Finite Element Simulation and Analysis of Local Stress Concentration in Polymers with a Nonlinear Viscoelastic Constitutive Model

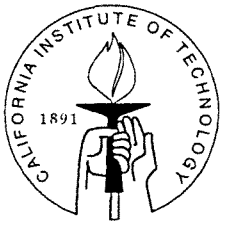
Thesis by

Limdara O. Chea

In Partial Fulfillment of the Requirements

for the Degree of

Aeronautical Engineer



California Institute of Technology

Pasadena, California

1997

(Submitted October 4th, 1996)

To a very special person, Mandy,

for all her help, patience and support.

Acknowledgements

I would like to express my gratitude to Professor Wolfgang G. Knauss for his guidance and advice throughout this project. His willingness to teach me viscoelasticity, and allowing me to pursue this work in this motivating matter, is greatly appreciated.

Some friends taught me more than what I could ever expect from classes. I would like to thank in particular Dr. Alfons Noe, for offering his precious advice that helped me greatly in writing my thesis and preparing my defense, and Demirkan Coker, my very good friend 'Iyi Arkadash Em', for being so available and complementing me with his strong experience in the field of Solid Mechanics.

I think both this dissertation, and I personally, have been enriched as a result of their wisdom. I would also like to thank the fantastic proofreading GCD team: Giorgio Isella, Claude Seywert and, especially, Demirkan Coker.

Abstract

Given a nonlinear viscoelastic (NLVE) constitutive model for a polymer, this numerical study aims at simulating local stress concentrations in a boundary value problem with a corner stress singularity. A rectangular sample of Polyvinyl Acetate (PVAc)-like cross-linked polymer clamped by two metallic rigid grips and subjected to a compression and tension load is numerically simulated.

A modified version of the finite element code FEAP, that incorporated a NLVE model based on the free volume theory, was used. First, the program was validated by comparing numerical and analytical results. Two simple mechanical tests (a uniaxial and a simple shear test) were performed on a Standard Linear Solid material model, using a linear viscoelastic (LVE) constitutive model. The LVE model was obtained by setting the proportionality coefficient δ to zero in the free volume theory equations. Second, the LVE model was used on the corner singularity boundary value problem for three material models with different bulk relaxation functions $K(t)$. The time-dependent stress field distribution was investigated using two sets of plots: the stress distribution contour plots and the stress time curves. Third, using the NLVE constitutive model, compression and tension cases were compared using the stress results (normal stress σ_{yy} and shear stress σ_{xy}). These two cases assessed the effect of the creep retardation-creep acceleration phenomena.

The shift between the beginning of the relaxation moduli was shown to play an important role. This parameter affects strongly the fluctuation pattern of the stress curves. For two different shift values, in one case, the stress response presents a 'double peak' and 'stress inversion' characteristic whereas, in the other case, it presents a 'single peak' and no 'inversion'. Another important factor was the material's compressibility. In the case of a nearly-incompressible material, the LVE and NLVE models yielded identical results; thus, the simpler LVE model is preferable.

However, in the case of sufficient volume dilatation (or contraction), the NLVE model predicted correct characteristic responses, whereas LVE results were erroneous. This proves the necessity of using the NLVE model over the LVE model.

Contents

List of Figures

Introduction

Given a nonlinear viscoelastic constitutive model for a polymer, this study aims at simulating a boundary value problem in which some inhomogeneous deformations, local stress concentrations occur. A corner singularity is likely to generate such stress concentration. For this purpose, the choice was made to consider the boundary value problem of a rectangular sample of polymer (Polyvinyl Acetate or PVAc) clamped by two metallic (Aluminium) rigid grips subjected to a compression and tension load. The idea was that this geometry might lead to the formation of shear bands within the polymer and consequently to their analyses.

For the numerical analyses, a finite element code, that incorporated the nonlinear viscoelastic (NLVE) model based on the free volume theory, was used. A finite element code based on the core program called FEAP had been developed and revised by successive Caltech graduate students. Unfortunately, no documentation existed for the program yet. The FE code FEAP is now treated in more detail in the Appendix.

First, we checked the code with two simple mechanical tests: a uniaxial compression test and a simple shear test. The boundary conditions were taken to be a step load, then a step boundary displacement in order to check respectively the strain creep behavior and the stress relaxation behavior. For comparison between analytical and numerical results, a simple standard solid material model associated with a linear viscoelastic (LVE) model was investigated.

Linear viscoelasticity was implemented in the finite element code by setting the coefficient δ to zero in the free volume equation. This way, the time shift factor is no more dependent on the local variations of volume. It becomes a homogeneous field distribution set to a trivial unit value 1. Consequently, the time shift factor was disabled in the program, which enabled a more straightforward comparison with the theoretical values.

Second, this LVE code was then used on the boundary value problem that in-

cluded a corner stress singularity. For further validation, the long-term stress field, that is an elastic solution, was double-checked with a commercial finite element software, ABAQUS. The time-dependent behavior that is between the instantaneous and the long-term responses is strictly characteristic to viscoelastic materials. This was investigated with a 'snapshot' (a contour plot) of the stress distribution at a middle time and with curves of the stresses and strains with respect to time.

Third, the time shift factor was then reintroduced in the code to investigate the full shift factor effect of nonlinear viscoelastic theory. With the NLVE code, a compression and tension case were simulated and their result were compared. Compression and tension affect in an opposite way the local variation of volume. And that local variation of volume commands the change in free volume. This way, we can study the effects of creep retardation and creep acceleration in, respectively, the compression and tension cases.

In the next chapter, the theory and the viscoelastic constitutive equations will be reviewed. Also the implementation of the boundary value problem in FEAP is summarized. In the third chapter, the material model characterization of three different materials is explained. In the fourth chapter, the results in terms of the normal stress σ_{yy} , shear stress σ_{xy} and normal strain ϵ_{yy} behavior are summarized.

Chapter 1 Theory and Methods

1.1 Constitutive Theory

The nonlinear viscoelastic constitutive theory based on the free volume model is briefly summarized. For more details, refer to the papers: Knauss and Emri (1987), Losi and Knauss (1992a). G. Losi extended the free volume model to temperatures below the glass transition, taking into account the fact that the instantaneous free volume only achieves a metastable equilibrium state below the glass transition temperature. That extended free volume theory is implemented in this thesis.

1.1.1 Stress-strain relations

A viscoelastic material stress-strain relation is of the form of a convolution product between the material's relaxation functions $K(t)$, $G(t)$ and the strain incremental history $d\epsilon_{ij}(t)$. The Cauchy stress is:

$$\begin{aligned}\sigma_{ij}(t) &= \int_{-\infty}^{t} 2G\left(\xi(t)-\xi(\tau)\right)\frac{\partial \epsilon_{ij}^{dev}}{\partial \tau}(\tau)d\tau + \delta_{ij}\int_{-\infty}^{t} K\left(\xi(t)-\xi(\tau)\right)\frac{\partial \epsilon_{kk}}{\partial \tau}(\tau)d\tau \\ &\quad -\delta_{ij}\hat{K}_{\infty}\alpha_{l_v}\triangle T(t) \qquad (1.1)\\ \xi(t) &= \int_{0}^{t}\frac{d\tau}{a_T} \qquad (1.2)\end{aligned}$$

The shear and bulk relaxation functions $G(t)$ and $K(t)$ are scaled in time by an 'internal' time function $\xi(t)$. In the most simple case of a thermo-rheologically simple viscoelastic material submitted to isothermal loading, $\xi(t)$ is just scaled by a constant coefficient a_T (or ψ) that is a function of temperature T: $\xi(t) = \frac{t}{a_T}$.

In an extension of this model, a_T becomes a function not only of the temperature history but also of the hydrostatic stresses history and the solvent concentration history (Ferry and Stratton, 1960; Knauss and Emri, 1987).

Coefficient a_T is called a time shift factor, because if a plot of σ_{ij} versus time is generated in logarithmic time, it appears as if the curve had been shifted in time by $\log(a_T)$.

1.1.2 Free volume theory extended to metastable equilibrium states

In the case of nonlinear viscoelasticity (NLVE), the time shift factor ψ is a function of the local (fractional) free volume f. The local free volume varies with all the applied conditions. The basic assumption is that the variation of the (local) free volume, df, is proportional to the variation of the (local) macroscopic volume, $d\epsilon_{kk}$. The proportionality coefficient δ is a function of the free volume. Basically, $\delta(f)$ is close to zero for small f and close to $\frac{\beta_{f_0}}{1+\beta_{f_0}}$ for large f, where β_{f_0} is the ratio between free and occupied volume changes above the glass transition.

Thus, below a certain value of f, any decrease in the macroscopic dilatation due to pressure or temperature change will not give a corresponding decrease in free volume. The polymer is, in this case, in the frozen state with a constant residual free volume.

The free volume theory equations are:

$$\xi(t) = \int_0^t \frac{d\tau}{\psi(\tau)} \tag{1.3}$$

$$\log \psi(t) = B\left(\frac{1}{f(t)} - \frac{1}{f_{ref}}\right) \tag{1.4}$$

$$f(t) = f_{init} + \int_0^t \delta(f(\tau)) d\epsilon_{kk}(\tau) \tag{1.5}$$

$$\delta(f) = \frac{\beta_{f_0}\phi(f)}{1+\beta_{f_0}\phi(f)} \tag{1.6}$$

$$\phi(f) = \frac{1}{1-\frac{A}{2}\left[1-\frac{B}{f}\right]\exp\left(\frac{B}{f}-\frac{B}{f_{ref}}\right)} \tag{1.7}$$

$$K(t) = K_\infty + \sum_{p=1}^{M} K_p \exp(-t/\tau_p'') \tag{1.8}$$

$$G(t) = G_\infty + \sum_{p=1}^{N} G_p \exp(-t/\tau_p') \tag{1.9}$$

$$\hat{K}_\infty = K_\infty \frac{1+\beta_{f_0}}{1+\beta_{f_0}\phi(f)} \tag{1.10}$$

$$\beta_{f_0} = \frac{\Delta V_{free}}{\Delta V_{occupied}} = \frac{\alpha_{l_v}}{\alpha_{g_v}} - 1 - \sum_{p=1}^{M} \frac{K_p}{K_\infty} \tag{1.11}$$

In the above equations, B is a material parameter, $f(t)$ and f_{ref} are the fractional free volume at current condition (time) due to temperature and hydrostatic stress changes and at reference condition (temperature) corresponding to the state in which the relaxation moduli were measured. f_{init} is the initial free volume; ψ is the time shift factor, and β_{f_0} is the ratio between free and occupied volume changes above the glass transition. $K(t)$ and $G(t)$ are the time-dependent bulk and shear moduli measured at the reference free volume f_{ref}. α_{l_v} and α_{g_v} are the rubbery and glassy value of the volumetric thermal expansion coefficient, respectively. $\phi(f)$ takes into account the metastable equilibrium state of the free volume that exists below the glass transition temperature.

From the above equations, understand that the time functions $\psi(t)$ and $f(t)$ are also field functions. Their values are dependent on the spatial position $\vec{x}$. A better notation would be $\psi(t,\vec{x})$ and $f(t,\vec{x})$. Consequently, the internal time $\xi(t,\vec{x})$ also has a spatial distribution and takes different values at a given time at two different points.

Remark: Linear viscoelasticity

1. If the proportionality coefficient δ is set to zero, then

$$f(t) = f_{init} \tag{1.12}$$

$$\log\psi(t) = B\left(\frac{1}{f_{init}} - \frac{1}{f_{ref}}\right) \tag{1.13}$$

2. Furthermore, if f_{init} is set equal to f_{ref} (which means that T_{init} equals T_{ref}) and the temperature T is kept constant, then

$$\log\psi(t) = 0 \tag{1.14}$$

$$\xi(t) = t \tag{1.15}$$

$$\triangle T = 0 \tag{1.16}$$

There is no shift factor anymore and the internal time $\xi(t)$ is just the normal time t.

We get the linear viscoelastic stress-strain relations:

$$\sigma_{ij}(t) = \int_{-\infty}^{t} 2G\left(t-\tau\right)\frac{\partial \epsilon_{ij}^{dev}}{\partial \tau}(\tau)d\tau + \delta_{ij}\int_{-\infty}^{t} K\left(t-\tau\right)\frac{\partial \epsilon_{kk}}{\partial \tau}(\tau)d\tau \tag{1.17}$$

The precedent remarks will be the assumptions of the linear viscoelastic (LVE) constitutive model.

1.1.3 Volumetric and deviatoric decomposition

The choice of the shear modulus $G(t)$ and bulk modulus $G(t)$ to describe the material lead naturally to the decomposition into deviatoric and volumetric parts. The relations for the deviatoric stress σ_{ij}^{dev} and the volumetric stress $\bar{\sigma}$ are:

$$\sigma_{ij}^{dev}(t) = \int_{-\infty}^{t} 2G\left(\xi(t)-\xi(\tau)\right)\frac{\partial \epsilon_{ij}^{dev}}{\partial \tau}(\tau)d\tau \tag{1.18}$$

$$\bar{\sigma}(t) = \int_{-\infty}^{t} K\left(\xi(t)-\xi(\tau)\right)\frac{\partial \epsilon_{kk}}{\partial \tau}(\tau)d\tau - \hat{K}_{\infty}\alpha_{l_v}\triangle T(t) \tag{1.19}$$

1.2 Validation of the Code

The nonlinear constitutive model (NLVE) was implemented in the current version of FEAP by former Caltech Graduate students. Before running it on a complex boundary value problem such as the one of this study, it needed to be validated. It was decided to use the code on some simple problems and to compare the numerical results with the 'hand-calculated' theoretical values.

1.2.1 Using the linear viscoelastic model

To simplify the problem for validation, the time shift factor effect has been deactivated. The linear viscoelastic model was taken so that local variations of volume would not induce a shift factor. This was made possible by setting the proportionality coefficient δ to zero in the constitutive relations of the program (see Constitutive theory).

Also, no time shift factor due to temperature variation was desired. Therefore, the temperature history was taken as constant and equal to the reference temperature T_{ref} (the temperature at which $K(t)$ and $G(t)$ were measured). This was done by setting the initial free volume f_{init} equal to the reference free volume f_{ref}.

1.2.2 A simple material model

To simplify the analytical calculation, a standard linear solid model was chosen for the material's shear relaxation. The bulk relaxation modulus is taken to be constant.

$$G(t) = G_\infty + G_1 \exp(-t/\tau) \tag{1.20}$$

$$K(t) = K_\infty \tag{1.21}$$

A real solid (the polymer considered afterwards) can be seen as a linear superposition of several standard linear solids.

1.2.3 Two simple mechanical tests

Simple boundary conditions were chosen for the material to undergo homogeneous deformations. Therefore, in the specimen, the strain and stress field distribution are homogeneous. Two simple mechanical tests were simulated on the standard linear solid: a step uniaxial load and a step shear load (see fig. 1.1 and fig. 1.2). The numerical creep results (see fig. 1.3 and fig. 1.4) were compared to the theoretical results.

Then, taking the real solid (the polymer described in the material data section), a

uniaxial step load and a uniaxial step displacement test were simulated (see fig. 1.5 and fig. 1.6). The respective uniaxial creep and relaxation response had their initial and long-term values compared (see fig. 1.7 and fig. 1.10) to the elastic theory, because linear viscoelastic materials in their instantaneous and long-term behavior act like elastic materials. It has also been checked that the normal stress σ_{yy} in the step load case and the normal strain ϵ_{yy} in the step displacement case remained constant and had the correct applied value (see fig. 1.8 and fig. 1.9).

1.3 The Boundary Value Problem of the Studied Case

1.3.1 Geometry

For the FE mesh modelisation, two homogeneous materials are considered: one viscoelastic (polymer) and one elastic (aluminium metal). The top layer of metal is bonded to the polymer. A vertical load (compression or tension) is then applied on the metallic layer.

The metal can be considered as a rigid body with respect to the polymer. The metallic layer keeps the top polymer mesh points horizontally aligned and 'locks' their x-displacements (see fig. 1.11). The metallic layer is so stiff compared to the polymer that its deformation can be neglected. Thus, it provides the wanted tangential constraint on the boundary of the polymer. This way, the polymer sample has a shear stress singularity in its corner. The top rigid layer applies reactional shear stresses on the polymer's top boundary while the polymer's right boundary is traction free. The study will focus on analyzing the stress distribution behavior with respect to time and space.

For symmetry reason, one quarter of the rectangular specimen is sufficient to study the problem. The precedent boundary conditions were chosen because they were thought to represent best a realistic laboratory manipulation.

1.3.2 Mesh refinement

The mesh was refined several times to make sure that it handles correctly the singularity at the corner. After each refinement, the stress field was compared to the previous one and the changes were noted. After the 7^{th} refinement, the results appeared to be stable. That mesh is the one used in fig. 1.11.

1.3.3 Simulated experiment

The purpose of this research is to study some inhomogeneous stress distribution in a nonlinear viscoelastic material (polymer).

A rectangular sample of polymer (close to PVAc) clamped by two rigid metallic grips on the top and bottom sides is submitted to a compression and tension load. The rigid grips develop some boundary tangential constraints, which in turn create a singularity point at the corner. This singularity point is a source of stress concentration.

1.3.4 Validation

The first test to be simulated was for the polymeric material under compression given a linear viscoelastic elastic (LVE) constitutive model. The long term contour plot of σ_{yy} for the LVE material (see fig. 1.12) is compared to the one of an elastic material (see fig. 1.13), provided with the long term material constants of the viscoelastic material ($K_\infty = 10$ Pa, $G_\infty = 3.16$ Pa). The two contour plots superpose perfectly. Also, the same elastic material has been simulated under compression with a commercial software called ABAQUS. The stress field contour plot is in Fig. 1.14 and coincides with both previous results. The case is a further validation of the code. This is in accordance with the linear viscoelasticity theory. In its instantaneous and long term response, a viscoelastic material acts like an elastic material.

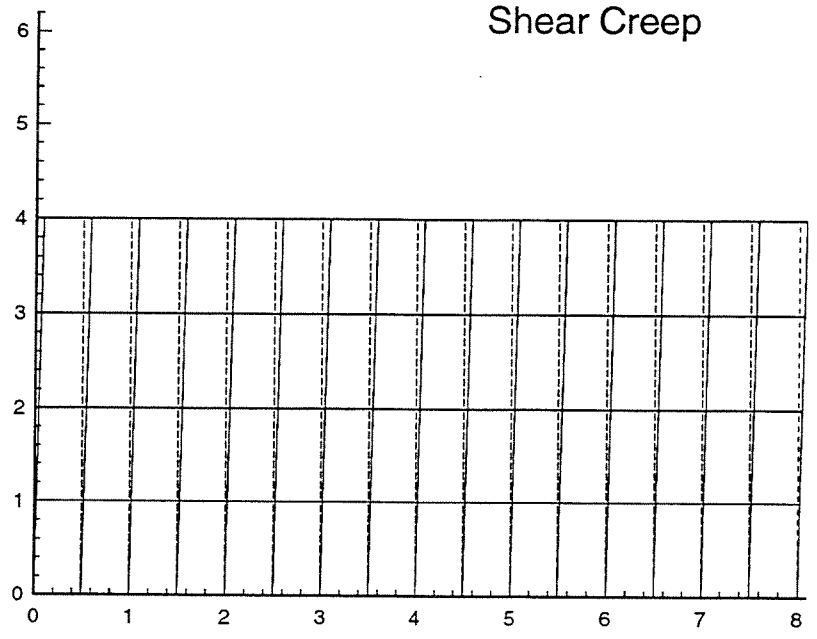


Figure 1.1: Long term deformation, simple shear, Standard Linear Solid

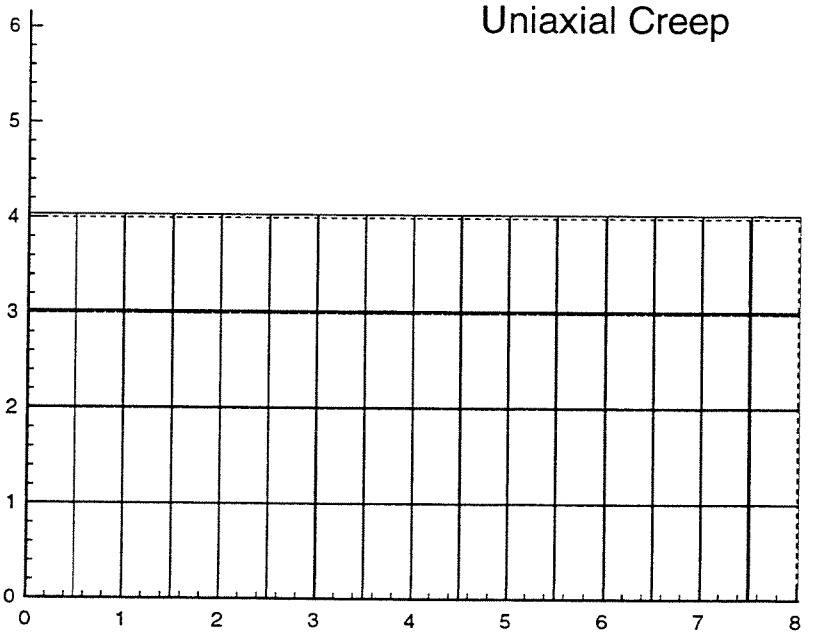


Figure 1.2: Long term deformation, uniaxial, Standard Linear Solid

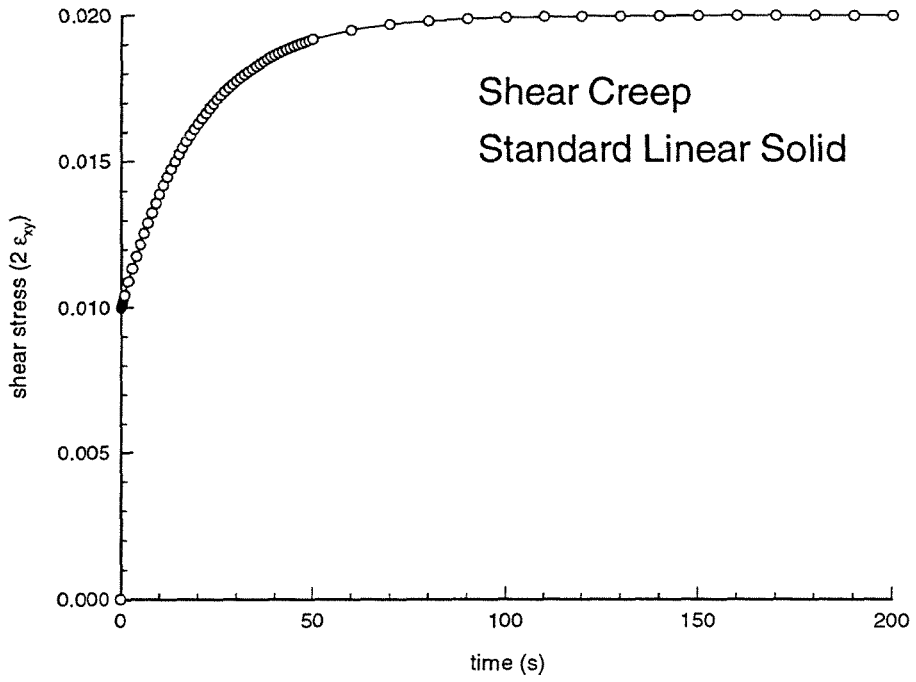


Figure 1.3: Shear stress $2\ \epsilon_{xy}(t)$, simple shear, Standard Linear Solid

