SANDIA REPORT
SAND2005-7357
Unlimited Release
Printed January 2006

Constitutive Models for Rubber Networks Undergoing Simultaneous Crosslinking and Scission

Joanne Budzien, David Chi S. Lo, John G. Curro, Aidan P. Thompson, Gary S. Grest, and Dana Rottach

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2005-7357
Unlimited Release
Printed January 2006

Constitutive Models for Rubber Networks Undergoing Simultaneous Crosslinking and Scission

Joanne Budzien
Computational Materials Science & Engineering

David Chi S. Lo
Material Mechanics

John G. Curro
Ceramic Processing & Inorganic Materials

Aidan P. Thompson
Multiscale Computational Material Methods

Gary S. Grest
Surface & Interface Sciences

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1411

Dana Rottach
Department of Chemical and Nuclear Engineering
University of New Mexico
Albuquerque, NM 87106

Abstract

Constitutive models for chemically reacting networks are formulated based on a generalization of the independent network hypothesis. These models account for the coupling between chemical reaction and strain histories, and have been tested by comparison with microscopic molecular dynamics simulations. An essential feature of these models is the introduction of stress transfer functions that describe the interdependence between crosslinks formed and broken at various strains. Efforts are underway to implement these constitutive models into the finite element code Adagio. Preliminary results are shown that illustrate the effects of changing crosslinking and scission rates and history.

CONTENTS

FIGURES

TABLES

LIST OF SYMBOLS

$\underline{\underline{A}}_i = \underline{\underline{\lambda}}\underline{\underline{\lambda}}_i^{-1}$	deformation tensor relative to stage i
$\underline{\underline{E}}$	identity matrix
F	Helmholtz free energy
n	number of states of ease
N_e	average chain length between entanglements
N_i	number of chains formed in stage i
p	indeterminate Langrangian multiplier
p_0	fractional conversion in the unstrained state (# of reacted sites/total # of sites capable of reaction)
p_c	fractional conversion at the gel point
R	gas constant
T	temperature
V	current volume
κ	bulk modulus of the material
λ	stretch ratio (deformed length/original length)
λ_x	stretch ratio based on stage x deformation (deformed length at stage x/original length)
$\underline{\underline{\lambda}}_i$	deformation gradient of stage i network relative to the state of ease
$\underline{\underline{\lambda}}$	deformation gradient relative to original state
ν	total chain density
ν_i	current chain density from stage i crosslinks
ν_i^{eff}	effective chain density resulting from stage i crosslinks and stress transfer function
ν_i^*	chain density that resulted from the original insertion of stage i crosslinks
ν_i^{Rj}	number of chains (per volume) that were created in stage i which were destroyed in stage j
σ	true stress
$\underline{\underline{\sigma}}$	true stress tensor
Φ_i^{Rj}	stress transfer function for stage i chains destroyed in stage j

1. BACKGROUND

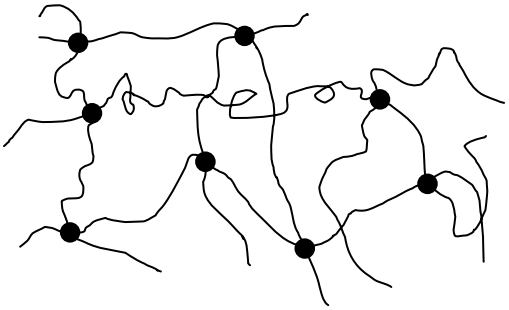
In many defense applications, elastomeric (typically butyl, ethylene-propylene, or silicone rubber) o-rings are used as barriers to protect sensitive components from air and moisture. While in use, o-rings experience chemical reactions that lead to a change in mechanical properties of the rubber with time. The effect of this chemical aging under typical stockpile conditions can be dramatic over the course of decades. Of greatest importance is the gradual loss of sealing force such that o-rings no longer perform their primary function after years of service. An important corollary behavior is that o-rings that have been compressed for years will not return to their original shape upon release from confinement. The change in dimensions is usually quantified by the rubber industry in terms of a permanent set. The goal of the present investigation is to formulate a constitutive model that accurately predicts the stress as a function of strain and chemical reaction history. In general, viscoelastic processes also have an effect on the time response of a rubber component; however, in this work, we will focus on the rubber elastic behavior arising from the chemical changes in the network. Although viscoelastic effects can contribute appreciably to the stress response, previous work[1, 2] has shown that ultimate failure of an o-ring, where the sealing force approaches zero, is traceable to the dependence of the equilibrium rubber elastic modulus on chemical aging.

Permanent set can be understood as a competition between crosslinks formed in the undeformed and deformed states. On a molecular level, rubbers are composed of long molecules connected through crosslinks (as seen in Figure 1). At equilibrium, these molecules have preferred mean squared end-to-end distances between crosslink points. For the entire sample, this leads to a zero normal stress state termed the state-of-ease. Stress is a measure of the restoring force exerted by the molecules when the sample is deformed from its state-of-ease. The restrictions placed on the chain motions by the crosslinks gives rise to the elastic behavior. Indeed, many of the molecular theories of rubber elasticity are based on the mean squared end-to-end distances of the strands between the crosslinks, and the possible changes in those distances. Thus, compressing or elongating the sample will change the mean squared end-to-end distance of the chains and the observed stress will change. This simple picture is complicated by the fact that rubber networks are not static entities. The network changes with time as crosslinks are added or scission occurs.

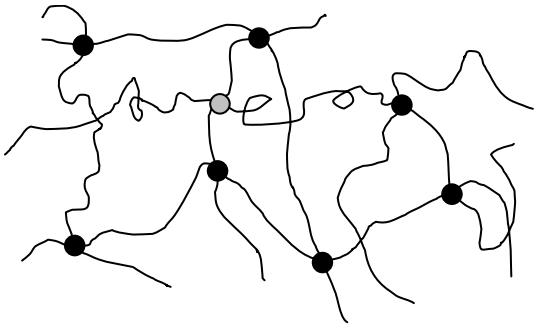
If a rubber sample is deformed for an extended time (as in a typical o-ring application), post-curing can occur resulting in additional crosslinks being introduced to the deformed sample. Crosslinks require proximity of specific moieties on chains in order to react. The positions of the new crosslinks are dependent on the positions of the already existing crosslinks. If some crosslinks have already reduced the number of possible conformations of the chains, then the new crosslinks have limited possibilities for reactions. The strands formed by the new crosslinks in a deformed state have a different state-of-ease than the previously existent strands. Many years ago, Tobolsky proposed

that networks crosslinked in various states of strain obey the two-network hypothesis[3] (termed the independent network hypothesis in generalizations by later researchers). The main idea is when crosslinks are added in different deformations (or stages), the resulting stress is as though separate networks are formed with states-of-ease at the deformations where the crosslinks were added. The overall response of these networks is the sum of the individual stresses for the different networks. Thus, the deformation history is very important to the prediction of the mechanical properties of rubber.

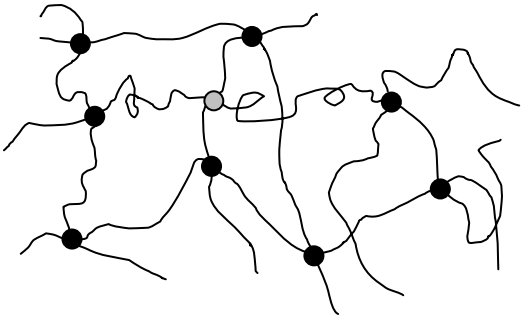
a. Stage 1 network forms



c. Stage 1 network scissions



b. Stage 2 network forms



d. Stress transfers from stage 2 to stage 1

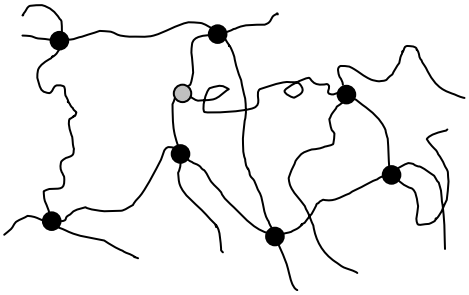


Figure 1. Schematic of a two-stage rubber network with scission and stress transfer.

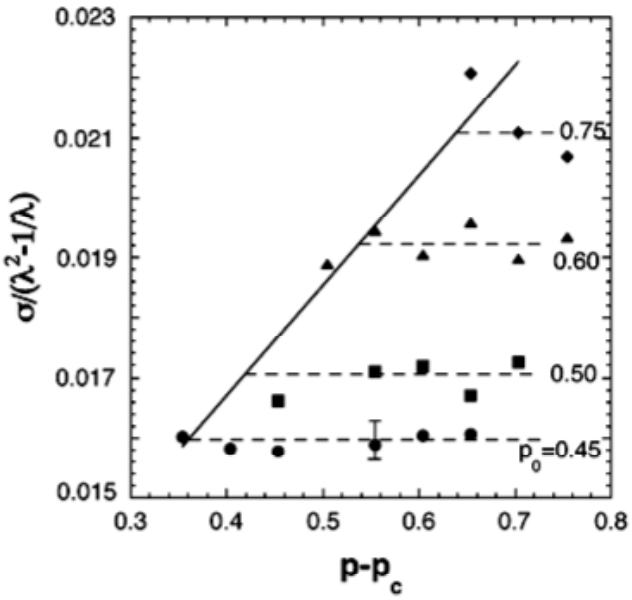


Figure 2. Molecular dynamics results of crosslinking to given extents in a strained state.
p_0 is the fractional conversion in the unstrained state, and p_c is the fractional conversion at the gel point, here equal to 0.096. The ordinate is the normalized deviatoric stress (σ) when the sample is held at the same stretch ratio (λ =1.5) where the second stage crosslinks are introduced (from Reference 4).

