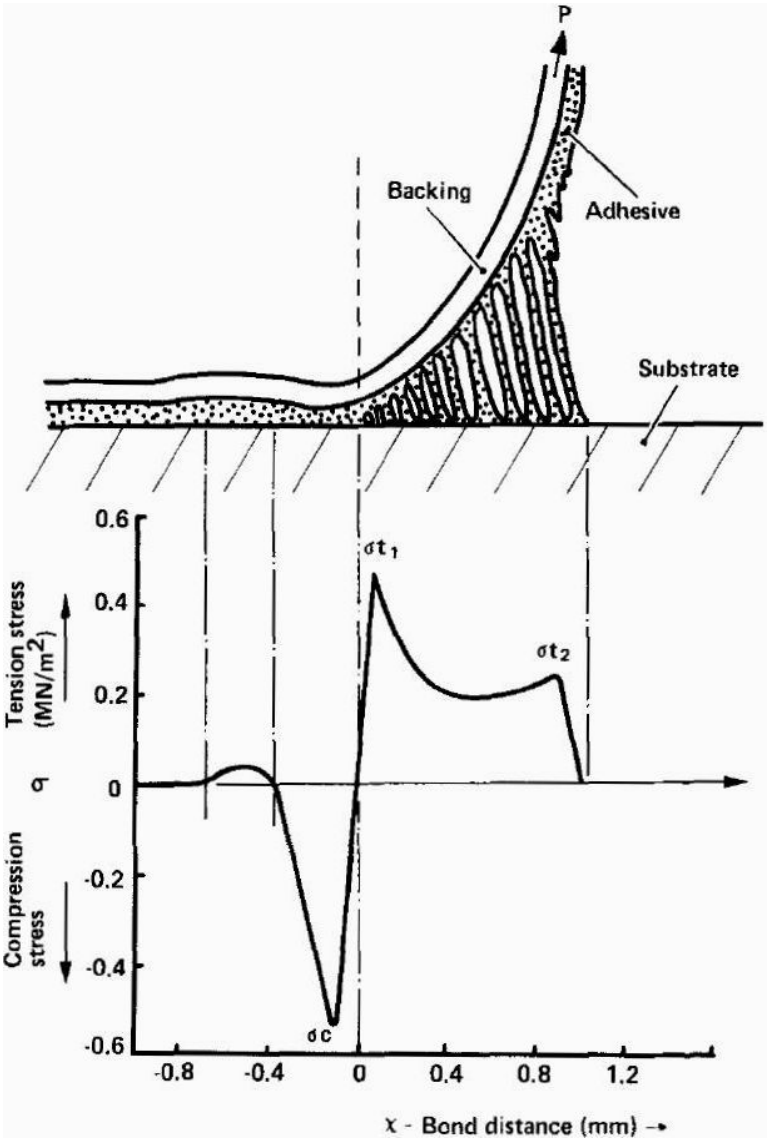


Figure 3. Experimental traction distribution obtained and profile proposed by Kaelble and Reylek [6].

give rise to a correspongding linear increase in the peel force. The predicted patterns were compared to those obtained experimentally. The results indicate that the traction distribution and thus the peel force are affected by fibrillation significantly. However, the proposed scale-up rules imply that the peel force is proportional to the backing thickness, the adhesive thickness and the peel rate, which is not generally the case. The proposed method is hereby incapable of analyzing the mechanics of the peel test accurately.

The propagation of the peel front is often accompanied by a process of cavitation and fibrillation of the adhesive [8]. The adhesive in the vicinity of the peel front is subjected to large hydrostatic tension due to the lateral constraint imposed by the adherends. For strong adhesives, the adhesive does not debond so that newly formed cavities grow vertically perpendicular to the interface. New cavities will continue to form as long as the average spacing between them is large enough to provide sufficient lateral constraint. These cavities grow rapidly to a critical level at which the hydrostatic tension is used to extend an array of fibrils.

The fibrils are often modeled as deformable strings, that is, the fibrils are subjected to uniaxial extension. Gent and Petrich [9] calculated the peel force by summing tensile stresses in fibrils of the adhesive across the peel front. The peel force was found to depend on such parameters as the adhesive thickness, the critical extension of fibril break or detachment, and the tension in the fibrils. Gutpa [10] pointed out that it is difficult to represent the peel process with controlled deformation as the peel geometry is not known a priori. However, Gent and Petrich [9] neglected the contribution of the

flexibility of the backing on the peel force. Generally, the peel force does not merely represent the true strength of the adhesive bond. The measured peel force may represent a combination of the true strength of the adhesive bond and other work expended in the elastic and plastic deformation of the adherends. This can be understood by considering a $180°$ peel test, where work is being done not only to rupture the adhesive bond but also to bend the backing through $180°$ [1].

Christensen et al. [11, 12] quantified the deformation of the adhesive in photo images. The relationship between the peel force and the dimensionless deformation of the fibrils was obtained. The tension in the fibrils was found to increase monotonically across the peel front, while the magnitude of the tension increases as the peel rate increases. They compared the theoretical peel forces with measurements and demonstrated the validity of the expression and the string assumption [13]. They also observed that the extension of a fibril in its longitudinal direction is accompanied by the extensive shrinkage in its transverse directions (see Figure 4).

Lin et al. [8] analyzed the failure of the adhesive in a $180°$ peel test. They compared the failure process in a peel test with that in a tack test and assumed that cavitation and fibrillation of the adhesive prevail in the vicinity of the peel front. Governing equations for the peel test were derived based on the fibrillated zone model. The peel force was found to be dependent on the adhesive thickness and also the peel rate. The logarithmic relationship between the peel force and the peel rate was shown to be linear. However, they failed to model the peel geometry accurately. In the work of Lin, the fibrils were modeled as joining the neutral axis of the backing and the substrate

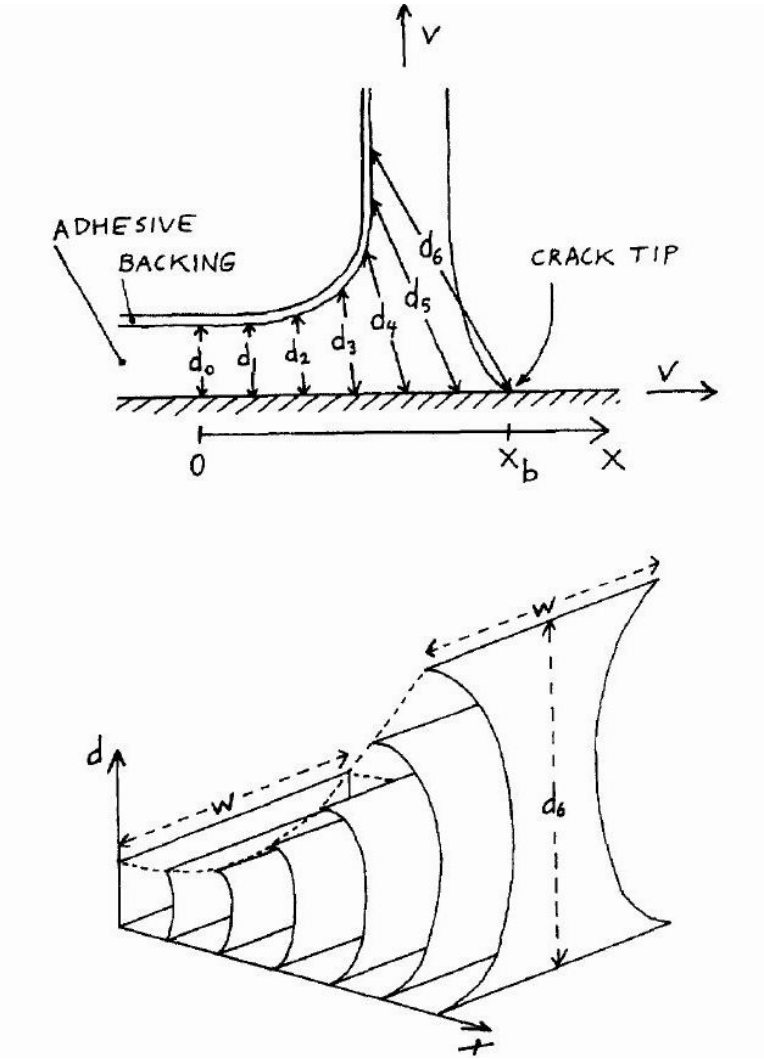


Figure 4. Schematics of the shrinkage in the transverse directions of the fibrils during the peel test [12].

surface, whereas the fibrils are in fact joining the lower surface of the backing and the substrate surface. Such an inadequacy may cause the elongation and orientation of the spring elements, the magnitude of surface traction acting on the backing, and thus the moment about the neutral axis of the backing produced by the fibrils to suffer a high degree of error compared with the actual values.

The flexibility, geometry and dimensions of the backing influence the peel force. Several authors [4, 14-17] studied the influence of flexible backings on the peel force. Wei and Hutchinson [15] investigated the relationship between the peel force and the adhesive fracture energy in the presence of the plastic deformation of the peel arm. The adhesive thickness was taken to be zero. A traction-separation relation was employed to govern the deformation of the adhesive. The macroscopic fracture energy was found to be the sum of the adhesive fracture energy and the plastic dissipation. However, the adhesive thickness cannot be neglected for PSA tapes. The influence of the adhesive thickness hereby should be evaluated. Furthermore, the traction-separation relation might not be able to characterize the viscoelastic behavior of PSAs. A time dependent creep function should be specified.

The nature of the substrate influences the peel force as well. Kaelble [18, 19] conducted a series of experiments in 1959. In the experiments, cellophane/rubber resin based adhesive tapes were peeled away from a cellophane surface at peel angles ranging from $20°$ to $170°$. A string wheel about 150mm in diameter was employed (see Figure 5). The peel rate was monitored by controlling the rotational speed of the wheel. Since the adhesive is much softer than the wheel, its deformation is strongly influenced by the

geometry of the wheel. As the adhesive thickness approaches the radius of the wheel, the influence of the radius on the peel force becomes significant.

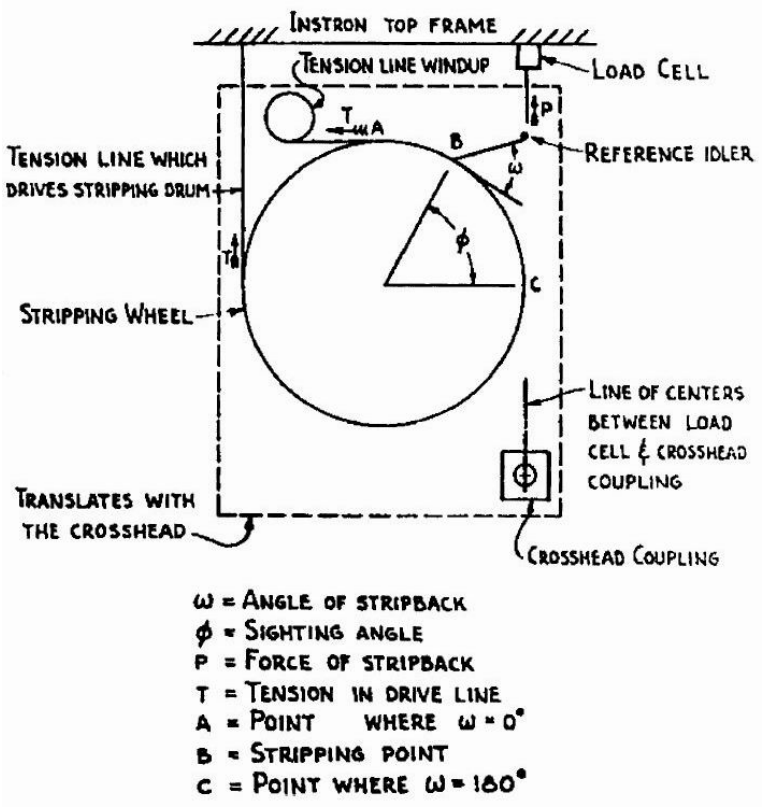


Figure 5. Schematics of the peeling device in the work of Kaelble [18, 19].

1.4 Research Objective

The objective of the present research is to provide a generalized model that is capable of characterizing the mechanics of the peel test. A cohesive zone model, in which the adhesive fails by splitting of the adhesive itself, is introduced. The cohesive

zone is modeled as a continuous fibrillated region, where the viscoelastic behavior of the adhesive is characterized by the Maxwell model. The peel geometry is modeled accurately, that is, the fibrils are modeled as joining the lower surface of the backing and the substrate surface. Under steady state conditions, the model predicts the peel force in terms of the peel rate, the peel angle, the nature of the adhesive, and the properties of the backing and the substrate. The mechanical response of the adhesive in the presence of a curved substrate surface is evaluated. The traction distribution on the substrate surface is calculated to investigate the influences of various factors. Finally, finite element analysis is performed using the commercial software package ABAQUS. The results from FEA are compared with those from the mathematical method to evaluate the validity of the present model.

2. THEORETICAL CONSIDERATIONS

2.1 Peel Geometry

The peel geometry is displayed in Figure 1. Take the origin of the coordinate system $O^{-}xy$ at current location of the peel front. The coordinates x and y are set to be in the tangential and normal directions of the substrate surface at the origin, respectively. The peel test is conducted at a rate V and an angle ψ. The peel angle ψ specifies the angle which the peel force acts with the x direction in the moving reference frame. The peel front propagates with a speed V^*, where as the peel arm moves at a relative speed V^* with respect to the moving origin. The peel rate V can be related to V^* by vector addition. The direction of the peel rate is as shown in Figure 1, while the magnitude of peel rate is given by

$$V = V^* \sqrt{2\,(1 - \cos\psi)}. \tag{1}$$

The adhesive thickness in the undeformed state is taken to be l_0. Generally, the peel force also depends on the adhesive thickness. The peel value increases as the adhesive thickness increases [1]. The adhesive thickness can be treated as a variable, but it will not be considered in the present work in the interest of reducing the number of variables. The adhesive is assumed to be not removable [20], that is, the adhesive does not debond from the backing and the substrate during the peel test. This assumption implies that cohesive failure or splitting of the adhesive itself prevails.

The width of the backing is assumed to be sufficiently large compared to its thickness $2h$, so that plane strain conditions are appropriate to describe the deformation of the backing. The backing is taken to be linearly elastic with Young's modulus E, Poisson's ratio ν and yield stress in shear τ. The peel force per unit width of the backing is F. The backing will not yield as long as

$$\sqrt{3E^*F/h} < \tau, \tag{2}$$

where $E^* = E/(1-\nu^2)$ for plane strain conditions. It is necessary to make the assumption that the backing is of a relatively high stiffness compared with the adhesive and undergoes negligible deformation in shear and extension. Specifically, consider a line element PP^+ that lies perpendicular to the neutral axis, where points $P\,(x,\ y)$ and $P^+\,(x^+,\ y^+)$ locate on the neutral axis and the lower surface, respectively (see Figure 6). The arc length distance between points O and P is s, and the arc length distance between points O^+ and P^+ is s^+. s^+ is related to s by

$$s^+ = s + h\theta, \tag{3}$$

where θ denotes the inclination of the neutral axis at P measured from the x axis. x^+ and y^+ are related to x and y by

$$x^+ = x + h\sin\theta, \tag{4}$$

$$y^+ = y - h\cos\theta, \tag{5}$$

The curvature of the neutral axis at P follows from the definition as

$$\kappa = \frac{d\theta}{ds}. \tag{6}$$

The backing is modeled as elastic so that the curvature of the neutral axis after bending is related to the applied bending moment per unit width by

$$M = E^* I \kappa, \tag{7}$$

where $I = (2h)^3 / 12$ is the moment of inertia per unit width.

The substrate is modeled as a singly curved adherend with a constant radius of curvature R, where R is much larger than l_0 and h. Specifically, consider a line element P^+P^- that lies perpendicular to the substrate surface before the detachment of the adherends, where point P^- $(x^-,\ y^-)$ locates on the substrate surface (see Figure 6). The line element P^+P^- can be treated as a fibril with its initial length l_0. Let s^- be the arc length distance between points O^- and P^- so that x^- and y^- can be written as

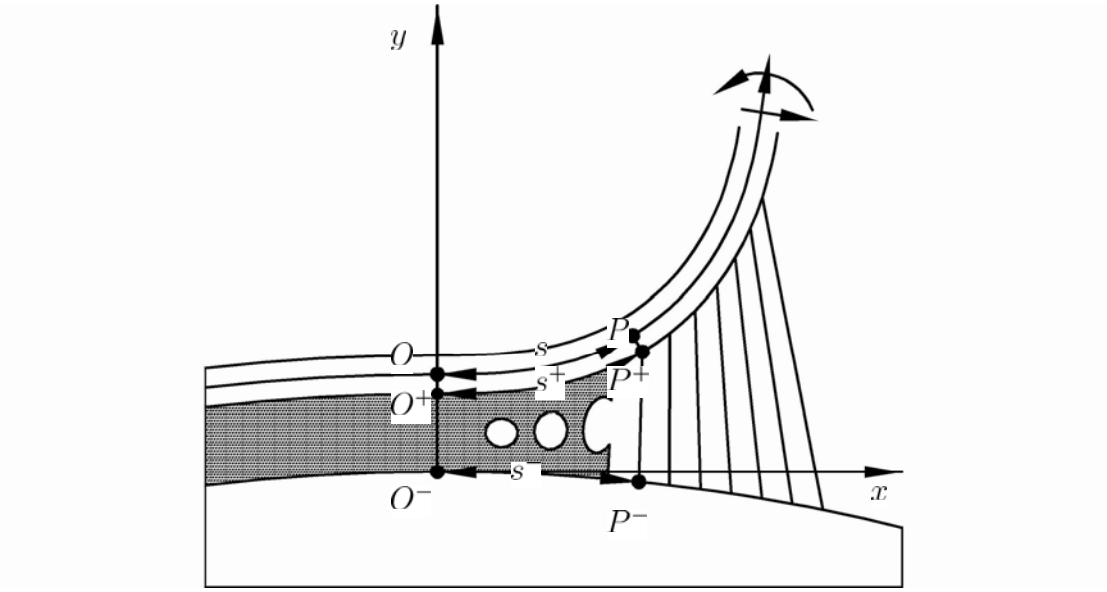
$$x^- = R \sin \varphi, \tag{8}$$

$$y^- = -R\left(1 - \cos \varphi\right), \tag{9}$$

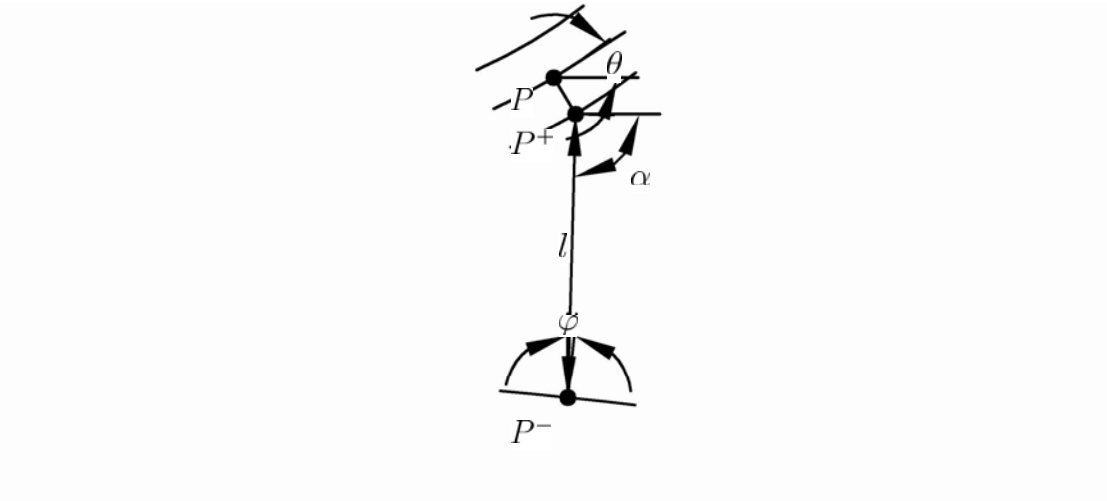
where $\varphi = s^- / R$.