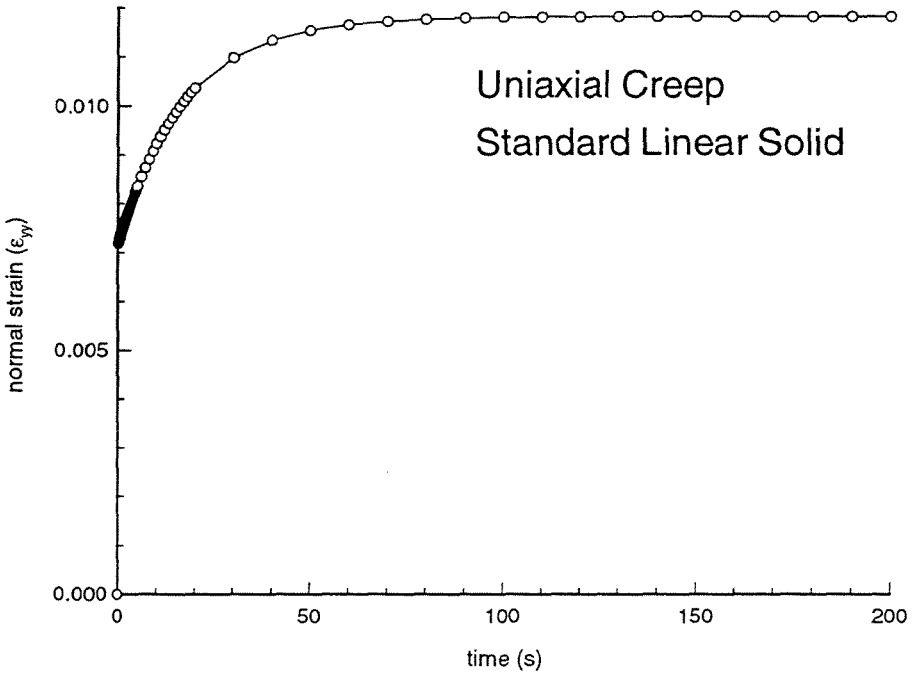


Figure 1.4: Normal stress $\epsilon_{yy}(t)$, uniaxial, Standard Linear Solid

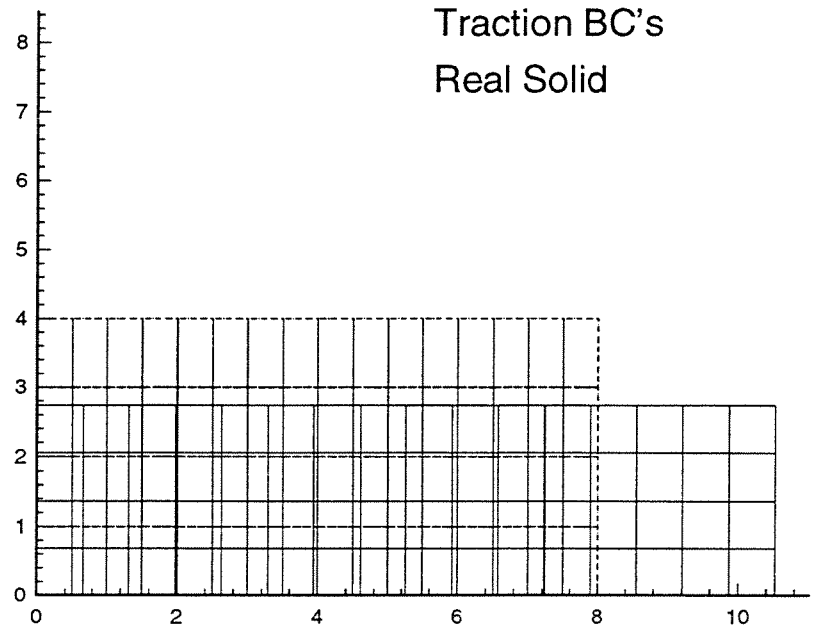


Figure 1.5: Initial and long term deformations, step load, Real Solid

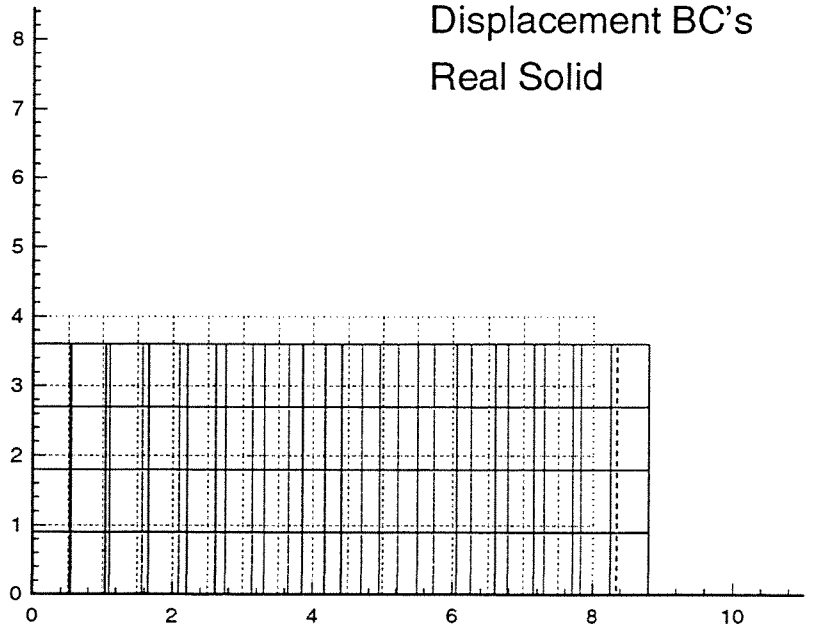


Figure 1.6: Initial and long term deformations, step displacement, Real Solid

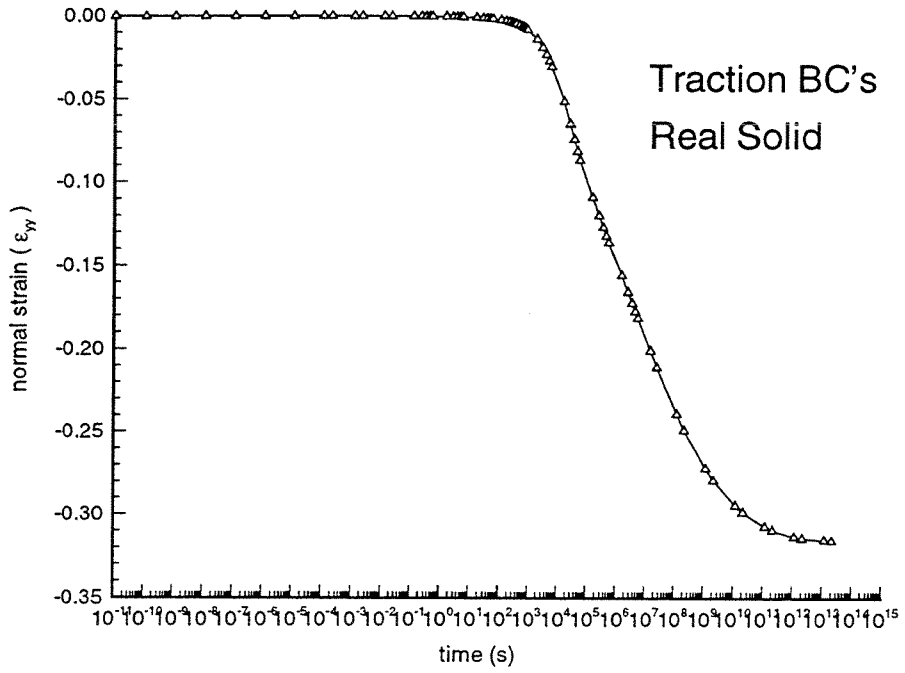


Figure 1.7: Normal strain $\epsilon_{yy}(t)$, step load, Real Solid

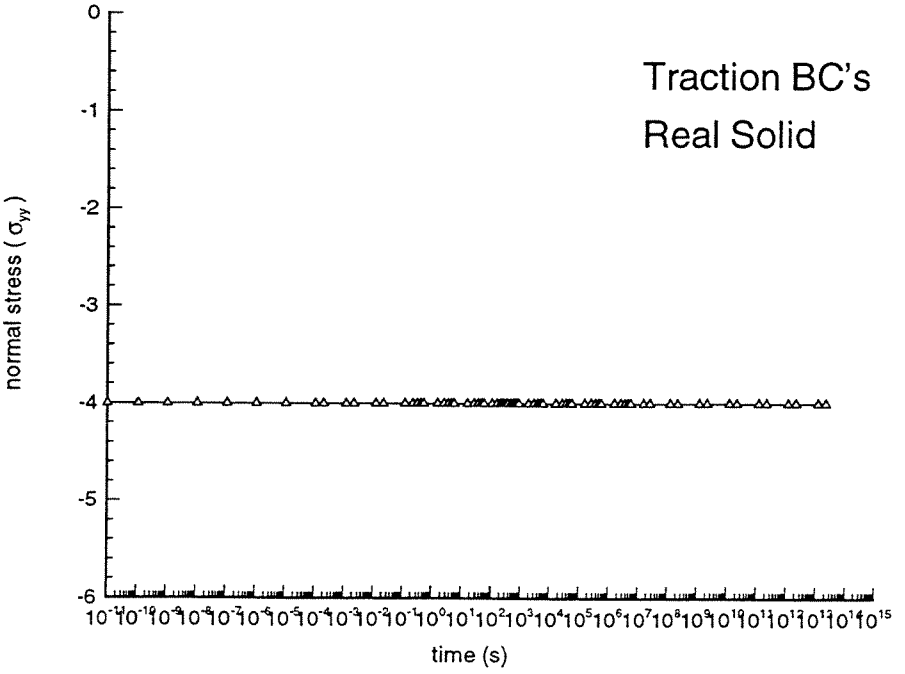


Figure 1.8: Normal stress $\sigma_{yy}(t)$, step load, Real Solid

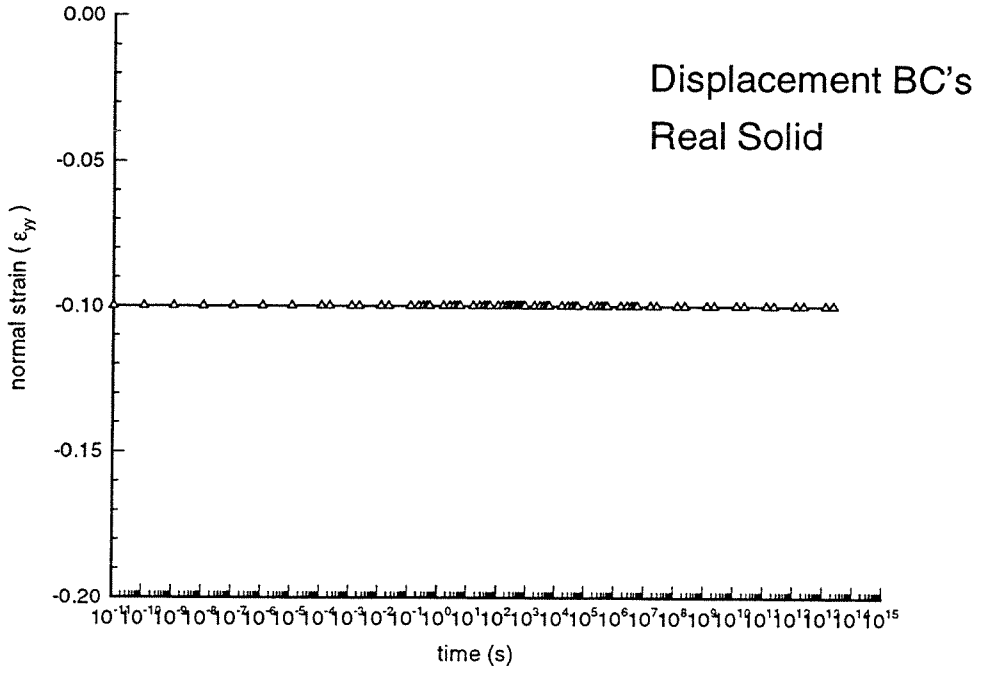


Figure 1.9: Normal strain $\epsilon_{yy}(t)$, step displacement, Real Solid

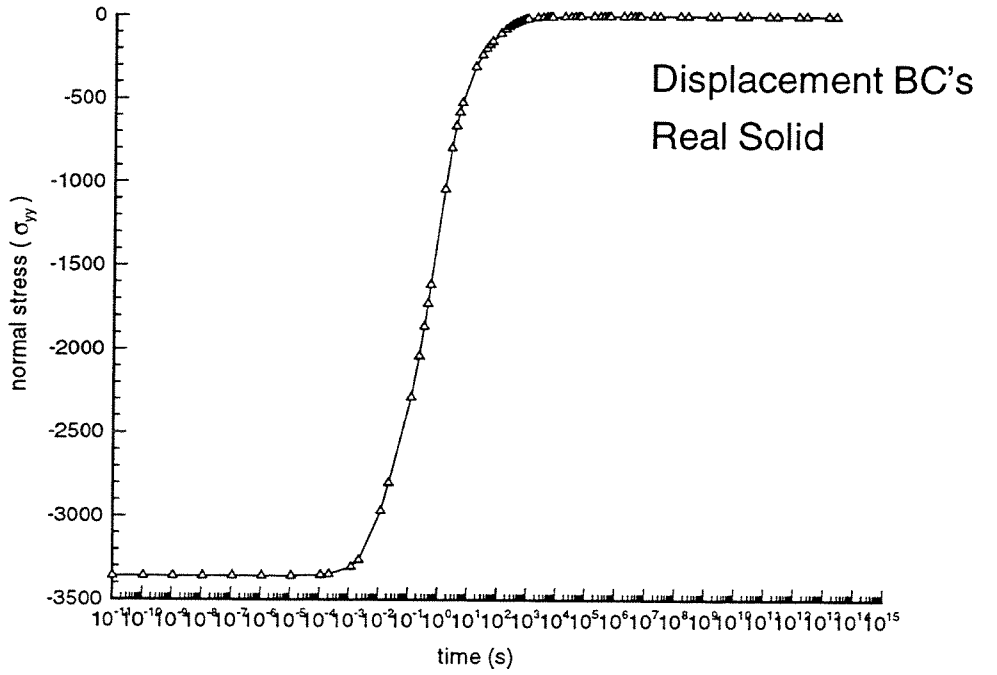


Figure 1.10: Normal strain $\sigma_{yy}(t)$, step displacement, Real Solid

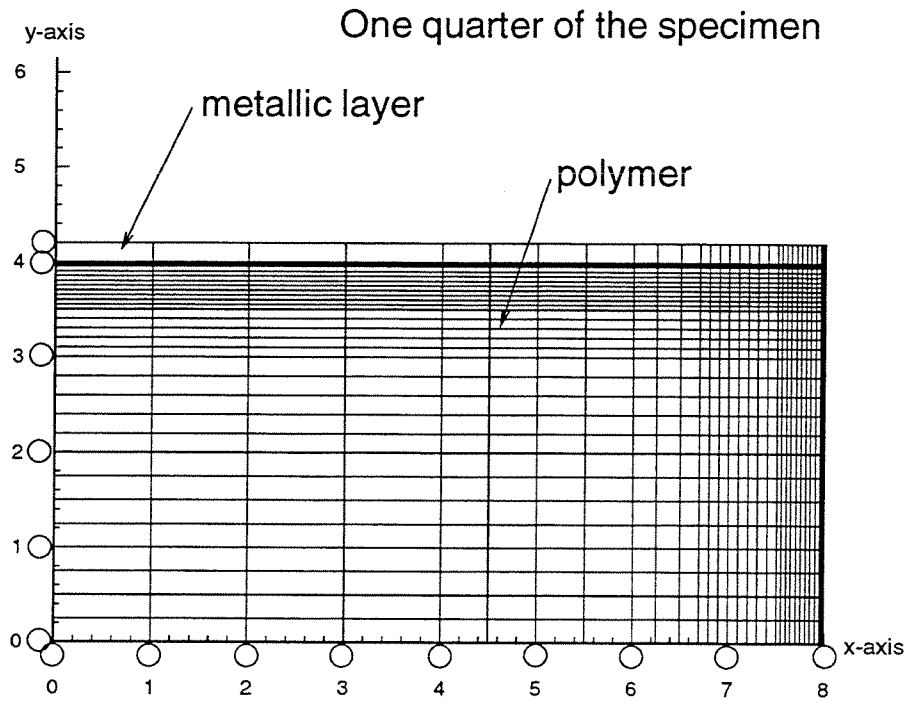


Figure 1.11: The boundary value problem

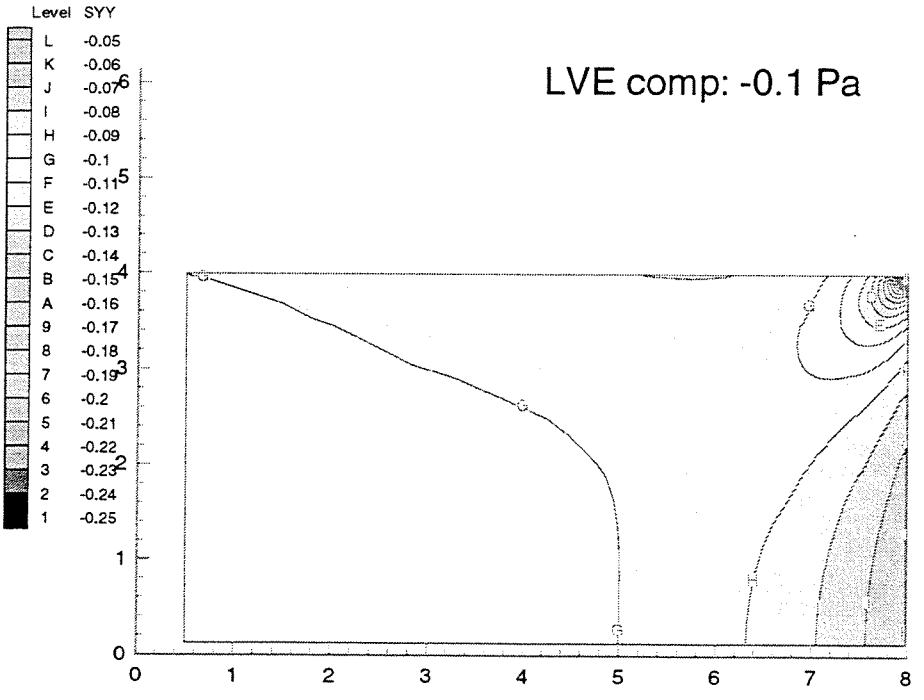


Figure 1.12: LVE, compression, long term, load= -0.1 Pa

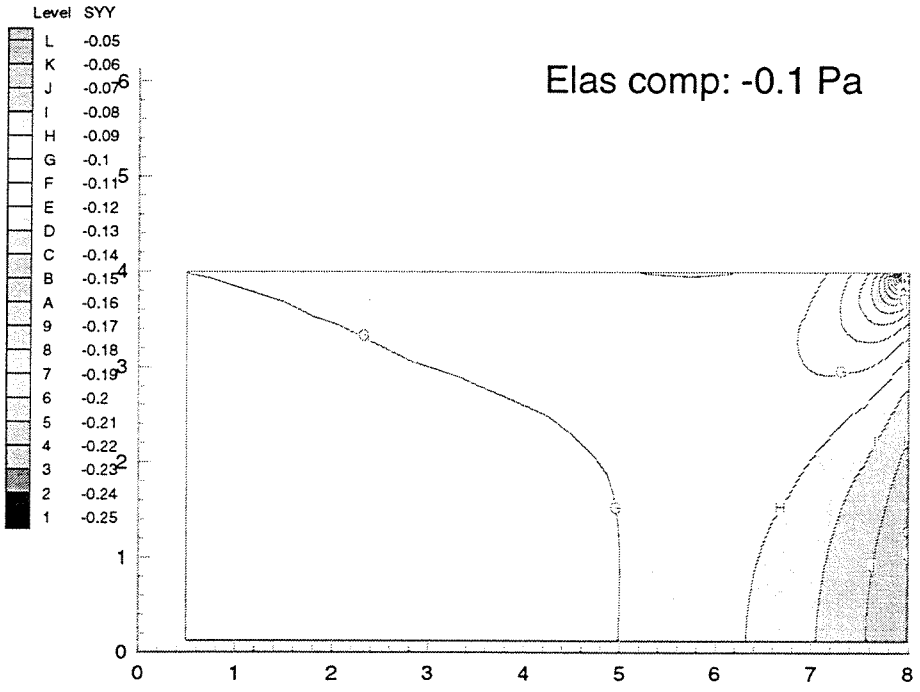


Figure 1.13: Elastic, compression, long term, load= -0.1 Pa

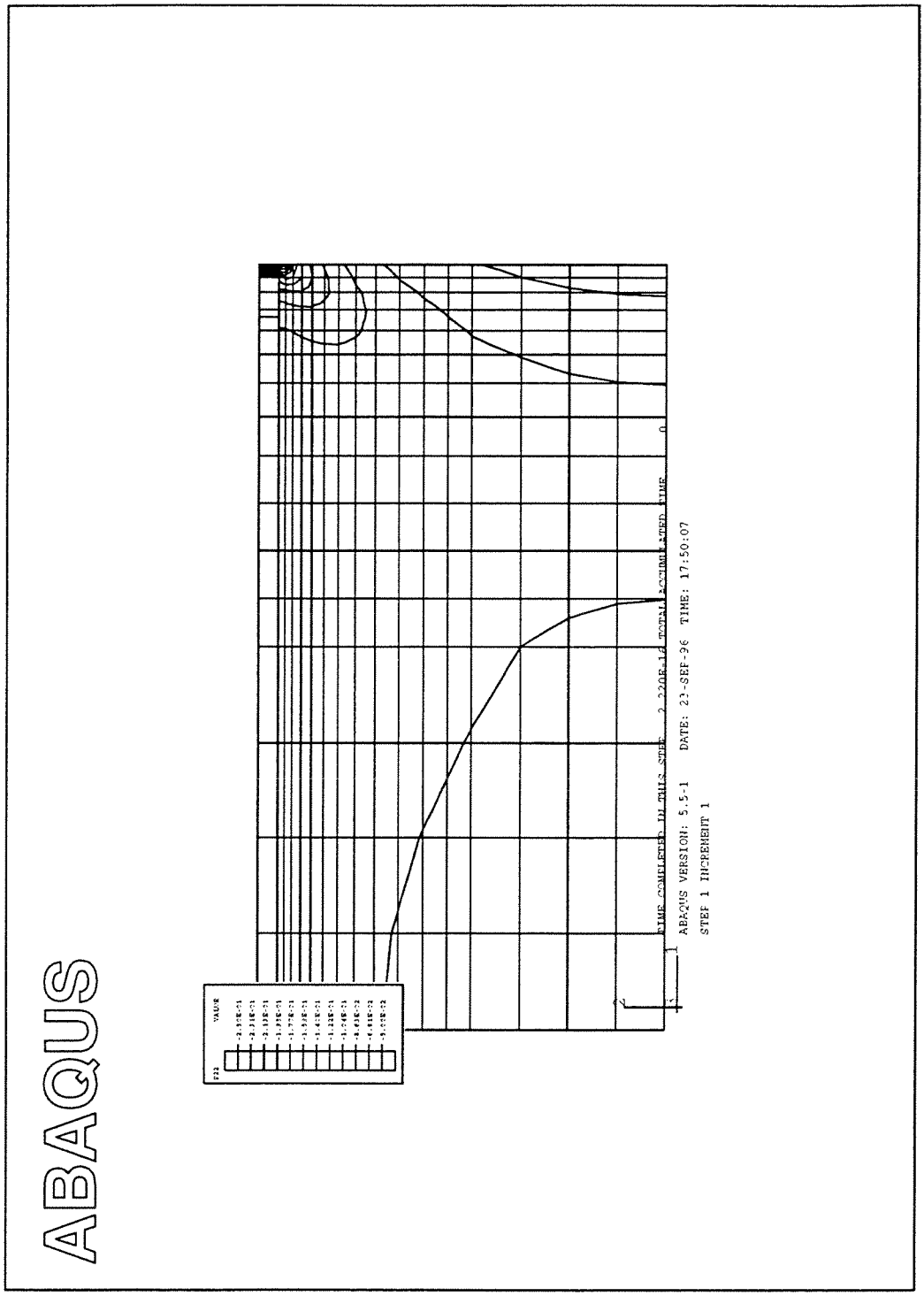


Figure 1.14: ABAQUS Elastic, compression, long term, load= -0.1 Pa

Chapter 2 Material Characterization and Constitutive Model

2.1 A Cross-linked Polymer

The cross-linked polymer used was extrapolated from the PVAc (Polyvinyl Acetate) material characteristics. PVAc is an uncross-linked polymer. It means that, when loaded in shear, the PVAc shear strain response reaches an asymptotic climbing line in the long term and creeps forever (see Fig. 2.1). That is called 'free dashpot' behavior. The extrapolation was achieved by subtracting the free dashpot asymptotic line from the PVAc creep curve in order to obtain the shear creep function $J(t)$ of a cross-linked polymer, the final material (see Fig. 2.2).

In order to input this material data for the FEAP code, it was necessary to get the shear relaxation function $G(t)$ and extract a proper Prony series representation for it. Inverting the 'smooth' creep curve into the relaxation curve was accomplished by using the program `invert.f`. The relaxation curve $G(t)$ (see Fig. 2.3) was smoothed by quadratic interpolation into a curve of at least 500 uniformly spaced points (see Fig. 2.4) by using the program `quadinterp.f`. A Prony series representation of 26 components (see `prony.dat`) was then extracted from the curve by using the program `prony.f`. The listing of these program can be in the Appendix.

The bulk relaxation function $K(t)$ initially remains the same as that of the PVAc material, then its value will be lowered to allow better compressibility, and finally the bulk relaxation function will be shifted so that its beginning matches with the beginning of the shear relaxation function.

2.2 Aluminium Metal

The grips are made of the metallic element aluminium. The metal can be considered to be a rigid body compared to the polymer. The elastic material constants are $K = 0.676 \times 10^9$ Pa and $G = 0.259 \times 10^6$ Pa.

2.3 Material Models in the Numerical Simulation

2.3.1 The three material models

Varying the parameters (such as the magnitude of the long term bulk modulus K_∞, the shift between the beginning of the relaxation moduli $K(t)$ and $G(t)$) resulted in three different material models. Each material model is described by its two relaxation moduli $K(t)$ and $G(t)$.

The shear relaxation function $G(t)$ remained unchanged for the three materials (see Fig. 2.4). This function was extrapolated from the shear relaxation of PVAc in order to model a cross-linked polymer, as explained in section 2.1.