The independent network hypothesis was demonstrated recently for a coarse-grained polymer model using molecular dynamics (MD) simulations[4]. Five hundred chains with five hundred sites each were simulated and a random 4% of the sites were designated as reactive. A fraction of the reactive sites was allowed to form crosslinks in an undeformed state. The sample was uniaxially deformed so that the stretch ratio λ was 1.5 and additional reactive sites were allowed to form crosslinks. Figure 2 shows the stress in the deformed state as additional crosslinks are introduced. As expected from the independent network hypothesis, the stress did not change with the addition of new crosslinks as long as the sample is held in the same deformed state. Thus, the stress depends solely on how many undeformed or first-stage crosslinks were formed. The deformed or second-stage crosslinks only come into play when the deformation state is changed. In particular, if the sample is allowed to return to zero stress, it will take on a permanent set.

In actual o-ring applications not only can crosslinks form in deformed states, but also scission of existing crosslinks or strands can occur. Overall, the naive way to treat the problem is to simply reduce the number of elastic chains by the number of scissions. However, a true independent network treatment does not take into account the history dependence of network formation. For a rubber component, generally, the first network (stage 1 network) is formed during vulcanization and processing. While the rubber component is in service (i.e., in a deformed state), a second network (stage 2 network) can be formed after the first network is in place. The topology of the first network

strongly influences the topology that successive networks can form. If the first network starts to degrade after the later network has a substantial gelation, the effect of the first network on the stress remains even after all the first stage crosslinks are removed. The later network will retain some memory of the first network and may be able to act as if some percentage of the first stage network remains. Thus, the independent network hypothesis is inapplicable when crosslinks are removed as well as formed. We have seen this effect clearly in the MD simulations shown in Figure 3 where all of the first stage crosslinks are removed after introduction of stage 2 crosslinks.

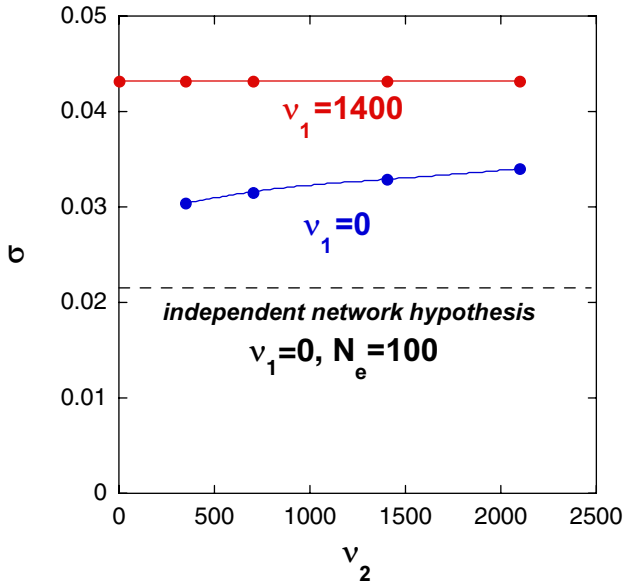


Figure 3. Molecular dynamics simulation results for a two-stage network with scission.

(λ_1=1,λ_2=2) with 80 initial chains of 3500 sites each. Stress is measured at λ_2 and is in reduced Lennard-Jones units. ν_x is the number of crosslinks formed in stage x that still remains. The dashed line indicates a value based on the independent network hypothesis with trapped entanglements considered a network.

Figure 3 shows the stress is larger than predicted from the independent network hypothesis. In effect, when the first stage crosslinks are removed, some of the second stage crosslinks take over the role of the first stage crosslinks thus preserving a memory of the initial network. As first proposed by Flory[5], the stress transfer function accounts for this memory effect. The stress transfer function allows a certain percentage of later stage crosslinks to act as though they were earlier stage crosslinks. The strength of this effect depends on the relative numbers of crosslinks in the networks at the current state of strain and in previous states of strain.

To get a qualitative feel for why this memory effect occurs, consider the network in Figure 1. In Figure 1a, the sample reacts in the undeformed state creating a network using stage 1 crosslinks. The sample is deformed in Figure 1b and some stage 2 crosslinks are inserted in the deformed state (one is shown as the grey circle). No chemical difference necessarily exists between stage 1 and stage 2 crosslinks; these crosslinks are differentiated by their state-of-ease. If one of the stage 1 crosslinks breaks as shown in Figure 1c, the resulting network is different from what would have resulted if the now-removed stage 1 crosslinks had not been present during the reactions that created the stage 2 crosslinks. The stage 2 crosslinks continue to enforce many of the restrictions on the chain conformations that were previously imposed by the stage 1 crosslinks. Note from Figure 1c that removing the stage 1 crosslink created a dangling end. Since this dangling end is elastically ineffective (i.e., will not contribute to the stress of the network), it can be removed from the network as in Figure 1d. We now see that the stage 2 crosslink takes the role of the first stage crosslink that was removed. This pro tempore transfer of crosslinks from one stage to another is probably not a one-for-one substitution in a real network, but is more likely a result of the combined effects of several later stage crosslinks functioning to hold the strands in conformations similar to that required by the vacated crosslinks.

Nothing restricts a sample to the simple two-stage strain history. Indeed, under some conditions, materials cycle repeatedly through a few extension states. Likewise, no theoretical barrier exists to prevent crosslinks from any later stage becoming effective earlier stage crosslinks so that second, third, and fourth stage crosslinks could compensate for breakage of the first stage network. Logically, earlier stage crosslinks cannot become effective later stage crosslinks because the transfer idea is based on faithfulness to the network structure in which the crosslink was formed. We emphasize that no chemical difference need exist between the crosslinks attributed to different stages; the difference results from the deformation at which the crosslinks form that leads to different states-of-ease.

Consequently, our idea is that stress can be calculated from any reasonable strain energy formulation. Stress transfer functions (the later stage crosslinks acting as earlier stage crosslinks) can be used to make the independent network hypothesis conform to what happens in real networks undergoing both scission and crosslinking for a series of independent networks formed in various strain states.

2. THE STRESS TRANSFER FUNCTION AND STRESS CALCULATION

2.1. Deriving the Stress Transfer Function

Flory was the first to explore removing some stage 1 crosslinks in a two-stage network and the resulting effect upon the stress[5]. Using an affine network of Gaussian chains, he speculated that the stress would be higher than expected from the simple independent network hypothesis and he attributed this to the effect of stage 1 crosslinks influencing the placement of stage 2 crosslinks. Flory used statistical mechanics to derive the conformations of the strands based upon forming one network, deforming, and crosslinking to form a second network, and finally removing some of the original crosslinks. By knowing the conformations of the strands, the free energy of the network can be determined, which is directly related to the stress. The stress for a rubber network is frequently written as a function of the chain density. The chain density (ν) is the number of network chains that can support stress (i.e., not dangling ends or loops) per sample volume.

The most convenient way to use the stress transfer function (Φ) is through the definition of an effective chain density (ν^{eff}). This effective chain density is a combination of the original network remaining after scission modified by contributions from later stage networks and contributions to earlier stage networks. Thus, while the chain densities of early stage networks are reduced by scission, some of this loss is offset by fractions of later stage networks acting as earlier stage networks. Effectively, the chain density for an early stage network is larger than its current value while the chain densities for later stage networks are correspondingly smaller. The stress transfer functions determine the fraction of chains that act as effective chains for other networks. Therefore, the effective chain density allows the stress transfer function to be applied in a physically meaningful way. For two stages, the effective chain densities can be written as:

$$\begin{aligned} \nu_1^{eff} &= \nu_1 + \Phi \nu_2 \\ \nu_2^{eff} &= \nu_2 - \Phi \nu_2 \end{aligned} \qquad (1)$$

where ν_x is the current chain density of the network formed in stage x (the original value reduced by scissions), ν_x^{eff} is the effective chain density for the network formed at stage x, and Φ is the stress transfer function.

Flory's original formulation[5] for the stress transfer function (Φ), which is exact for affine networks, is a complicated function of the original number of crosslinks and the remaining number of crosslinks. It only can be written in closed form for the complete removal of first stage crosslinks. Fricker[6] proposed that Φ is simply the fraction of stage 1 crosslinks removed leading to a much simpler expression for a two-stage phantom network as

$$\Phi = \frac{\nu_1^R}{\nu_1^R + \nu_1 + \nu_2} \tag{2}$$

where $0.5\,\nu_1^R$ is the density of first stage crosslinks removed and $0.5\nu_x$ is the density of stage x crosslinks remaining.

In this work, we extend Flory's and Fricker's analyses of the stress transfer functions for two-stage networks to an arbitrary number of stages. The basis of Flory's arguments stems from the history dependence of the network formations. We therefore propose that multiple stress transfer functions exist at a given stage with one stress transfer function for each earlier stage network.

Using this idea, we generalize the stress transfer functions from the Fricker formula to be

$$\Phi_i^{Rj} = \frac{\text{stage i crosslinks removed in stage j}}{\text{total number of crosslinks added (up through stage j)}} = \frac{\upsilon_i^{Rj}}{\sum_{k=1}^{j} \nu_k^*} \tag{3}$$

where ν_k^* is the initial chain density for the network formed in stage k, ν_i^{Rj} is the difference between the chain density of the stage i network at the beginning of stage j and at the end of stage j (i.e., the effect of the scissions for the stage i networks that occur during stage j), and Φ_i^{Rj} is the stress transfer function for the stage i network based on chemical changes during stage j.

This definition of the stress transfer function allows the effective strand density for n stages to be written as

$$\nu_p^{eff} = \nu_p \left(1 - \sum_{i=1}^{p-1} \sum_{k=p}^{n} \Phi_i^{Rk} \right) + \sum_{i=p+1}^{n} \nu_i \sum_{k=i}^{n} \Phi_p^{Rk} \tag{4}$$

This can also be written as

$$\nu_p^{eff} = \nu_p\left(1 - \sum_{i=1}^{p-1}\sum_{k=p}^{n}\Phi_i^{Rk}\right) + \sum_{i=p+1}^{n}\Phi_p^{Ri}\sum_{k=p+1}^{i}\nu_k \qquad (5)$$

where ν_x is the current chain density for the network formed at stage x and ν_x^{eff} is the effective current chain density for the network formed at stage x. For notational convenience, if the end index is smaller than the start index, no terms are summed.