2.2 Constitutive Model

Now consider the case of steady state conditions, in which the peel rate V and the peel angle ψ are independent of time. The peel front is assumed to have propagated a sufficient distance so that steady state conditions prevail in the vicinity of the peel front. Thus the peel shape and also the deformation of the fibrillated adhesive are independent of time with respect to the moving reference frame. After the detachment occurs, P^-



(a)



(b)

Figure 6. Schematics of (a) the cohesive zone and (b) the line elements PP^+ and P^+P^-.

stays on the substrate surface, whereas P^+ has moved with the lower surface of the backing (Figure 6). It is a valid approximation to model the fibrils as deformable strings. Let l be the length of the fibril joining P^+ and P^-. l can be given by

$$l = \sqrt{(x^- - x^+)^2 + (y^- - y^+)^2}. \tag{10}$$

Let α be the inclination of the fibril measured from the x axis. α can be represented by

$$\cos\alpha = \frac{x^- - x^+}{\sqrt{(x^- - x^+)^2 + (y^- - y^+)^2}}, \tag{11}$$

$$\sin\alpha = \frac{y^- - y^+}{\sqrt{(x^- - x^+)^2 + (y^- - y^+)^2}}. \tag{12}$$

As can be seen in eq (10), l is a function of the current positions of P^+ and P^-. Note that the backing undergoes negligible deformation in shear and extension so that $s = s^-$. This equation, together with eq (3), implies that l is a function of s and θ.

Now suppose that the fibril joining P^+ and P^- is a fictitious fibril to the left of $s = 0$ in the moving reference frame. Let the fibril translate to the right with P^+ and P^- moving to the right along their respective surfaces at a speed V^* synchronously. It is deformed to the right of $s = 0$. Failure occurs at $s = s_b$ after it has been extended in a homogeneous manner. Note that the deformation of the adhesive is independent of time with respect to the moving reference frame so that the fictitious fibril experienced identical strain history with a fibril in a fixed laboratory frame. Therefore, time t in the fixed laboratory frame can be expressed in terms of s and V^* in the moving reference frame as $t = s/V^*$. The rate of extension of the fibril can then be expressed in terms of s as

$$\dot{l} = \frac{\partial l}{\partial s}\dot{s} + \frac{\partial l}{\partial \theta}\theta'(s)\,\dot{s} = V^* \left(\frac{\partial l}{\partial s} + \kappa \frac{\partial l}{\partial \theta} \right), \tag{13}$$

where dot denotes d/dt. The derivatives $\partial l/\partial s$ and $\partial l/\partial \theta$ are calculated using chain rule:

$$\begin{aligned} \frac{\partial l}{\partial s} &= \frac{\partial l}{\partial x^+}\frac{\partial x^+}{\partial x}\frac{dx}{ds} + \frac{\partial l}{\partial y^+}\frac{\partial y^+}{\partial y}\frac{dy}{ds} + \frac{\partial l}{\partial x^-}\frac{dx^-}{ds} + \frac{\partial l}{\partial y^-}\frac{dy^-}{ds} \\ &= \cos(\alpha + \varphi) - \cos(\alpha - \theta), \end{aligned} \tag{14}$$

$$\frac{\partial l}{\partial \theta} = \frac{\partial l}{\partial x^+}\frac{\partial x^+}{\partial \theta} + \frac{\partial l}{\partial y^+}\frac{\partial y^+}{\partial \theta} = -h\cos(\alpha - \theta), \tag{15}$$

where $\cos\theta = dx/ds$ and $\sin\theta = dy/ds$. Eq (13) can then be rewritten as:

$$\dot{l} = V^* \left[\cos(\alpha + \varphi) - (1 + h\kappa)\cos(\alpha - \theta) \right]. \tag{16}$$

Note that the line element PP^+ does not deform after bending, so that it is equivalent to take $\cos\theta = dx^+/ds^+$ and $\sin\theta = dy^+/ds^+$ in eq (16). Furthermore, the curvature of the neutral axis is related to that of the lower surface by

$$\kappa = \frac{\kappa^+}{1 - h\kappa^+}. \tag{17}$$

Therefore, the rate of extension of the fibril can be written in terms of s^+ as

$$\dot{l} = V^* \left[\cos(\alpha + \varphi) - \frac{1}{1 - h\kappa^+}\cos(\alpha - \theta) \right], \tag{18}$$

where the term $h\kappa^+$ results from the change in length of the lower surface and is negligible in cases where the backing is of thin cross section.

The compliance function of the adhesive is assumed to have the form [21]:

$$C(t) = C_0 + C_1 t^m, \quad 0 < m \leq 1, \tag{19}$$

where C_0 is the instantaneous compliance, C_1 is a material constant, and m is the creep exponent. The special case of $m = 1$ corresponds to a Maxwell solid representing the

viscoelastic behavior of PSAs. Specifically, assume that $C_0 = 0$ and $C_1 = 1/\eta$, where η is the viscosity of the adhesive. The one-dimensional mechanical response of a Maxwell solid is generally represented by a mechanical analog – a viscous damper [22]. The response of the viscous damper is characterized by a relationship between stress and strain rate. Let $\sigma\,(t)$ and $\epsilon\,(t)$ be an arbitrary stress history and the corresponding strain response, respectively. The constitutive equation of the Maxwell solid has the form:

$$\dot{\epsilon} = \frac{\sigma}{\eta}. \tag{20}$$

The string assumption implies that the fibrils cannot support stresses normal to the plane of deformation. Therefore, even though plane strain conditions are assumed in the backing, plane stress conditions prevail in the adhesive. The constitutive equation of a fibril in the moving reference frame can be obtained by rewriting eq (20) as

$$\frac{\dot{l}}{l} = \frac{\sigma}{\eta}, \tag{21}$$

where the stress σ in the fibril is now a function of s^+. Constancy of volume is assumed in the fibril so that $l_0 A_0 = lA$, where A_0 and A are the initial and current cross section areas of the fibril, respectively. Let σ^* be the traction per unit area of the lower surface of the backing. Since the tension in the fibril must balance the surface traction on the lower surface, σ^* is related to σ by

$$\sigma^* = \frac{\sigma A}{A_0} = \frac{\sigma l_0}{l}. \tag{22}$$

This equation, together with eqs (16) and (21), gives a general representation of the traction:

$$\sigma^* = \frac{\eta l_0 \dot{l}}{l^2} = \frac{\eta l_0 V^*}{l^2}\left[\cos(\alpha+\varphi) - \frac{1}{1-h\kappa^+}\cos(\alpha-\theta)\right]. \tag{23}$$

As can be seen in eq (23), σ^* depends on the viscosity, the adhesive thickness, the speed of the peel front propagation, and parameters related to the peel geometry.

2.3 Governing Equations

A continuum analysis of the fibrils is required to obtain accurate results in the vicinity of the peel front. If the fibrils are assumed to be sufficiently close to each other, the cohesive zone can be treated as a continuum fibrillated region [8]. The peel geometry can then be sub-divided into two regions: the adhesive, the backing and the substrate $s < s_b$, and the detached infinite backing segment $s > s_b$, where s_b is the location of the last intact fibril on the substrate surface.

Now consider the moment per unit width about the neutral axis of the backing in the cohesive zone. The fibril joining P^+ and P^- occupies a small arc length ds^+ on the lower surface of the backing. Let $\vec{\sigma}^* ds^+$ be the surface traction acting on P^+, where $\vec{\sigma}^*$ is the traction vector and $|\vec{\sigma}^*| = \sigma^*$. The string assumption implies that $\vec{\sigma}^* ds^+$ is in the direction of the straight line join P^+ and P^+. $\vec{\sigma}^*$ can then be written as

$$\vec{\sigma}^* = \sigma^*\left(\cos\alpha\vec{i} + \sin\alpha\vec{j}\right), \tag{24}$$

where $\vec{i}$ and $\vec{j}$ are unit vectors in the x and y directions, respectively. The moment per unit width at P produced by $\vec{\sigma}^* ds^+$ is

$$\left|\vec{r}_{PP^+} \times \vec{\sigma}^* ds^+\right| = \left|\left[\left(x^+ - x\right)\vec{i} + \left(y^+ - y\right)\vec{j}\right] \times \sigma^* \left(\cos\alpha\vec{i} + \sin\alpha\vec{j}\right) ds^+\right|$$
$$= \sigma^* \left[\left(x^+ - x\right)\sin\alpha - \left(y^+ - y\right)\cos\alpha\right] ds^+. \tag{25}$$

where $\vec{r}_{PP^+}$ points from P to P^+; $\times$ is the vector cross product. Substitution of eq (23) into eq (25) gives

$$\left|\vec{r}_{PP^+} \times \vec{\sigma}^* ds^+\right| = \frac{\eta l_0 V^*}{l^2}\left[\cos\left(\alpha+\varphi\right) - \frac{1}{1-h\kappa^+}\cos\left(\alpha-\theta\right)\right]$$
$$\cdot\left[\left(x^+ - x\right)\sin\alpha - \left(y^+ - y\right)\cos\alpha\right] ds^+. \tag{26}$$

The resultant moment produced by the fibrils can be obtained by integrating $\left|\vec{r}_{PP^+} \times \vec{\sigma}^* ds^+\right|$ over $s + h\theta < s^+ < s_b + h\theta_b$, where θ_b is the value of θ at $s = s_b$ as shown in Figure 7. This integral plus the moment produced by the peel force F gives the moment per unit width at P as

$$M = F\left[L - x\sin\psi + y\cos\psi - \left(l_0 + h\right)\cos\psi\right]$$
$$+\eta l_0 V^* \int_{s+h\theta}^{s_b+h\theta_b}\left[\cos\left(\alpha+\varphi\right) - \frac{1}{1-h\kappa^+}\cos\left(\alpha-\theta\right)\right]$$
$$\cdot\frac{\left(x^+ - x\right)\sin\alpha - \left(y^+ - y\right)\cos\alpha}{\left(x^- - x^+\right)^2 + \left(y^- - y^+\right)^2} ds^+, \tag{27}$$

where L is the length of the peel arm. It should be noted that L is measured from O. Eqs (7) and (27), together with

$$\kappa = \frac{d^2y/dx^2}{\left[1 + \left(dy/dx\right)^2\right]^{3/2}}, \tag{28}$$

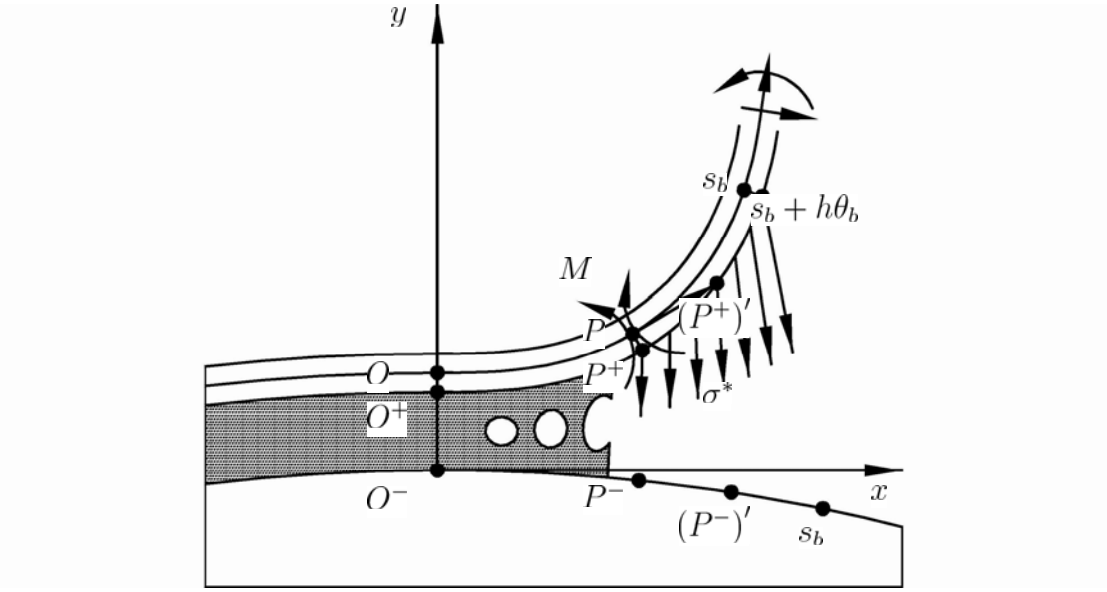


Figure 7. Moment balance in the cohesive zone.

determine the peel shape in the cohesive zone. Combining eqs (7), (27) and (28) gives

$$\frac{d^2y/dx^2}{\left[1+(dy/dx)^2\right]^{3/2}} = \frac{F\left[L - x\sin\psi + y\cos\psi - (l_0+h)\cos\psi\right]}{E^*I} + \frac{\eta l_0 V^*}{E^*I}\int_{s+h\theta}^{s_b+h\theta_b}\left[\cos(\alpha+\varphi) - \frac{1}{1-h\kappa^+}\cos(\alpha-\theta)\right] \cdot\frac{(x^+-x)\sin\alpha - (y^+-y)\cos\alpha}{(x^- - x^+)^2 + (y^- - y^+)^2}ds^+, \tag{29}$$

which is the governing equation for the peel shape in the cohesive zone. Once the peel rate and the peel angle are specified, the unknowns in eq (29) are F, L, s_b and the peel shape $(x,\ y)$. The boundary conditions at O are

$$\theta_0 = 0, \tag{30}$$

$$M_0 = 0. \tag{31}$$

Continuity of displacement, inclination, force and moment are required at $s = s_b$. Let f and υ be the horizontal and vertical forces acting on the backing, respectively. The supplemental boundary conditions at the end of the cohesive zone are

$$f_b = -F\cos\psi, \tag{32}$$

$$\upsilon_b = -F\sin\psi, \tag{33}$$

$$M_b = F\left[L - x_b\sin\psi + y_b\cos\psi - (l_0 + h)\cos\psi\right]. \tag{34}$$

The continuum analysis of the cohesive zone is matched to a bending problem for the detached backing at $s = s_b$. The peel shape is obtained by integrating the governing equation in the detached zone $s > s_b$, where the fibrils are broken. In this region, the surface traction vanishes, so that eq (27) reduces to

$$M = F\left[L - x\sin\psi + y\cos\psi - (l_0 + h)\cos\psi\right] \tag{35}$$

and eq (7) becomes

$$\kappa = -\frac{d^2x/dy^2}{\left[1 + (dx/dy)^2\right]^{3/2}} = \frac{F\left[L - x\sin\psi + y\cos\psi - (l_0 + h)\cos\psi\right]}{E^*I}. \tag{36}$$

Eq (36) can be rewritten in a concise form by transforming the coordinate system O^-xy into the coordinate system $Ox'y'$ as shown in Figure 8. The new coordinates x' and y' are related to the coordinates x and y by

$$x' = -x\cos\psi - y\sin\psi + (l_0 + h)\sin\psi, \tag{37}$$

$$y' = x\sin\psi - y\cos\psi + (l_0 + h)\cos\psi. \tag{38}$$

F is now in the $-x'$ direction. Substitution of eqs (37) and (38) into eq (36) gives

$$-\frac{d^2x'/dy'^2}{\left[1 + (dx'/dy')^2\right]^{3/2}} = \frac{F\left(L - y'\right)}{E^*I}. \tag{39}$$

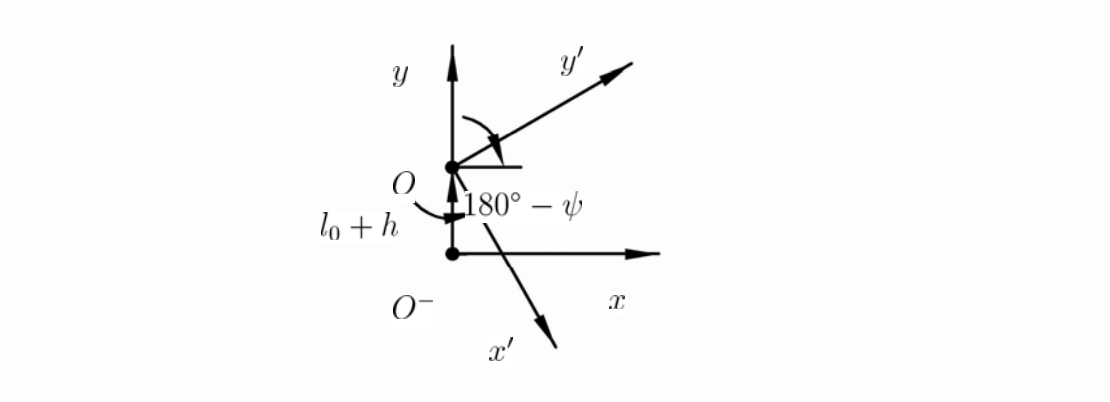


Figure 8. Transformation of the coordinate system.