The bulk relaxation function $K(t)$ was initially the original bulk modulus of PVAc with a long term value K_∞ of 252245 Pa (material #1, see Fig. 2.5), then the value K_∞ was decreased to 10 Pa (material #2, see Fig. 2.6) and finally the whole curve $K(t)$ is shifted 5 decades to the 'right' so that the beginning of $K(t)$ and $G(t)$ match (material #3, see Fig. 2.7).

The results for each one of these material models will be presented, below.

2.3.2 The two constitutive models: LVE and NLVE

Both constitutive models: linear viscoelasticity and nonlinear viscoelasticity are used for this boundary value problem. The difference between the two models lies in the presence or absence of the time shift factor ψ, that scales the 'internal time' function $\xi(t)$. Variations of the shift factor ψ are related to the local variation of dilatation, $d\epsilon_{kk}$. The shift factor effect, that is connected to the local variation of volume, exists

only in the case of the NLVE model.

Using the LVE constitutive model, the compression results for the three materials were compared with the relaxation moduli. Because the shift factor is disabled, that enables us to study the impact of the material model, i.e., the specific contributions of the relaxation moduli $K(t)$ and $G(t)$ on the normal stress $\sigma_{yy}(t)$ and shear stress $\sigma_{xy}(t)$ behavior.

Using the NLVE constitutive model, compression and tension results were compared in the case of each material model to study the impact of the shift factor ψ on the stress response $\sigma_{yy}(t)$.

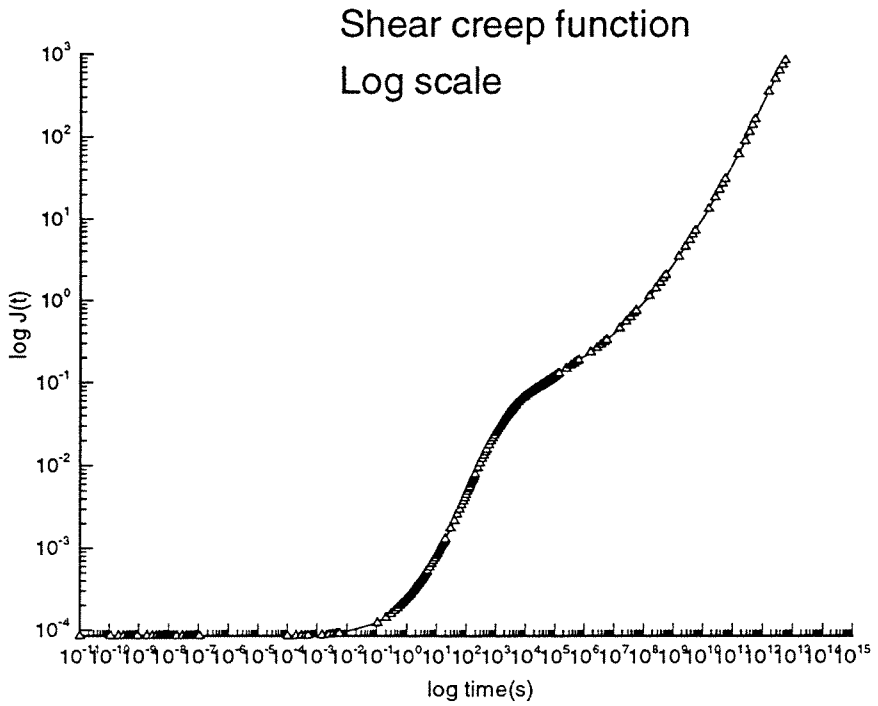


Figure 2.1: Original Shear creep function $J(t)$, PVAc

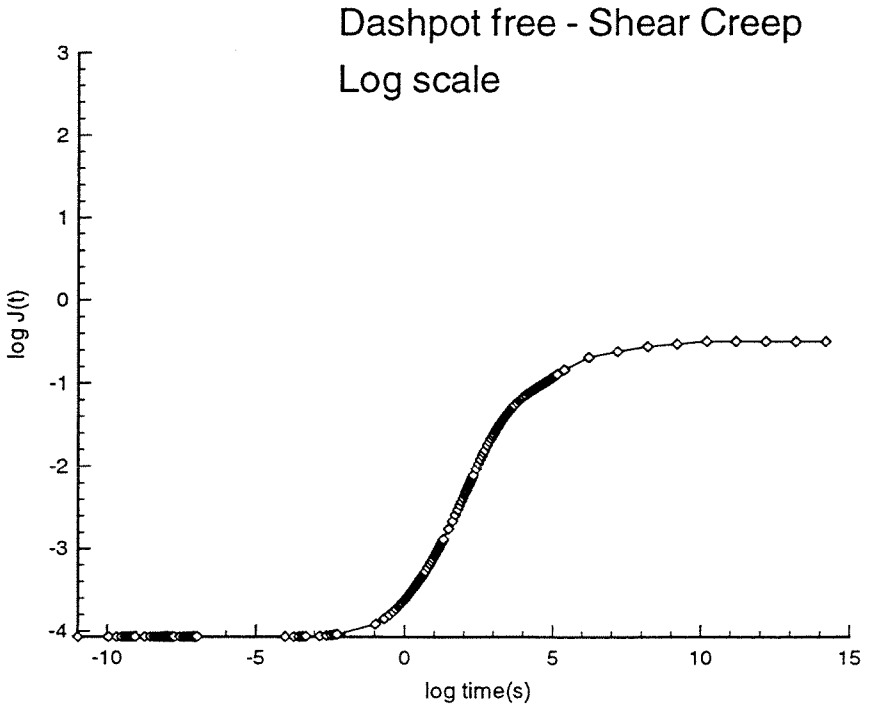


Figure 2.2: Shear creep function $J(t)$, cross-linked polymer

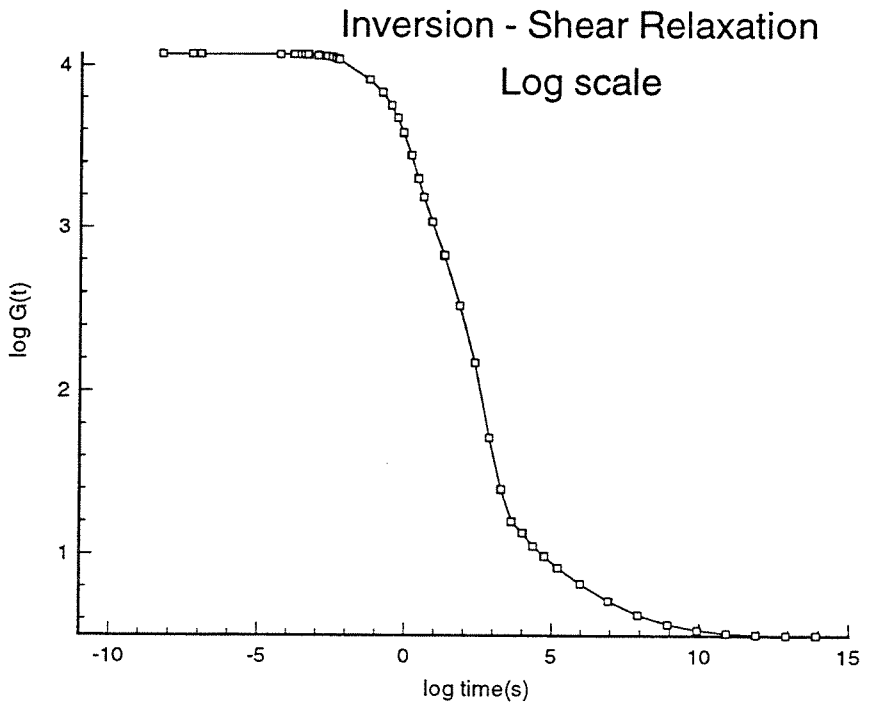


Figure 2.3: Shear relaxation function $G(t)$, inverted from the shear creep

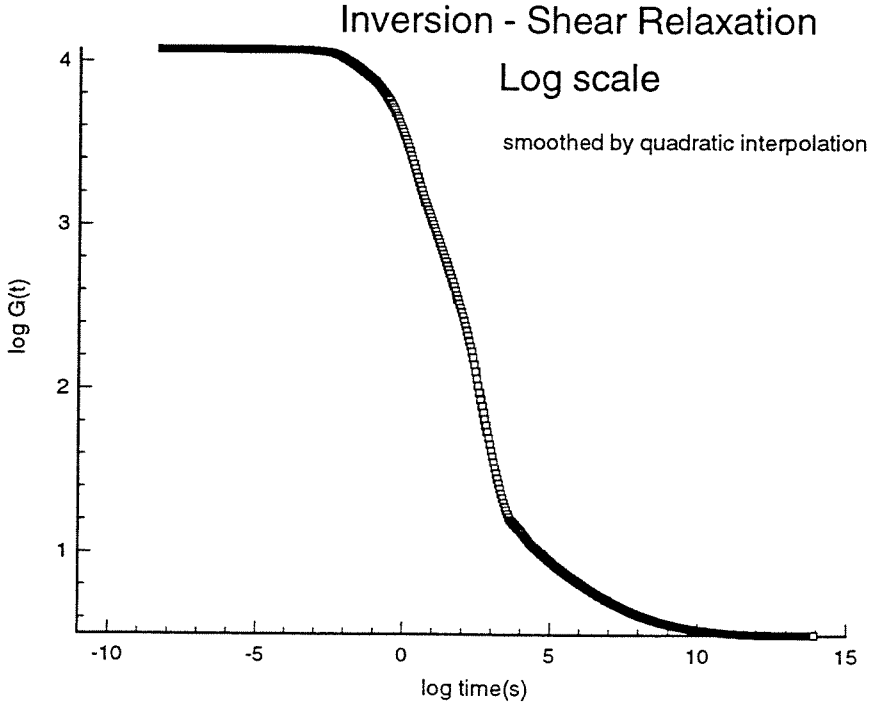


Figure 2.4: Shear relaxation function $G(t)$, uniformly spaced points

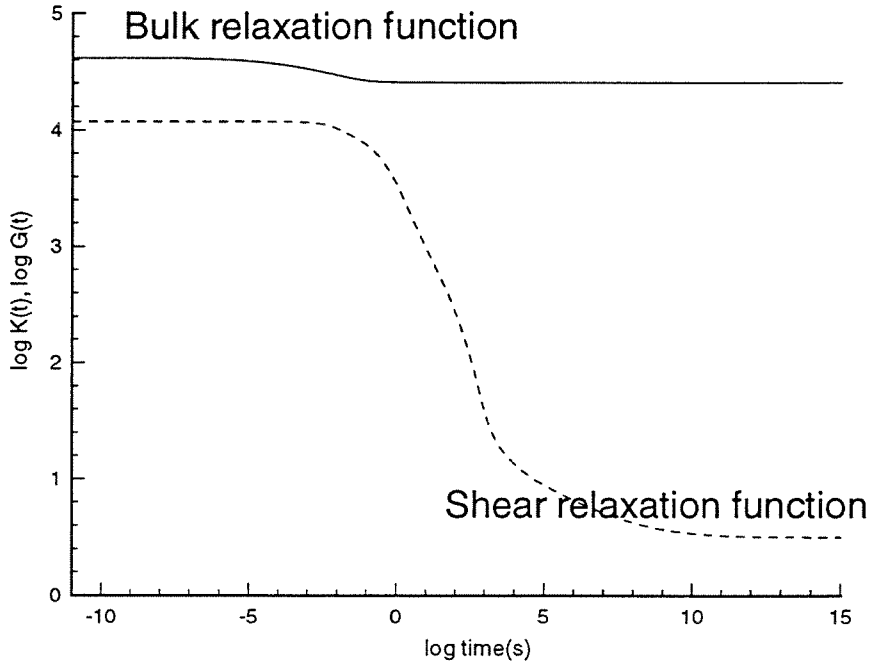


Figure 2.5: Original Shear and Bulk relaxation functions

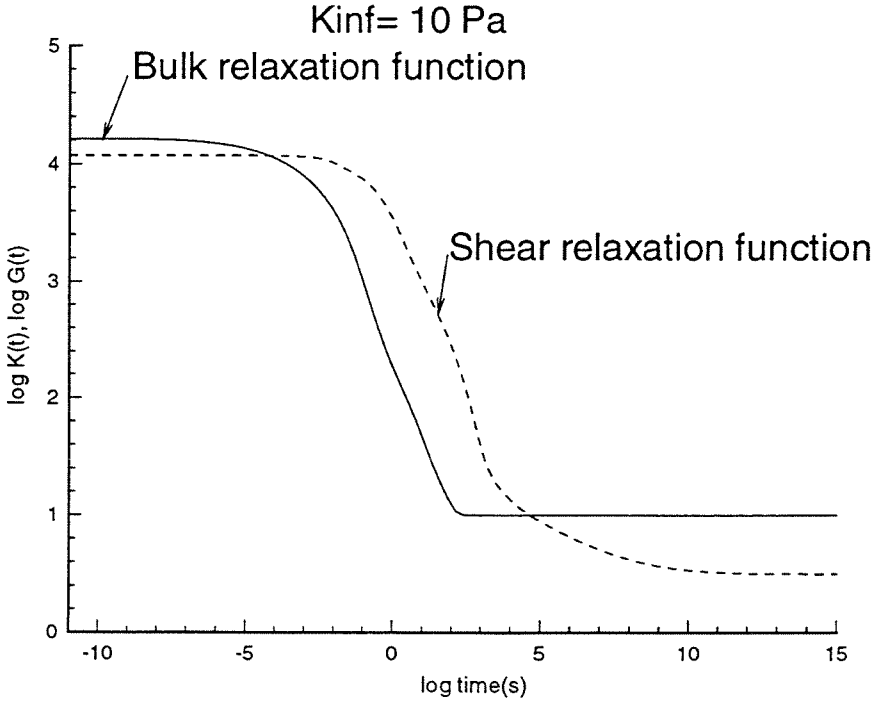


Figure 2.6: New Shear and Bulk relaxation, $K_\infty = 10$ Pa, better compressibility

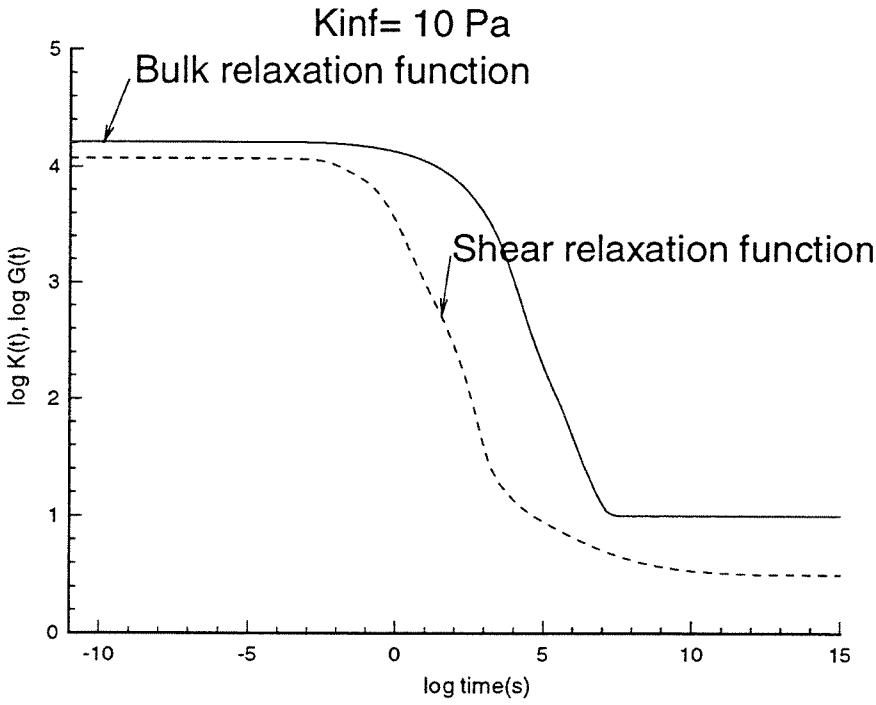


Figure 2.7: Matching Shear and Bulk relaxation, $K_\infty = 10$ Pa

Chapter 3 Results and Discussion

The results from two basic mechanical tests (uniform vertical compressive loading and uniform vertical tensile loading) are presented here. The tests were simulated on a time range from $10^{-11}s$ to 10^{15} seconds.

Three different material models were tested: first, with the original PVAc-based 'high' bulk modulus $K(t)$; second, with a 'lower' bulk modulus $K(t)$; third, with matching relaxation moduli $K(t)$ and $G(t)$. Initially, the normal stress $\sigma_{yy}(t)$ and shear stress $\sigma_{xy}(t)$ response of the three materials for a LVE constitutive model were investigated. Then, with a NLVE constitutive model, the results of a compression and tension tests for each of the three materials were compared.

3.1 Output Variables

The first decision to be made about the data output was to choose the relevant physical values. The study aims at analyzing the possible generation of deformation localizations in the polymer for a given compression with lateral constraints on the specimen. Deformation gradient localization can be related to inhomogeneity in the stress field. Therefore, the study will concentrate on the stress field. Given the boundary conditions and the importance of the y-direction, mostly the yy- and xy-components of the stresses are represented.

The study of the normal stress in the y-direction, σ_{yy}, was relevant. Furthermore, as the problem depends on two characteristic features, a vertical compression and a lateral constraint, the shear stress in the y-direction, σ_{xy}, should also be discussed.

3.2 Output Plots

Assessing stresses in a viscoelastic material requires dealing with two different parameters, space and time. Therefore, the results are gathered in two series of plots, that are given with respect to space and time: the spatial distribution plots (contour plots) and the stress time curves, respectively. The time-dependence of the stress solutions is a characteristic of viscoelastic materials.

A spatial distribution plot (or contour plot) is like a 'snapshot' of the stress field, at a given time. Each stress field is depicted by a figure at initial ($10^{-11} sec.$), intermediate ($10^{3} sec.$) and final time ($10^{14} sec.$). However, a series of 'snapshots' cannot recover the continuous time behavior of a stress field.

The normal stress $\sigma_{yy}(t)$ and shear stress $\sigma_{xy}(t)$ are calculated at different points located on the bottom row of the mesh (which actually is the middle row of the whole specimen). Each one of those fixed points is associated with a stress curve (see Fig. 3.1). Variables V2 to V8 correspond to the points near the y-axis going towards the right boundary. Hence, the highest compression σ_{yy} (in the compression case) occurs at the point closest to the y-axis (see Fig. 3.1). That holds true for all the cases under consideration. Also, the highest tension σ_{yy} (in the tension case) is at the center of the specimen.

The two sets of data contribute and provide feedback to one another. The time stress curves indicate the best intermediate time (a 'peak', see Fig. 3.2) for the stress distribution plots. The stress distribution plots indicate the overall 'picture' of the stress field at those given times, an information that cannot be gathered from the time curves.

3.3 Influence of the Parameters

Several parameters influence the results strongly, as the following results show. Basically, three parameters were varied and their influence was studied; these parameters are as follows:

- the magnitude of the applied vertical load
- the magnitude of the long term bulk modulus K_∞
- the log scale time shift between the beginning of the relaxation moduli $K(t)$ and $G(t)$.

Load :

The viscoelastic theory formulated in this program is restricted to small strain theory (see Constitutive theory). Hence, large deformations that yield more than about 10% of strain cannot be correctly calculated by this version of FEAP. Therefore, the applied load must be chosen such that the strains do not exceed 10%. A realistic load magnitude must be determined with respect to the material constants. The polymer described in chapter Theory and Methods has the following initial and long term shear and bulk moduli:

K_{init} = 41343 Pa, G_{init} = 11790 Pa and K_∞ = 25245 Pa, G_∞ = 3.16 Pa, respectively.

That leads to a long term Young's modulus E_∞ of roughly 10 Pa. Given that an average strain of about 1% is desirable, the load should be no larger than +0.1 Pa (tension) or −0.1 Pa (compression).

Long term bulk modulus :

First, the original value of K_∞ was that of PVAc: 25245 Pa. Later, the value of the long term bulk modulus K_∞ was lowered to 10 Pa in order to allow better compressibility.

The time shift between the two relaxation moduli :

For the first set of simulations (Material #1 and #2), the bulk relaxation modulus $K(t)$ preceded the shear relaxation modulus $G(t)$ by roughly five time decades. Later, it was decided to match the beginning of the two relaxation functions in order to help determine the relative contribution of $K(t)$ and $G(t)$ to the stress response.

3.4 Contribution of the Relaxation Moduli to the Stress Curves in Linear Viscoelasticity

The linear viscoelastic model eliminates the effect of the shift factor. Thus, the time behavior of $\sigma_{yy}(t)$ has the same scaling as the original relaxation functions $K(t)$ and $G(t)$. The contributions of each of the relaxation functions to the stress response can be directly determined by relating $K(t)$ and $G(t)$ to the stress curves $\sigma_{yy}(t)$ and $\sigma_{xy}(t)$.

3.4.1 Normal stresses $\sigma_{yy}(t)$

The stress time behavior curves $\sigma_{yy}(t)$ for the material models #1, #2 and #3 are shown in Figures 3.2, 3.3 and 3.4, respectively. Their respective relaxation functions $K(t)$ and $G(t)$ are depicted in Fig. 2.5, 2.6 and 2.7.

For Material #1, the stresses, initially, slowly converge to -0.1 Pa when the $K(t)$ value begins dropping (see Fig. 2.5). But the stresses abruptly diverge away from the -0.1 Pa line as soon as the shear modulus $G(t)$ begins dropping. Also, $G(t)$ variation is much more significant than $K(t)$ variation. That may explain the faster divergence from the -0.1 Pa line than the initial convergence. The stress σ_{yy} has a non-monotonic behavior. The stress curves present a 'peak' (a local extremum) between their instantaneous and long term values.

For Material #2, the existence of a 'double peak' in the σ_{yy} stress curve has to be pointed out in Fig. 3.2. Furthermore, a sort of 'inversion' in the order of the stress values is visible. At the initial time, the highest σ_{yy} stress (in the compression case) is close to the y-axis and the stress gradually drops, as the right edge is approached, and reaches the lowest compression value at the right edge. However, that order of the stress values is inverted during the transitional period (from 10^{-3} to 10^{5} seconds) between initial and long term response. The bulk modulus $K(t)$'s variation is now of same magnitude than the variation of $G(t)$.

For short times, the stress curves seem influenced by the decrease of $K(t)$. When $K(t)$ begins decreasing, the stresses converge to -0.1 Pa and even cross one another to invert their numerical values order ('stress inversion'). Then, for the intermediate time, $G(t)$ begins dropping. The fact that $K(t)$ and $G(t)$ have the same order of magnitude may explain the transitional behavior of $\sigma_{yy}(t)$, strongly non-monotonic in Fig. 3.2. The 'double peak' of $\sigma_{yy}(t)$ may be traced back to the changing relative contributions of $K(t)$ and $G(t)$ in this transitional period. For the final time, $\sigma_{yy}(t)$ is influenced by $G(t)$. While $K(t)$ reaches a plateau, $G(t)$ keeps decreasing. The stress curves $\sigma_{yy}(t)$ re-converge to -0.1 Pa, intersect one another and diverge away from -0.1 Pa.

For Material #3, the 'double peak' and the 'stress inversion' have disappeared (as seen in Fig. 3.4). In this case, the $K(t)$ and $G(t)$ influence begins at the same time. The $\sigma_{yy}(t)$ response is dominated by $G(t)$'s influence, in a similar way as in Material #1. Note that when $K(t)$ reaches a plateau, around 10^7 seconds, the stresses re-diverge away from -0.1 Pa. That proves the influence of $G(t)$. Also, the slopes of the stress curves in their re-convergence, before the final time, are steeper than in case #1. That traces back to the influence of $K(t)$, that has a stronger magnitude variation in this case (3 decades) than in case of Material #1 (1/5th of a decade) as seen in Fig. 2.5 and Fig. 2.6.

3.4.2 Shear stresses $\sigma_{xy}(t)$

The maximum shear stress σ_{xy} value is located at the point V4 (around the middle of the mesh bottom row), but not at the point closest to the y-axis. The points close to the y-axis (point V2) and to the right boundary (point V8) have shear stress values that are nearly zero (see Fig. 3.5, 3.6, 3.7). By looking at the σ_{xy} contour plots in Figs. 3.14, 3.15 and 3.16 for the case of Material #2, the fact is explained by the 'half-egg' shape of the σ_{xy} contours. Maximum shear stresses σ_{xy} occur near the middle of the central row. The shear stresses decrease to zero as the y-axis and the right boundary are approached. The shear stresses σ_{xy} are zero on the left and right

boundaries, in accordance with the given boundary conditions.

Most of the precedent remarks formulated for the normal stress $\sigma_{yy}(t)$ (non-monotonic behavior, peak, inversion) apply to the shear stresses $\sigma_{xy}(t)$, too. The one difference is that there is no 'double peak' for shear stress in the case of Material #2 compared with the normal stresses.