A consequence of this general formulation is that networks from all later stages can contribute to the effective strand density of a given network. As an example of the consequences of this transfer, we show here a thought experiment that demonstrates the use of Equation 5. Table 1 shows a hypothetical data set of chain densities for a system subjected to five stages of deformation. This example qualitatively corresponds to a system with excess curing agent so that crosslinking occurs much faster than scission. The system is held at some fixed deformation for a given time period with chemistry allowed to occur. At the end of that time, the chain density corresponding to each network is recorded for the current deformation (Table 1a) and the system is deformed to the next stage. The scissions that occur can be calculated by simple subtraction from the remaining chains (Table 1b). This allows the use of Equation 3 to calculate the stress transfer function (Table 1c). Then, the effective chain densities can be calculated using Equation 5 (Table 1d). For illustrative purposes, the chain density values have been selected as simple numbers in consistent units and no attempt has been made to match with experimental data.

To view the effect of the transfers, consider what happens to the stage three network. The first term in Equation 5 is the transfer from the network formed at stage p to earlier networks. At the end of deformation stage five, this term for the stage three network reads

$$\nu_3(1 - \phi_1^{R3} - \phi_1^{R4} - \phi_1^{R5} - \phi_2^{R3} - \phi_2^{R4} - \phi_2^{R5}) \qquad (6)$$

Upon substituting the values of the stress transfer functions, the result is that almost half (47.2%) of the stage three network is acting as an earlier stage network. The second term in Equation 5 is the contribution to the stage p network from all later stage networks. At the end of deformation stage five, this second term for the stage three network reads

$$\nu_4\left(\phi_3^{R4} + \phi_3^{R5}\right) + \nu_5\phi_3^{R5} \qquad (7)$$

By substituting the data from the table into this equation, we see that the stage three network retains 86% of its current chain density as an effective value. This is a result of the difference between losing a chain density of 2.36 and gaining a chain density of 1.67 so that overall little loss has occurred.

Table 1. Example of stress transfer functions for five stage deformation.

(a) Current chain density

	ν_1	ν_2	ν_3	ν_4	ν_5	**total** ν
stage 1	15	0	0	0	0	15
stage 2	10	15	0	0	0	25
stage 3	5	10	15	0	0	30
stage 4	0	5	10	15	0	30
stage 5	0	0	5	10	0	15
$\mathbf{v_j^*}$	15	15	15	15	0	60

(b) Breakage at each stage

	ν^R_1	ν^R_2	ν^R_3	ν^R_4	ν^R_5
stage 1	0	0	0	0	0
stage 2	5	0	0	0	0
stage 3	5	5	0	0	0
stage 4	5	5	5	0	0
stage 5	0	5	5	5	0

(c) Stress transfer functions

j=	**1**	**2**	**3**	**4**	**5**
$\sqrt{}_i^{R1}$	0	0	0	0	0
$\sqrt{}_i^{R2}$	0.167	0	0	0	0
$\sqrt{}_i^{R3}$	0.111	0.111	0	0	0
$\sqrt{}_i^{R4}$	0.083	0.083	0.083	0	0
$\sqrt{}_i^{R5}$	0	0.083	0.083	0.083	0

(d) Effective chain density

	ν^1	ν^2	ν^3	ν^4	ν^5	**total** ν
stage 1	15	0	0	0	0	15
stage 2	12.5	12.5	0	0	0	25
stage 3	9.44	8.89	11. 7	0	0	30
stage 4	5	6.39	7.36	11.2	0	30
stage 5	1.80	3.06	4.30	5.83	0	15

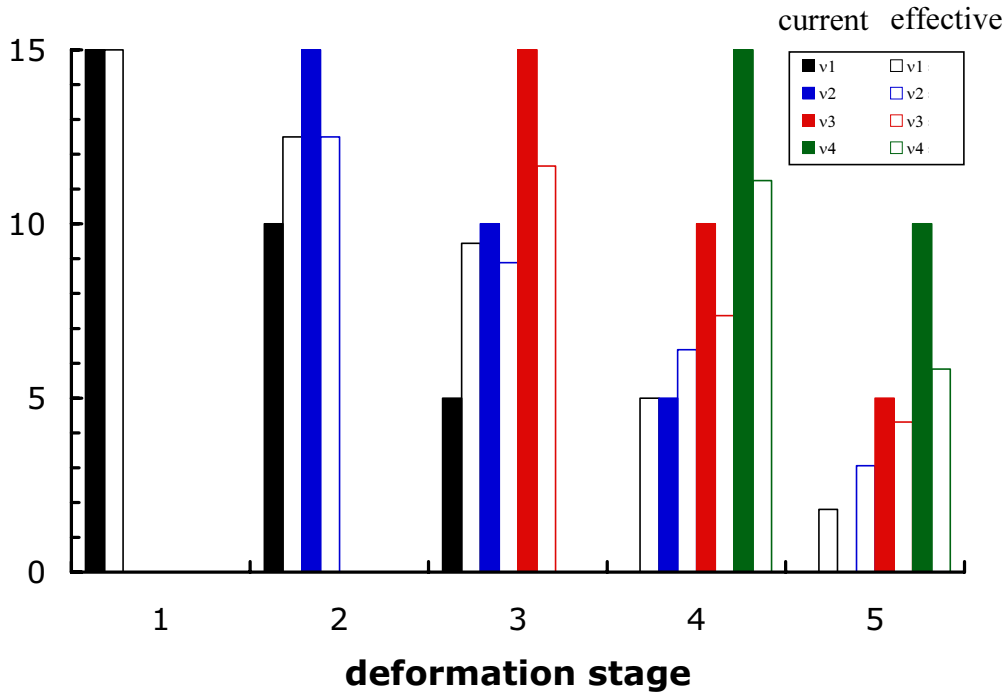


Figure 4. Comparing effective chain density to current chain density for five stage deformation as given in Table 1.
Chain density is given in consistent arbitrary units.

Several important points about using the stress transfer function as defined in Equation 3 and used with Equation 5 can be made. First, this stress transfer function conserves the total number of crosslinks. The effective chain density sums to the same value as the total chain density (compare the last columns of Tables 1a and 1d).

Second, at no point do the effective chain densities for a particular network exceed their original values, although, by design, the effective chain densities often exceed the current values. Figure 4 compares the current chain density for each network with its effective value at each deformation stage. Crossover from current density larger than effective density to effective density larger can be observed for the stage two network at deformation stages three and four.

Third, while the early stage crosslinks become less important to the overall state-of-ease as later stage crosslinks are added, the memory of the initial stage remains past the point at which all the initial crosslinks are removed. As can be seen in Figure 4, networks formed in stages one and two have no remaining chains by deformation stage five.

However, their effective chain densities, although small, are non-zero. For the permanent set, this memory will have a noticeable effect.

Fourth, the formulation we developed allows for arbitrary reaction kinetics. Crosslinking and scission do not need to occur at equal rates. Physically, situations involving rubbers exist where scission occurs with little crosslinking, converting an originally elastic material to a weak, low-modulus material. Crosslinking much faster than scission also occurs, especially in materials that have excess curing agent, leading to hardening and large permanent sets in short times. For illustrative purposes, the thought experiment described here had a simple deformation and reaction history to provide insight. Nothing intrinsic to the calculation of the stress transfer function requires such simple kinetics. All that is required are reasonable estimates of the crosslinking and scission rates during the deformation time history. This chemical reaction kinetic information is typically obtained experimentally. In the future, molecular dynamics simulation with reactive force fields could also be useful for estimating reaction kinetics.

2.2. Using the Stress Transfer Function in a Stress Calculation

Rubbers fall into the class of materials that can have the stress calculated from knowledge of the free energy. In this case, the stress tensor ($\underline{\underline{\sigma}}$) can be calculated from

$$\underline{\underline{\sigma}} = \left[\frac{2}{V} \sum_{i=1}^{n} \underline{\underline{A}}_i \frac{\partial F_i}{\partial \underline{\underline{A}}_i^T \underline{\underline{A}}_i} \underline{\underline{A}}_i^T \right] + p\underline{\underline{E}} \tag{8}$$

where V is the current volume, F is the free energy in terms of the right Cauchy-Green[7] strain measure $\underline{\underline{A}}^T \underline{\underline{A}}$, p is an indeterminate Langrangian multiplier that depends on the specific boundary conditions for a given problem, and $\underline{\underline{E}}$ is the unit matrix. Each network created at stage i has a deformation gradient tensor

$$\underline{\underline{A}}_i = \underline{\underline{\lambda}} \left(\underline{\underline{\lambda}}_i^{-1} \right) \tag{9}$$

with $\underline{\underline{\lambda}}_i$ denoting the deformation gradient tensor at the state of ease for network i, and $\underline{\underline{\lambda}}$, the deformation gradient tensor relative to the initial state.

To use the effective chain densities in a stress calculation, some model of the free energy must be assumed. The elastic contribution to the free energy is a point of much contention. Multiple strain energy density functions (free energy per volume) have been proposed. Flory originally used an affine model[5]; however, no theoretical barrier exists to using other functions provided they are consistent with the independent network

hypothesis. In this work, we have chosen the simple affine model for testing purposes. Other nonaffine elastic strain energy models that also incorporate entanglement effects are being considered, but are not yet implemented and will be one of the priorities of the Level 2 ASC milestone in FY06.

The free energy (F_i) given by the affine model for a network formed at stage i can be written as

$$F_i = \frac{RTN_i}{2} Tr\left[\underline{\underline{A}}_i^T \underline{\underline{A}}_i - \underline{\underline{E}}\right] \tag{10}$$

where R is the gas constant, T is the temperature, and N_i is the number of chains attributed to the network formed at stage i.