Let $\theta' = \cot^{-1}(dx'/dy')$ be the inclination of the neutral axis measured from the x' axis. Eq (39) can be rewritten in terms of θ' and integrated exactly as

$$\int_{\theta'}^{180°} \sin\theta' d\theta' = \int_{y'}^{L} \frac{F(L-y')}{E^* I} dy'. \tag{40}$$

The relationship between θ' and y' can then be obtained as

$$1 + \cos\theta' = \frac{F(L-y')^2}{2E^* I}. \tag{41}$$

The rotation of the neutral axis are required to be continuous at $s = s_b$ so that $\theta'_b < \theta' < 180°$, where θ'_b is the value of θ' at $s = s_b$. It is obvious that $y' = L$ when $\theta' = 180°$. Note that $\theta = \tan^{-1}(dy/dx)$ is the inclination of the neutral axis in the coordinate system $O^- xy$ so that θ' is related to θ by

$$\theta' = \theta + 180° - \psi. \tag{42}$$

The peel shape in the detached zone in $O^- xy$ can then be obtained as

$$1 + \cos(\theta + 180° - \psi) = \frac{F[L - x\sin\psi + y\cos\psi - (l_0 + h)\cos\psi]^2}{2E^* I}, \tag{43}$$

where $\theta_b < \theta < \psi$. L can be expressed in terms of x_b, y_b and θ_b as

$$L = x_b \sin\psi - y_b \cos\psi + (l_0 + h)\cos\psi + \sqrt{2E^* I\,[1 + \cos(\theta_b + 180° - \psi)]/F}. \quad (44)$$

Substitution of eq (44) into eq (34) gives

$$M_b = \sqrt{2FE^* I\,[1 + \cos(\theta_b + 180° - \psi)]}. \quad (45)$$

Eq (29) and boundary conditions (30) – (34) provide a partial description of the peel mechanics. A fibril failure criterion is needed to determine the extent of the cohesive zone.

2.4 Failure Criterion

Following the work of Lin et al. [8], the fibrils are assumed to break at a critical stretch ratio. Let λ_b be the critical stretch ratio of the fibril just before cohesive failure. λ_b can be written as

$$\lambda_b = \frac{l_b}{l_0}. \quad (46)$$

Once λ_b is specified, s_b is uniquely determined. It is observed that the critical stretch ratio does vary with the peel rate, but the dependence on the peel rate is weak [12]. This phenomenon is related to the cross-linking of macromolecules of PSAs. The internal motion of cross-linked macromolecules is a relative slow process, that is, the molecular rearrangement requires a certain time [1]. This time is much longer than the duration of the external load in a peel test. Therefore, the molecular rearrangement will not

contribute significantly to the critical stretch ratio within the time scale relevant in peel tests at typical peel rates.

2.5 Dimensional Analysis

Dimensional analysis can be performed to reduce the number of material parameters. Following the work of Lin et al. [8], s_b and $E^* s_b/2$ are employed as primary parameters. Let these two parameters normalize lengths and forces in eqs (23), (29), (30) – (34), respectively. These equations can then be rewritten in terms of the following dimensionless parameters:

$$\psi, \quad \bar{l}_0 = \frac{l_0}{s_b}, \quad \bar{l}_b = \frac{l_b}{s_b}, \quad \bar{h} = \frac{h}{s_b}, \quad \bar{R} = \frac{R}{s_b}, \tag{47}$$

where ψ specifies the peel angle; $\bar{l}_0$ and $\bar{l}_b$ account for the viscoelastic behavior of the adhesive; $\bar{h}$ characterizes the flexibility and dimensions of the backing; $\bar{R}$ characterizes the geometry of the substrate surface. This independent dimensionless group, together with the peel rate, characterizes the peel test. Dimensional analysis implies the solution for the dimensionless peel force $\bar{F} = 2F/E^* s_b$ must have the general form:

$$\bar{F} = \bar{F}\left(\psi,\ \bar{l}_0,\ \bar{l}_b,\ \bar{h},\ \bar{R},\right). \tag{48}$$

Once the dimensionless group is specified, the dimensionless parameters $\bar{\sigma}$, $\bar{F}$, $\bar{L}$ and the dimensionless peel shape $(\bar{x},\ \bar{y})$ are completely determined.

It should be pointed out that the solutions for different sets of dimensionless groups only provide combinations of the dimensionless parameters with which steady

state conditions are achieved. The dimensionless parameters can be related to the dimensional parameters by solving for s_b in the solutions. Specifically, each solution gives a dimensionless peel rate $\bar{V} = V\eta/E^* s_b$, and the dependence of the peel rate on the peel angle and the critical stretch ratio gives the corresponding dimensional peel rate. s_b can then be calculated for given E^* and η.

3. NUMERICAL METHOD

3.1 Discretized Governing Equations

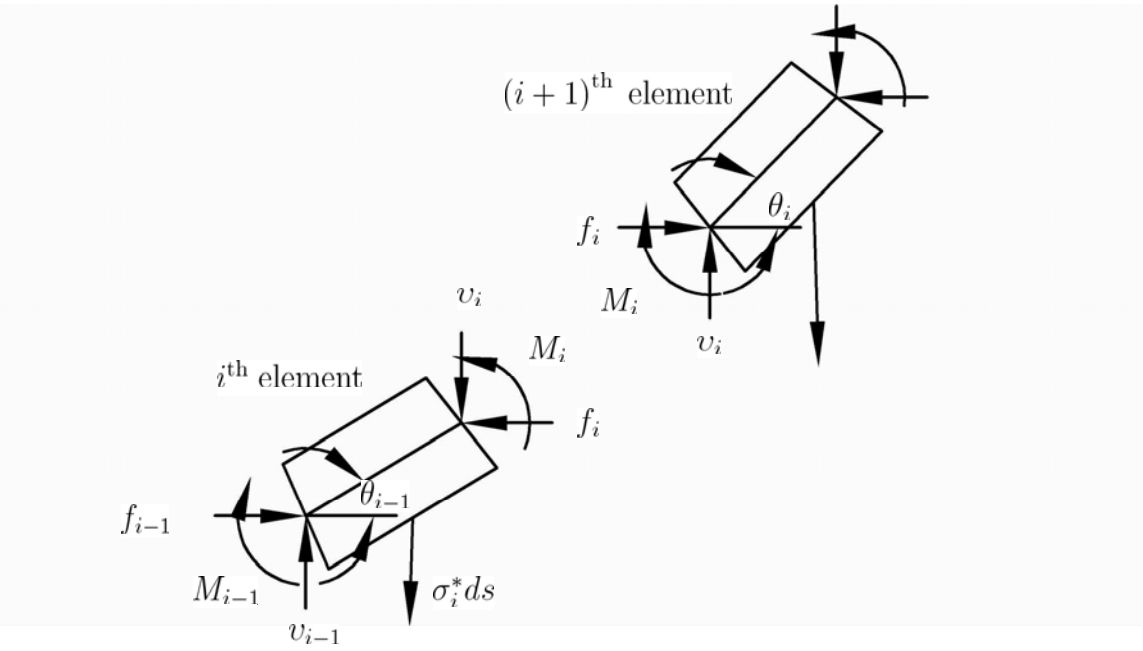


Figure 9. Free body diagram of two neighboring elements of the backing.

Eq (29) provides the basis for the analytical model. However, exact solutions for this problem cannot be found explicitly. To solve the problem numerically, a series of alternate but completely equivalent formulations are proposed. The backing in the cohesive zone is subdivided into N differential elements with the same length ds. Two neighboring elements (the i^{th} and the $(i+1)^{\text{th}}$ element) are illustrated schematically in Figure 5. The horizontal and vertical forces acting on the i^{th} element are f_{i-1}, f_i, v_{i-1}

and v_i. The moments about the neutral axis of the i^{th} element are M_{i-1} and M_i. The surface traction produced by the fibrils is denoted by $\sigma_i^* ds$. $\sigma_i^* ds$ is approximated to act on the midpoint of the lower surface of the i^{th} element. The coordinate of the midpoint can be given as

$$x_i' = x_{i-1} + 0.5ds\cos\theta_{i-1} + h\sin\theta_{i-1}, \tag{49}$$

$$y_i' = y_{i-1} + 0.5ds\sin\theta_{i-1} - h\cos\theta_{i-1}, \tag{50}$$

where $(x_{i-1},\ y_{i-1})$ is the coordinate of the $(i-1)^{\text{th}}$ node; θ_{i-1} is the inclination of the i^{th} element. Let l_i be the length of the fibril that occupies the i^{th} element. l_i is given by

$$l_i = \sqrt{\left(R\sin\varphi_i - x_i'\right)^2 + \left[-R\left(1-\cos\varphi_i\right) - y_i'\right]^2}. \tag{51}$$

where $\varphi_i = (i-0.5)\,ds/R$. Let α_i be the inclination of the fibril measured from the x axis. α_i can be represented by

$$\cos\alpha_i = \frac{R\sin\varphi_i - x_i'}{\sqrt{\left(R\sin\varphi_i - x_i'\right)^2 + \left[-R\left(1-\cos\varphi_i\right) - y_i'\right]^2}}, \tag{52}$$

$$\sin\alpha_i = \frac{-R\left(1-\cos\varphi_i\right) - y_i'}{\sqrt{\left(R\sin\varphi_i - x_i'\right)^2 + \left[-R\left(1-\cos\varphi_i\right) - y_i'\right]^2}}. \tag{53}$$

The force balance for the i^{th} element requires

$$f_{i-1} - f_i + \sigma_i ds\cos\alpha_i = 0, \tag{54}$$

$$v_{i-1} - v_i + \sigma_i ds\sin\alpha_i = 0. \tag{55}$$

The moment balance about $(x_{i-1},\ y_{i-1})$ requires

$$M_{i-1} - M_i - f_i ds\sin\theta_{i-1} + v_i ds\cos\theta_{i-1} + \sigma_i ds\cos\alpha_i\left(0.5ds\sin\theta_{i-1} - h\cos\theta_{i-1}\right)$$
$$-\sigma_i ds\sin\alpha_i\left(0.5ds\cos\theta_{i-1} + h\sin\theta_{i-1}\right) = 0. \tag{56}$$

An alternate formulation of eq (7) is

$$\theta_i - \theta_{i-1} = \frac{M_{i-1} + M_i}{2E^*I} ds. \tag{57}$$

The traction per unit length of the element can be obtained by rewriting eq (23):

$$\sigma_i^* = \frac{\eta l_0 V^*}{l_i^2} \left[\cos\left(\alpha_i + \varphi_i\right) - \left(1 + h\kappa_i\right) \cos\left(\alpha_i - \theta_{i-1}\right) \right], \tag{58}$$

where

$$\kappa_i = \frac{M_{i-1}}{E^*I}. \tag{59}$$

The boundary conditions at O are

$$\theta_0 = 0, \tag{60}$$

$$M_0 = 0. \tag{61}$$

The supplemental boundary conditions at the end of the cohesive zone are

$$f_N = -F \cos\psi, \tag{62}$$

$$v_N = -F \sin\psi, \tag{63}$$

$$M_N = F \left[L - x_N \sin\psi + y_N \cos\psi - \left(l_0 + h\right) \cos\psi \right], \tag{64}$$

$$\lambda_b = \frac{l_N}{l_0}. \tag{65}$$

The length of the peel arm can be obtained by rewriting eq (44):

$$L = x_N \sin\psi - y_N \cos\psi + \left(l_0 + h\right) \cos\psi + \sqrt{2E^*I \left[1 + \cos\left(\theta_N + 180^\circ - \psi\right)\right]/F}. \tag{66}$$

3.2 Dimensionless Discretized Governing Equations

Now consider the dimensionless versions of eqs (51) – (66). Let primary parameters s_b and $E^* s_b/2$ normalize lengths and forces in eqs (51) – (66), respectively. The dimensionless parameters can then be obtained as

$$\bar{x}_i = \frac{x_i}{s_b}, \quad \bar{y}_i = \frac{y_i}{s_b}, \quad \bar{l}_0 = \frac{l_0}{s_b}, \quad \bar{h} = \frac{h}{s_b}, \quad \bar{R} = \frac{R}{s_b},$$
$$\bar{f}_i = \frac{2f_i}{E^* s_b}, \quad \bar{v}_i = \frac{2v_i}{E^* s_b}, \quad \bar{M}_i = \frac{M_i s_b}{E^* I}, \quad \bar{\sigma}_i = \frac{\sigma_i}{E^*}, \quad \bar{V}^* = \frac{V^* \eta}{E^* s_b}. \tag{67}$$

Substitution of these dimensionless parameters in to eqs (51) – (59) gives the dimensionless versions of these equations:

$$\bar{l}_i = \sqrt{\left(\bar{R}\sin\varphi_i - \bar{x}_i'\right)^2 + \left[-\bar{R}\left(1-\cos\varphi_i\right) - \bar{y}_i'\right]^2}, \tag{68}$$

$$\cos\alpha_i = \frac{\bar{R}\sin\varphi_i - \bar{x}_i'}{\sqrt{\left(\bar{R}\sin\varphi_i - \bar{x}_i'\right)^2 + \left[-\bar{R}\left(1-\cos\varphi_i\right) - \bar{y}_i'\right]^2}}, \tag{69}$$

$$\sin\alpha_i = \frac{-\bar{R}\left(1-\cos\varphi_i\right) - \bar{y}_i'}{\sqrt{\left(\bar{R}\sin\varphi_i - \bar{x}_i'\right)^2 + \left[-\bar{R}\left(1-\cos\varphi_i\right) - \bar{y}_i'\right]^2}}, \tag{70}$$

$$\bar{f}_{i-1} - \bar{f}_i + 2\bar{\sigma}_i d\bar{s}\cos\alpha_i = 0, \tag{71}$$

$$\bar{v}_{i-1} - \bar{v}_i + 2\bar{\sigma}_i d\bar{s}\sin\alpha_i = 0, \tag{72}$$

$$\bar{M}_{i-1} - \bar{M}_i - \beta\bar{f}_i d\bar{s}\sin\theta_{i-1} + \beta\bar{v}_i d\bar{s}\cos\theta_{i-1} + 2\beta\bar{\sigma}_i d\bar{s}\cos\alpha_i\left(0.5d\bar{s}\sin\theta_{i-1} - \bar{h}\cos\theta_{i-1}\right)$$
$$-2\beta\bar{\sigma}_i d\bar{s}\sin\alpha_i\left(0.5d\bar{s}\cos\theta_{i-1} + \bar{h}\sin\theta_{i-1}\right) = 0, \tag{73}$$

$$\theta_i - \theta_{i-1} = \frac{\bar{M}_{i-1} + \bar{M}_i}{2} d\bar{s}, \tag{74}$$

$$\bar{\sigma}_i^* = \frac{\bar{l}_0 \bar{V}^*}{\bar{l}_i^2}\left[\cos\left(\alpha_i + \varphi_i\right) - \left(1 + \bar{h}\kappa_i\right)\cos\left(\alpha_i - \theta_{i-1}\right)\right], \tag{75}$$

where

$$\kappa_i = \bar{M}_{i-1}. \tag{76}$$

The dimensionless versions of eqs (60) – (65) can be similarly obtained as

$$\bar{f}_N = -\bar{F}\cos\psi, \tag{77}$$

$$\bar{v}_N = -\bar{F}\sin\psi, \tag{78}$$

$$\bar{M}_N = \beta\bar{F}_N\left[\bar{L} - \bar{x}_N\sin\psi + \bar{y}_N\cos\psi - \left(\bar{l}_0 + \bar{h}\right)\cos\psi\right], \tag{79}$$

$$\lambda_b = \frac{\bar{l}_N}{\bar{l}_0}, \tag{80}$$

where $\beta = s_b^3/2I$. The dimensionless version of eq (66) is

$$\bar{L} = \bar{x}_N\sin\psi - \bar{y}_N\cos\psi + \left(\bar{l}_0 + \bar{h}\right)\cos\psi + \sqrt{2\left[1 + \cos\left(\theta_N + 180^\circ - \psi\right)\right]/\bar{F}\beta}. \tag{81}$$

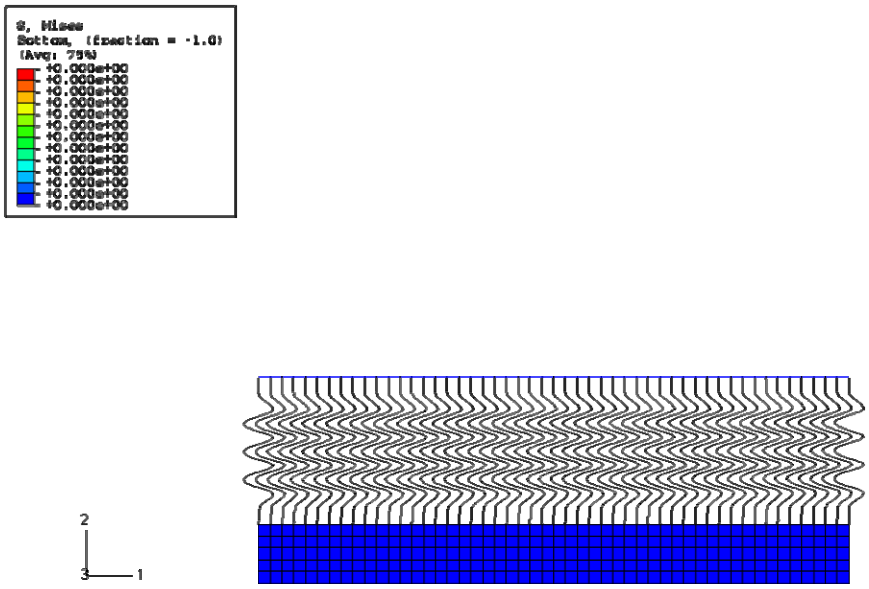
Solutions for the dimensionless group can then be obtained using the dimensionless versions of the governing equations and the boundary conditions.