3.5 Comparison of Tension and Compression Results for Nonlinear Viscoelasticity

The results for compression and tension of a nonlinear viscoelastic (NLVE) material are compared for the material models #1, #2 and #3 in Fig. 3.17 and 3.18, 3.19 and 3.20, 3.21 and 3.22 for the normal stresses $\sigma_{yy}(t)$ and Fig. 3.23 and 3.24, 3.25 and 3.26 for the shear stresses $\sigma_{xy}(t)$. Essentially, nonlinear viscoelasticity reintroduces the time shift factor.

3.5.1 Normal stresses $\sigma_{yy}(t)$

For Material #1, the time curves $\sigma_{yy}(t)$ for compression in Fig. 3.17 and tension in Fig. 3.18 are very similar, almost identical. Essentially they superpose nearly exactly if you 'flip' the curves over the time axis to the other side. Also, the stress curves $\sigma_{yy}(t)$ corresponding to the NLVE compression and tension tests are almost identical to the $\sigma_{yy}(t)$ curves of the LVE compression test (see Fig. 3.2). The time behavior is the same for the three tests, namely NLVE compression test, NLVE tension test and LVE compression test. The time scaling has not been modified by the time shift factor. Therefore, the shift factor ψ must be trivially equal to 1, which means that the fractional free volume $f(t)$ remains at its initial value f_{init}.

The negligible variation of the fractional free volume (variation proportional to the variation of ϵ_{kk}) can be linked to the high values of the bulk modulus $K(t)$ (K_{init}= 41343 Pa, K_∞= 25245 Pa) with respect to the values of the shear modulus $G(t)$ (G_{init} = 11790 Pa, G_∞ = 3.16 Pa). The material has a 'low' compressibility.

For instance, a hydrostatic pressure P of 0.1 Pa and a bulk modulus K of 20000 Pa leads a volume contraction $|\epsilon_{kk}| = \frac{\Delta V}{V} = \frac{P}{K}$ of 0.0005%. The changes in free volume are therefore negligible.

For Material #1, the LVE or NLVE model do not give any substantial difference in the material response.

For Material #2, the value of the bulk modulus $K(t)$ was lowered to allow better compressibility (K_{init}= 16108 Pa, K_∞= 10 Pa). In this case, we see a different time scaling in the stress response between the compression and tension case. The long term behavior (where the asymptotic flat lines appear, c.f. Fig. 3.19 and 3.20) is reached much earlier in the tension case than in the compression case. The compression stress curves are stretched in time while the tension stress curves are contracted. In the long term, a hydrostatic pressure P of 0.1 Pa and a bulk modulus K of 10 Pa provoke a volume contraction $|\epsilon_{kk}| = \frac{P}{K}$ of 1%, which is of the same order than the initial fractional free volume (f_{init}= 0.95%).

In the compression test, the free volume decreases. That brings a retardation effect into the relaxation (or creep) behavior. The range of the transitional stress state (where the stresses are fluctuating and strongly non-monotonic) is longer than in the case of the LVE compression test (see Fig. 3.19 and Fig. 3.3).

The 'double peak' in the compression test is more clearly marked. The relaxation functions $K(\xi(t))$ and $G(\xi(t))$ time scale has been modified by the internal time function $\xi(t)$. Their relative contributions are different from those of the LVE compression test: these new contributions make the 'double peak' more pronounced. As time goes on, $K(t)$ drops, the compressibility increases, the volume contraction $|\epsilon_{kk}|$ increases, the free volume f decreases and the retardation effect grows more important. Consequently, the retardation effect is more important around the 2^{nd} 'peak' period than for an earlier time. That may explain why the 2^{nd} 'peak' is more pronounced.

On the contrary, in the tension test, the transition range has shrinked and the 2^{nd} 'peak' is less pronounced (see Fig. 3.20). Thus, the time scaling (represented by the internal time function $\xi(t)$) is affected by an acceleration effect.

For Material #3, the beginning of the bulk relaxation $K(t)$ was matched to the beginning of the shear relaxation $G(t)$ (see Fig. 2.7). The conclusions drawn for Material #2 apply once more. Notice that the stress curves convergence then divergence before the final time (where it looks like a 'neck' in Fig. 3.22) is more pronounced in the tension case, while this 'neck' disappears in the compression case (see Fig. 3.21).

3.5.2 Shear stress $\sigma_{xy}(t)$

Most of the characteristics of the normal stresses $\sigma_{yy}(t)$ hold also to the shear stresses $\sigma_{xy}(t)$: the retardation or the acceleration effect (see Figs. 3.23, 3.24), the neck appearance for Material #3 (see Figs. 3.25, 3.26).

3.6 Using Contour Plots for σ_{yy} and σ_{xy}

The 'stress inversion' can also be recognized in the contour plots (NLVE compression -0.1 Pa) in Figs. 3.34, 3.35 and 3.36 for the normal stresses σ_{yy} and Figs. 3.37, 3.38 and 3.39 for the shear stresses σ_{xy}, that are given for Material #2 case. Each stress component of the case (NLVE, compression, Material #2) has three contour plots corresponding to the stress distribution field at the initial (10^{-11} s), intermediate (10^{3} s) and final time (10^{14} seconds).

The long term σ_{yy} contour plot for the NLVE model (Fig. 3.36) in compression test is compared to σ_{yy} contour plot in compression test for an elastic model (Fig. 1.13), provided with the long term material constants, in Fig. 3.33. The interesting point is that the stress σ_{yy} contours do not superpose exactly. In its long term behavior, NLVE materials do not act like elastic materials. The time shift factor effect modifies the response, somewhat. This property is very different from linear viscoelasticity. In linear viscoelasticity, theory predicted and results confirmed that elastic and LVE models in the long term give the same result. This proves the necessity of using the NLVE model over the LVE model, for this corner stress singularity boundary value problem.