The stress transfer function from Equation 5 can be used in conjunction with Equation 10 and Equation 8 to calculate stress for a given material with a particular deformation and reaction history. This affine model has been implemented into the finite element code Adagio under the SIERRA framework. Specifically, the stress at each element is calculated from

$$\underline{\underline{\sigma}}^{elem} = \underline{\underline{\sigma}}^{affine} - \frac{1}{3}\left[Tr\underline{\underline{\sigma}}^{affine}\right]\underline{\underline{E}} + \sum_{i=1}^{n} \frac{\nu_i^{eff}}{\nu} \kappa \ln\left(\frac{\det \underline{\underline{\lambda}}}{\det \underline{\underline{\lambda}}_{i-1}}\right)\underline{\underline{E}} \tag{11}$$

$$\underline{\underline{\sigma}}^{affine} = \frac{RT}{\det \underline{\underline{\lambda}}} \sum_{i=1}^{n} \nu_i \underline{\underline{A}}_i \underline{\underline{A}}_i^T \tag{12}$$

where ν_i is the number of chains formed in stage i per initial, unstrained volume, ν is the current total number of chains per initial unstrained volume, and κ is the bulk modulus of the material. Separating the stress in this manner allows the indeterminate multiplier to be eliminated and yet includes the slight compressibility of these materials.

Using the simple affine model for the strain energy function, an incompressible material undergoing a uniaxial extension has a true stress of

$$\sigma = RT \sum_{k=1}^{n} \nu_k^{eff} \left(\left(\frac{\lambda}{\lambda_k}\right)^2 - \frac{\lambda_k}{\lambda}\right) \tag{13}$$

where λ is the stretch ratio in the extended direction.

3. PRELIMINARY RESULTS FROM FINITE ELEMENT CALCULATIONS

As a preliminary check, a cube of eight elements has been modeled with an edge length of ten centimeters. A Poisson's ratio of 0.499 and a Young's modulus of 1.0 x 106 Pa were used giving a bulk modulus of 16.7 MPa. These values were chosen from handbook values[8] for butyl rubber. In order to verify that the constitutive model has been implemented correctly, we first compared numerical results for uniaxial and biaxial deformations where analytical solutions could be obtained. Sample input decks with the mesh are given in Appendices A and B. For testing purposes, highly idealized systems were used with chain densities specified as linear or even step functions of time. This allows testing of the finite element code to compare with simple analytic solutions. Comparisons with more realistic systems from simulation and experimental results are planned.

Consider the strain and reaction history shown in Figure 5. A cube starts with the first stage network having a value of 0.34 MPa for $\nu_1 RT$ and, in the period of half a day, is stretched to twice its original length in the z-direction. A second stage network with $\nu_2 RT$= 0.34 MPa is added while the sample is held at the stretch ratio λ_2=2. Then, the first stage crosslinks are removed with a linear time dependence while the sample is maintained at λ_2=2. Figure 6 compares the stress versus time from the finite element calculation and the analytic solution. Results are shown for calculations that include the stress transfer function and those that have a pure independent network assumption with no stress transfer function. The finite element results match the corresponding analytic results within 1.5%. As required for the independent network, the stage two network has zero stress when formed. As the first stage chains break, the stress becomes smaller. Without stress transfer, the stress decays to zero as the first stage chains break. With stress transfer, the stress remains larger and reaches half its maximum value upon removal of all the first stage chains. This result is expected because equal numbers of chains were formed in both stages.

The equibiaxial system follows the same time schedule for adding new networks and the same chain densities were used; however, the y and z directions were deformed to stretch ratios of 1.35. A comparison of finite element results and analytic solution both with and without stress transfer is shown in Figure 7. Again, the finite element solution matches the analytic solution well (within 5%). As seen with the uniaxial elongation, the stress decays more rapidly without a stress transfer than with stress transfer active. The ending stress values are zero for the independent network model corresponding to no stress transfer and half of the maximum value when stress transfer is included in the calculation.

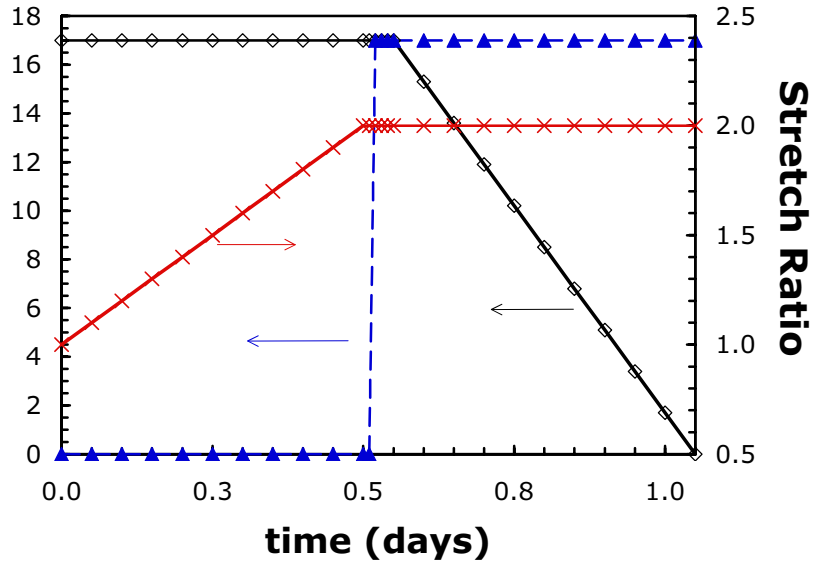


Figure 5. Uniaxial extension for a two-stage network deformation and reaction history.

Shown are ν_1RT (black diamonds), ν_2RT (blue triangles), and z-direction stretch ratio (red x's).

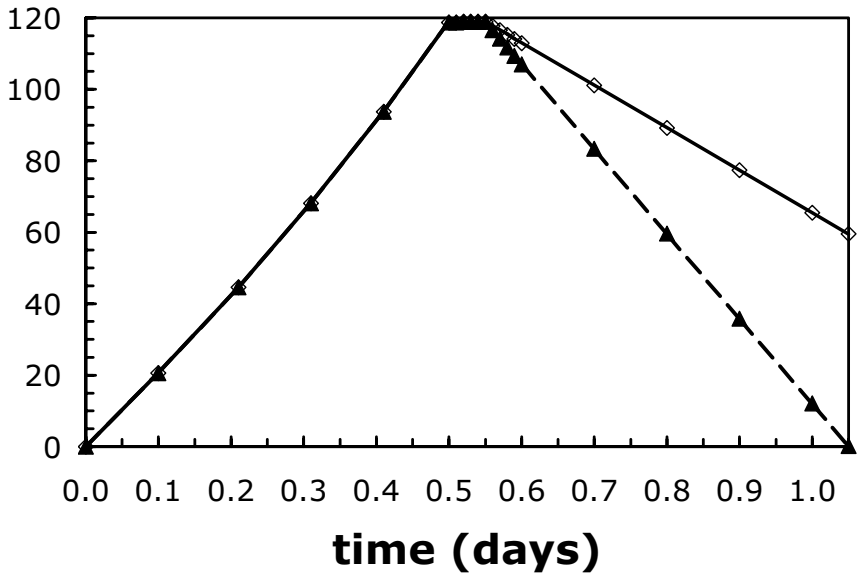


Figure 6. Stress versus time for a uniaxial two-stage network.

Finite element results are diamonds for active stress transfer and triangles for no stress transfer. Analytic results are solid line with active stress transfer and dashed line with no stress transfer.

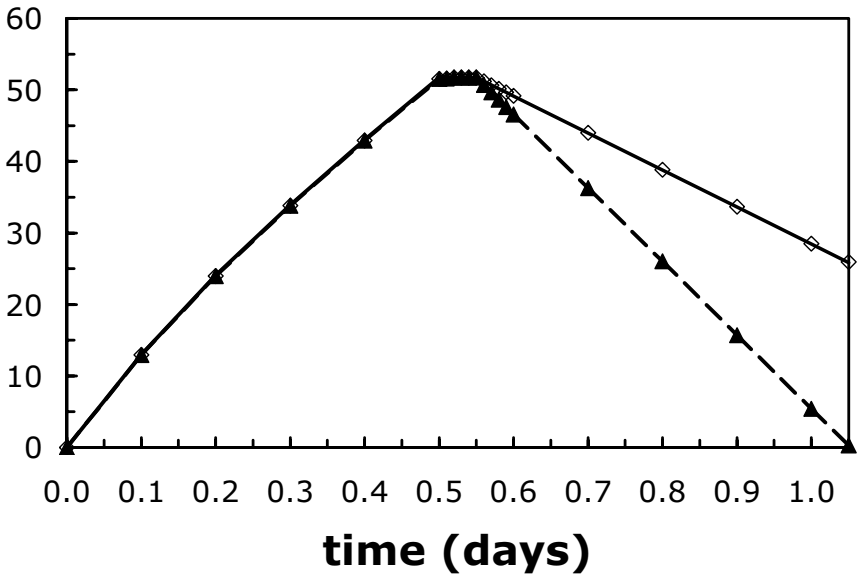


Figure 7. Stress versus time for a biaxial stretch with a two-stage network.

Finite element results are diamonds with stress transfer and triangles for no stress transfer. Analytic results have a solid line for active stress transfer and a dashed line for no stress transfer.

Uniaxial extensions were used to study the effects of adding more stages, breaking both first and second stage chains, and allowing crosslinking and scission to have different rate constants. A three-stage system is shown in Figure 8. The deformation time schedule is the same as shown in Figure 5, however, the stage two network is formed at time = 0.25 days (λ_2=1.5) as a step function with a ν_2RT value of 0.17 MPa. The stage three network forms at time = 0.52 days (λ_3=2.0) with ν_3RT of 0.17 MPa. Starting at time = 0.55, stage one chains break causing the stress to decay until, at time = 1.05, no stage one chains remain. Again, the finite element results for stress match the analytic values (within 1%). However, now with no stage one chains remaining in the independent network model with no stress transfer included, the stress remains non-zero because the stage two chains are not in their state of ease. Including the stress transfer functions in the calculations leads to a significantly higher stress. Although the total number of crosslinks after time 0.55 is the same for both the two-stage and three-stage uniaxial extension, the result of adding half the later stage crosslinks at λ=1.5 and half at λ=2.0 is a significantly higher stress than adding all the later stage crosslinks at λ=2.0. This comparison is shown in Figure 9.