Boundary conditions (60) – (65) imply that this problem is a two boundary value problem, where the dimensionless group can be treated as a set of eigenvalues of the governing equations. The shooting method is often employed to solve a two boundary value problem [23]. Specifically, $\bar{f}_0$, $\bar{v}_0$, $\bar{V}$ and $\bar{F}$ are initially guessed. Solutions for the problem can then be obtained with initial value methods. In general, $\bar{f}_b$, $\bar{v}_b$, θ_b and $\bar{l}_b$ obtained in this manner may deviate form the desired boundary values at $\bar{s} = 1$. The problem can then be solved by finding the adjustment of $\bar{f}_0$, $\bar{v}_0$, $\bar{V}$ and $\bar{F}$ that zeros this deviation. Details of the shooting method are given in Appendix A, while the C program developed to perform the shooting method is given in Appendix B.

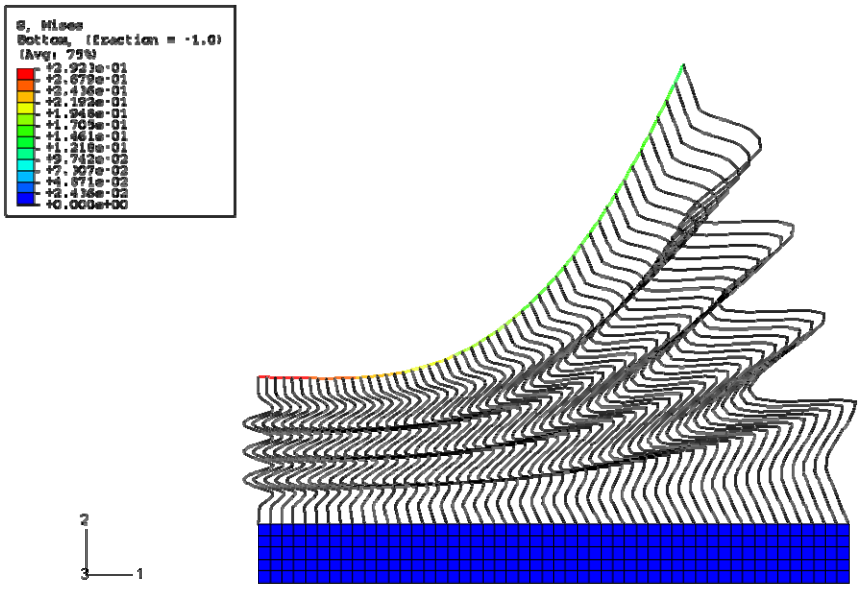
4. FINITE ELEMENT ANALYSIS

The commercial software package ABAQUS is used in this research. The finite element geometry consists of the backing, the substrate and the adhesive (see Figure 10(a)). The backing is discretized with the beam element ABAQUS type B21. A beam element is a one-dimensional line element in three-dimensional space or in the xy plane that has stiffness associated with deformation of the line [24]. These deformations consist of axial stretch, bending and torsion. The main advantage of beam elements is that they are geometrically simple and have few degrees of freedom. The length of the backing (and also the extent of the cohesive zone) is set to be 100, while the backing is discretized into 50 elements. The modulus E^* is set to be 1 so that the results can then be converted to dimensionless results easily. The substrate is modeled as flat and of high modulus and identical length compared with the backing. The two-dimensional quadratic element ABAQUS type CPS4R is employed to discretize the substrate. To model the adhesive accurately, the number of elements in the longitudinal direction of the substrate is set to be 50, which is identical of that of the backing.

The string assumption implies that the fibrils are subjected to uniaxial extension. The fibrils are hereby modeled as a set of strings joining corresponding pairs of nodes of the backing and the substrate, respectively. Spring elements are used to model actual physical springs as well as idealizations of axial or torsional components, while their behavior can be linear or nonlinear [24]. Nonlinear spring behavior can be defined by



(a)



(b)

Figure 10. Geometry of FEA model: (a) undeformed; (b) deformed.

giving pairs of force–relative displacement values. The spring element ABAQUS type SpringA is employed to characterize the viscoelastic response of each string. The SpringA element acts between two nodes, with its line of action being the line joining the two nodes, so that this line of action can rotate in large-displacement analysis.

As mentioned above, each fibril experiences identical strain history under steady state conditions, and the mechanical response of each fibril depends on its position (see eq (23)). Since the peel shape is independent of time with respect to the moving reference frame, the behavior of the fibrils can be characterized by a force–relative displacement relationship. The set of spring element consists of 51 elements. Since the backing and the substrate will be set to be fully constrained at the peel front, the spring element there can be neglected. The arc length that each spring element occupies ds are then approximated to be 2. The nonlinear force–relative displacement relationship of the spring elements can then be obtained by plotting $l - l_0$ versus $\sigma^* ds$. The elongation of a typical fibril $l - l_0$ and the traction per unit area of the backing produced by the fibril σ^* are given by numerical results from the mathematical method.

The backing is set to be fully constrained at the peel front. The substrate is modeled as rigid so that it is set to be fully constrained both at the peel front and on its upper surface. Then a displacement and rotation boundary condition is imposed to the end of the backing. The displacement and rotation of the backing are also given by numerical results from the mathematical method. The peel shape and the traction distribution on the substrate surface can then be obtained (see Figure 10(b)). The

agreement between these results and those from the mathematical method will be evaluated. The input file of the FEA model is given in Appendix C.

One issue of the model is that the beam element is a one-dimensional line element. As a result, the spring elements are joining the neutral axis of the backing and the substrate surface, whereas the fibrils are in fact joining the lower surface of the backing and the substrate surface. Such an issue may cause the elongation and orientation of the spring elements, the magnitude of surface traction acting on the backing, and thus the moment about the neutral axis of the backing produced by the fibrils to deviate from the actual values. The effect of this issue on the peel shape and the traction distribution will also be discussed.

5. RESULTS AND DISCUSSION

Numerical results are obtained for peel tests in which cohesive failure prevails. Following experiments and observations by Christensen et al. [11-13], the dimensionless group is set as follows:

$$\bar{l}_0 = 0.2, \quad 0.4 \leq \bar{l}_b \leq 0.8, \quad 0.03 \leq \bar{h} \leq 0.05,$$
$$5.0 \leq \bar{R} < \infty, \quad 90^\circ \leq \psi \leq 180^\circ. \tag{82}$$

Since the critical stretch ratio depends on the peel rate and the peel angle, the extent of the cohesive zone also varies with these two parameters. Here, $\bar{s}_b$ is maintained to be constant in most cases in the interest of analyzing the influences of various parameters.

Figure 11 compares the dimensionless peel shapes from the mathematical method and those from FEA for several values of λ_b, while Figure 12 depicts the corresponding traction distributions from the mathematical method and FEA. As can be seen in Figure 11, the peel curves from the mathematical method and FEA agree with each other well for $\lambda_b = 3$. For $\lambda_b = 4$, the deviation of the peel curve from FEA becomes noticeable. Correspondingly, the traction distributions from the mathematical method and FEA agree with each other well for $\lambda_b = 3$ in Figure 12. The traction peak from FEA moves apart from the peel front for $\lambda_b = 4$. The agreement between results from the mathematical method and those from FEA demonstrates that both sets of results are valid for low values of λ_b, whereas the difference between two sets of results implies that at least one set of results is not valid for high values of λ_b.

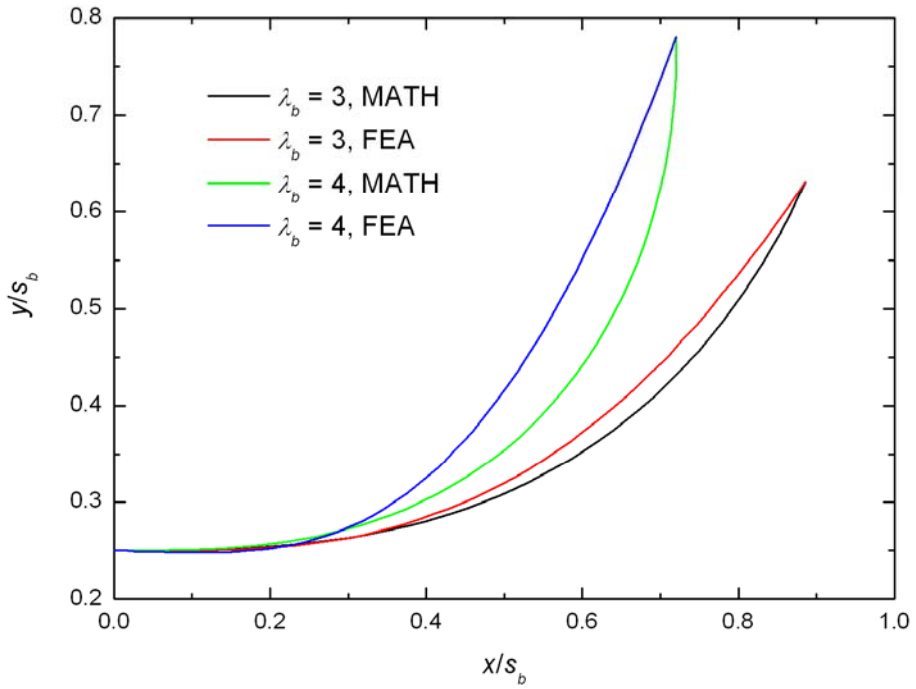


Figure 11. Dimensionless peel shapes from the mathematical method and FEA. The dimensionless group is taken to be $\psi = 180°$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

The difference between two sets of results can be quantified by the shift of the traction peak and the difference between values of tractions at $\bar{s} = 1$ in Figure 12. Note that a displacement and rotation boundary condition is imposed to the end of the backing. For identical λ_b, the last intact fibrils in both models (the mathematical method and FEA) are of identical elongation. Also note that the behaviors of the fibrils for both methods are characterized by identical force–relative displacement relationship. The two values of tractions at $\bar{s} = 1$ for both methods should hereby be the same. As can be seen in Figure 12, the difference between the two values, however, is noticeable. This difference

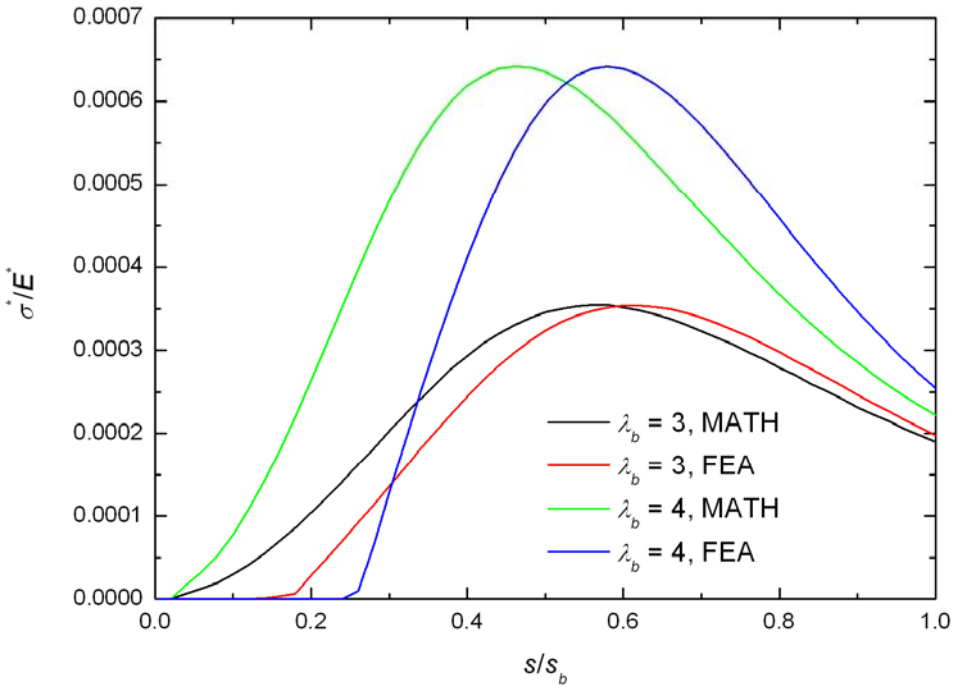


Figure 12. Dimensionless traction distributions from the mathematical method and FEA. The dimensionless group is taken to be $\psi = 180°$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

results from that the peel geometry is not modeled accurately from FEA. As mentioned above, the spring elements are modeled as joining the neutral axis of the backing and the substrate surface. The elongation and orientation of the spring elements, the magnitude of surface traction acting on the backing, and thus the moment about the neutral axis of the backing produced by the fibrils to deviate from the true values. As λ_b increases, $\bar{\sigma}^*$ increases correspondingly, while the backing becomes more deformed. The calculated moment produced by the fibrils may hereby suffer a high degree of error. As a result, the peel curve from FEA deviates from the actual values, while the traction peak shifts

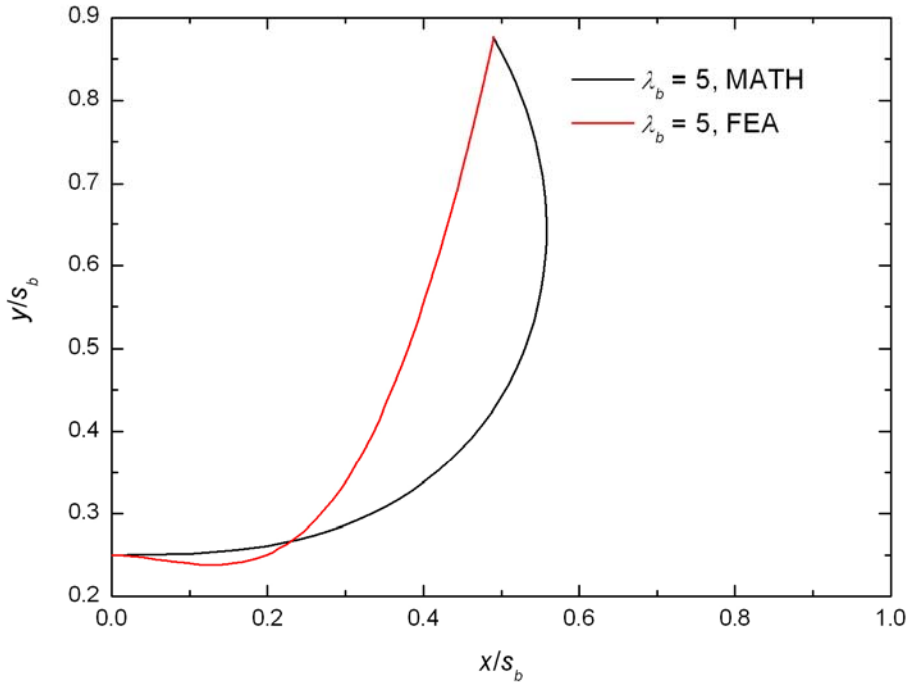


Figure 13. Dimensionless peel shapes from the mathematical method and FEA for high dimensionless critical fibril length. The dimensionless group is taken to be $\psi = 180°$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

correspondingly. Nevertheless, both the mathematical method and FEA provide a solid framework to characterize the mechanics of the peel test for low values of λ_b.

The dimensionless critical fibril length $\bar{l}_b = l_b/s_b$ is another factor that affects the validity of results from not only FEA but also the mathematical method. As can be seen in Figure 13, the peel curve from FEA is no long a monotonically increasing curve for $\lambda_b = 5$ and $\bar{l}_b = 1$, that is, buckling occurs in the backing. Note that the extent of the cohesive is determined by eq (30), which requires that the inclination of the backing is

zero at the peel front, and eq (46), which determines the location of the last intact fibril. As buckling occurs in the backing, there is one point other than the peel front where the inclination of the backing is also zero. This implies that the cohesive zone is resized. Furthermore, the adhesive at that point is subject to compression. This violates the string assumption. Therefore, the present model is unable to analyze the mechanics of peel tests in the presence of buckling in the backing. The reason for the occurrence of buckling here is that the issue of the FEA model causes the system to suffer a high degree of error and thus to be unstable for extremely high values of λ_b.

Figure 13 also shows conditions in which the mathematical method might be unable to model the peel test. The backing can deform to such an extent that the fibrils overlap each other in the vicinity of $\bar{s}_b = 1$. In this case, there will be interaction between neighboring fibrils. This violates the assumption that the cohesive zone is a continuum fibrillated region. Numerical results from the mathematical method are hereby no long the solution for the physical problem. To avoid the overlapping of the fibrils, the dimensionless critical fibril length $\bar{l}_b$ should be limited to a certain extent. The effective range of the present model is suggested to be $\bar{l}_b \leq 0.8$.