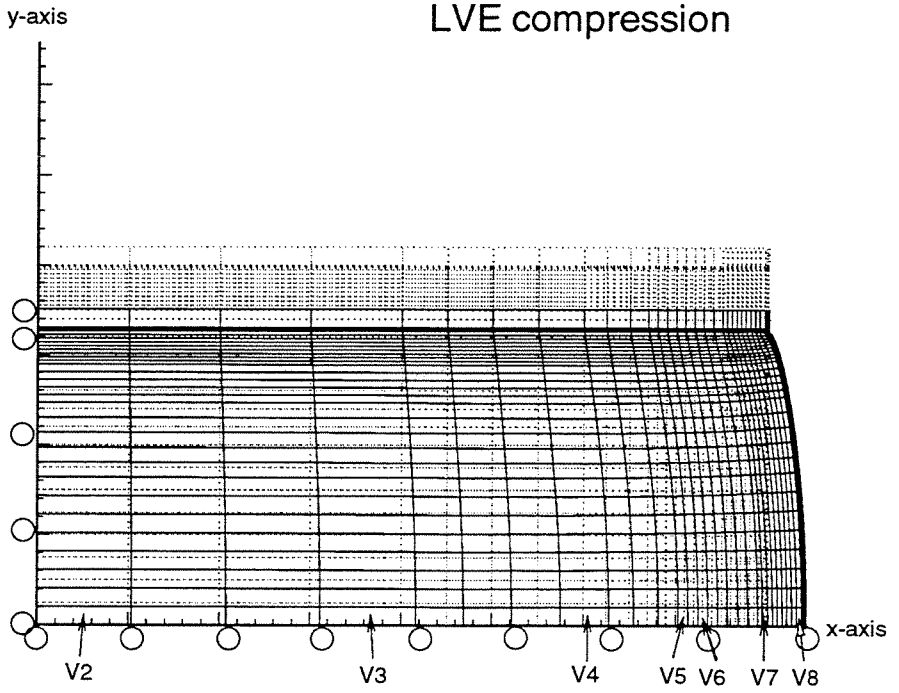


Figure 3.1: Stress points V2 to V8 in the mesh

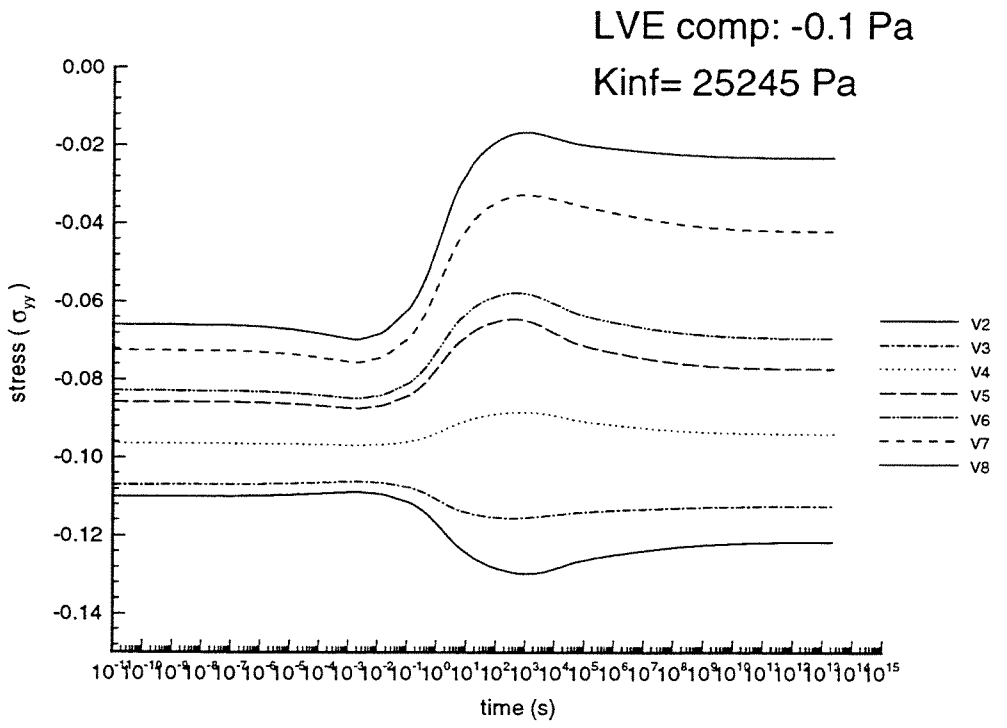


Figure 3.2: Time stress curve $\sigma_{yy}(t)$, LVE, compression, Material#1

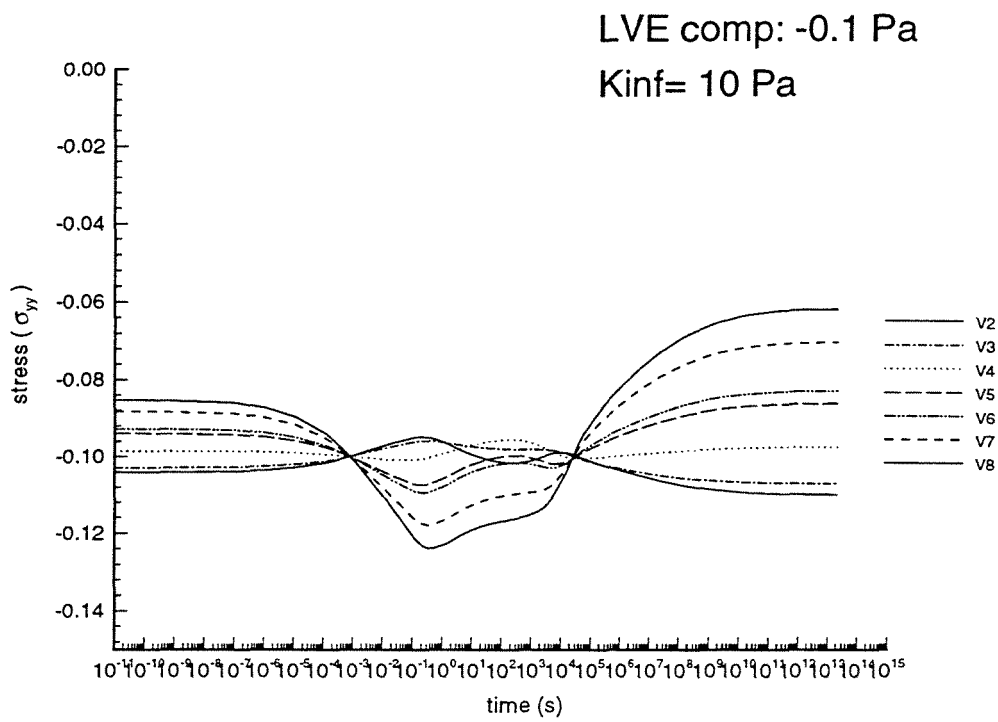


Figure 3.3: $\sigma_{yy}(t)$, LVE, compression, Material#2

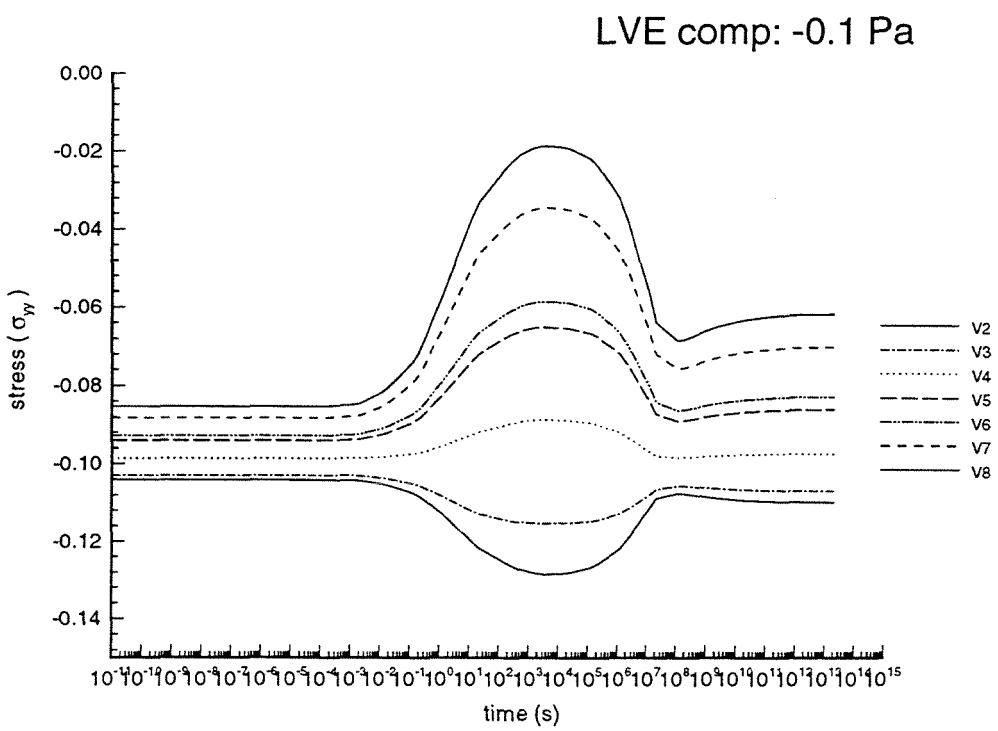


Figure 3.4: $\sigma_{yy}(t)$, LVE, compression, Material#3

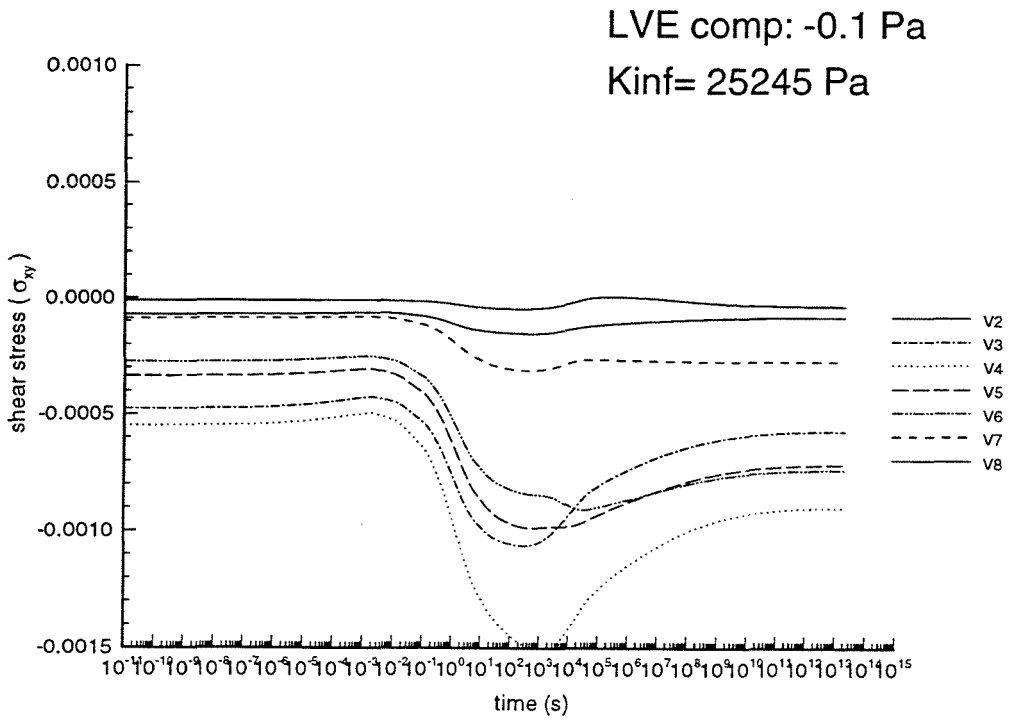


Figure 3.5: $\sigma_{xy}(t)$, LVE, compression, Material#1

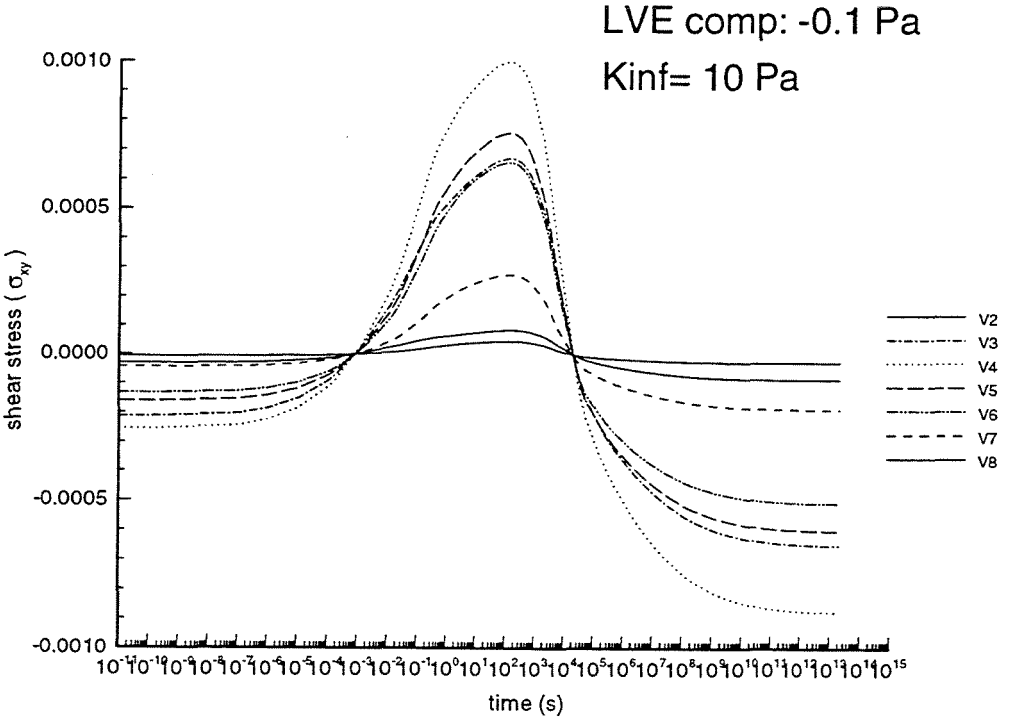


Figure 3.6: $\sigma_{xy}(t)$, LVE, compression, Material#2

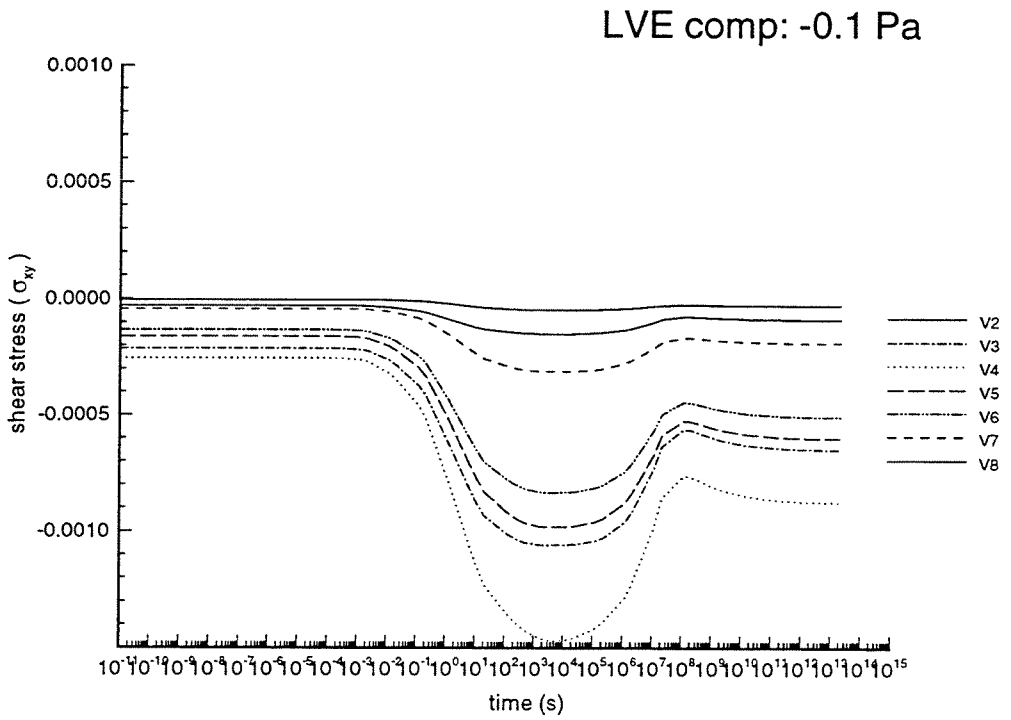


Figure 3.7: $\sigma_{xy}(t)$, LVE, compression, Material#3

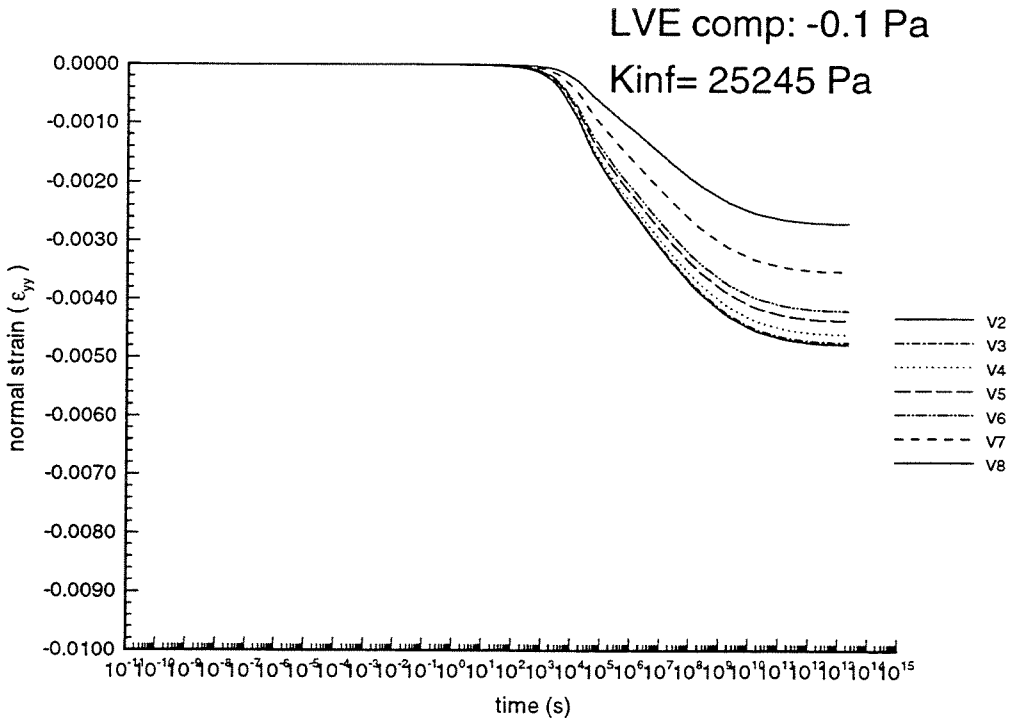


Figure 3.8: $\epsilon_{yy}(t)$, LVE, compression, Material#1

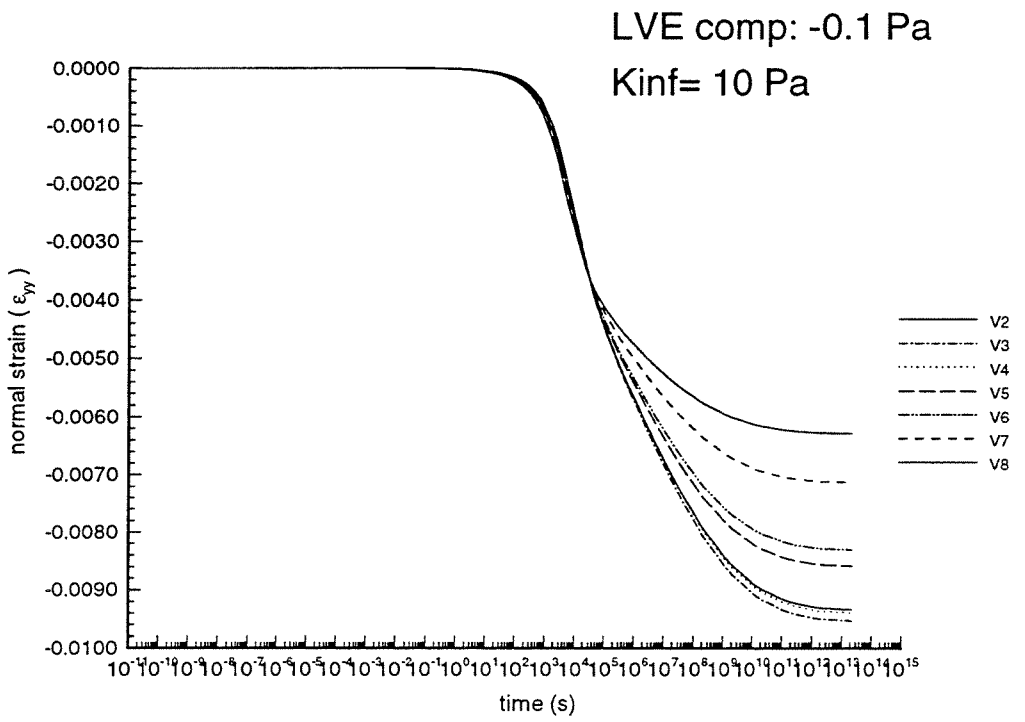


Figure 3.9: $\epsilon_{yy}(t)$, LVE, compression, Material#2

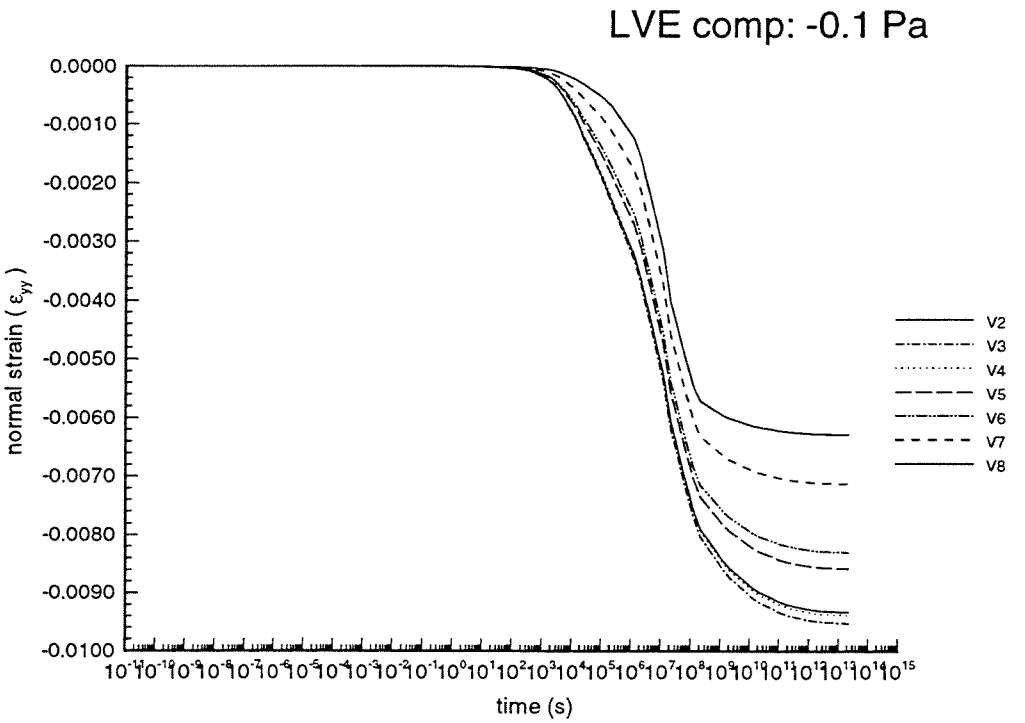


Figure 3.10: $\epsilon_{yy}(t)$, LVE, compression, Material#3

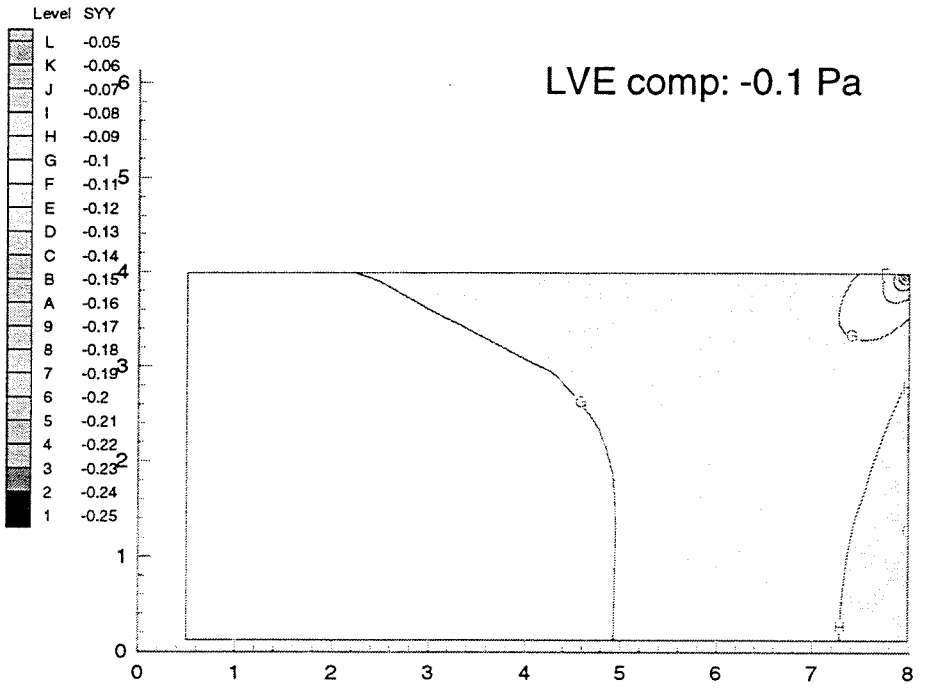


Figure 3.11: Contour plot σ_{yy}, LVE, compression, initial time 10^{-11} s