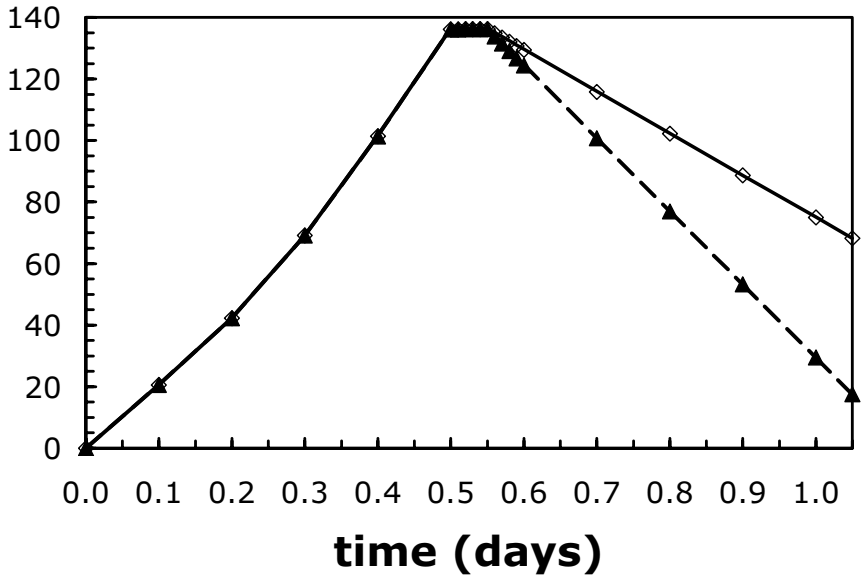


Figure 8. Stress versus time for a uniaxial stretch of a three-stage network

(stage one at time = 0 days, stage two at time = 0.25 days, and stage three at time = 0.52 days). Finite element results are diamonds with stress transfer and triangle for no stress transfer. Analytic results have a solid line for active stress transfer and a dashed line for no stress transfer.

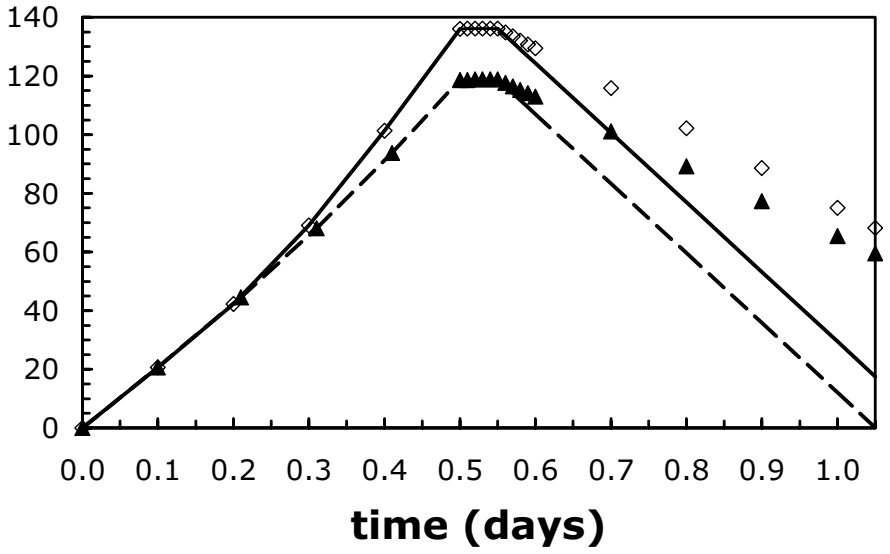


Figure 9. Comparing the finite element results of three-stage and two-stage uniaxial extension.

The three-stage system is diamonds with stress transfer active and a solid line for no stress transfer. The two-stage system has triangles with stress transfer active and a dashed line for no stress transfer.

Of course, in physical systems, the more common case is to have scission and crosslinking occurring concurrently rather than consecutively. The effect of this was investigated using the uniaxial extension with two stages of crosslinking. Five cases were investigated. All had the same starting deformation history with $\nu_1 RT$=0.34 MPa at time = 0 and stretching the sample until λ_2=2.0 at time =0.50 days. Descriptions of the five cases are given in the following list:

Case 1: This is the original situation with all the stage two chains formed at time = 0.52 days. The scission of stage one chains begins at time = 0.55 days and reaches $\nu_1 RT$ = 0 MPa at time = 1.05.

Case 2: As in Case 1, the second stage chains are formed at time = 0.52 days and scission begins starting at time = 0.55 days. However, now both the first and second stage chains scission at the same rate until they reach half their original νRT values at time = 1.05.

Case 3: No stage two chains form until time = 0.55. Then, second stage chains form at the same rate that the first stage chains disappear so that at time = 1.05 $\nu_1 RT$ = 0 MPa and $\nu_2 RT$ = 0.34 MPa.

Case 4: This case is the same situation as Case 3 except that the stage two chains are formed at twice the rate that the stage one chains are removed. Thus, $\nu_2 RT$ increases from zero to 0.68 MPa as $\nu_1 RT$ decreases from 0.34 to zero MPa during the time between 0.55 and 1.05 days.

Case 5: This case is the reverse of Case 4, with the scission faster than the crosslinking so that $\nu_2 RT$ increases from zero to 0.17 MPa during the time that $\nu_1 RT$ decreases from 0.34 to zero MPa.

Figure 10 shows the results of these five cases. As one would expect, for all cases in which the $\nu_1 RT$ decreases from 0.34 to zero MPa with no stress transfer, the resulting stress as a function of time is the same. The nonpreferential breaking of chains but at the same total crosslink density result (Case 2) with no stress transfer is the same as Case 1 with stress transfer active. This result is exactly what is expected because of the equal values of stage one and stage two crosslink densities and the definition of the stress transfer function.

Activating the stress transfer functions does give different results for the stress as a function of time. As one would expect, at short times, scission at equal rates by both fully formed networks (Case 2) has the highest stress. However, near the end of the calculation Case 4 (faster crosslinking than scission) has a slightly higher stress than Case 2. Faster scission (Case 5) has the smallest stress, but it is still larger than the results without stress transfer and the difference becomes greater as the calculation time becomes longer. At time = 1.05 days, Case 5 with active stress transfer has a stress of 40 x 10^4 Pa while without stress transfer, the stress is zero. However, Case 4 (faster crosslinking) has an ending value of 80 x 10^4 Pa.

The simple examples considered here clearly demonstrate that the constitutive model that accounts for coupling between crosslinks formed at different stages gives substantially different results than the independent network model with no stress transfer included.

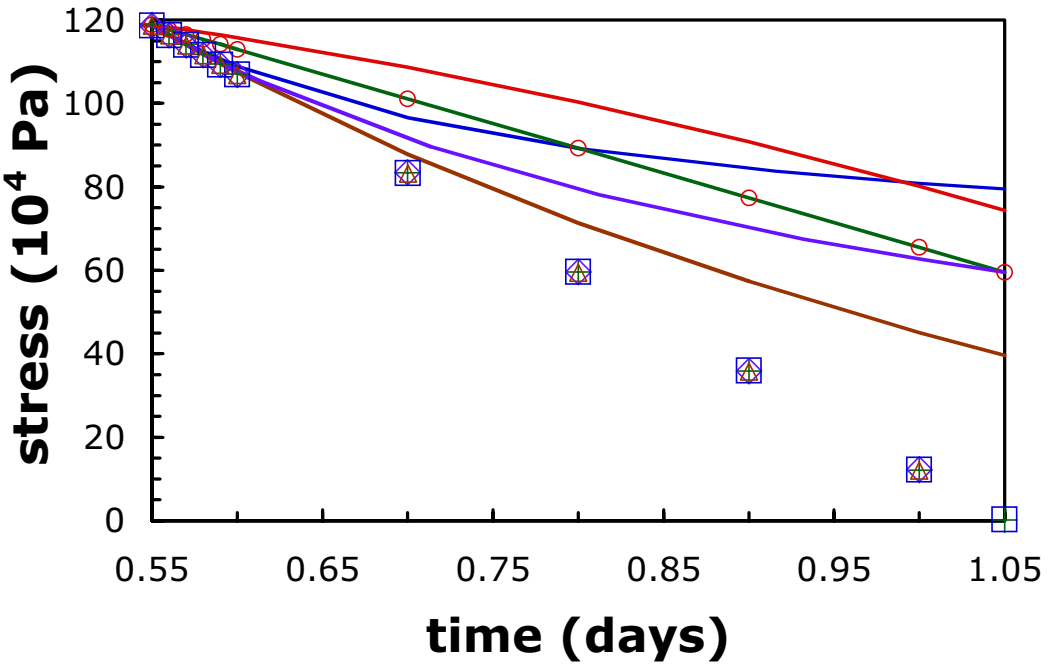


Figure 10. Comparing stress from the finite element calculation as a function of relative scission and crosslinking rates.

Lines have active stress transfer, points indicate calculations for purely independent networks, and colors are consistent for the reaction history. All stage two chains formed prior to scission (Case 1: green pluses), scission from both stages at equal rates (Case 2: red circles), equal rates crosslinking and scission (Case 3: purple triangles), faster crosslinking (Case 4: blue squares), and faster scission (Case 5: brown triangles).

.

4. CONCLUSIONS AND FUTURE WORK

The examples shown in this report are greatly simplified models to investigate the effect of the stress transfer function. Additional complexities must be implemented to obtain results relevant to physical systems. In order to do this, we plan to:

1) Add the effect of entanglements.