Figure 14 shows the dimensionless traction distribution on the substrate surface. The dimensionless traction distribution, which in this case corresponds to λ_b, shows that the dimensionless traction $\bar{\sigma}^*$ increases with increasing λ_b. $\bar{\sigma}^*$ attains a peak value when $\bar{s} = 0.76$ for $\lambda_b = 2$. As λ_b increases, the tensile traction peak becomes more prominent and move towards to the peel front. This result is mostly in good agreement with

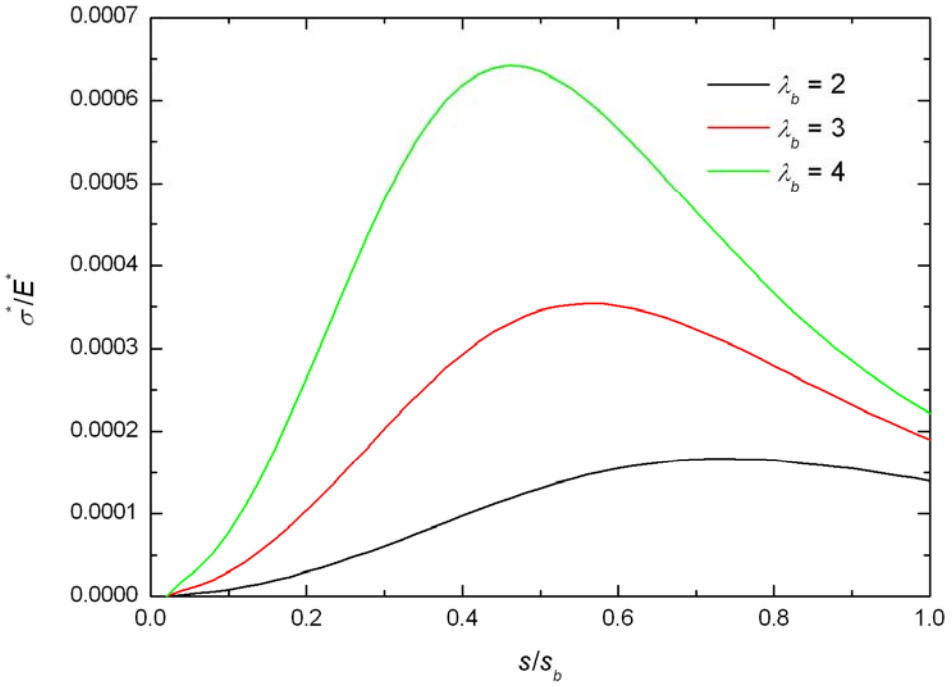


Figure 14. Dimensionless traction distributions for different critical stretch ratios. The dimensionless group is taken to be $\psi = 180°$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

previous works [5-7]. However, two distinct differences are observed. First, a prominent compressive traction peak was observed by Kaelble and Reylek [6], as well as by Niesiolowski and Aubrey [7], but is absent in the present result. The reason for this discrepancy is that the length of the peel arm is coupled with the peel force in the present work, whereas they are independent of each other in the previous works [5-7]. The inclination of the backing at the peel front is generally nonzero, that is, the backing resists against being bent during the peel test [12]. As a result, a compressive traction peak appears ahead of the peel front due to the resistance. However, Eq (30) requires

that the inclination be zero, and also that the length of the peel arm depend on the peel force. Since the resistance is now acting parallel to the tangential line of the substrate surface at the peel front, no compressive traction peak will be produced. Suppose that eq (30) is not satisfied so that the length of the peel arm and the peel force are independent on each other. The peel shape ahead of the peel front can then be approximately sinusoidal. The sinusoidal peel shape may produce a compressive traction peak and a secondary tensile traction peak [6, 7]. It should be pointed out that eq (30) is automatically satisfied in the case of a $90°$ peel test. Therefore, the compressive traction peak was also absent in the work of Christensen. Second, no tensile traction peak was observed by Christensen et al. [12]. Christensen suggested that the absent of the tensile traction peak was due to the fact that the process of cavitation of the adhesive did not occur. This statement, however, is not generally the case. Adhesive failure was observed in a series of $90°$ peel tests by Christensen, that is, the adhesive debonded from the substrate before the traction attained a peak value. Furthermore, the backing chosen by Christensen was so flexible that the resistance of the backing against the adhesive could be neglected during the test.

Figure 15 depicts the influence of $\bar{V}$ on the dimensionless traction distribution. Here, $\bar{\sigma}^*$ and $\bar{F}$ are normalized by E^* and $E^* s_0/2$, respectively, where s_0 is a referential extent of the cohesive zone. As can be seen, $\bar{\sigma}^*$ increases with increasing $\bar{V}$, while the traction peak becomes more prominent for high values of $\bar{V}$. However, the extent of the cohesive zone decreases as $\bar{V}$ increases. This can be understood by considering a peel test at a high peel rate, where the backing is bent to such an extent that the critical stretch

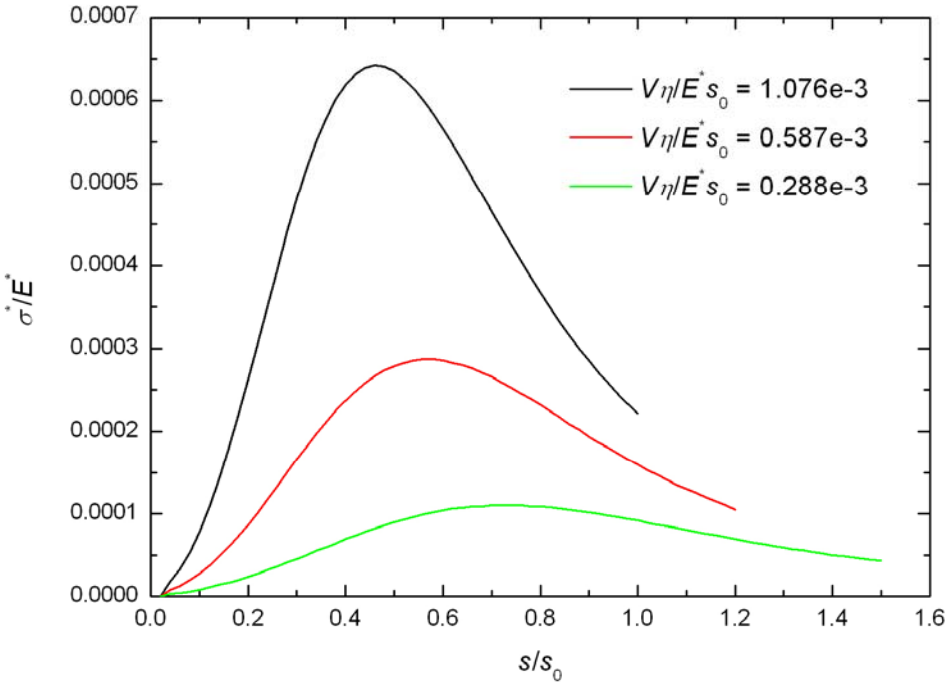


Figure 15. Dimensionless traction distributions for different peel rates. The dimensionless group is taken to be $\lambda_b = 4$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

ratio is attained and the extent of the cohesive is relatively small. Therefore, the extent of the cohesive zone is determined by both the dimensionless peel rate and the critical stretch ratio. However, there exists a lower bound of the extent of the cohesive zone for given peel geometry and nature of the adhesive. As mentioned above, the mathematical method is unable to model the peel test as the dimensionless critical fibril length exceeds its limitation. As a result, the lower bound of the extent of the cohesive zone varies for various critical stretch ratios.

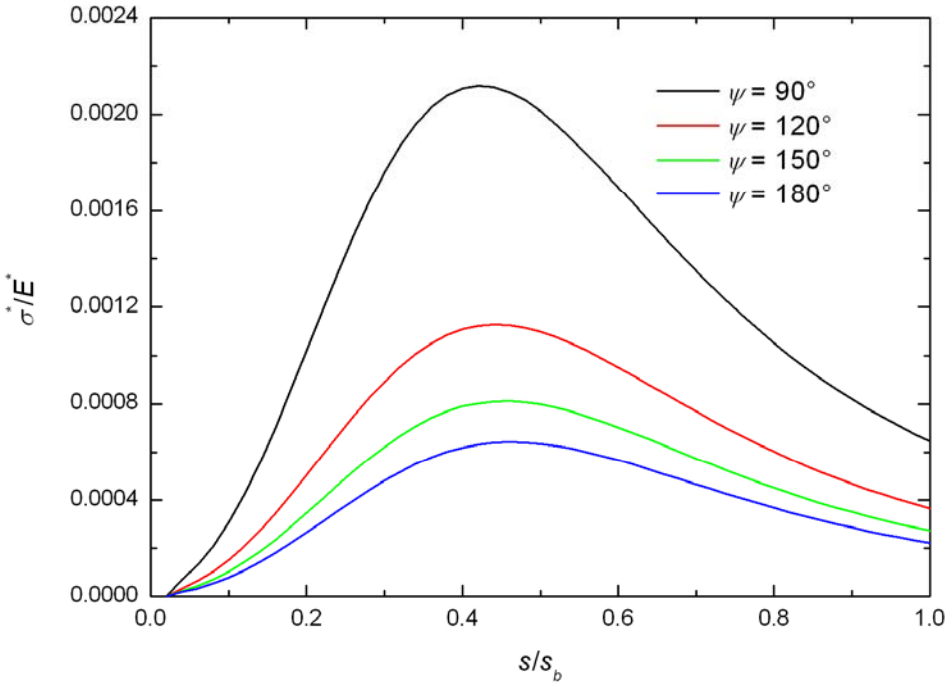


Figure 16. Dimensionless traction distributions for different peel angles. The dimensionless group is taken to be $\lambda_b = 4$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

The dependence of the dimensionless traction distribution on ψ is illustrated in Figure 16. As can be seen, $\bar{\sigma}^*$ decreases with increasing ψ. Note that the work expended during the peel test represents a combination of the strength of the adhesive bond and the work expended in the elastic deformation of the backing. The backing has to be bent to such an extent that the critical stretch ratio is attained at the end of the cohesive zone. This requirement is easier to be satisfied for high values of ψ. Therefore, high values of ψ causes low values of dimensionless speeds of the peel front propagation $\bar{V}^*$. Since eq

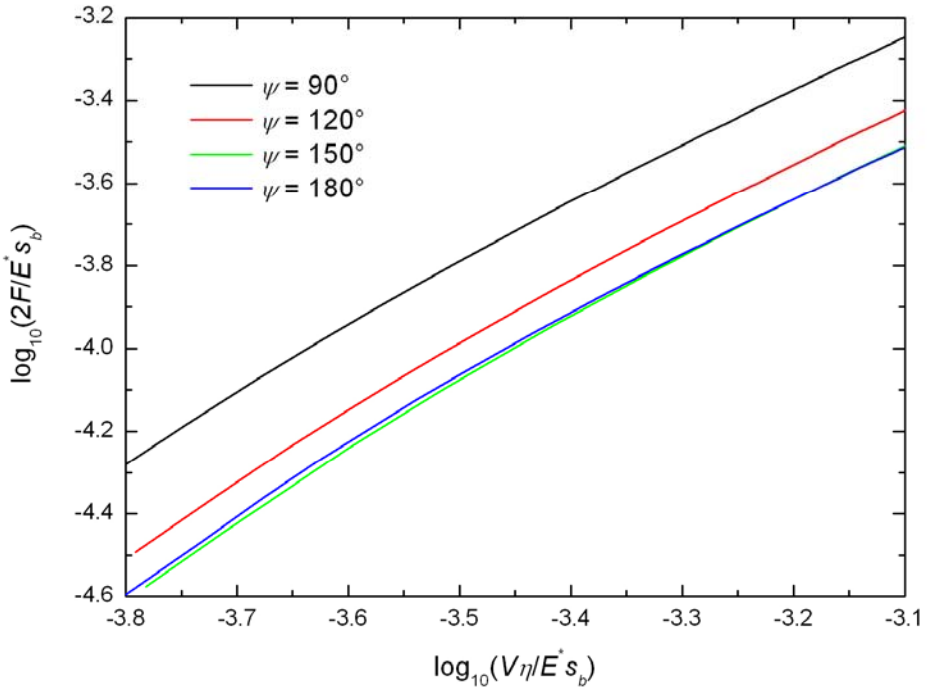


Figure 17. Dimensionless peel rate versus dimensionless peel force for different peel angles. The dimensionless group is taken to be $\bar{h} = 0.05$ and $\bar{R} = \infty$.

(23) implies that $\bar{\sigma}^*$ is proportional to $\bar{V}^*$, low values of $\bar{\sigma}^*$ are obtained for low values of $\bar{V}^*$ and thus high values of ψ. For extremely low values of ψ, numerical results cannot be obtained. Since the backing is assumed to undergo negligible deformation in extension, the critical stretch ratio cannot be attained in these cases.

Figure 17 plots $\log_{10} \bar{V}$ versus $\log_{10} \bar{F}$ for several values of ψ. As expected, $\bar{F}$ increases with increasing $\bar{V}$. For identical critical stretch ratio, low values of $\bar{V}$ and $\bar{F}$ are required for high values of ψ. The influence of the peel angle on the peel shape is weak for high values of ψ. As a result, the curves for $150°$ and $180°$ peel tests are quite

close to each other. The relationship between $\log_{10} \bar{F}$ and $\log_{10} \bar{V}$ is not strictly linear, whereas the peel rate is proportional to $F^{3/4}$ in the work of Lin et al. [8]. This discrepancy might due to the fact that Lin failed to model the peel geometry accurately. As mentioned above, the fibrils are joining the lower surface of the backing and the substrate surface. However, they were modeled as joining the neutral axis of the backing and the substrate surface in the work of Lin. The elongation and orientation of the spring elements, the magnitude of surface traction acting on the backing, and thus the moment about the neutral axis of the backing produced by the fibrils may hereby suffer a high degree of error in the presence of a thick backing.

The dependence of the dimensionless traction distribution on η and E^* is shown in Figure 18. The ratio of the viscosity of the adhesive to the modulus of the backing η/E^* (and thus the dimensionless peel rate $\bar{V}$) is assumed to remain constant in each curve. Here, $\bar{\sigma}^*$ and $\bar{F}$ are normalized by E_0 and $E_0 s_b/2$, respectively, where E_0 is a referential modulus. As can be seen, $\bar{F}$ increases proportionally as η and E^* increase. The trends of the curves are similar, whereas $\bar{\sigma}^*$ increases proportional to η and E^*. The peel force also depends on the nature of the adhesive, as well as the flexibility of the backing. High peel forces are required to bend stiff backings and rupture strong adhesive bonds. Stiff backings and strong adhesive bonds also produce high tractions on the substrate surface increase. As mentioned above, the measured peel force represents a combination of the strength of the adhesive bond and other work expended in the elastic deformation of the backing. For low values of η/E^*, the peel force depends on the

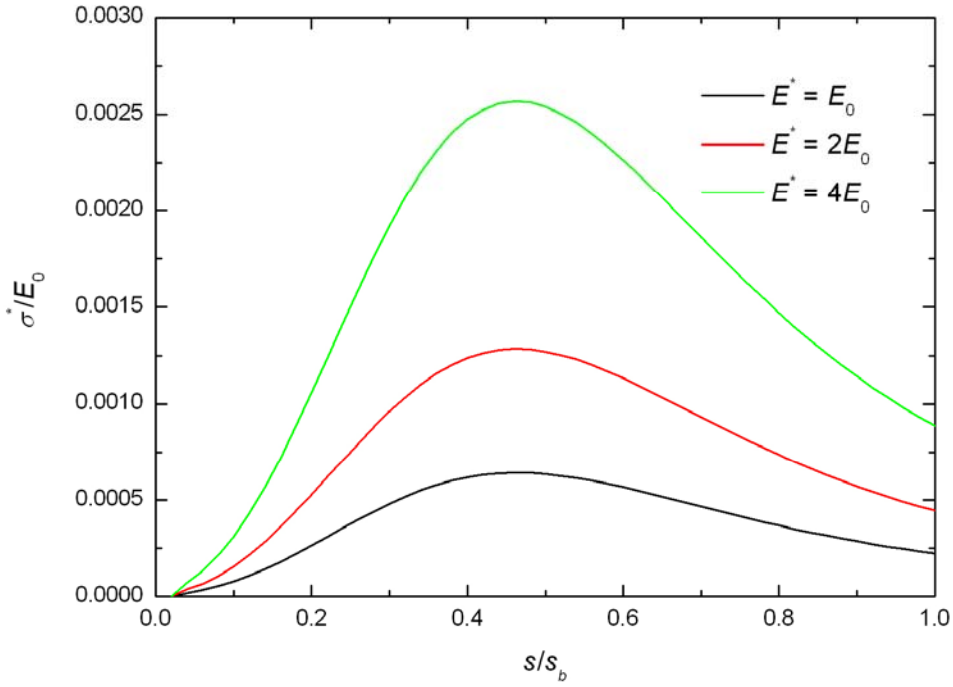


Figure 18. Dimensionless traction distributions for different viscosities of the adhesive and modulii of the backing. The dimensionless group is taken to be $\psi = 180°$, $\lambda_b = 4$, $\bar{h} = 0.05$ and $\bar{R} = \infty$.

mechanical properties of the backing significantly. To evaluate the contribution of the adhesive bond on the peel force, the elastic and plastic deformation of the backing needs to be analyzed in detail. For high values of η/E^*, the peel force can be approximated as the representation of the true strength of the adhesive bond.

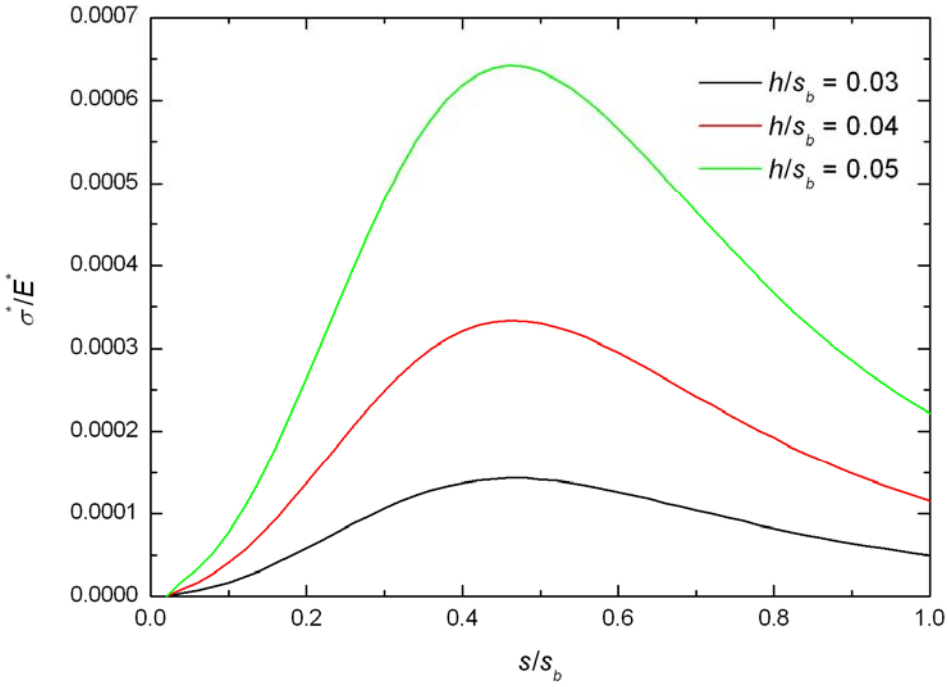


Figure 19. Dimensionless traction distributions for different backing thicknesses. The dimensionless group is taken to be $\psi = 180°$, $\lambda_b = 4$ and $\bar{R} = \infty$.