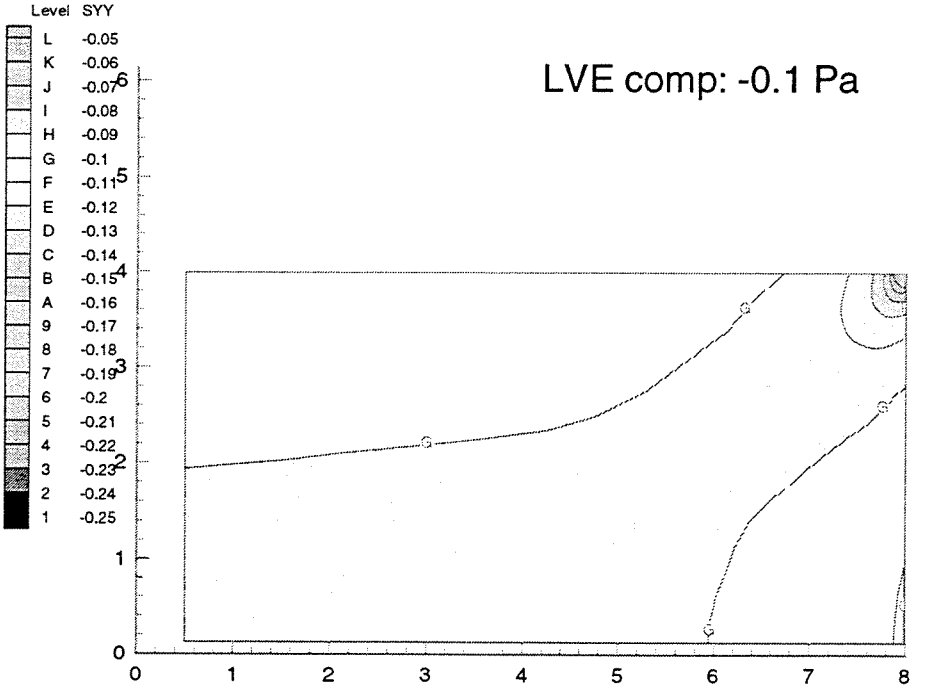


Figure 3.12: Contour plot σ_{yy}, LVE, compression, middle time 10^3 s

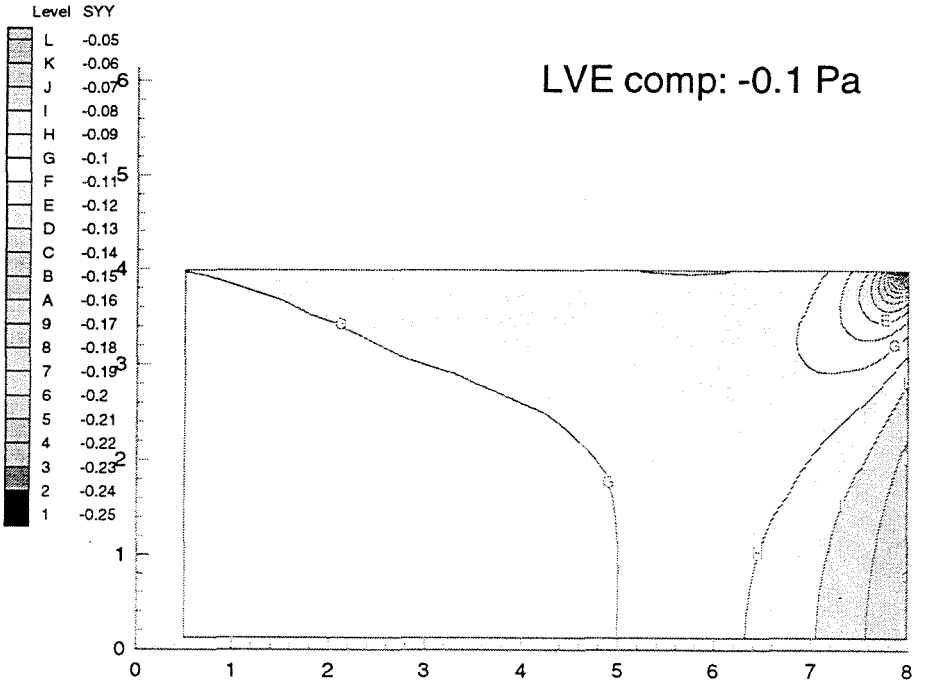


Figure 3.13: Contour plot σ_{yy}, LVE, compression, final time 10^{14} s

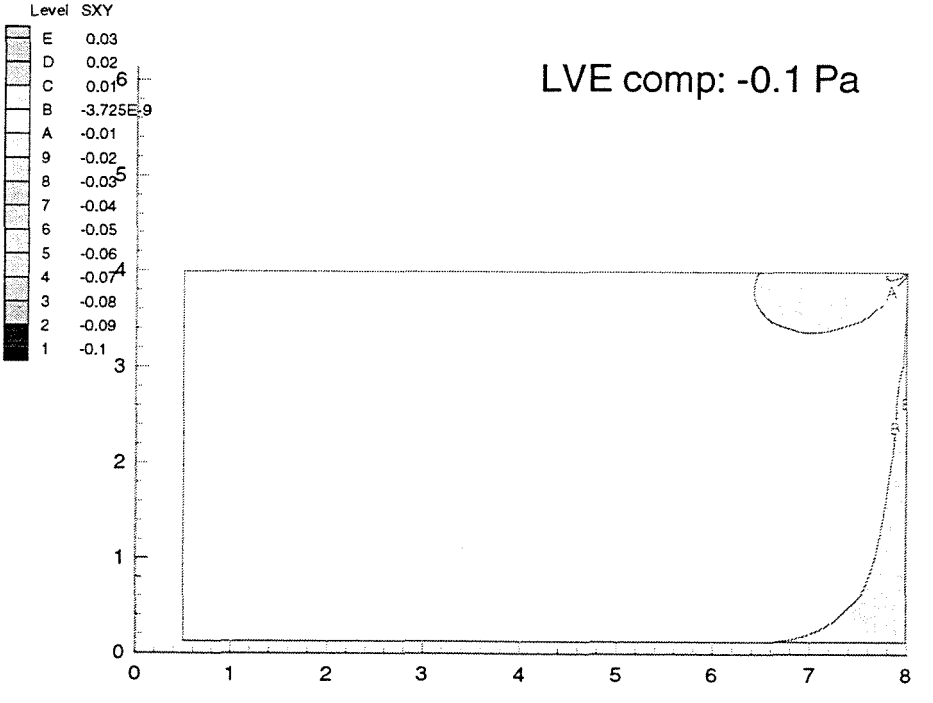


Figure 3.14: Contour plot σ_{xy}, LVE, compression, initial time 10^{-11} s

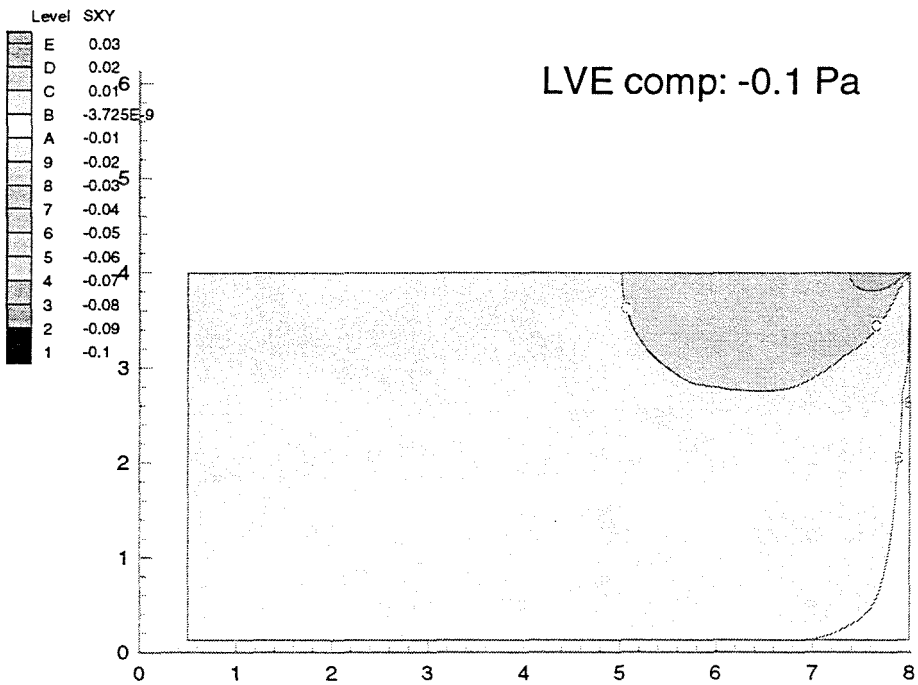


Figure 3.15: Contour plot σ_{xy}, LVE, compression, middle time 10^3 s

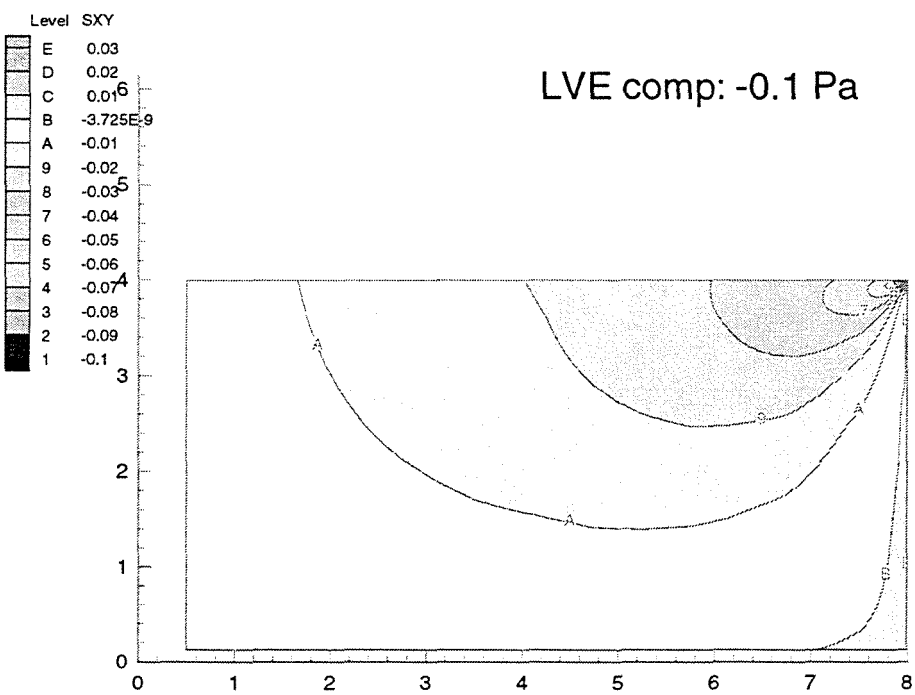


Figure 3.16: Contour plot σ_{xy}, LVE, compression, final time 10^{14} s

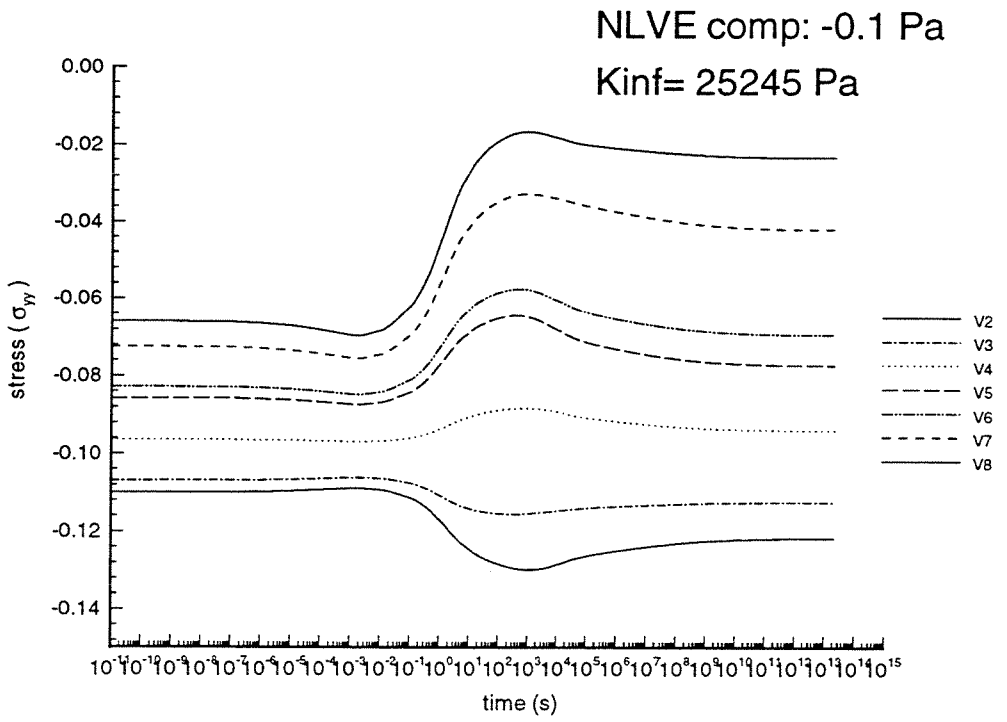


Figure 3.17: $\sigma_{yy}(t)$, NLVE, compression, Material#1

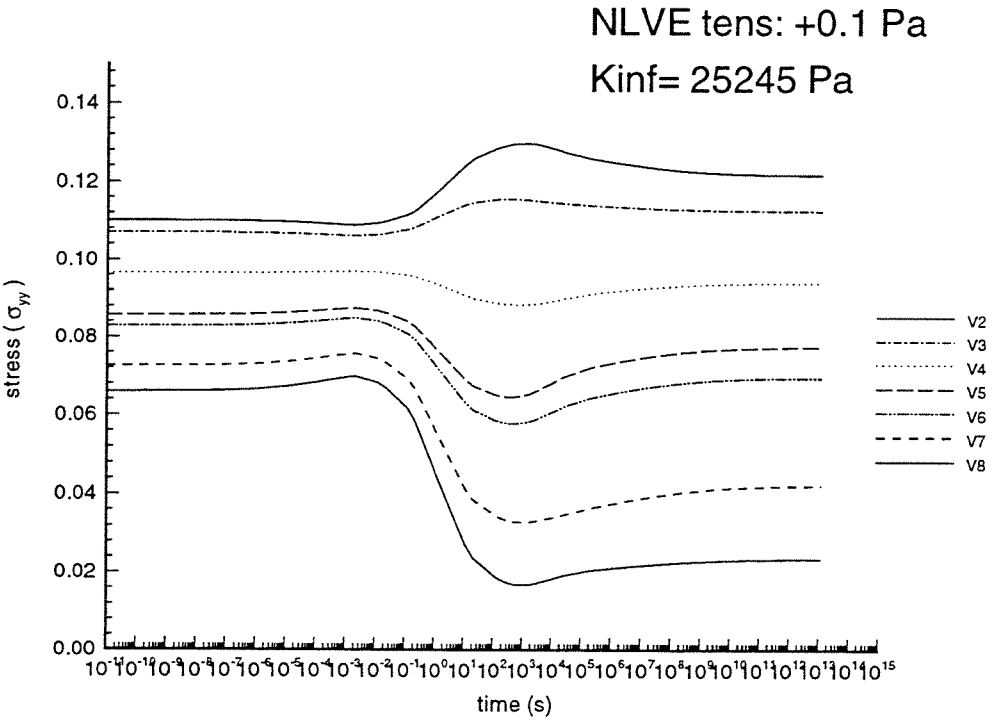


Figure 3.18: $\sigma_{yy}(t)$, NLVE, tension, Material#1

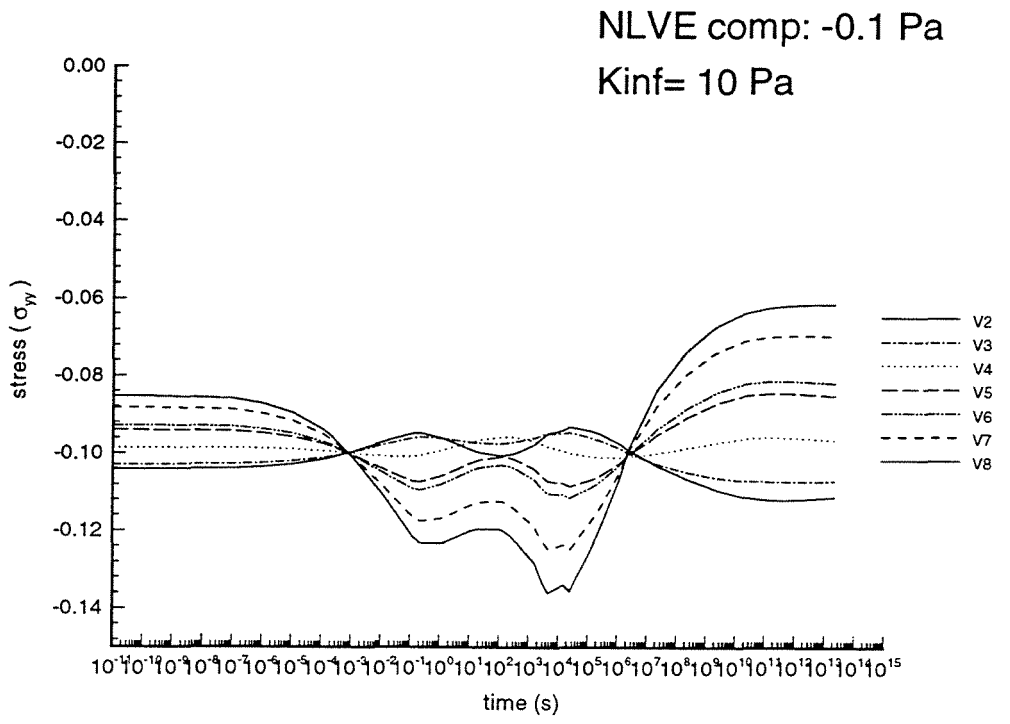


Figure 3.19: $\sigma_{yy}(t)$, NLVE, compression, Material#2

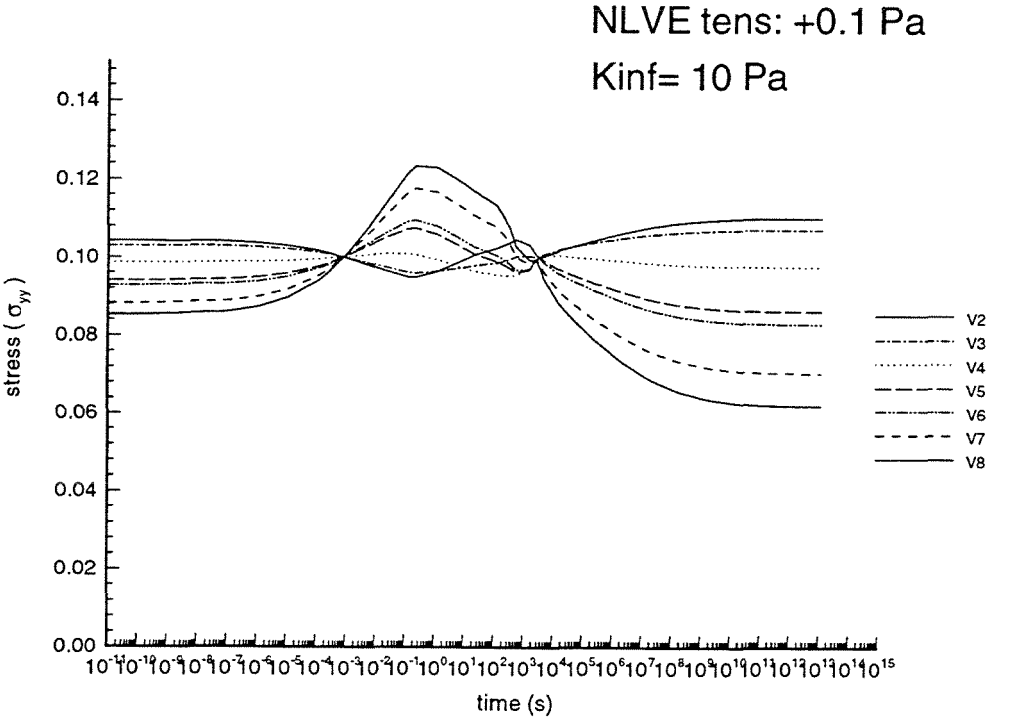


Figure 3.20: $\sigma_{yy}(t)$, NLVE, tension, Material#2

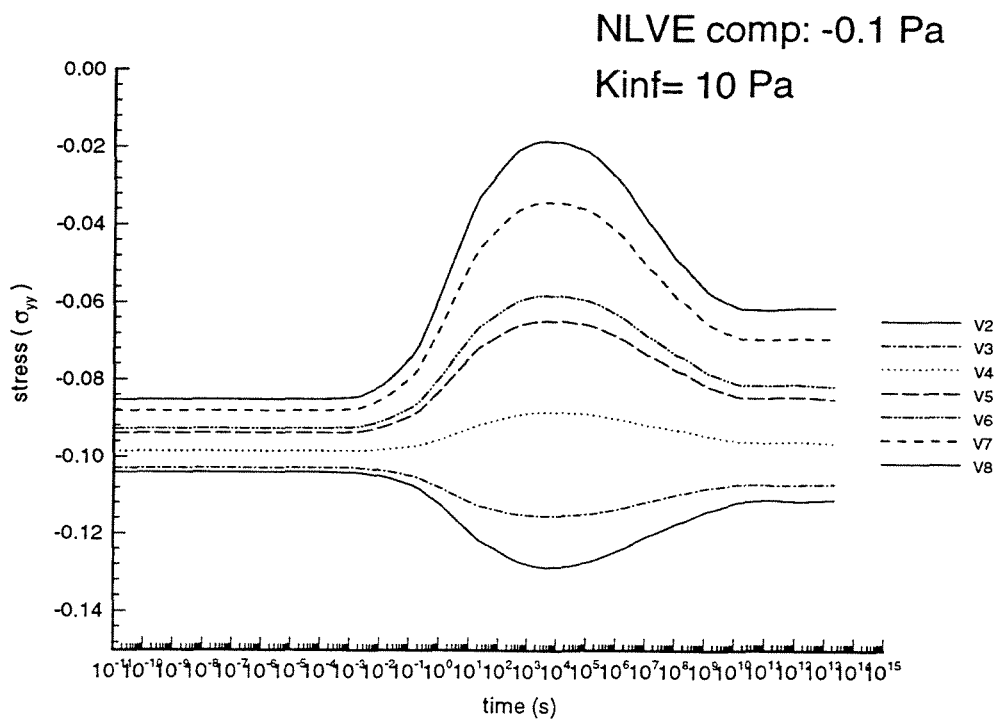


Figure 3.21: $\sigma_{yy}(t)$, NLVE, compression, Material#3

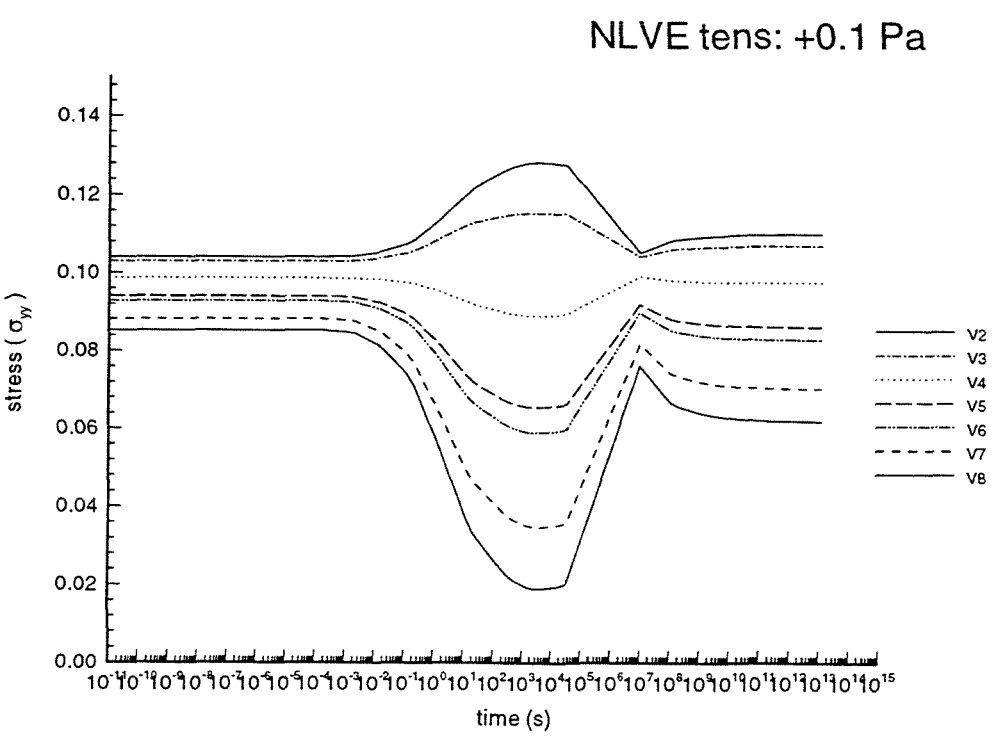


Figure 3.22: $\sigma_{yy}(t)$, NLVE, tension, Material#3