One dramatic oversimplification is the neglect of the effect of trapped entanglements. Long polymer chains tend to be jumbled together so that loops and knots form. Crosslinking may prevent loops from disentangling. These trapped entanglements may contribute to the stress in a similar manner that crosslinks do. Molecular dynamics simulations are being performed to investigate what effect these trapped entanglements have on the stress and what contribution they may make to the stress transfer function. The resulting information can then be incorporated into the finite element code.

2) Add additional strain energy functions.

The affine model is used primarily for its simplicity, not its accuracy. More accurate, nonaffine theories have been developed that include the effects of entanglements on junction fluctuations[9] and chain fluctuations[10, 11] and the result of these fluctuations on the constitutive behavior. The Ogden strain energy form[12] is a phenomenological strain energy that has, as special cases, neo-Hookean behavior, Mooney-Rivlin expressions, as well as some of the modern theories that incorporate entanglement effects.

3) Use geometries more relevant to the o-ring problem including compressed states and o-ring segments.

The results shown here are all for cubes of material, most in a state of uniaxial extension. For o-rings, the later stage crosslinking and scission primarily occur in compressed states. Molecular dynamics simulations are also being performed in compression.

4) Use reaction and scission rates that are more realistic.

Chemical reactions rarely proceed linearly with time. In general, we will examine more realistic crosslinking and scission kinetics based on available experimental data. Ultimately, we hope to gain further insights into crosslinking and scission from oxidative degradation by MD simulations using reactive force fields[13] as implemented in the GRASP computer code.

With the incorporation of the above four points, we can make comparisons with experimental results from literature and with the rubber aging model that was previously implemented in the finite element code JAS3D.

5. REFERENCES

1) Curro, J. G. and Salazar, E. A. Rubber Chem Technol. 46 (2), 530 (1973).

2) Salazar, E.A., Curro, J.G., and Gillen, K.T. J. Appl. Polym. Sci. 21, 1597 (1977).

3) Andrews, R. D., Tobolsky, A. V., and Hanson, E. E. J. Appl. Phys. 17, 352 (1946).

4) Rottach, D. R., Curro, J. G., Grest, G. S., and Thompson, A. P. Macromolecules 37, 5468 (2004).

5) Flory, P. J. Trans. Faraday Soc. 56, 722 (1960).

6) Fricker, H.S. Proc. Royal Soc. London. Series A, Math. Phys. Sci. 335, (1602), 267 (1973).

7) Ogden, R. W. Non-linear Elastic Deformations Dover Publications, Inc. Mineola, N.Y. (1997). p. 85.

8) Brandrup, J., Immergut, E.H., and Grulke, E.A. Polymer Handbook 4th ed. Wiley-Interscience, New York (1999) pp. V/4-V/5.

9) Flory, P. J. and Erman, B. Macromolecules 15, 800 (1982)

10) Mergell, B. and Everaers, R. Macromolecules 34, 5675 (2001).

11) Rubinstein, M. and Panyukov, S. Macromolecules 35, 6670 (2003).

12) Ogden, R. W. Non-linear Elastic Deformations Dover Publications, Inc. Mineola, N.Y. (1997). p. 221.

13) van Duin, A. C. T., Dasgupta, S., Lorant, F., and Goddard, W. A. J. Phys. Chem. A 105 (41), 9396 (2001)

APPENDIX A: SAMPLE ADAGIO INPUT DECKS.

A.1. Uniaxial extension, two stages, active stress transfer function

```
begin sierra blocky
        begin title Flory_model: uniaxial extension, with phi
        end title


###------------------------###
### Function definitions###
###------------------------###

  begin definition for function stretch
    type is piecewise linear
    begin values
          0.00      0.000
          0.50      10.0
          1.05      10.0
    end values
  end definition for function stretch

  begin definition for function time_step
    type is piecewise constant
    begin values
          0.0       1.0e-3
          0.01      0.01
          0.1       0.1
          0.40      0.01
          0.50      0.01
          0.55      1.0e-04
          0.551     0.001
          0.56      0.01
          0.97      0.001
          1.02      0.001
          1.038     1.0e-04
    end values
  end definition for function time_step
```

```
begin definition for function zero
  type is constant
  begin values
              0.0
  end values
end definition for function zero

begin definition for function c1func
  type is piecewise linear
  ordinate is crosslinkdensity
  abscissa is time
  begin values
              0.0       17.0
              0.55      17.0
              1.05      0.0
  end values
end definition for function c1func

begin definition for function c2func
  type is piecewise constant
  ordinate is crosslinkdensity
  abscissa is time
  begin values
              0.0       0.0
              0.50      0.0
              0.52      17.0
              1.05      17.0
  end values
end definition for function c2func

begin definition for function c3func
  type is piecewise linear
  ordinate is crosslinkdensity
  abscissa is time
  begin values
              0.0       0.0
              0.5       0.0
              0.52      0.0
              1.05      0.0
  end values
end definition for function c3func
```

```
  begin definition for function sfunc
    type is piecewise constant
    ordinate is numberofrefstates
    abscissa is time
    begin values
                0.0         1.0
                0.50        2.0
    end values
  end definition for function sfunc

  begin definition for function tiny
    type is piecewise constant
    ordinate is tolfunction
    abscissa is time
    begin values
                0.0         1.0e-06
                365.0       1.0e-06
    end values
  end definition for function tiny

  define direction x with vector 1.0 0.0 0.0
  define direction y with vector 0.0 1.0 0.0
  define direction z with vector 0.0 0.0 1.0


###-------------------------###
### Material Properties ###
###-------------------------###

  begin property specification for material rPIB
#density units are g/cm^3
    density = 0.933
    begin parameters for model flory
#units for Ym and crosslink density prefactor is 10^4 Pa #(i.e. Ey = 1 MPa)
youngs modulus  = 1.0E2
poissons ratio = 0.499
#cross x function is 0.5νxRT
                cross 1 function = c1func
                cross 2 function = c2func
                cross 3 function = c3func
                cross 4 function = c3func
                cross 5 function = c3func
#state function indicates when new states of ease are set
                state function = sfunc
#phi onoff toggles the stress transfer functions.  1 is active and 0 is inactive.
                phi onoff = 1
```

```
#To avoid numerical inaccuracies, a check is made on ln(det(lambda)/det(lambdaref)).
#For the state of ease, the value should be zero.  Tol function indicates what value is
#near enough to zero.  If the material were truly incompressible, this would not be an
#issue.
                tol function = tiny
#Multiple strain energies will be available soon.  Currently, only the affine model is
#implemented as type 1
                strain energy type = 1
   end parameters for model flory
  end property specification for material rPIB

  begin solid section solid_1
   strain incrementation = strongly_objective
  end solid section solid_1

  begin finite element model mesh1
   Database Name = manyedges.g
   Database Type = exodusII
   begin parameters for block block_1
material rPIB
solid mechanics use model flory
                section = solid_1
   end parameters for block block_1
  end finite element model mesh1


###-------------------------###
### Adagio Procedures ###
###-------------------------###

  begin adagio procedure Agio_Procedure
   begin time control
begin time stepping block p1
  start time =  0.0
  begin parameters for adagio region adagio
   time increment function = time_step
  end parameters for adagio region adagio
end time stepping block p1

begin time stepping block p2
  start time =  1.03
  begin parameters for adagio region adagio
   time increment function = time_step
  end parameters for adagio region adagio
end time stepping block p2
```

```
begin time stepping block p3
  start time =  1.035
  begin parameters for adagio region adagio
    time increment function = time_step
  end parameters for adagio region adagio
                end time stepping block p3
                termination time = 1.05
    end time control

    begin adagio region adagio
use finite element model mesh1


###----------------------###
### Output definition ###
###----------------------###

  begin Results Output output_adagio
                Database Name = manyedges.e
                Database Type = exodusII
                At time 0.0 increment = 0.1
                At time 0.50 increment = 0.01
                At time 0.60 increment = 0.1
                nodal Variables = displacement as displ
                element Variables = rotated_stress as stress
                element variables = stretch as stretch
                global Variables = timestep as timestep
  end results output output_adagio


###-----------------###
### Restart Data ###
###----------------###

  begin restart data newstart
                Output Database Name = newrestart.e
                Input Database Name = restartinitial.e
                Database Type = exodusII
                at time 0.0 increment =0.01
                Termination Time = 1.05
  end restart data newstart
```

```
###--------------------###
### BC descriptions###
###--------------------###

                begin fixed displacement
                  node set = nodelist_4
                  components = Z
                end fixed displacement

                begin fixed displacement
                  node set = nodelist_32
                  components = y
                end fixed displacement

                begin fixed displacement
                  node set = nodelist_42
                  components = y
                end fixed displacement

                begin fixed displacement
                  node set = nodelist_31
                  components = x
                end fixed displacement

                begin fixed displacement
                  node set = nodelist_41
                  components = x
                end fixed displacement

                begin prescribed displacement
                  node set = nodelist_3
                  direction = z
                  function = stretch
                  scale factor = 1.0
                end prescribed displacement
```

```
### ---------------------###
### Solver definition ###
### ---------------------###

                Loadstep predictor using line search type secant
                begin adagio solver cg
                  target relative residual = 1.0e-05 during p1
                  target residual = 3.1e-03 during p2
                  target residual = 4.0e-03 during p3
                  acceptable relative residual = 1.0e-05
                  Maximum Iterations = 5000
                  Orthogonality measure for reset = 0.5
                  Line Search type secant
                  preconditioner = elastic
                end adagio solver cg

    end adagio region adagio
  end adagio procedure Agio_Procedure
end sierra blocky
```