Figure 19 shows the influence of $\bar{h}$ on the dimensionless traction distribution. As mentioned above, the dimensional analysis only provides combinations of the dimensionless parameters with which steady state conditions are achieved. For high values of $\bar{h}$, high values $\bar{F}$ and $\bar{V}$ are required to bend the backing through ψ. Therefore, increasing $\bar{h}$ causes significant increase in $\bar{\sigma}^*$. As can be seen in eq (23), the term $h\kappa^+$, which results from the change in length of the lower surface of the backing, affects $\bar{\sigma}^*$ as well. Eq (23) reduces to

$$\sigma^* = \frac{\eta l_0 V^*}{l^2} \left[\cos\left(\alpha + \varphi\right) - \cos\left(\alpha - \theta\right)\right] \tag{83}$$

for negligible h or κ^+, whereas eq (23) becomes

$$\sigma^* = \frac{\eta l_0 V^*}{l^2} \left[\cos\left(\alpha + \varphi\right) - 2\cos\left(\alpha - \theta\right)\right] \tag{84}$$

as $h\kappa^+$ approaches to a limit of $1/2$. The case of $h\kappa^+ = 1/2$ corresponds to peel tests in which the backing is thick compared to the extent of the cohesive zone and bent to a large extent. Lin et al. [8] derived an expression that is equivalent to eq (83), whereas the backing thickness was assumed to be much greater than the adhesive thickness. It is obvious that results obtained using eq (83) will deviate from the exact values when $h\kappa^+$ is no longer negligible.

The influence of $\bar{R}$ on the dimensionless traction distribution is illustrated in Figure 20. The last intact fibril is extended to identical stretch ratio in each curve. Since the adhesive is much softer the substrate, the deformation of the adhesive strongly depends on the geometry of the substrate. For high values of $\bar{R}$, fibrils in the cohesive zone are subjected to larger deformation. Correspondingly, $\bar{\sigma}^*$ increases with increasing $\bar{R}$. For high values of $\bar{R}$, the dependence of $\bar{\sigma}^*$ on $\bar{R}$ becomes weak. The curves approach to the curve of $\bar{R} = \infty$. However, this does not mean that the substrate can always be treated as flat when the extent of the cohesive zone is much smaller than the radius of the substrate. Sometimes the roughness of the substrate surface needs to be considered. Now suppose that the substrate surface is approximated as a random rough surface with asperities of constant radius of curvature [2]. As the extent of the cohesive

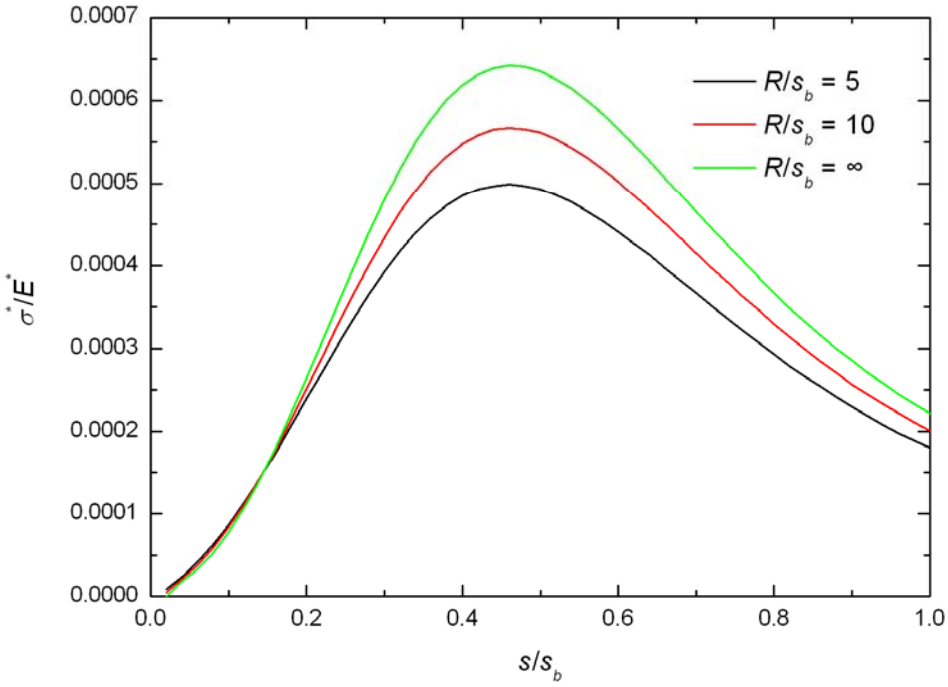


Figure 20. Dimensionless traction distributions for different dimensionless radii of curvature of the substrate surface. The dimensionless group is taken to be $\psi = 180°$, $\lambda_b = 4$ and $\bar{h} = 0.05$.

becomes comparable with the radius of these asperities, the traction distribution will be influenced the geometry of these asperities.

6. CONCLUSION

The model presented in this work provides a cohesive zone model to characterize the mechanics of the peel test. For given peel geometry, this model can be used to predict the dimensionless peel force in terms of the dimensionless peel rate and the peel angle. The dimensionless traction distribution is found to depend on such factors as the critical stretch ratio, the dimensionless peel rate, the peel angle, the mechanical properties of the adhesive and the backing, the geometries of the backing and the substrate. The dimensionless parameters can be easily related to dimensional experimental results for given peel geometry and material parameters. The results from FEA are compared with those from the mathematical method to evaluate the validity of the present model. The effective range of the present model is found to be related to the critical fibril length to the extent of the cohesive zone.

In the previous studies, the backing thickness was assumed to be much greater than the adhesive thickness, whereas the fibrils in the cohesive zone were modeled as joining the neutral axis of the backing and the substrate surface. In addition, the influence of the change in length of the lower surface on the strain rates of the fibrils was neglected. In the present work, the peel geometry is modeled accurately, while the influence of the deformation backing on the peel test is also evaluated. As a result, the logarithmic relationship between the dimensionless peel force and the dimensionless peel rate becomes not strictly linear. More realistic models for viscoelastic materials can be employed to characterize the behavior of the adhesive during peel test. The present

model, however, provides a solid framework to better understand the mechanics of peeling pressure sensitive adhesives.

REFERENCES

[1] Benedek I. Pressure-sensitive adhesives and applications. 2nd ed. Marcel Dekker Inc 2004.

[2] Creton C, Leibler L. How does tack depend on time of contact and contact pressure? J Polym Sci Pol Phys 1996;34:545–54.

[3] Hui CY, Lin YY, Baney JM. The mechanics of tack: viscoelastic contact on a rough surface. J Polym Sci Pol Phys 2000;38(11):1485–95.

[4] Williams JA, Kauzlarich JJ. Peeling shear and cleavage failure due to tape prestrain. J Adhesion 2004;80(5):433–58.

[5] Kaelble DH. Peel adhesion: influence of surface energies and adhesive rheology. J Adhesion 1969;1:102–23.

[6] Kaelble DH, Reylek RS. Peel adhesion: rate dependence of microfracture processes. J Adhesion 1969;1:124–35.

[7] Niesiolowski F, Aubrey DW. Stress distribution during peeling of adhesive Tapes. J Adhesion 1981;13:87–89.

[8] Lin YY, Hui CY, Wang YC. Modeling the failure of an adhesive layer in a peel test. J Polym Sci Pol Phys 2002;40(19):2277–91.

[9] Gent AN, Petrich RP. Adhesion of viscoelastic materials to rigid substrates. Proc R Soc A 1969;310(1502):433–48.

[10] Gupta RK. Comments on “prediction of peel adhesion using extensional rheometry” by Connelly, Parsons, and Pearson. J Rheol1983;27(2):171–75.

[11] Christensen SF, Flint SC. Practical criterion for rheological modeling of the peeling of pressure sensitive adhesives. J Adhesion 2000;72(2):177–207.

[12] Christensen SF, Everland H, Hassager O, Almdal K. Observations of peeling of a polyisobutylene-based pressure-sensitive adhesive. Int J Adhes Adhes 1998;18(2):131–37.

[13] Christensen SF, McKinley GH. Rheological modelling of the peeling of pressure-sensitive adhesives and other elastomers. Int J Adhes Adhes 1998;18(5):333–43.

[14] Williams JA, Kauzlarich JJ. The influence of peel angle on the mechanics of peeling flexible adherends with arbitrary load-extension characteristics. Tribol Int 2005;38(11-12 SPEC ISS):951–58.

[15] Wei Y, Hutchinson JW. Interface strength, work of adhesion and plasticity in the peel test. Int J Fracture 1998;93(1-4): 315–33.

[16] Kinloch AJ, Lau CC, Williams JG. The peeling of flexible laminates. Int J Fracture 1994;66(1):45–70.

[17] Kim J, Kim KS, Kim YH. Mechanical effects in peel adhesion test. J Adhes Sci Technol 1989;3(3):175–87.

[18] Kaelble DH. Theory and analysis of peel adhesion: bond stresses and distributions. Trans Soc Rheol 1960;4:45–73.

[19] Kaelble DH. Theory and analysis of peel adhesion: mechanisms and mechanics. Trans Soc Rheol 1959;3:161–80.

[20] Makati AC. Development of waterborne laminating adhesive systems. Tappi J 1988;71(6):145–50.

[21] Hui CY, Baney JM, Kramer EJ. Contact mechanics and adhesion of viscoelastic spheres. Langmuir 1998;14(22):6570–78.

[22] Wineman AS, Rajagopal KR. Mechanical response of polymers: an introduction. Cambridge Univ Pr 2000.

[23] Press WH, Teukolsky SA, Vetterling WT. Numerical recipes in C: the art of scientific computing. 2nd ed. Cambridge Univ Pr 1992

[24] ABAQUS user's manual. © ABAQUS Inc 2004;http://sc.tamu.edu:2080/v6.5/.

APPENDIX A

THE SHOOTING METHOD

A typical two point boundary value problem is comprised of a set of N coupled first-order ordinary differential equations, n_1 boundary conditions at the starting point $x = x_1$, and a remaining set of $n_2 = N - n_1$ boundary conditions at the final point $x = x_2$. The differential equations can be represented as

$$\frac{dy_i(x)}{dx} = g_i(x,\ y_1,\ y_2,\ \ldots,\ y_N), \quad i = 1,\ 2,\ \ldots,\ N. \tag{85}$$

The solution for the differential equations is supposed to satisfy

$$B_{1j}(x_1,\ y_1,\ y_2,\ \ldots,\ y_N) = 0, \quad j = 1,\ \ldots,\ n_1 \tag{86}$$

at $x = x_1$, while it is also supposed to satisfy

$$B_{2k}(x_2,\ y_1,\ y_2,\ \ldots,\ y_N) = 0, \quad k = 1,\ \ldots,\ n_2 \tag{87}$$

at $x = x_2$.

There are N starting values y_i to be specified at the starting point $x = x_1$, while the starting values are subject to n_1 conditions. Therefore, there are $n_2 = N - n_1$ freely specifiable starting values. Suppose that these freely specifiable values are components of a vector $\mathbf{V}$, of dimension n_2. The boundary condition (86) can then be rewritten as

$$y_i(x_1) = y_i(x_1;\ V_1,\ \ldots,\ V_{n_2}), \quad i = 1,\ \ldots,\ N. \tag{88}$$

Below, the function that implements (88) will be called load.

As $\mathbf{y}(x_1)$ has been generated for given initial guesses for $\mathbf{V}$, $\mathbf{y}(x_2)$ can then be obtained by integrating the ODEs to $x = x_2$ as an initial value problem. Now define a

discrepancy vector $\mathbf{F}$, also of dimension n_2, whose components indicates how far the freely specifiable values are from satisfying the n_2 boundary conditions at $x = x_2$. The boundary condition (87) can then be rewritten as

$$F_k = B_{2k}\left(x_2,\ \mathbf{y}\right), \quad k = 1,\ \ldots,\ n_2. \tag{89}$$

All components of $\mathbf{F}$ are equal to zero if and only if the boundary conditions at $x = x_2$ are satisfied. Therefore, finding a vector value of $\mathbf{V}$ that zeros the vector value of $\mathbf{F}$ will solve the problem. The initial guesses for $\mathbf{V}$ can be adjusted by

$$\mathbf{V}^{\text{new}} = \mathbf{V}^{\text{old}} + \delta\mathbf{V}. \tag{90}$$

The correction $\delta\mathbf{V}$ is obtained by solving the set of n_2 linear equations

$$\mathbf{J} \cdot \delta\mathbf{V} = -\mathbf{F}, \tag{91}$$

The Jacobian matrix **J** has components as follows:

$$J_{ij} = \frac{\partial F_i}{\partial V_j}. \tag{92}$$

Since these partial derivatives generally cannot be computed analytically, an alternative form of eq (92) is given as follows:

$$\frac{\partial F_i}{\partial V_j} \approx \frac{F_i\left(V_1,\ \ldots,\ V_j + \Delta V_j,\ \ldots\right) - F_i\left(V_1,\ \ldots,\ V_j\ \ldots\right)}{\Delta V_j}. \tag{93}$$

The solution procedure is completed when $\delta\mathbf{V} = 0$.

APPENDIX B

C PROGRAM

The C programs include three files: nutril.h, nutril.cpp and shoot.cpp. All the subroutines are discussed in detail in [23].

nrutil.h

```
#ifndef _NR_UTILS_H_
#define _NR_UTILS_H_

static float sqrarg;
#define SQR(a) ((sqrarg=(a)) == 0.0 ? 0.0 : sqrarg*sqrarg)

static double dsqrarg;
#define DSQR(a) ((dsqrarg=(a)) == 0.0 ? 0.0 : dsqrarg*dsqrarg)

static double dmaxarg1,dmaxarg2;
#define DMAX(a,b) (dmaxarg1=(a),dmaxarg2=(b),(dmaxarg1) > (dmaxarg2) ?\
        (dmaxarg1) : (dmaxarg2))

static double dminarg1,dminarg2;
#define DMIN(a,b) (dminarg1=(a),dminarg2=(b),(dminarg1) < (dminarg2) ?\
        (dminarg1) : (dminarg2))

static float maxarg1,maxarg2;
#define FMAX(a,b) (maxarg1=(a),maxarg2=(b),(maxarg1) > (maxarg2) ?\
        (maxarg1) : (maxarg2))

static float minarg1,minarg2;
#define FMIN(a,b) (minarg1=(a),minarg2=(b),(minarg1) < (minarg2) ?\
        (minarg1) : (minarg2))

static long lmaxarg1,lmaxarg2;
#define LMAX(a,b) (lmaxarg1=(a),lmaxarg2=(b),(lmaxarg1) > (lmaxarg2) ?\
        (lmaxarg1) : (lmaxarg2))

static long lminarg1,lminarg2;
```

```
#define LMIN(a,b) (lminarg1=(a),lminarg2=(b),(lminarg1) < (lminarg2) ?\
        (lminarg1) : (lminarg2))

static int imaxarg1,imaxarg2;
#define IMAX(a,b) (imaxarg1=(a),imaxarg2=(b),(imaxarg1) > (imaxarg2) ?\
        (imaxarg1) : (imaxarg2))

static int iminarg1,iminarg2;
#define IMIN(a,b) (iminarg1=(a),iminarg2=(b),(iminarg1) < (iminarg2) ?\
        (iminarg1) : (iminarg2))

#define SIGN(a,b) ((b) >= 0.0 ? fabs(a) : -fabs(a))

#if defined(__STDC__) || defined(ANSI) || defined(NRANSI) /* ANSI */

void nrerror(char error_text[]);
float *vector(long nl, long nh);
int *ivector(long nl, long nh);
float **matrix(long nrl, long nrh, long ncl, long nch);
void free_vector(float *v, long nl, long nh);
void free_ivector(int *v, long nl, long nh);
void free_matrix(float **m, long nrl, long nrh, long ncl, long nch);

#else /* ANSI */
/* traditional - K&R */

void nrerror();
float *vector();
float **matrix();
int *ivector();
void free_vector();
void free_ivector();
void free_matrix();

#endif /* ANSI */

#endif /* _NR_UTILS_H_ */
```

nutril.cpp

```
#include <stdio.h>
#include <stddef.h>
#include <stdlib.h>
#define NR_END 1
```

```
#define FREE_ARG char*

void nrerror(char error_text[])
/* Numerical Recipes standard error handler */
{
	fprintf(stderr,"Numerical Recipes run-time error...\n");
	fprintf(stderr,"%s\n",error_text);
	fprintf(stderr,"...now exiting to system...\n");
	exit(1);
}

double *vector(long nl, long nh)
/* allocate a double vector with subscript range v[nl..nh] */
{
	double *v;

	v=(double *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(double)));
	if (!v) nrerror("allocation failure in vector()");
	return v-nl+NR_END;
}

int *ivector(long nl, long nh)
/* allocate an int vector with subscript range v[nl..nh] */
{
	int *v;

	v=(int *)malloc((size_t) ((nh-nl+1+NR_END)*sizeof(int)));
	if (!v) nrerror("allocation failure in ivector()");
	return v-nl+NR_END;
}

double **matrix(long nrl, long nrh, long ncl, long nch)
/* allocate a double matrix with subscript range m[nrl..nrh][ncl..nch] */
{
	long i, nrow=nrh-nrl+1,ncol=nch-ncl+1;
	double **m;

	/* allocate pointers to rows */
	m=(double **) malloc((size_t)((nrow+NR_END)*sizeof(double*)));
	if (!m) nrerror("allocation failure 1 in matrix()");
	m += NR_END;
	m -= nrl;

	/* allocate rows and set pointers to them */
```

```
        m[nrl]=(double *) malloc((size_t)((nrow*ncol+NR_END)*sizeof(double)));
        if (!m[nrl]) nrerror("allocation failure 2 in matrix()");
        m[nrl] += NR_END;
        m[nrl] -= ncl;

        for(i=nrl+1;i<=nrh;i++) m[i]=m[i-1]+ncol;

        /* return pointer to array of pointers to rows */
        return m;
}

void free_vector(double *v, long nl, long nh)
/* free a double vector allocated with vector() */
{
        free((FREE_ARG) (v+nl-NR_END));
}

void free_ivector(int *v, long nl, long nh)
/* free an int vector allocated with ivector() */
{
        free((FREE_ARG) (v+nl-NR_END));
}

void free_matrix(double **m, long nrl, long nrh, long ncl, long nch)
/* free a double matrix allocated by matrix() */
{
        free((FREE_ARG) (m[nrl]+ncl-NR_END));
        free((FREE_ARG) (m+nrl-NR_END));
}
```

shoot.cpp

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include "nrutil.h"
#include "nrutil.cpp"
#define TINY 1.0e-20 //A small number.
#define ALF 1.0e-4 /*Ensures sufficient decrease in function value.*/
#define TOLX 1.0e-7 /*Convergence criterion on Δx.*/
#define MAXITS 500
#define TOLF 1.0e-6
#define TOLMIN 1.0e-8
#define EPS 1.0e-8
```