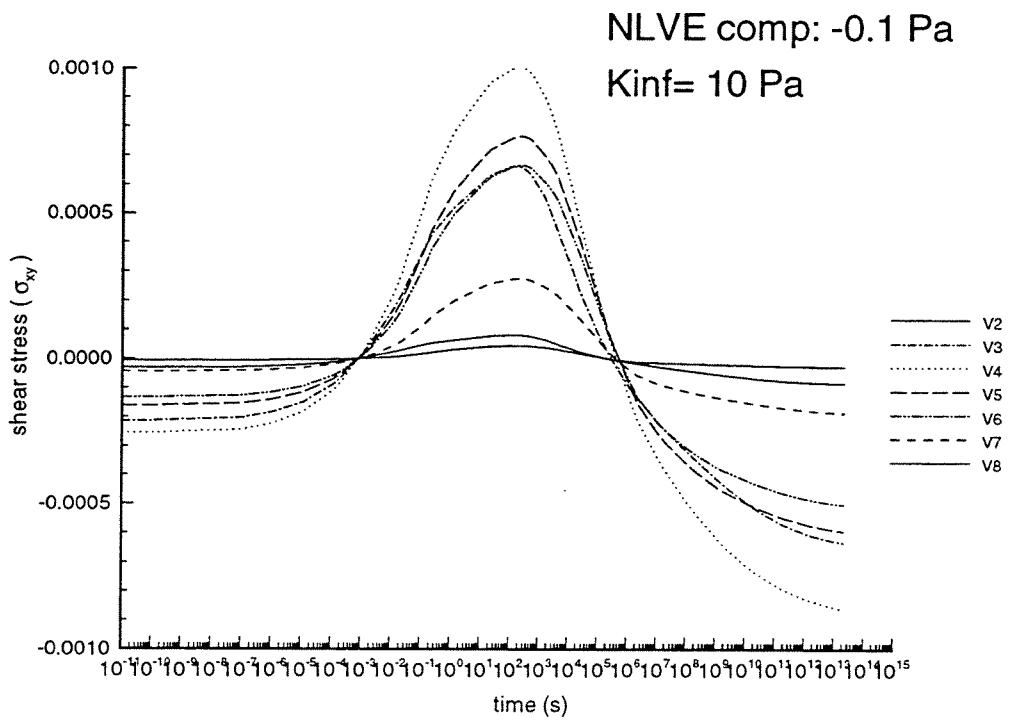


Figure 3.23: $\sigma_{xy}(t)$, NLVE, compression, Material#2

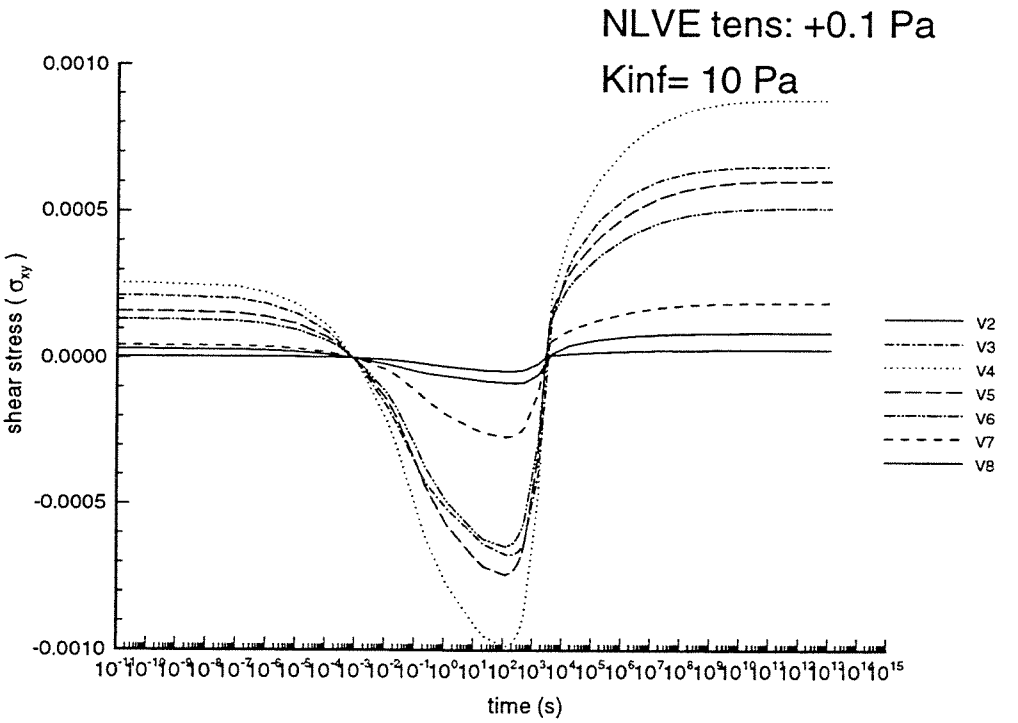


Figure 3.24: $\sigma_{xy}(t)$, NLVE, tension, Material#2

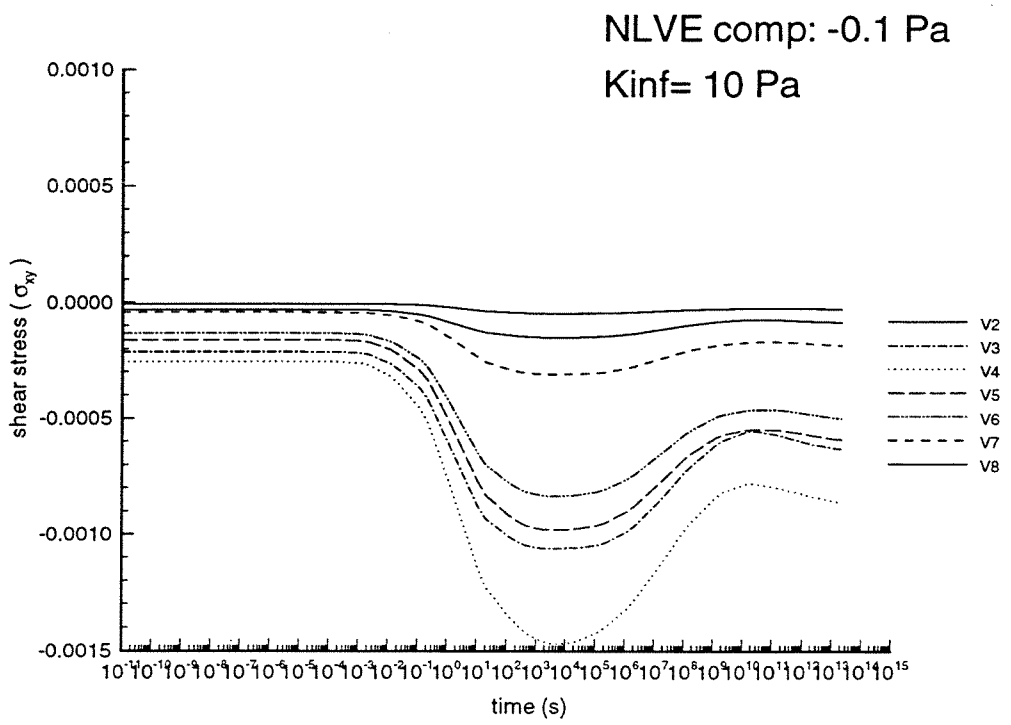


Figure 3.25: $\sigma_{xy}(t)$, NLVE, compression, Material#3

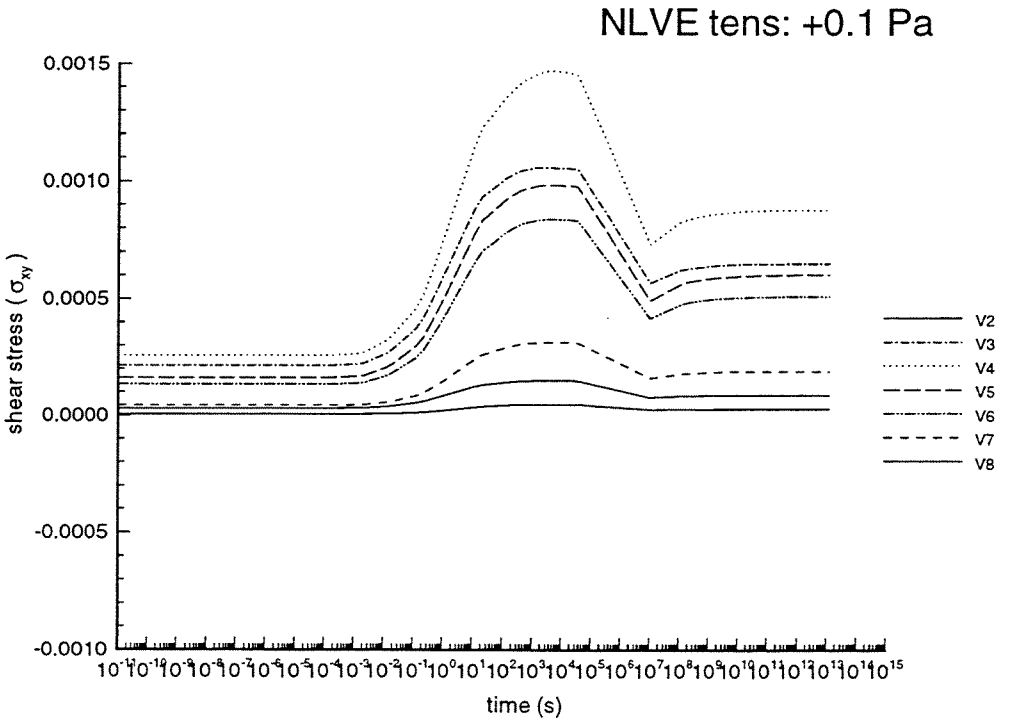


Figure 3.26: $\sigma_{xy}(t)$, NLVE, tension, Material#3

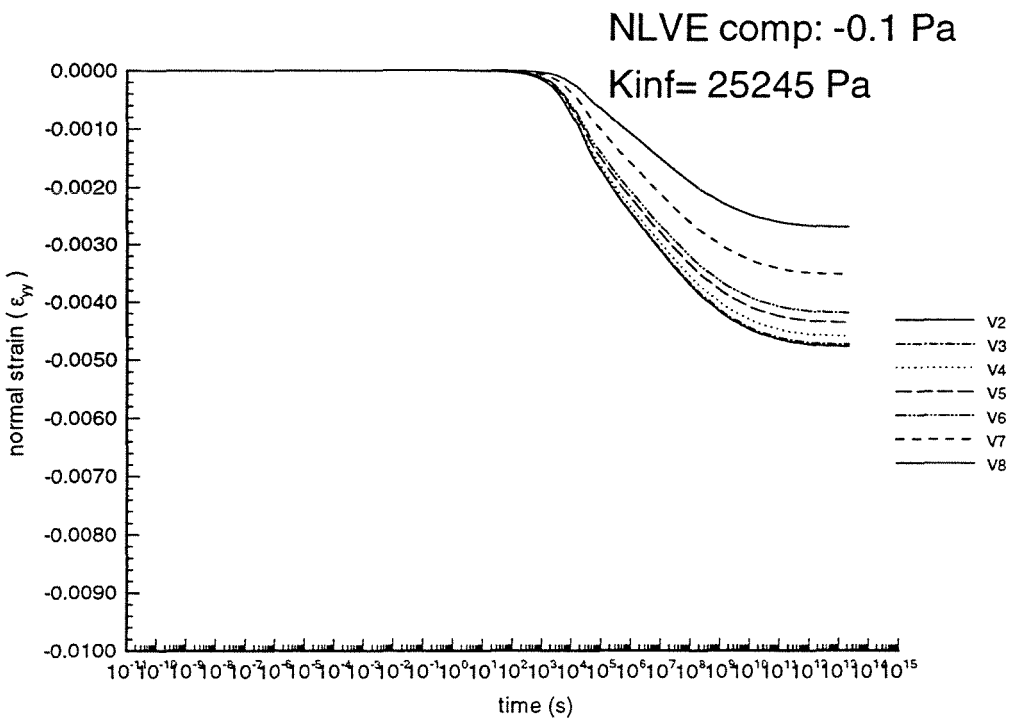


Figure 3.27: $\epsilon_{yy}(t)$, NLVE, compression, Material#1

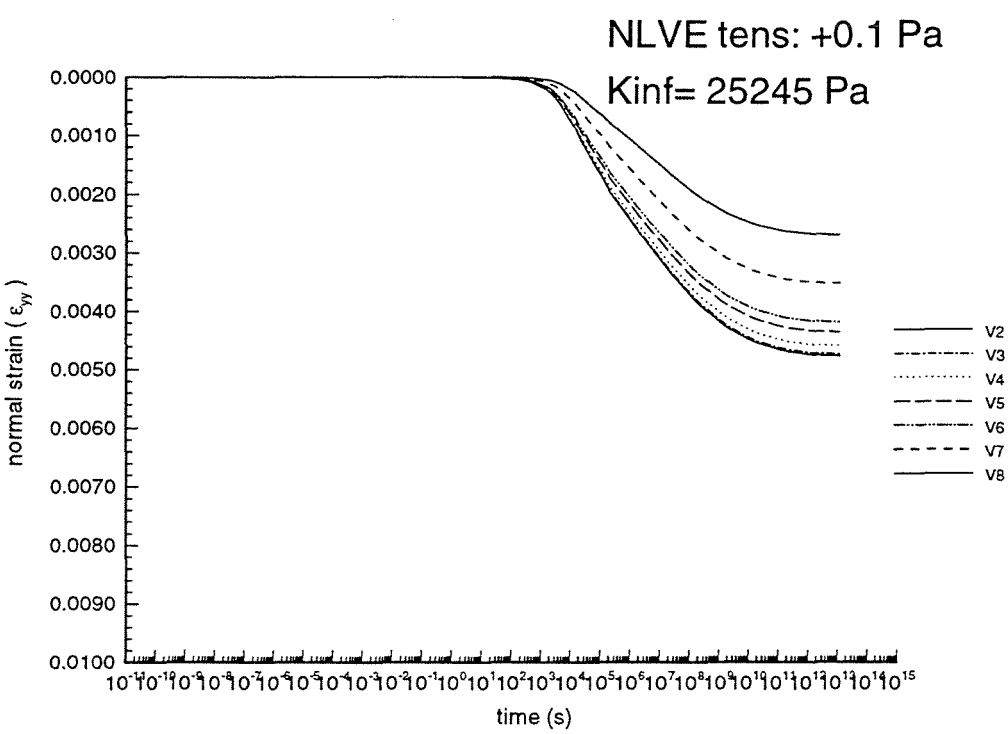


Figure 3.28: $\epsilon_{yy}(t)$, NLVE, tension, Material#1

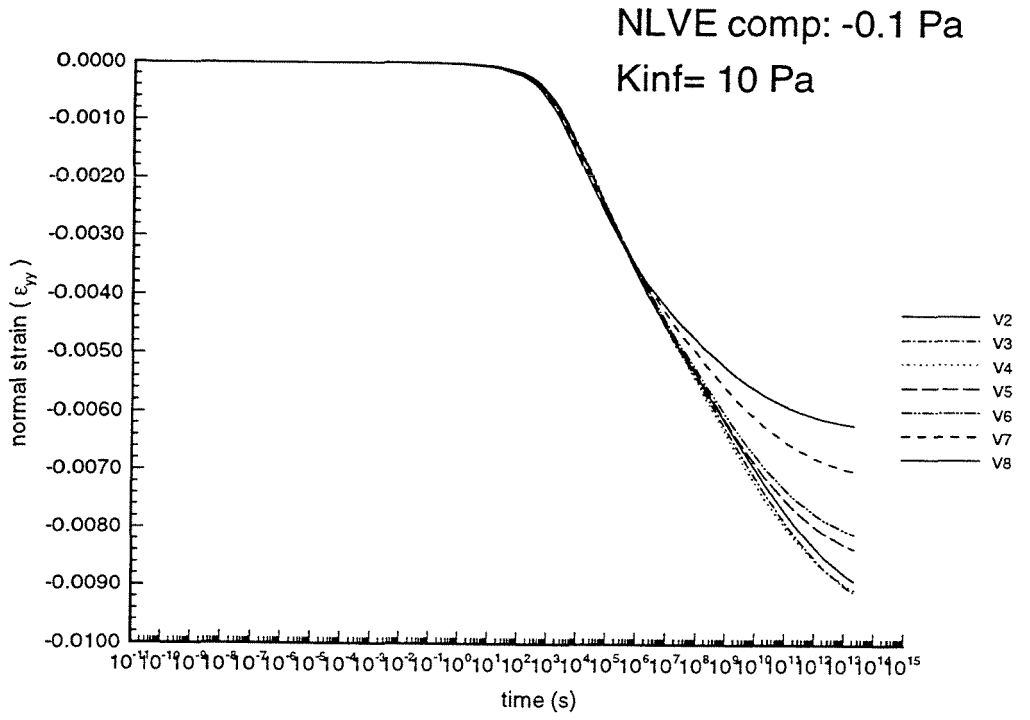


Figure 3.29: $\epsilon_{yy}(t)$, NLVE, compression, Material#2

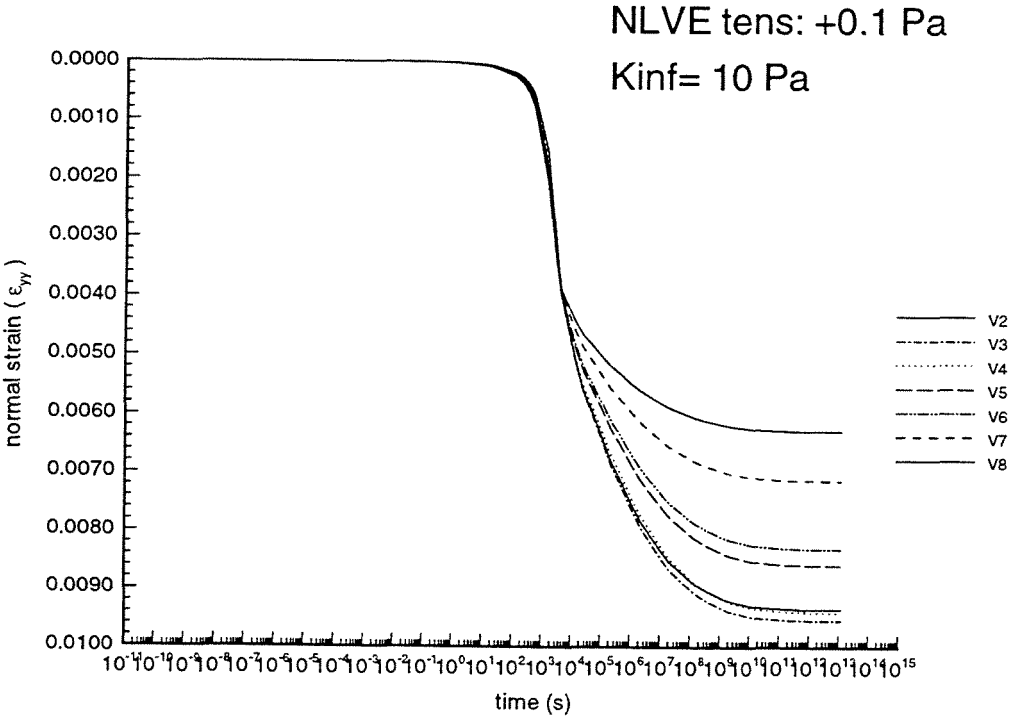


Figure 3.30: $\epsilon_{yy}(t)$, NLVE, tension, Material#2

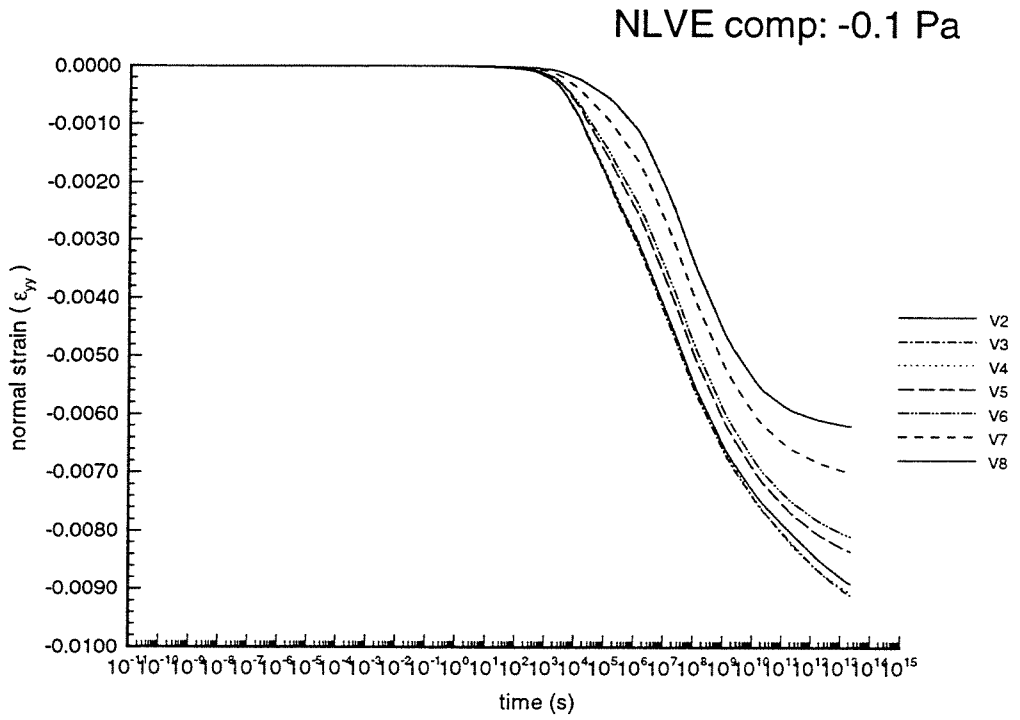


Figure 3.31: $\epsilon_{yy}(t)$, NLVE, compression, Material#3

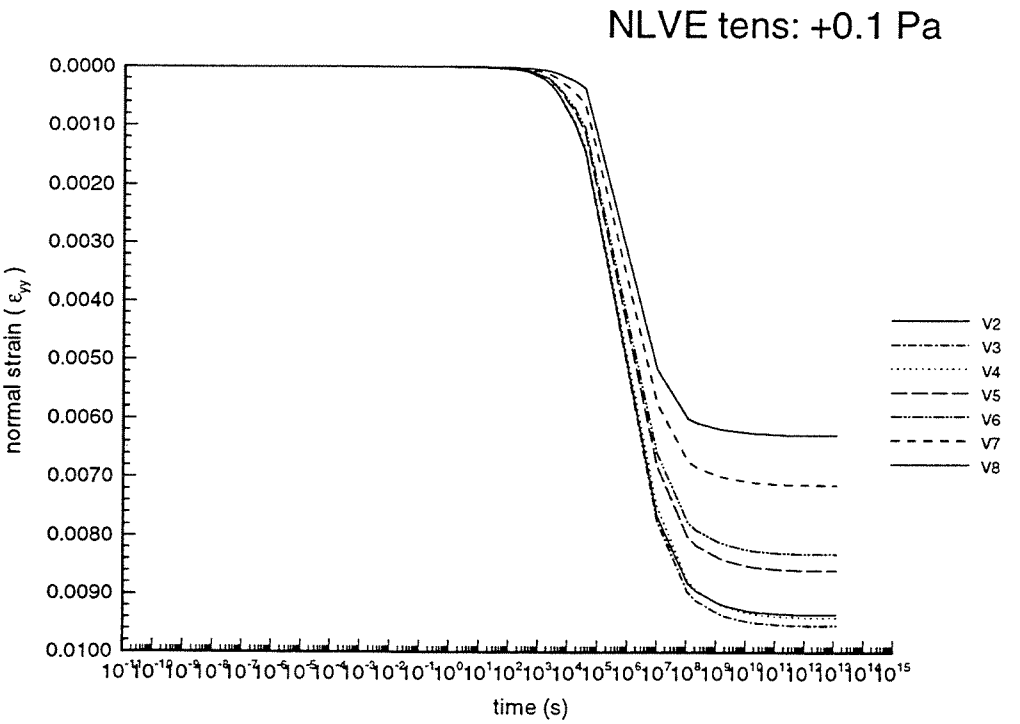


Figure 3.32: $\epsilon_{yy}(t)$, NLVE, tension, Material#3

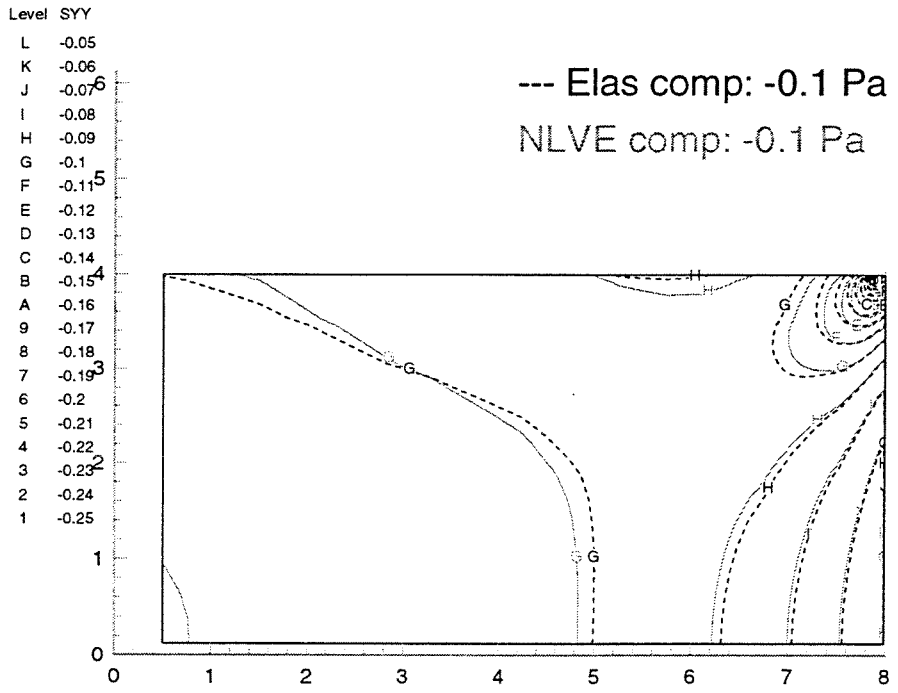


Figure 3.33: Contour plot σ_{yy} , Comparison of NLVE and Elastic solution, compression, long term