A.2. Biaxial extension, two stages, active stress transfer function

```
begin sierra blocky
  begin title Flory_model: biaxial extension, active phi
  end title


###-------------------------###
### Function definitions###
###-------------------------###

  begin definition for function stretch
    type is piecewise linear
    begin values
                0.00        0.000
                0.50         3.5
                0.55         3.5
                1.05         3.5
    end values
  end definition for function stretch

  begin definition for function zero
    type is constant
    begin values
                0.0
    end values
  end definition for function zero

  begin definition for function c1func
    type is piecewise linear
    ordinate is crosslinkdensity
    abscissa is time
    begin values
                0.0         17.0
                0.50        17.0
                0.55        17.0
                1.05        0.0
    end values
  end definition for function c1func
```

```
begin definition for function c2func
  type is piecewise constant
  ordinate is crosslinkdensity
  abscissa is time
  begin values
              0.0       0.0
              0.50      0.0
              0.52      17.0
  end values
end definition for function c2func

begin definition for function c3func
  type is piecewise linear
  ordinate is crosslinkdensity
  abscissa is time
  begin values
              0.0       0.0
              0.5       0.0
              0.52      0.0
              1.05      0.0
  end values
end definition for function c3func

begin definition for function sfunc
  type is piecewise constant
  ordinate is numberofrefstates
  abscissa is time
  begin values
              0.0       1.0
              0.50      2.0
  end values
end definition for function sfunc

begin definition for function tiny
  type is piecewise constant
  ordinate is tolfunction
  abscissa is time
  begin values
              0.0       1.0e-06
              0.52      1.0e-06
  end values
end definition for function tiny

define direction x with vector 1.0 0.0 0.0
define direction y with vector 0.0 1.0 0.0
define direction z with vector 0.0 0.0 1.0
```

```
###------------------------###
### Material Properties ###
###------------------------###

begin property specification for material rPIB
#density units are g/cm^3
   density = 0.933
   begin parameters for model flory
#units for Ym and crosslink density prefactor is 10^4 Pa #(i.e. Ey = 1 MPa)
youngs modulus  = 1.0E2
poissons ratio = 0.499
#cross x function is 0.5νxRT
                cross 1 function = c1func
                cross 2 function = c2func
                cross 3 function = c3func
                cross 4 function = c3func
                cross 5 function = c3func
#state function indicates when new states of ease are set
                state function = sfunc
#phi onoff toggles the stress transfer functions.  1 is active and 0 is inactive.
                phi onoff = 1
#To avoid numerical inaccuracies, a check is made on ln(det(lambda)/det(lambdaref)).
#For the state of ease, the value should be zero.  Tol function indicates what value is
#near enough to zero.  If the material were truly incompressible, this would not be an
#issue.
                tol function = tiny
#Multiple strain energies will be available soon.  Currently, only the affine model is
#implemented as type 1
                strain energy type = 1
   end parameters for model flory
  end property specification for material rPIB

  begin solid section solid_1
   strain incrementation = strongly_objective
  end solid section solid_1

  begin finite element model mesh1
   Database Name = manyedges.g
   Database Type = exodusII
   begin parameters for block block_1
                material rPIB
                solid mechanics use model flory
                section = solid_1
   end parameters for block block_1
  end finite element model mesh1
```

```
###------------------------###
### Adagio Procedures ###
###------------------------###

  begin adagio procedure Agio_Procedure
    begin time control

                begin time stepping block p1
                  start time =  0.0
                  begin parameters for adagio region adagio
                    time increment = 1.0e-03
                  end parameters for adagio region adagio
                end time stepping block p1

                begin time stepping block p1a
                  start time =  0.01
                  begin parameters for adagio region adagio
                    time increment = 0.01
                  end parameters for adagio region adagio
                end time stepping block p1a

                begin time stepping block p1b
                  start time =  0.1
                  begin parameters for adagio region adagio
                    time increment = 0.01
                  end parameters for adagio region adagio
                end time stepping block p1b

                begin time stepping block p2
                  start time =  0.5
                  begin parameters for adagio region adagio
                    time increment = 0.01
                  end parameters for adagio region adagio
                end time stepping block p2

                begin time stepping block p3
                  start time =  0.55
                  begin parameters for adagio region adagio
                    time increment = 1.0e-04
                  end parameters for adagio region adagio
                end time stepping block p3
```

```
    begin time stepping block p4
      start time =  0.551
      begin parameters for adagio region adagio
        time increment = 0.001
      end parameters for adagio region adagio
    end time stepping block p4

    begin time stepping block p5
      start time =  0.56
      begin parameters for adagio region adagio
        time increment = 0.01
      end parameters for adagio region adagio
    end time stepping block p5

    begin time stepping block p6
      start time =  0.97
      begin parameters for adagio region adagio
        time increment = 0.001
      end parameters for adagio region adagio
    end time stepping block p6

    begin time stepping block p7
      start time =  1.035
      begin parameters for adagio region adagio
       time increment = 0.001
      end parameters for adagio region adagio
    end time stepping block p7
    termination time = 1.05
end time control
```

```
  begin adagio region adagio
            use finite element model mesh1

###---------------------###
### Output definition ###
###---------------------###

 begin Results Output output_adagio
              Database Name = manyedges.e
              Database Type = exodusII
              At time 0.0 increment = 0.1
              At time 0.50 increment = 0.01
              At time 0.60 increment = 0.1
              nodal Variables = displacement as displ
              element Variables = rotated_stress as stress
              element variables = stretch as stretch
              global Variables = timestep as timestep
 end results output output_adagio

###-----------------###
### Restart Data ###
###---------------###

 begin restart data newstart
              Output Database Name = newrestart.e
              Input Database Name = restartsecondcomplete.e
              Database Type = exodusII
              at time 0.0 increment =0.01
              at time 0.55 increment = 0.01
              Termination Time = 1.05
 end restart data newstart

###-------------------###
### BC descriptions###
###-------------------###

            begin fixed displacement
              node set = nodelist_4
              components = Z
            end fixed displacement

            begin fixed displacement
              node set = nodelist_32
              components = y
            end fixed displacement
```

```
begin fixed displacement
  node set = nodelist_42
  components = y
end fixed displacement

begin fixed displacement
  node set = nodelist_31
  components = x
end fixed displacement

begin fixed displacement
  node set = nodelist_41
  components = x
end fixed displacement

begin prescribed displacement
  node set = nodelist_3
  direction = z
  function = stretch
  scale factor = 1.0
end prescribed displacement

begin fixed displacement
  node set = nodelist_5
  components = y
end fixed displacement

begin fixed displacement
  node set = nodelist_52
  components = z
end fixed displacement

begin fixed displacement
  node set = nodelist_22
  components = z
end fixed displacement

begin fixed displacement
  node set = nodelist_51
  components = x
end fixed displacement

begin fixed displacement
  node set = nodelist_21
  components = x
end fixed displacement
```

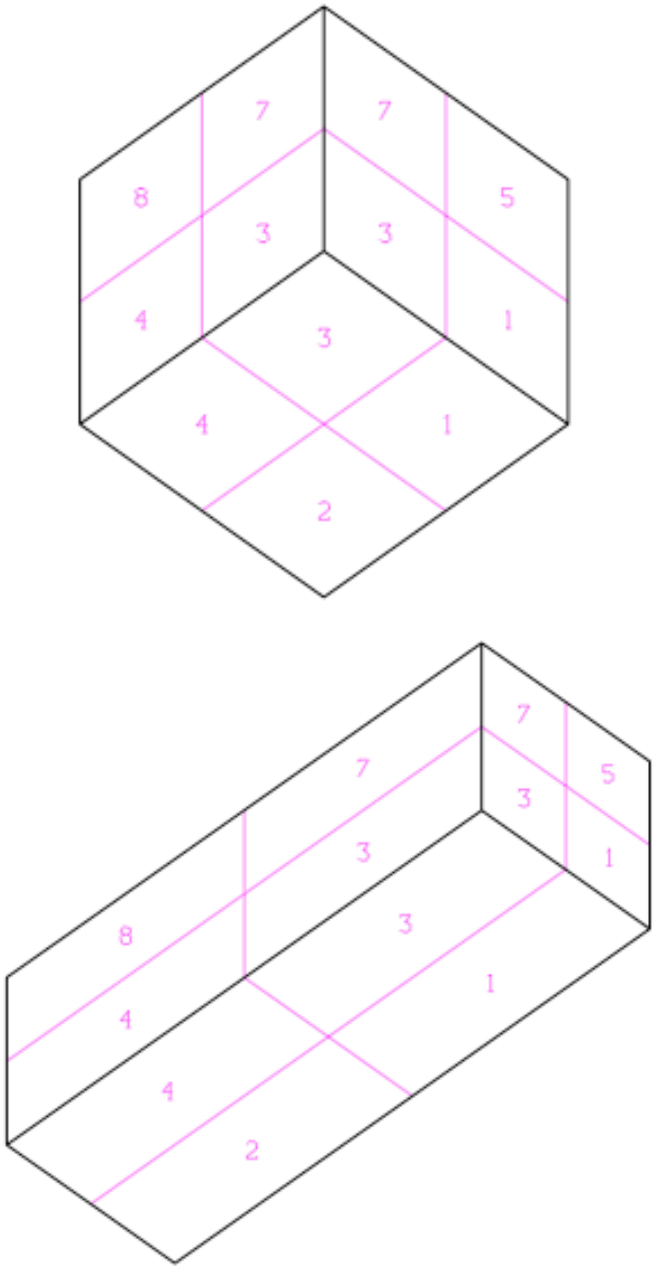
```
                begin prescribed displacement
                  node set = nodelist_2
                  direction = y
                  function = stretch
                  scale factor = 1.0
                end prescribed displacement


### ---------------------###
### Solver definition ###
### ---------------------###

                Loadstep predictor using line search type secant
                Begin adagio solver cg
                  target relative residual = 1.0e-04
                  Maximum Iterations = 5000
                  Minimum Iterations = 0
                  Orthogonality measure for reset = 0.5
                  Line Search type secant
                  preconditioner = elastic
                end adagio solver cg
    end adagio region adagio
  end adagio procedure Agio_Procedure
end sierra blocky
```

APPENDIX B: MESH FOR EXAMPLES.

Figure 11. Picture of the eight-element block
(initially and for λ=2.0 in uniaxial extension).