```
#define STPMX 100.0
#define PI 3.14159265
/*Here MAXITS is the maximum number of iterations; TOLF sets the convergence
criterion on function values; TOLMIN sets the criterion for deciding whether spurious
convergence to a minimum of fmin has occurred; TOLX is the convergence criterion on
δx; STPMX is the scaled maximum step length allowed in line searches.*/

int nn; //Global variables to communicate with fmin.
double *fvec;
void (*nrfuncv)(int n, double v[], double f[]);
#define FREERETURN {free_vector(fvec,1,n);free_vector(xold,1,n);\
        free_vector(p,1,n);free_vector(g,1,n);free_matrix(fjac,1,n,1,n);\
free_ivector(indx,1,n);return;}

FILE *fp;

void ludcmp(double **a, int n, int *indx, double *d)
/*Given a matrix a[1..n][1..n], this routine replaces it by the LU decomposition of a
rowwise permutation of itself. a and n are input. a is output; indx[1..n] is an output
vector that records the row permutation effected by the partial pivoting; d is output as ±1
depending on whether the number of row interchanges was even or odd, respectively.
This routine is used in combination with lubksb to solve linear equations or invert a
matrix.*/
{
        int i,imax,j,k;
        double big,dum,sum,temp;
        double *vv; //vv stores the imphicit scaling of each row.
        vv=vector(1,n);
        *d=1.0; //No row interchanges yet.
        for (i=1;i<=n;i++) { //Loop over rows to get the imphicit scaling information.
                big=0.0;
                for (j=1;j<=n;j++) {
                        if ((temp=fabs(a[i][j])) > big) big=temp;
                }
                if (big == 0.0) nrerror("Singular matrix in routine ludcmp");
                //No nonzero largest element.
                vv[i]=1.0/big; //Save the scaling.
        }
        for (j=1;j<=n;j++) { //This is the loop over columns of Crout's method.
                for (i=1;i<j;i++) {
                        sum=a[i][j];
                        for (k=1;k<i;k++) sum -= a[i][k]*a[k][j];
                        a[i][j]=sum;
                }
```

```
                big=0.0; //Initialize for the search for largest pivot element.
                for (i=j;i<=n;i++) {
                        sum=a[i][j];
                        for (k=1;k<j;k++)
                                sum -= a[i][k]*a[k][j];
                        a[i][j]=sum;
                        if ( (dum=vv[i]*fabs(sum)) >= big) {
                                //Is the figure of merit for the pivot better than the best so
far?
                                big=dum;
                                imax=i;
                        }
                }
                if (j != imax) { //Do we need to interchange rows?
                        for (k=1;k<=n;k++) { //Yes, do so...
                                dum=a[imax][k];
                                a[imax][k]=a[j][k];
                                a[j][k]=dum;
                        }
                        *d = -(*d); //...and change the parity of d.
                        vv[imax]=vv[j]; //Also interchange the scale factor.
                }
                indx[j]=imax;
                if (a[j][j] == 0.0) a[j][j]=TINY;
                //If the pivot element is zero the matrix is singular (at least to the
precision of
                // the algorithm). For some apphications on singular matrices, it is
desirable to
                // substitute TINY for zero.
                if (j != n) { //Now, finally, divide by the pivot element.
                        dum=1.0/(a[j][j]);
                        for (i=j+1;i<=n;i++) a[i][j] *= dum;
                }
        } //Go back for the next column in the reduction.
        free_vector(vv,1,n);
}

void lubksb(double **a, int n, int *indx, double b[])
/*Solves the set of n linear equations A·X = B. Here a[1..n][1..n] is input, not as the
matrix A but rather as its LU decomposition, determined by the routine ludcmp.
indx[1..n] is input as the permutation vector returned by ludcmp. b[1..n] is input as the
right-hand side vector B, and returns with the solution vector X. a, n, and indx are not
modified by this routine and can be left in place for successive calls with different right-
```

```
hand sides b. This routine takes into account the possibility that b will begin with many
zero elements, so it is efficient for use in matrix inversion.*/
{
        int i,ii=0,ip,j;
        double sum;
        for (i=1;i<=n;i++) { /*When ii is set to a positive value, it will become the
                                                index of the first nonvanisling element of b.
The
                                                only new wrinkle is to unscramble the
permutation
                                                as we go.*/
                ip=indx[i];
                sum=b[ip];
                b[ip]=b[i];
                if (ii)
                        for (j=ii;j<=i-1;j++) sum -= a[i][j]*b[j];
                        else if (sum) ii=i; /*A nonzero element was encountered, so from
now on
                                                                we will have to do the sums
in the loop above.*/
                        b[i]=sum;
        }
        for (i=n;i>=1;i--) {
                sum=b[i];
                for (j=i+1;j<=n;j++) sum -= a[i][j]*b[j];
                b[i]=sum/a[i][i]; /*Store a component of the solution vector X.*/
        } /*All done!*/
}

void fdjac(int n, double x[], double fvec[], double **df,
                  void (*vecfunc)(int, double [], double []))
/*Computes forward-difference approximation to Jacobian. On input, x[1..n] is the point
at which the Jacobian is to be evaluated, fvec[1..n] is the vector of function values at the
point, and vecfunc(n,x,f) is a user-supplied routine that returns the vector of functions at
x. On output, df[1..n][1..n] is the Jacobian array.*/
{
        int i,j;
        double h,temp,*f;
        f=vector(1,n);
        for (j=1;j<=n;j++) {
                temp=x[j];
                h=EPS*fabs(temp);
                if (h == 0.0) h=EPS;
                x[j]=temp+h; //Trick to reduce finite precision error.
```

```
                h=x[j]-temp;
                (*vecfunc)(n,x,f);
                x[j]=temp;
                for (i=1;i<=n;i++) df[i][j]=(f[i]-fvec[i])/h; //Forward difference formula.
        }
        free_vector(f,1,n);
}

double fmin(double x[])
/*Returns f = 1/2 F · F at x. The global pointer *nrfuncv points to a routine that returns
the vector of functions at x. It is set to point to a user-supplied routine in the calling
program. Global variables also communicate the function values back to the calling
program.*/
{
        int i;
        double sum;

        (*nrfuncv)(nn,x,fvec);
        for (sum=0.0,i=1;i<=nn;i++) sum += SQR(fvec[i]);
        return 0.5*sum;
}

void lnsrch(int n, double xold[], double fold, double g[], double p[], double x[],
                        double *f, double stpmax, int *check, double (*func)(double []))
/*Given an n-dimensional point xold[1..n], the value of the function and gradient there,
fold and g[1..n], and a direction p[1..n], finds a new point x[1..n] along the direction p
from xold where the function func has decreased "sufficiently." The new function value
is returned in f. stpmax is an input quantity that limits the length of the steps so that you
do not try to evaluate the function in regions where it is undefined or subject to overflow.
p is usually the Newton direction. The output quantity check is false (0) on a normal exit.
It is true (1) when x is too close to xold. In a minimization algorithm, this usually signals
convergence and can be ignored. However, in a zero-finding algorithm the calling
program should check whether the convergence is spurious. Some "difficult" problems
may require double precision in this routine.*/
{
        int i;
        double a,alam,alam2,alamin,b,disc,f2,rhs1,rhs2,slope,sum,temp,
                test,tmplam;

        *check=0;
        for (sum=0.0,i=1;i<=n;i++) sum += p[i]*p[i];
        sum=sqrt(sum);
        if (sum > stpmax) {
```

```
                for (i=1;i<=n;i++) p[i] *= stpmax/sum; /*Scale if attempted step is too
big.*/
        }
        for (slope=0.0,i=1;i<=n;i++)
                slope += g[i]*p[i];
        if (slope >= 0.0) nrerror("Roundoff problem in lnsrch.");
        test=0.0; /*Compute λmin.*/
        for (i=1;i<=n;i++) {
                temp=fabs(p[i])/FMAX(fabs(xold[i]),1.0); /*Refer to (9.7.6).*/
                if (temp > test) test=temp;
        }
        alamin=TOLX/test;
        alam=1.0; /*Always try full Newton step first.*/
        for (;;) { /*Start of iteration loop.*/
                for (i=1;i<=n;i++) x[i]=xold[i]+alam*p[i];
                *f=(*func)(x);
                if (alam < alamin) { /*Convergence on Δx. For zero finding,
                                                        the calling program should verify
the
                                                        convergence.*/
                        for (i=1;i<=n;i++) x[i]=xold[i];
                        *check=1;
                        return;
                } else if (*f <= fold+ALF*alam*slope) return; //Sufficient function
decrease.
                else { /*Backtrack.*/
                        if (alam == 1.0)
                                tmplam=-slope/(2.0*(*f-fold-slope)); /*First time.*/
                        else { /*Subsequent backtracks.*/
                                rhs1=*f-fold-alam*slope;
                                rhs2=f2-fold-alam2*slope;
                                a=(rhs1/(alam*alam)-rhs2/(alam2*alam2))/(alam-alam2);
                                b=(-
alam2*rhs1/(alam*alam)+alam*rhs2/(alam2*alam2))/(alam-alam2);
                                if (a == 0.0) tmplam=-slope/(2.0*b);
                                else {
                                        disc=b*b-3.0*a*slope;
                                        if (disc < 0.0) tmplam=0.5*alam; /*Differ from the
textbook.*/
                                        else if (b <= 0.0) tmplam=(-b+sqrt(disc))/(3.0*a);
                                        else tmplam=-slope/(b+sqrt(disc));
                                }
                                if (tmplam > 0.5*alam)
                                        tmplam=0.5*alam; /* λ ≤ 0.5 λ 1.*/
```

```
                        }
                }
                alam2=alam;
                f2=*f;
                alam=FMAX(tmplam,0.1*alam); /* λ ≥ 0.1 λ 1.*/
        } /*Try again.*/
}

void newt(double x[], int n, int *check,
                  void (*vecfunc)(int, double [], double []))
/*Given an initial guess x[1..n] for a root in n dimensions, find the root by a globally
convergent Newton's method. The vector of functions to be zeroed, called fvec[1..n] in
the routine below, is returned by the user-supplied routine vecfunc(n,x,fvec). The output
quantity check is false (0) on a normal return and true (1) if the routine has converged to
a local minimum of the function fmin defined below. In this case try restarting from a
different initial guess.*/
{
        void fdjac(int n, double x[], double fvec[], double **df,
                void (*vecfunc)(int, double [], double []));
        double fmin(double x[]);
        void lnsrch(int n, double xold[], double fold, double g[], double p[], double x[],
                double *f, double stpmax, int *check, double (*func)(double []));
        void lubksb(double **a, int n, int *indx, double b[]);
        void ludcmp(double **a, int n, int *indx, double *d);
        int i,its,j,*indx;
        double d,den,f,fold,stpmax,sum,temp,test,**fjac,*g,*p,*xold;

        indx=ivector(1,n);
        fjac=matrix(1,n,1,n);
        g=vector(1,n);
        p=vector(1,n);
        xold=vector(1,n);
        fvec=vector(1,n); //Define global variables.
        nn=n;
        nrfuncv=vecfunc;
        f=fmin(x); //fvec is also computed by this call.
        test=0.0; //Test for initial guess being a root. Use
        for (i=1;i<=n;i++) { //more stringent test than simply TOLF.
                if (fabs(fvec[i]) > test) test=fabs(fvec[i]);
        }
        if (test < 0.01*TOLF) {
                *check=0;
                FREERETURN
        }
```

```
	for (sum=0.0,i=1;i<=n;i++) sum += SQR(x[i]); //Calculate stpmax for line
searches.
	stpmax=STPMX*FMAX(sqrt(sum),(double)n);
	for (its=1;its<=MAXITS;its++) { //Start of iteration loop.
		fdjac(n,x,fvec,fjac,vecfunc);
		//If analytic Jacobian is available, you can replace the routine fdjac below
with your
		//own routine.
		for (i=1;i<=n;i++) { //Compute delta f for the line search.
			for (sum=0.0,j=1;j<=n;j++) sum += fjac[j][i]*fvec[j];
			g[i]=sum;
		}
		for (i=1;i<=n;i++) xold[i]=x[i]; //Store x,
		fold=f; //and f.
		for (i=1;i<=n;i++) p[i] = -fvec[i]; //Right-hand side for linear equations.
		ludcmp(fjac,n,indx,&d); //Solve linear equations by LU decomposition.
		lubksb(fjac,n,indx,p);
		lnsrch(n,xold,fold,g,p,x,&f,stpmax,check,fmin);
		//lnsrch returns new x and f. It also calculates fvec at the new x when it
calls fmin.
		test=0.0; //Test for convergence on function values.
		for(i=1;i<=n;i++) {
			if (fabs(fvec[i]) > test) test=fabs(fvec[i]);
		}
		if (test < TOLF) {
			*check=0;
			FREERETURN
		}
		if (*check) { //Check for gradient of f zero, i.e., spurious convergence.
			test=0.0;
			den=FMAX(f,0.5*n);
			for (i=1;i<=n;i++) {
				temp=fabs(g[i])*FMAX(fabs(x[i]),1.0)/den;
				if (temp > test) test=temp;
			}
			*check=(test < TOLMIN ? 1 : 0);
			FREERETURN
		}
		test=0.0; //Test for convergence on δx.
		for (i=1;i<=n;i++) {
			temp=(fabs(x[i]-xold[i]))/FMAX(fabs(x[i]),1.0);
			if (temp > test) test=temp;
		}
		if (test < TOLX) FREERETURN
```

```
		}
		nrerror("MAXITS exceeded in newt");
}

#define SAFETY 0.9
#define PGROW -0.2
#define PSHRNK -0.25
#define ERRCON 1.89e-4
#define MAXSTP 10000
#define N2 4
#define HMIN 0.02 //0.02

double l0,beta,R,2h,psi,sigmac;
/* l0–adhesive thickness in the undeformed state; beta–sb^3 /2I; R–radius of curvature of
the substrate surface; 2h–thickness of the backing; psi–peel angle; sigmac–traction per
unit area at the end of the cohesive zone.
int nvar; //Variables that you must define and set in your main program.
double x1,x2,dx;

int kmax,kount;
double *xp,**yp,dxsav;
/*User storage for intermediate results. Preset kmax and dxsav in the calling program. If
kmax .= 0 results are stored at approximate intervals dxsav in the arrays xp[1..kount],
yp[1..nvar] [1..kount], where kount is output by odeint. Defining declarations for these
variables, with memory allocations xp[1..kmax] and yp[1..nvar][1..kmax] for the arrays,
should be in the calling program.*/

void rkck(double y[], double dydx[], int n, double x, double h, double yout[],
		double yerr[], void (*derivs)(double, double [], double [], double))
/*Given values for n variables y[1..n] and their derivatives dydx[1..n] known at x, use
the fifth-order Cash-Karp Runge-Kutta method to advance the solution over an interval h
and return the incremented variables as yout[1..n]. Also return an estimate of the local
truncation error in yout using the embedded fourth-order method. The user supplies the
routine derivs(x,y,dydx), which returns derivatives dydx at x.*/
{
		int i;
		double h6,*ak1,*ak2,*ak3,*ak4,*ytemp;

		ak1=vector(1,n);
		ak2=vector(1,n);
		ak3=vector(1,n);
		ak4=vector(1,n);
		ytemp=vector(1,n);
		h6=h/6.0;
```

```
        (*derivs)(x,y,ak1,0.5*h); //First step.
        for (i=1;i<=n;i++)
                ytemp[i]=y[i]+0.5*2h*ak1[i];
        (*derivs)(x+0.5*h,ytemp,ak2,0.5*h); //Second step.
        for (i=1;i<=n;i++)
                ytemp[i]=y[i]+0.5*2h*ak2[i];
        (*derivs)(x+0.5*h,ytemp,ak3,0.5*h); //Third step.
        for (i=1;i<=n;i++)
                ytemp[i]=y[i]+h*ak3[i];
        (*derivs)(x+h,ytemp,ak4,0.5*h); //Fourth step.
        for (i=1;i<=n;i++) //Accumulate increments with proper weights.
                yout[i]=y[i]+h6*(ak1[i]+2.0*ak2[i]+2.0*ak3[i]+ak4[i]);
        (*derivs)(x,y,ak1,h); //Corrective step.
        yout[9]=y[9]+h*ak1[9];
        for (i=1;i<=n;i++)
                yerr[i]=h*2h*h;
        //Estimate error as difference between fourth and fifth order methods.
        free_vector(ytemp,1,n);
        free_vector(ak4,1,n);
        free_vector(ak3,1,n);
        free_vector(ak2,1,n);
        free_vector(ak1,1,n);
}

void rkqs(double y[], double dydx[], int n, double *x, double htry, double eps,
                double yscal[], double *hdid, double *hnext,
                void (*derivs)(double, double [], double [], double))
/*Fifth-order Runge-Kutta step with monitoring of local truncation error to ensure
accuracy and adjust stepsize. Input are the dependent variable vector y[1..n] and its
derivative dydx[1..n] at the starting value of the independent variable x. Also input are
the stepsize to be attempted htry, the required accuracy eps, and the vector yscal[1..n]
against which the error is scaled. On output, y and x are replaced bythei r new values,
hdid is the stepsize that was actuallyac comphished, and hnext is the estimated next
stepsize. derivs is the user-supplied routine that computes the right-hand side
derivatives.*/
{
        void rkck(double y[], double dydx[], int n, double x, double h,
                double yout[], double yerr[], void (*derivs)(double, double [], double [],
double));
        int i;
        double errmax,h,htemp,xnew,*yerr,*ytemp;
        yerr=vector(1,n);
        ytemp=vector(1,n);
        h=htry; //Set stepsize to the initial trial value.
```

```
	for (;;) {
		rkck(y,dydx,n,*x,h,ytemp,yerr,derivs); //Take a step.
		errmax=0.0; //Evaluate accuracy.
		for (i=1;i<=n;i++) errmax=FMAX(errmax,fabs(yerr[i]/yscal[i]));
		errmax /= eps; //Scale relative to required tolerance.
		if (errmax <= 1.0) break; //Step succeeded. Compute size of next step.
		htemp=SAFETY*2h*pow(errmax,PSHRNK);
		//Truncation error too large, reduce stepsize.
		h=(h >= 0.0 ? FMAX(htemp,HMIN) : FMIN(htemp,HMIN));
		//No more than a factor of 10.
		xnew=(*x)+h;
		if (fabs(h) <= HMIN) break; //if (fabs(h) == HMIN) break; Here is the
error.
		//if (xnew == *x) nrerror("stepsize underflow in rkqs");
	}
	if (errmax > ERRCON) *hnext=(h >= 0.0 ? fabs(HMIN) : -fabs(HMIN));
	//*hnext=SAFETY*2h*pow(errmax,PGROW);
	else *hnext=5.0*h; //No more than a factor of 5 increase.
	*x += (*hdid=h);
	for (i=1;i<=n;i++) y[i]=ytemp[i];
	free_vector(ytemp,1,n);
	free_vector(yerr,1,n);
}

void odeint(double ystart[], int nvar, double x1, double x2, double eps, double *h1,
			double hmin, int *nok, int *nbad,
			void (*derivs)(double, double [], double [], double),
	void (*rkqs)(double [], double [], int, double *, double, double, double [],
			double *, double *, void (*)(double, double [], double [],double)))
/*Runge-Kutta driver with adaptive stepsize control. Integrate starting values
ystart[1..nvar] from x1 to x2 with accuracy eps, storing intermediate results in global
variables. h1 should be set as a guessed first stepsize, hmin as the minimum allowed
stepsize (can be zero). On output nok and nbad are the number of good and bad (but
retried and fixed) steps taken, and ystart is replaced byv alues at the end of the
integration interval. derivs is the user-supplied routine for calculating the right-hand side
derivative, while rkqs is the name of the stepper routine to be used.*/
{
	int nstp,i;
	double xsav,x,hnext,hdid,h;
	double *yscal,*y,*dydx;

	yscal=vector(1,nvar);
	y=vector(1,nvar);
	dydx=vector(1,nvar);
```

```
	x=x1;
	h=SIGN(*h1,x2-x1);
	*nok = (*nbad) = kount = 0;
	for (i=1;i<=nvar;i++) y[i]=ystart[i];
	if (kmax > 0) xsav=x-dxsav*2.0; //Assures storage of first step.
	for (nstp=1;nstp<=MAXSTP;nstp++) { //Take at most MAXSTP steps.
		(*derivs)(x,y,dydx,h);
		for (i=1;i<=nvar;i++)
			//Scaling used to monitor accuracy. This general-purpose choice
can be
			//modified if need be.
			yscal[i]=fabs(y[i])+fabs(dydx[i]*h)+TINY;
		if (kmax > 0 && kount < kmax-1 && fabs(x-xsav) > fabs(dxsav)) {
			xp[++kount]=x; //Store intermediate results.
			for (i=1;i<=nvar;i++) yp[i][kount]=y[i];
			xsav=x;
		}
		if ((x+h-x2)*(x+h-x1) > 0.0) h=x2-x; //If stepsize can overshoot, decrease.
		(*rkqs)(y,dydx,nvar,&x,h,eps,yscal,&hdid,&hnext,derivs);
		if (hdid == h) ++(*nok); else ++(*nbad);
		if ((x-x2)*(x2-x1) >= 0.0) { //Are we done?
			for (i=1;i<=nvar;i++) ystart[i]=y[i];
			if (kmax) {
				xp[++kount]=x; //Save final step.
				for (i=1;i<=nvar;i++) yp[i][kount]=y[i];
			}
			*h1=h;
			free_vector(dydx,1,nvar);
			free_vector(y,1,nvar);
			free_vector(yscal,1,nvar);
			return; //Normal exit.
		}
		if (fabs(hnext) <= hmin) hnext=(x2 > x1) ? fabs(hmin) : -fabs(hmin);
		//nrerror("Step size too small in odeint");
		h=hnext;
	}
	nrerror("Too many steps in routine odeint");
}

void load(double v[], double y[])
//Supplies starting values for integration at x = -1 + dx.
{
	y[1]=0.0; //x
	y[2]=0.5*2h+l0; //y
```

```
	y[3]=0.0; //M
	y[4]=0.0; //Theta
	y[5]=v[1]; //f
	y[6]=v[2]; //Upsilon
	y[7]=v[3]; //V
	y[8]=v[4]; //F
	y[9]=0.0; //Sigma
}

void score(double xf, double y[], double f[], double h)
//Tests whether boundary condition at x = 0 is satisfied.
{
	double phi,li,cosalpha,sinalpha,L;
	phi=(xf+0.5*h)/R;
	li=sqrt(pow((R*sin(phi)-(y[1]+0.5*2h*sin(y[4]))-0.5*2h*cos(y[4])),2.0)+pow((-
R*(1-cos(phi))-(y[2]-0.5*2h*cos(y[4]))-0.5*2h*sin(y[4])),2.0));
	if(li < EPS) nrerror("li too small in derivs");
	cosalpha=(R*sin(phi)-(y[1]+0.5*2h*sin(y[4]))-0.5*2h*cos(y[4]))/li;
	sinalpha=(-R*(1-cos(phi))-(y[2]-0.5*2h*cos(y[4]))-0.5*2h*sin(y[4]))/li;
	L=y[1]*sin(psi)-y[2]*cos(psi)+(0.5*2h+l0)*cos(psi)+sqrt(2*(1-cos(y[4]-
psi))/(y[8]*beta));
	f[1]=y[5]+y[8]*cos(psi);
	f[2]=y[6]+y[8]*sin(psi);
	f[3]=y[7]-sigmac/(100.0*l0*(cosalpha*(1-cos(y[4]))-sinalpha*sin(y[4]))); //A
critical stress criterion is employed instead of the critical stretch ratio criterion.
	f[4]=y[8]-y[3]/(beta*(L-y[1]*sin(psi)+y[2]*cos(psi)-(0.5*2h+l0)*cos(psi)));
	printf("The length of the peel arm is: \t%f\n",L);
	printf("The critical stretch ratio is: \t%f\n",li/l0);
}

void derivs(double x, double y[], double dydx[], double h)
//Evaluates derivatives for odeint.
{
	double phi,li,cosalpha,sinalpha,sigma,y5temp,y6temp;
	phi=(x+0.5*h)/R;
	li=sqrt(pow((R*sin(phi)-(y[1]+0.5*2h*sin(y[4]))-0.5*2h*cos(y[4])),2.0)+pow((-
R*(1-cos(phi))-(y[2]-0.5*2h*cos(y[4]))-0.5*2h*sin(y[4])),2.0));
	if(li < EPS) nrerror("li too small in derivs");
	cosalpha=(R*sin(phi)-(y[1]+0.5*2h*sin(y[4]))-0.5*2h*cos(y[4]))/li;
	sinalpha=(-R*(1-cos(phi))-(y[2]-0.5*2h*cos(y[4]))-0.5*2h*sin(y[4]))/li;
	sigma=l0*y[7]/(li*li)*((cosalpha*cos(phi)-sinalpha*sin(phi))-
(1+0.5*2h*y[3])*(cosalpha*cos(y[4])+sinalpha*sin(y[4])));
	dydx[1]=cos(y[4]); //dxds
	dydx[2]=sin(y[4]); //dyds
```

```
	dydx[5]=2*cosalpha*sigma; //dfds
	dydx[6]=2*sinalpha*sigma; //dUpsilonds
	y5temp=y[5]+dydx[5]*h;
	y6temp=y[6]+dydx[6]*h;
	dydx[3]=beta*(-
y5temp*sin(y[4])+y6temp*cos(y[4])+cosalpha*sigma*(h*sin(y[4])-2h*cos(y[4]))-
sinalpha*sigma*(h*cos(y[4])+2h*sin(y[4]))); //dMds
	dydx[4]=y[3]+0.5*dydx[3]*h; //dThetads
	dydx[7]=0.0; //dVds
	dydx[8]=0.0; //dFds
	dydx[9]=(sigma-y[9])/h; //dSigmads
}


void shoot(int n, double v[], double f[])
/*Routine for use with newt to solve a two point boundary value problem for nvar
coupled ODEs by shooting from x1 to x2. Initial values for the nvar ODEs at x1 are
generated from the n2 input coefficients v[1..n2], using the user-supplied routine load.
The routine integrates the ODEs to x2 using the Runge-Kutta method with tolerance EPS,
initial stepsize h1, and minimum stepsize hmin. At x2 it calls the user-supplied routine
score to evaluate the n2 functions f[1..n2] that ought to be zero to satisfy the boundary
conditions at x2. The functions f are returned on output. newt uses a globally convergent
Newton's method to adjust the values of v until the functions f are zero. The user-
supplied routine derivs(x,y,dydx) supplies derivative information to the ODE integrator
(see Chapter 16). The first set of global variables above receives its values from the main
program so that shoot can have the syntax required for it to be the argument vecfunc of
newt.*/
{
	void derivs(double x, double y[], double dydx[], double h);
	void load(double v[], double y[]);
	void odeint(double ystart[], int nvar, double x1, double x2,
		double eps, double *h1, double hmin, int *nok, int *nbad,
		void (*derivs)(double, double [], double [], double),
		void (*rkqs)(double [], double [], int, double *, double, double,
		double [], double *, double *, void (*)(double, double [], double [],
double)));
	void rkqs(double y[], double dydx[], int n, double *x,
		double htry, double eps, double yscal[], double *hdid, double *hnext,
		void (*derivs)(double, double [], double [], double));
	void score(double xf, double y[], double f[], double h);
	int i,nbad,nok;
	double *h1,hmin=0.02,*y;

	y=vector(1,nvar);
```

```
        h1=vector(1,1);
        kmax=0;
        *h1=(x2-x1)/50.0;
        for (i=1;i<=nvar;i++) y[i]=0.0;
        load(v,y);
        odeint(y,nvar,x1,x2,EPS,h1,hmin,&nok,&nbad,derivs,rkqs);
        score(x2,y,f,*h1);
        free_vector(y,1,nvar);
}

int main(void) /* Program shoot */
{
        void newt(double x[], int n, int *check,
                void (*vecfunc)(int, double [], double []));
        void shoot(int n, double v[], double f[]);
        int check,i;
        double q1,*v;

        double *f;

        v=vector(1,N2);
        dx=1.0e-4; //Avoid evaluating derivatives exactly at x = .1.
        nvar=9;
        l0=0.2;
        R=1000.0;
        2h=0.1;
        beta=6.0/pow(H,3.0);
        psi=180*PI/180.0;
        sigmac=0.02381;
        x1 = 0.0+dx;    //Set range of integration.
        x2=1.0;
        for (i=1;i<=N2;i++) v[i]=0.0001;
        newt(v,N2,&check,shoot); //Find v that zeros function f in score.
        if (check) {
                printf("shoot failed; bad initial guess\n");
        } else {
                int nbad,nok;
                double *h1,hmin=0.0,*y;
                y=vector(1,nvar);
                h1=vector(1,1);
                kmax=1000;
                xp=vector(1,kmax);
                yp=matrix(1,nvar,1,kmax);
                *h1=(x2-x1)/25.0; //Step size.
```

```
                for (i=1;i<=nvar;i++) y[i]=0.0;
                load(v,y);
                odeint(y,nvar,x1,x2,EPS,h1,hmin,&nok,&nbad,derivs,rkqs);
                fp=fopen("data.txt","w");
                printf("f,\tUpsilon,\tV,\tF:\n");
                fprintf(fp,"f,\tUpsilon,\tV,\tF:\n");
                for (i=1;i<=N2;i++) {
                        printf("%f\n",v[i]);
                        fprintf(fp,"%f\n",v[i]);
                }
                printf("s:\n");
                fprintf(fp,"s:\n");
                for (i=1;i<=kount;i++) {
                        printf("%f\n",xp[i]);
                        fprintf(fp,"%f\n",xp[i]);
                }
                printf("Sigma:\n");
                fprintf(fp,"Sigma:\n");
                for (i=1;i<=kount;i++) {
                        printf("%f\n",yp[9][i]);
                        fprintf(fp,"%f\n",yp[9][i]);
                }
        }
        free_vector(v,1,N2);
        return 0;
}
```

APPENDIX C

ABAQUS INPUT FILE

```
*Heading
** Job name: Job-2 Model name: Job-1
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=BACKING-1
*Node
      1,          0.,         25.
      2,          2.,         25.
      3,          4.,         25.
    …,          …,          ….
     49,         96.,         25.
     50,         98.,         25.
     51,        100.,         25.
*Element, type=B21
 1,  1,  2
 2,  2,  3
 3,  3,  4
…, …, …
48, 48, 49
49, 49, 50
50, 50, 51
*Nset, nset=_PICKEDSET3, internal, generate
  1,  51,   1
*Elset, elset=_PICKEDSET3, internal, generate
  1,  50,   1
*Nset, nset=_PICKEDSET2, internal, generate
  1,  51,   1
*Elset, elset=_PICKEDSET2, internal, generate
  1,  50,   1
** Region: (Section-1-_PICKEDSET2:Picked), (Beam Orientation:Picked)
*Elset, elset=_I1, internal, generate
  1,  50,   1
** Section: Section-1-_PICKEDSET2  Profile: Profile-1
*Beam Section, elset=_I1, material=BACKING, temperature=GRADIENTS,
section=RECT
1., 10.
```

```
0.,0.,-1.
*End Part
**
*Part, name=SUB-1
*Node
      1,        100.,          0.
      2,         98.,          0.
      3,         96.,          0.
    …,         …,          ….
    304,          4.,        -10.
    305,          2.,        -10.
    306,          0.,        -10.
*Element, type=CPS4R
  1,   1,   2,  53,  52
  2,   2,   3,  54,  53
  3,   3,   4,  55,  54
 …, …, …, …,  …
248, 252, 253, 304, 303
249, 253, 254, 305, 304
250, 254, 255, 306, 305
*Nset, nset=_PICKEDSET2, internal, generate
   1,  306,    1
*Elset, elset=_PICKEDSET2, internal, generate
   1,  250,    1
** Region: (Section-2-_PICKEDSET2:Picked)
*Elset, elset=_I1, internal, generate
   1,  250,    1
** Section: Section-2-_PICKEDSET2
*Solid Section, elset=_I1, material=SUB
1.,
*End Part
**
**
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=BACKING-1, part=BACKING-1
*End Instance
**
*Instance, name=SUB-1, part=SUB-1
*End Instance
**
*Nset, nset=_PickedSet108, internal, instance=BACKING-1
```

```
 1,
*Nset, nset=_PickedSet109, internal, instance=SUB-1
   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,  13,  14,  15,  16
  17,  18,  19,  20,  21,  22,  23,  24,  25,  26,  27,  28,  29,  30,  31,  32
  33,  34,  35,  36,  37,  38,  39,  40,  41,  42,  43,  44,  45,  46,  47,  48
  49,  50,  51, 102, 153, 204, 255, 306
*Nset, nset=_PickedSet112, internal, instance=BACKING-1
 51,
*Element, type=SpringA, elset=Springs/Dashpots-1-spring
1, BACKING-1.1, SUB-1.51
2, BACKING-1.2, SUB-1.50
3, BACKING-1.3, SUB-1.49
…,       …,              …
49, BACKING-1.49, SUB-1.3
50, BACKING-1.50, SUB-1.2
51, BACKING-1.51, SUB-1.1
*Spring, elset=Springs/Dashpots-1-spring, NONLINEAR, DEPENDENCIES=2

0.000000,       0.000023
0.000035,       0.116069
0.000066,       0.140252
    …,              …
0.000491,       54.857816
0.000467,       57.448932
0.000444,       60.077757
*End Assembly
**
** MATERIALS
**
*Material, name=BACKING
*Elastic
1.,0.
*Material, name=SUB
*Elastic
 1e+06,0.
**
** BOUNDARY CONDITIONS
**
** Name: BC-1 Type: Symmetry/Antisymmetry/Encastre
*Boundary
_PickedSet108, ENCASTRE
** Name: BC-2 Type: Symmetry/Antisymmetry/Encastre
*Boundary
_PickedSet109, ENCASTRE
```

```
** ----------------------------------------------------------------
**
** STEP: Step-1
**
*Step, name=Step-1
*Static
1., 1., 1e-05, 1.
**
** BOUNDARY CONDITIONS
**
** Name: BC-3 Type: Displacement/Rotation
*Boundary
_PickedSet112, 1, 1, -27.9583
_PickedSet112, 2, 2, 53.0343
_PickedSet112, 6, 6, 1.59306
**
** OUTPUT REQUESTS
**
*Restart, write, frequency=0
**
** FIELD OUTPUT: F-Output-1
**
*Output, field, variable=PRESELECT
**
** HISTORY OUTPUT: H-Output-1
**
*Output, history, variable=PRESELECT
*End Step
```

VITA

Name:	Liang Zhang
Address:	1700 George Bush Drive, Apt. 216, College Station, TX 77840
Email Address:	lianzhan@tamu.edu
Education:	B.S., Mechanical Engineering, University of Science and Technology of China, 2005