Mesh file for the eight-element block

```
! Database initial variables
       3     3.01            ! dimensions, version number
      27        8        1    ! nodes, elements, element blocks
      18        0             ! #node sets, #side sets
      90       90             ! len: node set list, dist fact length
       0        0        0    ! side sets len: element, node , dist fact
! Coordinate names
x                         y                         z
! Coordinates
 -5.0000000E+00  -5.0000000E+00   5.0000000E+00
 -5.0000000E+00  -5.0000000E+00   0.0000000E+00
 -5.0000000E+00   0.0000000E+00   0.0000000E+00
 -5.0000000E+00   0.0000000E+00   5.0000000E+00
  0.0000000E+00  -5.0000000E+00   5.0000000E+00
  0.0000000E+00  -5.0000000E+00   0.0000000E+00
  0.0000000E+00   0.0000000E+00   0.0000000E+00
  0.0000000E+00   0.0000000E+00   5.0000000E+00
 -5.0000000E+00  -5.0000000E+00  -5.0000000E+00
 -5.0000000E+00   0.0000000E+00  -5.0000000E+00
  0.0000000E+00  -5.0000000E+00  -5.0000000E+00
  0.0000000E+00   0.0000000E+00  -5.0000000E+00
 -5.0000000E+00   5.0000000E+00   0.0000000E+00
 -5.0000000E+00   5.0000000E+00   5.0000000E+00
  0.0000000E+00   5.0000000E+00   0.0000000E+00
  0.0000000E+00   5.0000000E+00   5.0000000E+00
 -5.0000000E+00   5.0000000E+00  -5.0000000E+00
  0.0000000E+00   5.0000000E+00  -5.0000000E+00
  5.0000000E+00  -5.0000000E+00   5.0000000E+00
  5.0000000E+00  -5.0000000E+00   0.0000000E+00
  5.0000000E+00   0.0000000E+00   0.0000000E+00
  5.0000000E+00   0.0000000E+00   5.0000000E+00
  5.0000000E+00  -5.0000000E+00  -5.0000000E+00
  5.0000000E+00   0.0000000E+00  -5.0000000E+00
  5.0000000E+00   5.0000000E+00   0.0000000E+00
  5.0000000E+00   5.0000000E+00   5.0000000E+00
  5.0000000E+00   5.0000000E+00  -5.0000000E+00
```

```
! Node number map
explicit map
       6       19       23        7        8       21       27        9
      13       16       14       18       22        4       25        5
      15       17        1       20       26        3       11       12
      24        2       10
! Element number map
sequence 1..numel
! Element order map
sequence 1..numel
! Element block    1
       1        8     HEX8      ! ID, elements, name
       8        0     ! nodes per element, attributes
! Connectivity
       1        2        3        4        5        6        7        8
       2        9       10        3        6       11       12        7
       4        3       13       14        8        7       15       16
       3       10       17       13        7       12       18       15
       5        6        7        8       19       20       21       22
       6       11       12        7       20       23       24       21
       8        7       15       16       22       21       25       26
       7       12       18       15       21       24       27       25
! Node sets                       18
! Len node set node list          90
! Len node set dist fact list     90
! Nodal point set         1
       1        9        9     ! ID, nodes, dist factors
      20     1.0000000E+00
      24     1.0000000E+00
      25     1.0000000E+00
      22     1.0000000E+00
      19     1.0000000E+00
      23     1.0000000E+00
      27     1.0000000E+00
      26     1.0000000E+00
      21     1.0000000E+00
```

```
! Nodal point set         2
         2         9         9     ! ID, nodes, dist factors
        25     1.0000000E+00
        18     1.0000000E+00
        13     1.0000000E+00
        16     1.0000000E+00
        26     1.0000000E+00
        27     1.0000000E+00
        17     1.0000000E+00
        14     1.0000000E+00
        15     1.0000000E+00
! Nodal point set         3
         3         9         9     ! ID, nodes, dist factors
        22     1.0000000E+00
        16     1.0000000E+00
         4     1.0000000E+00
         5     1.0000000E+00
        19     1.0000000E+00
        26     1.0000000E+00
        14     1.0000000E+00
         1     1.0000000E+00
         8     1.0000000E+00
! Nodal point set         4
         4         9         9     ! ID, nodes, dist factors
        24     1.0000000E+00
        11     1.0000000E+00
        10     1.0000000E+00
        18     1.0000000E+00
        27     1.0000000E+00
        23     1.0000000E+00
         9     1.0000000E+00
        17     1.0000000E+00
        12     1.0000000E+00
! Nodal point set         5
         5         9         9     ! ID, nodes, dist factors
         2     1.0000000E+00
        11     1.0000000E+00
        20     1.0000000E+00
         5     1.0000000E+00
         1     1.0000000E+00
         9     1.0000000E+00
        23     1.0000000E+00
        19     1.0000000E+00
         6     1.0000000E+00
```

```
! Nodal point set        6
       6        9        9     ! ID, nodes, dist factors
      13    1.0000000E+00
      10    1.0000000E+00
       2    1.0000000E+00
       4    1.0000000E+00
      14    1.0000000E+00
      17    1.0000000E+00
       9    1.0000000E+00
       1    1.0000000E+00
       3    1.0000000E+00
! Nodal point set        7
      11        3        3     ! ID, nodes, dist factors
      19    1.0000000E+00
      20    1.0000000E+00
      23    1.0000000E+00
! Nodal point set        8
      12        3        3     ! ID, nodes, dist factors
      27    1.0000000E+00
      24    1.0000000E+00
      23    1.0000000E+00
! Nodal point set        9
      21        3        3     ! ID, nodes, dist factors
      26    1.0000000E+00
      25    1.0000000E+00
      27    1.0000000E+00
! Nodal point set       10
      22        3        3     ! ID, nodes, dist factors
      17    1.0000000E+00
      18    1.0000000E+00
      27    1.0000000E+00
! Nodal point set       11
      31        3        3     ! ID, nodes, dist factors
      19    1.0000000E+00
      22    1.0000000E+00
      26    1.0000000E+00
! Nodal point set       12
      32        3        3     ! ID, nodes, dist factors
       1    1.0000000E+00
       5    1.0000000E+00
      19    1.0000000E+00
```

```
! Nodal point set        13
       41         3         3      ! ID, nodes, dist factors
       27     1.0000000E+00
       24     1.0000000E+00
       23     1.0000000E+00
! Nodal point set        14
       42         3         3      ! ID, nodes, dist factors
       23     1.0000000E+00
       11     1.0000000E+00
        9     1.0000000E+00
! Nodal point set        15
       51         3         3      ! ID, nodes, dist factors
       19     1.0000000E+00
       20     1.0000000E+00
       23     1.0000000E+00
! Nodal point set        16
       52         3         3      ! ID, nodes, dist factors
       23     1.0000000E+00
       11     1.0000000E+00
        9     1.0000000E+00
! Nodal point set        17
       61         3         3      ! ID, nodes, dist factors
        1     1.0000000E+00
        2     1.0000000E+00
        9     1.0000000E+00
! Nodal point set        18
       62         3         3      ! ID, nodes, dist factors
        9     1.0000000E+00
       10     1.0000000E+00
       17     1.0000000E+00
! Properties
        1           ! Number of ELEMENT BLOCK Properties
! Property Name:
ID
! Property Value(s):
        1
        1           ! Number of NODE SET Properties
! Property Name:
ID
! Property Value(s):
        1         2         3         4         5         6        11
       12        21        22        31        32        41        42
       51        52        61        62
        0           ! Number of SIDE SET Properties
```

```
! QA Records
        2       ! QA records
CUBIT
9.2b
07/25/2005
13:29:07
exo2txt
 1.16
20050815
15:56:37
! Information Records
        0       ! information records
! Variable names
        0       0       0       ! global, nodal, element variables
```

APPENDIX C: MATERIAL MODEL INPUT FUNCTIONS

Keyword	Argument Type	Summary
Cross X Function	time-dependent function	The current stage X cross-link density multiplied by RT. Should be in a pressure unit (e.g. kPa). Five stages should be given even if some are zero.
State Function	time-dependent function	The current number of reference states based on deformation
Phi Onoff	integer	This command controls if the stress transfer function is active. A value of 0 indicates use no stress transfer function. A value of 1 is use stress transfer functions.
Tol Function	time-dependent function	A numerical accuracy value to determine if the current state is close enough to be a state of ease. The comparison is $\ln\left[\frac{\det \underline{\underline{\lambda}}}{\det \underline{\underline{\lambda}}_{i-1}}\right] > tol$. False indicates a state of ease and the bulk modulus term is zero. True indicates a different state, which includes a bulk modulus term in the stress.
Strain Energy Type	integer	Integer flag for the type of strain energy to use in calculating the stress. Currently, only the affine strain energy is available and it is marked as 1.

DISTRIBUTION

1	MS0139	Paul Yarrington	1902
1	MS0139	Robert K. Thomas	1904
1	MS0321	Jennifer E. Nelson	1430
1	MS0557	Daniel Segalman	1524
1	MS0847	Peter J. Wilson	1520
1	MS0885	Justine Johannes	1810
1	MS0885	Richard Salzbrenner	1820
1	MS0887	Duane Dimos	1800
1	MS0888	Doug Adolf	1821
1	MS0888	Robert Bernstein	1821
1	MS0888	Roger Clough	1821
1	MS0893	Robert Chambers	1523
5	MS0893	Chi S. Lo	1523
1	MS0893	John Pott	1523
1	MS1110	John Aidun	1435
1	MS1110	Aidan Thompson	1435
5	MS1349	John Curro	1815
1	MS1349	Dana Rottach	1815
1	MS1411	Roger Assink	1821
5	MS1411	Joanne Budzien	1814
1	MS1411	Mathias Celina	1821
1	MS1411	H. Eliot Fang	1814
1	MS1411	Ken Gillen	1821
1	MS1415	Carlos Gutierrez	1114
1	MS1415	Gary Grest	1114
1	MS9042	Davina M. Kwon	8774
1	MS9042	Vera Revelli	8774
1	MS9403	Linda Domeier	8762
2	MS9018	Central Technical Files	8945-1
2	MS0899	Technical Library	4536


Sandia National Laboratories