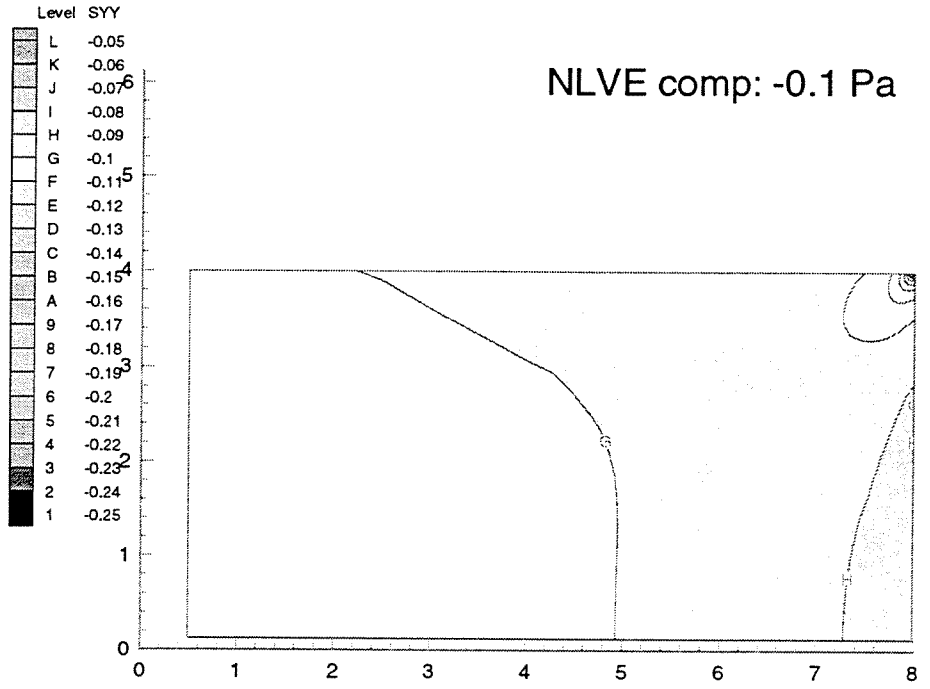


Figure 3.34: Contour plot σ_{yy}, NLVE, compression, initial time

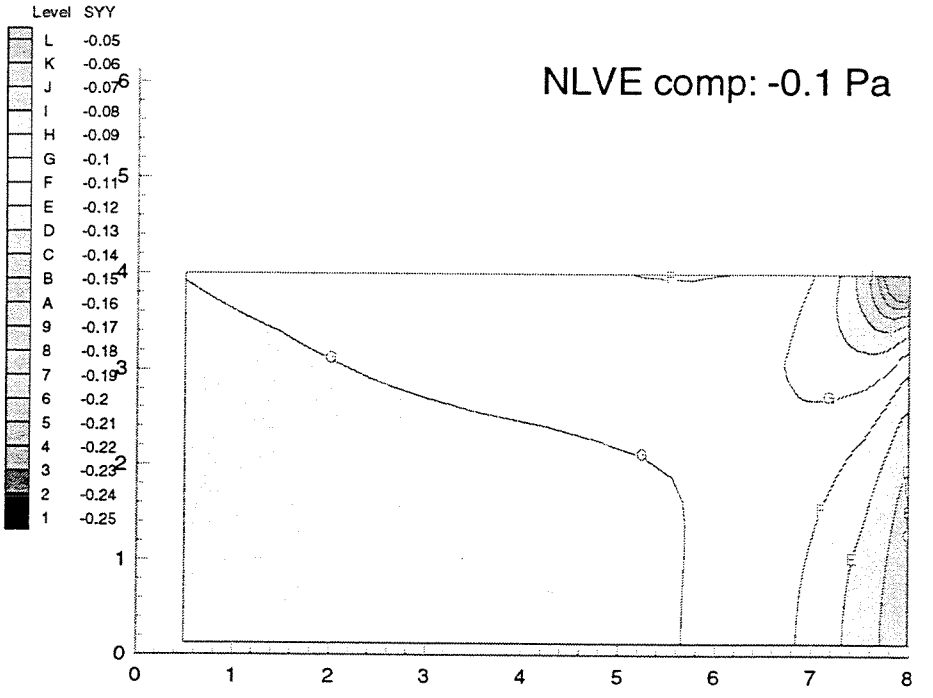


Figure 3.35: Contour plot σ_{yy}, NLVE, compression, middle time

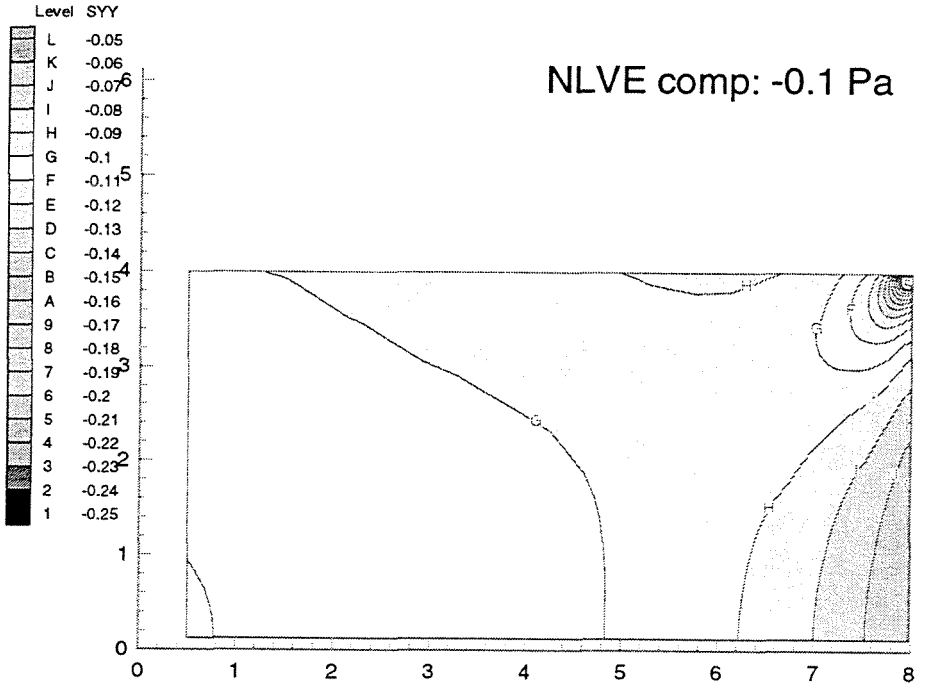


Figure 3.36: Contour plot σ_{yy}, NLVE, compression, final time

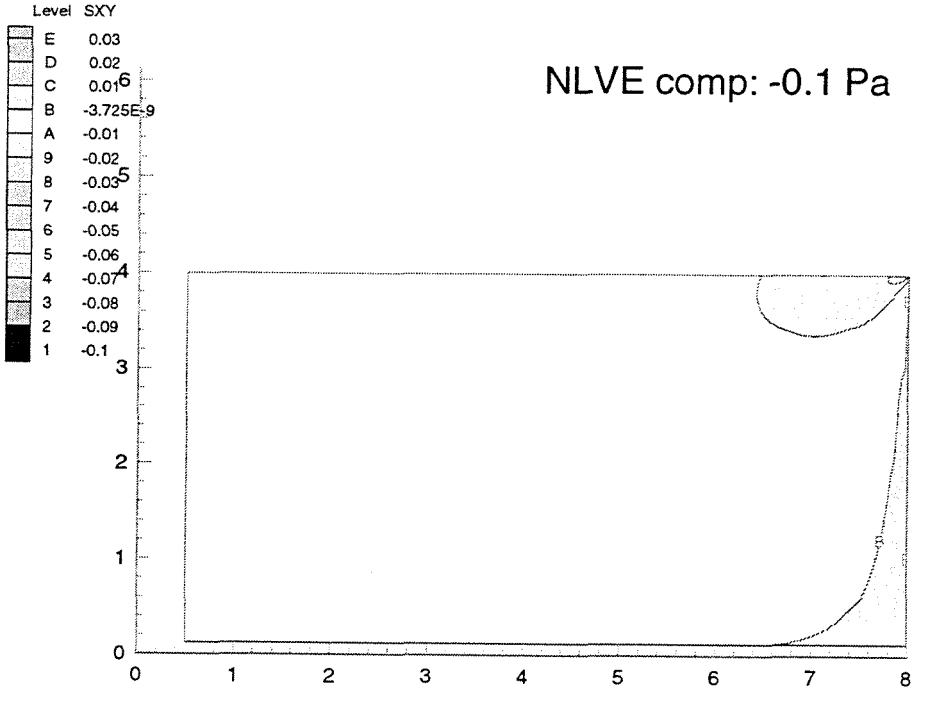


Figure 3.37: Contour plot σ_{xy}, NLVE, compression, initial time

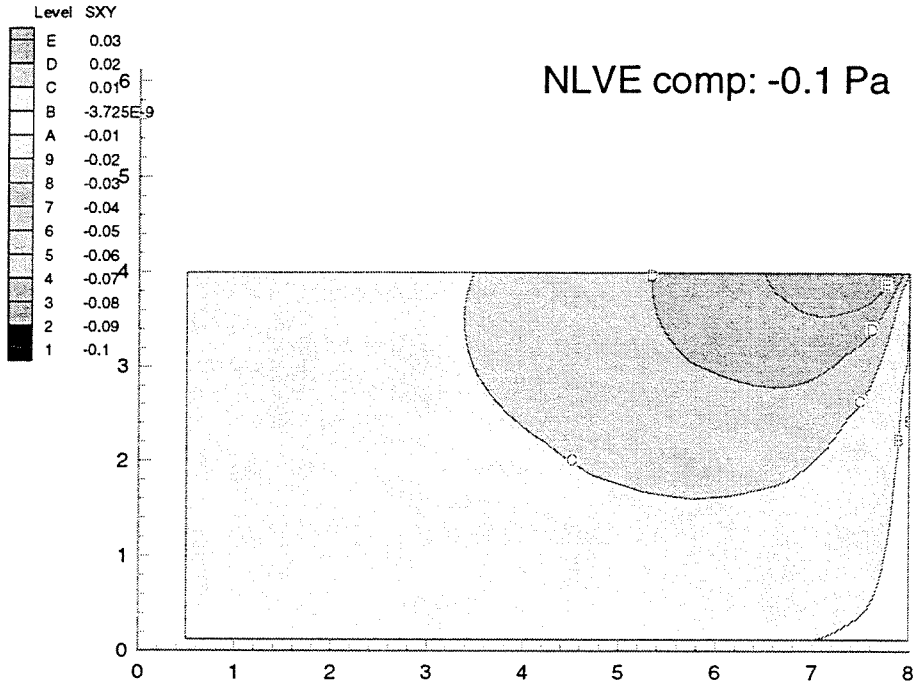


Figure 3.38: Contour plot σ_{xy}, NLVE, compression, middle time

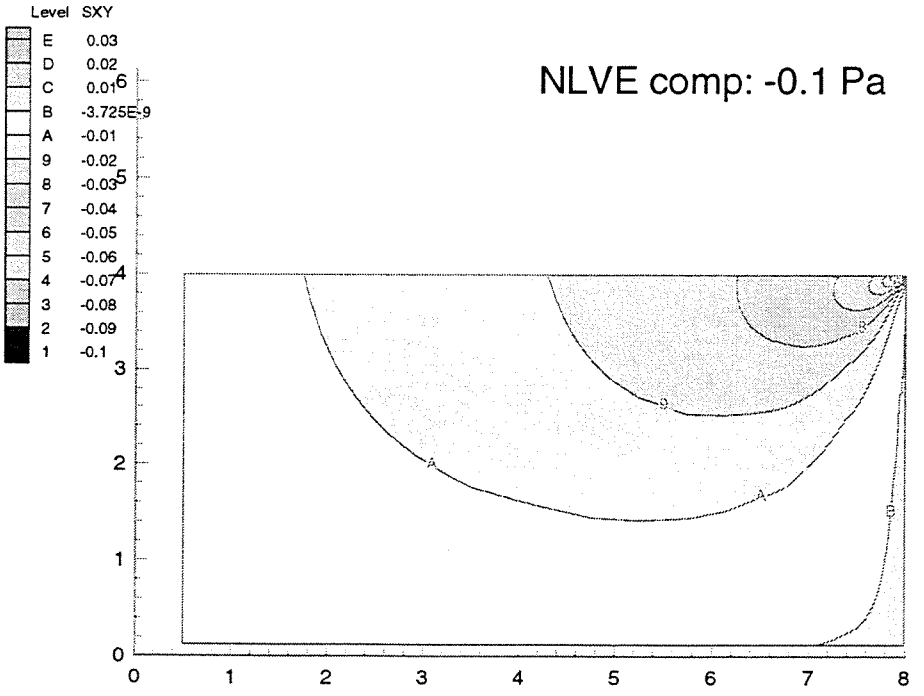


Figure 3.39: Contour plot σ_{xy}, NLVE, compression, final time

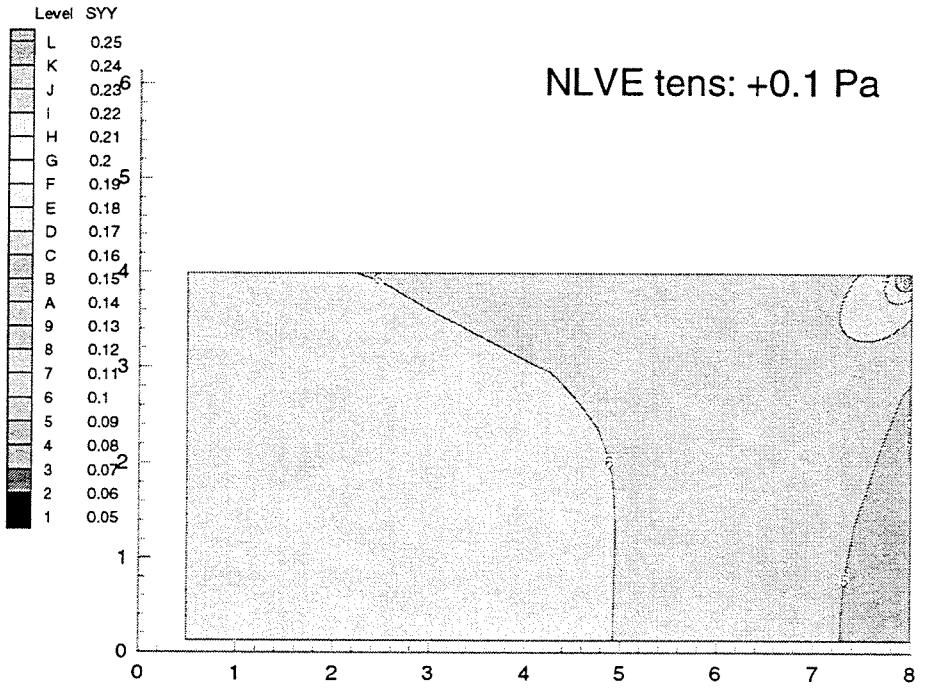


Figure 3.40: Contour plot σ_{yy}, NLVE, tension, initial time

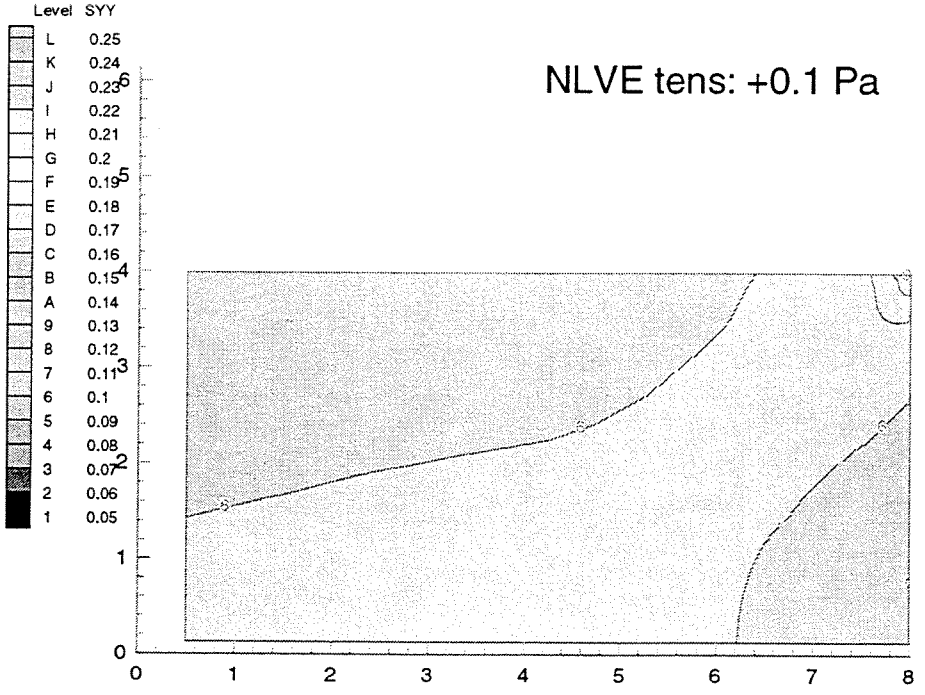


Figure 3.41: Contour plot σ_{yy}, NLVE, tension, middle time

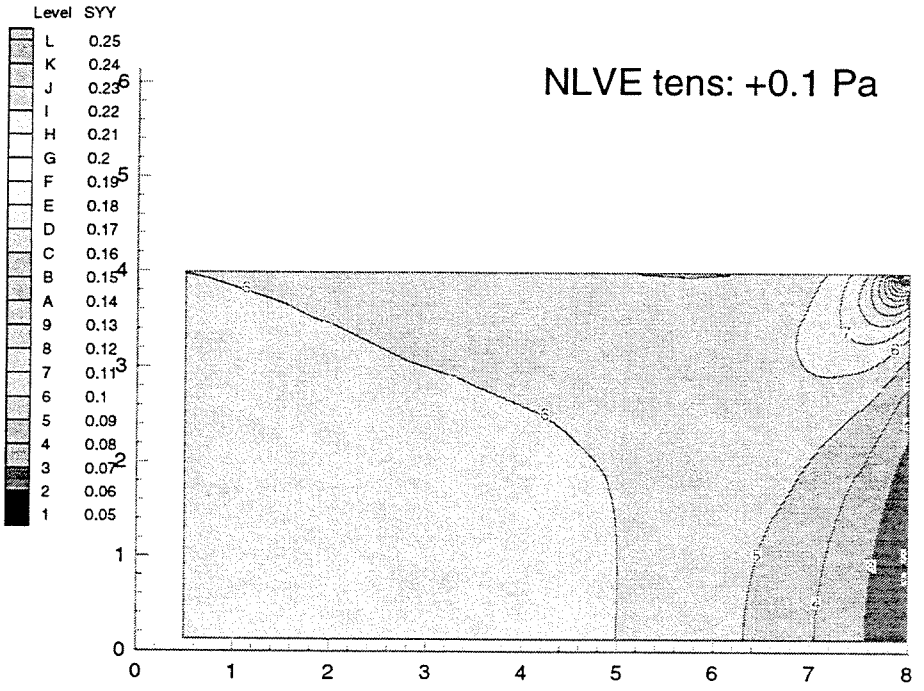


Figure 3.42: Contour plot σ_{yy}, NLVE, tension, final time

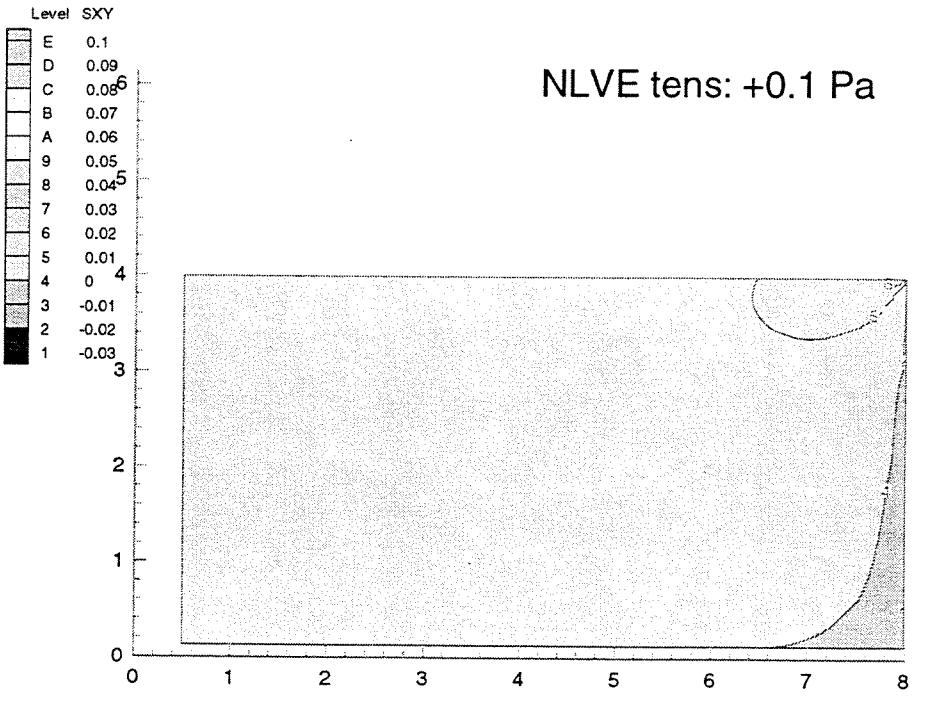


Figure 3.43: Contour plot σ_{xy}, NLVE, tension, initial time

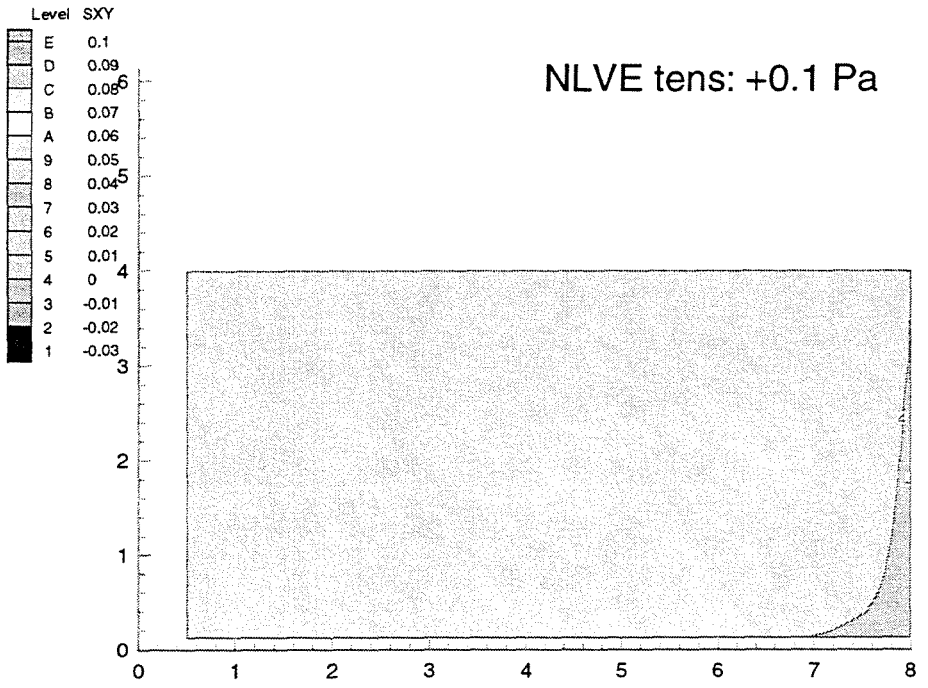


Figure 3.44: Contour plot σ_{xy}, NLVE, tension, middle time

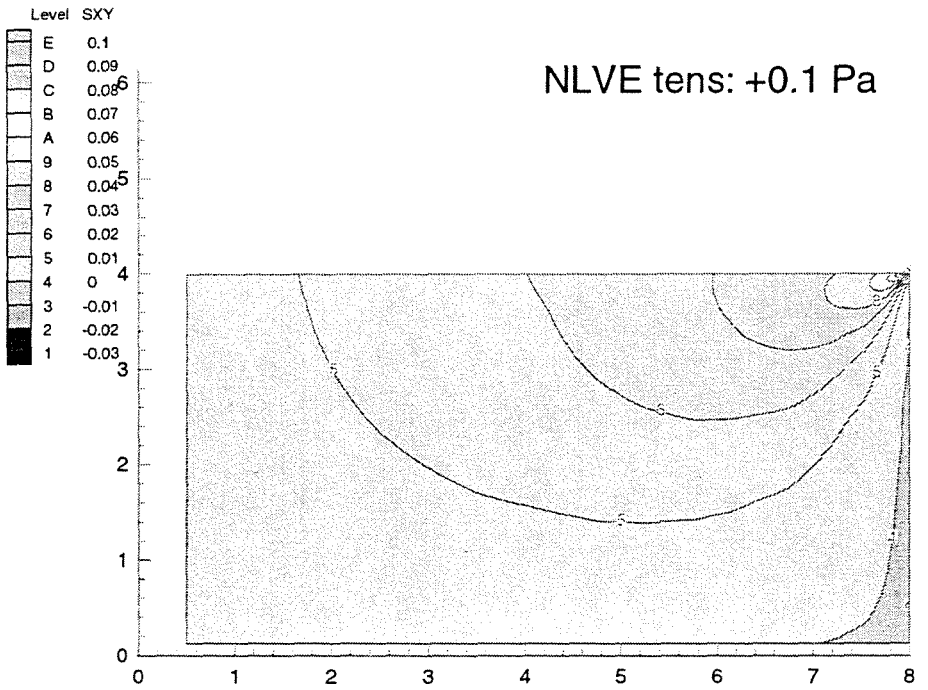


Figure 3.45: Contour plot σ_{xy}, NLVE, tension, final time

Chapter 4 Conclusion

A boundary value problem of a rectangular sample of polymer (PVAc) clamped by two metallic (Aluminium) rigid grips subjected to a compression and tension load was simulated with a finite element code. Three different material models (with the original bulk modulus $K(t)$ of PVAc, with a lower bulk modulus $K(t)$ and with matching relaxation moduli $K(t)$ and $G(t)$) were compared.

The time behavior of the normal stress $\sigma_{yy}(t)$ and shear stress $\sigma_{xy}(t)$ show that decreasing $K(t)$ values command the stress values along the central row to 'converge' close to -0.1 Pa (the applied load) and even intersect and invert their order. On the other hand, decreasing $G(t)$ values command the same stress values to 'scatter' away from -0.1 Pa and, in this way, the differences between the stress values along the central line get larger. Given these opposite effects of the relaxation moduli, the stress response presents a highly non-monotonic behavior. The stress curves even exhibit local 'peak(s)' between their initial and long term values.

The positions of the relaxation moduli with respect to one another also plays an important role in the time-dependent stress response. And two different values for the shift between the beginning of $K(t)$ and $G(t)$ lead to radically different stress responses. For instance, Material #2 (which has the beginning of its relaxation moduli separated by five time decades) presents a 'double peak' and 'stress inversion' whereas Material #3 (which has the beginning of its relaxation moduli at the same time) has a 'single peak' and no 'inversion' (see Fig. 3.3 and 3.4). The only difference between Material #2 and #3 is the shift. All other parameters (K_∞, G_∞, Prony components K_p and G_p, applied load) are identical.

Another important factor is the material's compressibility, namely the inverse of the bulk modulus $K(t)$. A material presenting a high bulk modulus, that would lead to a negligible volume dilatation (or contraction) $\triangle\epsilon_{kk}$, is not affected by the creep acceleration or retardation phenomena. In this situation, tension or compression tests

give identical results (after changing the signs) as seen in Figs. 3.2, 3.17 and 3.18. The LVE model is perfectly valid in the case of a nearly-incompressible material.

However, in the case of sufficient volume dilatation (or contraction), the NLVE constitutive model affects the response time scaling differently according to the loading that is applied. For instance, the stress responses of the compressive and tensile tests present noticeably strong differences in the time range and the fluctuation pattern (see Figs. 3.19 and 3.20). The compressive and tensile tests are no more just 'simple mirrors' of one another. Notice also that the long term stress values of the NLVE model differ from the stress values of an elastic model, provided with the long term material constants K_∞, G_∞ (see Fig. 3.36 and 1.13). In contrast, the long term stress values for the LVE model and elastic model are identical (see Fig. 3.13 and 1.13), because the long term LVE response is an elastic response. This proves the necessity of using the NLVE model over the LVE model.

One restriction in this current NLVE constitutive model based on the free volume theory is that the inherent variables (free volume, shift factor, internal time) depend only on the volumetric deformations. For instance, simple shear deformations conserve the volume and they cannot create any shift factor effect. Additionally superposed to another boundary value problem, shear deformations should not modify the response. But some recent multi-axial experiments (tension-torsion, compression-torsion) [Lu, 1996] suggest that the presence of shear strain does actually affect the material response.

One direction for future research could be to model the influence of the shear deformations in an extension of the free volume theory. This could give a more general nonlinear viscoelastic constitutive behavior.

Bibliography

[1] G. U. Losi and W. G. Knauss (1992), *Free volume theory and Nonlinear Thermoviscoelasticity*, Polymer Eng. and Science, Vol 32, pp 542-557.

[2] C. N. Duong and W. G. Knauss (1995), *A Nonlinear Thermoviscoelastic Stress and Fracture Analysis of an Adhesive Bond*, J. Mech. Phys. Solids, Vol 43, pp 1505-1549.

[3] W. G. Knauss and I. J. Emri (1987), *Volume Change and the Nonlinear Thermoviscoelastic Constitutive Relations of Polymers*, Polymer Eng. and Science, Vol 27, pp 86-100.

[4] J. D. Ferry and R. A. Stratton (1960) , Kolloid-Zeitschrift, Vol 171, pp 107.

[5] C. N. Duong (1994), *A Nonlinear Thermoviscoelastic Stress and Fracture Analysis of an Adhesive Bond*, Ph.D. thesis, California Institute of Technology.

[6] G. U. Losi (1990), *Nonlinear Thermoviscoelastic Behavior of Polymers*, Ph.D. thesis, California Institute of Technology.

[7] R. L. Taylor and K. L. Pister and G. L. Goudreau (1970), *Thermomechanical Analysis of Viscoelastic Solids*, Intern. J. for Num. Meth. in Engr, Vol 2, pp 44-59.

[8] T. J. R. Hughes (1980), *Generalization of Selective Integration Procedure to Anisotropic and Nonlinear Media*, Intern. J. for Num. Meth. in Engr, Vol 15, pp 1413-1418.

[9] T. Belytschko, T. Tsay and W. K. Liu (1981), *A Stabilization Matrix for the Bilinear Mindlin Plate Element*, Comp. Meth. App. Mech. Engr, Vol 20, pp 313-327.

[10] O. C. Zienkiewicz and R. L. Taylor (1994), *The Finite Element Method*, McGraw-Hill Book Company, Chap 15, p 436.

Appendix A The Finite Element Program FEAP

A.1 The FE Code Input and Output

The FE code used during this research project is originally called FEAP. The core of the program originated from the University of California, Berkeley (Zienkiewicz and Taylor, 1967). After, the code was enhanced at Brown University, Providence, RI, and finally arrived at the California Institute of Technology, Pasadena, to be further extended. The nonlinear viscoelastic constitutive model was then added to the program by Caltech Graduate students, G. Losi (1990) and C. Duong (1994). This program, modified three-four times by different persons, is the version used in this thesis project. In its current version, the source code consists of roughly 10,100 lines.

A.1.1 Input file

The basic information to be fed to the executable code is in the input file called `input4.d`. FEAP is commanded by a macro-command language. This macro-language entered in `input4.d` is used to specify all the needed information and all the necessary instructions to be accomplished, in order to solve a specific problem. The input file is separated in two parts: first, a set of geometrical data and second, a set of instructions. A simple input example file is presented below.

Set of geometrical and material data

The input file begins with FEAP. The macro FEAP must `always` be placed at the very beginning of the input file. After, it can be followed by any title. For instance,

`FEAP -nonlinear viscoelastic model`

FEAP does not have a capability of automatic mesh refinement. The mesh is fixed after its definition in the input file and cannot be changed. These constants characterizing the mesh are detailed and explained below. They are to be entered in the specific order:

```
(format 9i5, 2i5)
numnp numel nummat ndm ndf nen nsdm nqdm nquad
nalum nfirst
```

where

`numnp`	=	number of nodal points
`numel`	=	# of elements
`nummat`	=	# of different materials
`ndm`	=	dimension of space (2 usually)
`ndf`	=	# of degrees of freedom (2 usually)
`nen`	=	# of nodes per element (4 for rectangular elements)
`nsdm`	=	dimension of stress array (4 for plane problems)
`nqdm`	=	# of internal variables
`nquad`	=	# of Gauss quadrature pts (5 for rectangular elements)
`nalum`	=	# of elastic elements in the mesh
`nfirst`	=	global # of the first elastic element

The number of internal variables `nqdm` varies with the number of Prony components of the polymer's shear and bulk moduli `(numshr, numblk)`. Use the formula:

$$\texttt{nqdm} = 14 + 4 \times \texttt{numshr} + \texttt{numblk} + 1 \tag{A.1}$$

If elastic elements (e.g., aluminium) are used, they must be grouped together in a consecutive numbering. `nalum` is the total number of elastic elements, while `nfirst` is the number of the first one. If no elastic element is used, set `nalum` to zero and `nfirst` greater than numnp.

The information concerning the mesh nodes coordinates, the element connectivities, the material constants, the boundary conditions and the applied boundary forces/displacements need to be preceded by the corresponding macros COOR, ELEM, MATE, BOUN and FORC. This information is to be put in given formats:

COOR (format 3i5, 2f15.0)

node# generation_flag generation_increment x-coord. y-coord.

If the generation flag is set to 1, FEAP generates uniformly spaced points on the segment between this node and the next listed node. The generated nodes are numbered according to the generation increment step.

For instance,

```
 1    1    2   0.00000   0.00000
11    0    0  10.00000   0.00000
```

creates 6 nodes 1,3,5,7,9,11 uniformly on the x-axis from position 0. to 10.

ELEM (format 7i5)

elem# material# node1 node2 node3 node4 generation_flag

A material# is assigned to each element (see MATE). The generation flag works the same way as for COOR.

MATE has two different formats for viscoelastic and elastic materials:

Viscoelastic material :

material_set# element_type# (5 for viscoelastic)

x-gravity y-gravity mode mass_density **spar**

B_factor f_{ref} f_{init}

where the B_factor is a material constant used in the free volume theory equations, f_{ref} is the reference free volume, f_{init} is the initial free volume and **spar** is the Belytchko integration parameter (for nearly-incompressible solids: Belytschko, 1981).

The Prony series components (K_p, τ_p' and G_p, τ_p'') of the relaxation functions $K(t)$ and $G(t)$ follow:

MODU

α_{l_v} K_∞ #_blk_comp.

K_p τ_p'

SHEA

α_{l_v} G_∞ #_shr_comp.

G_p τ_p''

PHI

α_{l_v} θ_∞ #_comp. (put zero here)

C_{p_l} C_{p_g} f_{crit} T_{ref} k

where α_{l_v} is the rubbery volumetric coefficient of thermal expansion. K_∞ is the long-term bulk modulus; K_p and τ_p' are respectively the modulus and time relaxation for the bulk p-component. G_∞ is the long-term shear modulus; G_p and τ_p'' are respectively the modulus and time relaxation for the shear p-component; and C_{p_l}, C_{p_g}, f_{crit}, T_{init} and k are the rubbery heat capacity per unit volume, the glassy heat capacity per unit volume, the critical free volume (free volume at the glass transient temperature), the initial temperature and the heat conductivity, respectively.

Elastic material :

The beginning is similar to the viscoelastic format.

material_set# element_type# (5 for viscoelastic)

x-gravity y-gravity mode mass_density

α k shear_mod bulk_mod ρC_v

where α is the coefficient of thermal expansion, k is the conductivity and ρC_v is the heat capacity per unit mass.

BOUN (format 2i5, 2f10.0)

node# generation_increment x-dir_flag y-dir_flag

The flags can be set to
- 1 displacement constrained
- 0 displacement free
- −1 same as 1 but can be automatically generated

FORC (format 2i5, 2f10.0)

node# generation_increment x-dir. y-dir.

The values in x- and y-direction represent either an applied nodal force or a displacement. It is a displacement if the corresponding flag in BOUN is set to 1 or -1. Otherwise, it represents an applied nodal force. When a node is not specified in BOUN, the flags are set to 0 by default.

END marks the set of geometrical data.

Set of instructions

The macro language enables FEAP to solve different kinds of problems without having to modify the code each time. Different combinations of macro-instructions can be tailored to solve any specific problem. Here the typical set of instructions for a viscoelastic problem is presented.

MACR	indicates the beginning of the instructions
AUTO	starts a restart file called 'rest4.d'
TOL	sets the tolerance
TTAB	reads the time temperature table
DT	sets the time increment
TIME	increments the time by the DT value
TDIS	sets uniform temperature in the solid
CEQS	integrates the constitutive equations to get the stresses
LOOP n	repeats n times the set of instructions enclosed between LOOP, NEXT
NEXT	

To solve one time increment for a viscoelastic problem:

LOOP 25	
TANG	builds the tangent stiffness matrix
FORM	builds the out-of-balance (or residual) forces vector
SOLV	solves the system for the displacements
CEQS	recalculates the internal time and updates the stresses
NEXT	
UPDT	updates all variables for all elements including the stresses

The Newton-Raphson method tries to iteratively solve the nonlinear system of equations with a maximum of 25 iterations. After 25 unsuccessful iterations, the program considers the case as non-convergent and stops. The convergence criterion, based on the residual vector, is tested each time in the FORM macro. If the criterion is satisfied before 25 iterations, the code exits the LOOP (to go to UPDT).

The macros CEQS and UPDT are programmed very similarly (see the code excerpts) and may seem to be doing the same function. However, they are to be inserted at very different parts of the instruction set. CEQS calculates the stress and updates some variables for a Newton-Raphson iteration; it needs to be put in the iteration loop. UPDT updates the stress and all other element variables after a time step. It has to be understood that several Newton-Raphson iterations are necessary to solve one incremental time step.

The time-temperature table, entered after the end of the macros (END), describes the temperature history of the simulation. The first column is the time and the second represents the difference $\triangle T$ between the current temperature and the initial temperature. The initial temperature was entered in the program through another variable, the initial free volume f_{init}. Only f_{init} matters! The temperature T_{init} entered after macro MATE is of no importance (a dummy). In the isothermal case, the differences $\triangle T$'s are all zeros.

The other macro-instructions MA01, HIST STRA, STRE are used for graphics data purpose (see Graphic data output).

A.1.2 Output file

After running the program, a good idea is to examine the output file called 'outpu4.d'. Most of the instructions and data are rewritten in it. If the case did not run properly, the output file should be checked to verify that the data was entered correctly.

The output file also enables to monitor the code's execution after each time increment step-by-step. The number of iterations for the previous time step is written. The current time is indicated. And for each iteration, $RNMAX$ represent the maximum residual norm, RN, the current residual norm and TOL, the tolerance. Convergence is reached when $RN/RNMAX < TOL$ is fulfilled.

When there is no convergence, the time increment DT should be checked. It may be too big and should be decreased. Or DT may be too small. For that case, it appears that the mechanical variations during one time increment are very small. Consequently, the maximal residual norm RNMAX is also very small (around 10^{-2}). And RN cannot go really smaller. The tolerance TOL should be decreased or the time increment DT increased.

A.1.3 Graphic data output

Some macros create the output files that are used for the graphics display. The basic information is stress and strains. The graphic software used is TECPLOT.

The files `syy.d`, `sxx.d`, `eyy.d` contain respectively the curves of the stresses σ_{yy}, σ_{xy} and ϵ_{yy} versus time. They can be visualized in the XY-Plot of TECPLOT.

The files `distsyy.d` and `distsxy.d` contain the field spatial distribution of σ_{yy} and σ_{xy}. The stress fields can be visualized by Contour-Plot in TECPLOT. Some headers, specific to the Contour-Plot format, also need to be added before each TECPLOT zone; they are of the form:

```
ZONE
VARIABLE
```

The file `fort.10` contains the deformed meshes at different times and is visualized through Mesh in TECPLOT.

The graphic data macros are:

MA01	prints the deformed mesh information at the given time (one zone) in the file `fort.10`.
HIST STRE	elmt# component# prints the current time and the current stress value (one line) in the respective file.
HIST STRA	elmt# component# works for strain the same way as for stress.
STRE	prints the field distribution of σ_{yy}, σ_{xy} and ϵ_{yy} at the given time in the respective file.

The stresses can only be calculated at the Gauss points of the elements. You need to specify which element you want and which component at which Gauss point you want. There are five Gauss points in a rectangular element; the 5^{th} Gauss point is the center point. The 4 stress components are in the order σ_{xx}, σ_{yy}, σ_{zz}, σ_{xy}. For example, the stress components σ_{xx}, σ_{yy}, σ_{zz}, σ_{xy} at the center point of the element are respectively numbered $17, 18, 19, 20$.

These macros can be modified in the file `pmacr.f` or `ma01.f`. The subroutine `pmacr.f` is like a library for macros. The subroutines used by the macros are called from within `pmacr.f`. By analyzing the code in `pmacr.f`, one can get to understand all these macros. Then, the macros can be adapted to any specific need.

A.2 Structure

As received, FEAP used to be stored in one unique gigantic source file called `feap.f`, which contained the main and all the subroutines, sub-subroutines, etc. ... Unfortunately, no structure was clear. Now, the finite element source has been separated and structured into several files:

`pmain.f, pmesh.f, pmacr.f, pform.f, pelmt05.f, pnonlinve.f and ma01.f.`

These files gather all the program's subroutines into groups dedicated to one general task in the code. Each task is more or less described by the name of the file.

Let's explain their functions:

- `pmain.f` contains -as you expect- the main, but also some other subroutines high in the hierarchy. For instance, the subroutine `pcontr.f` controls the memory allocation and generates the code arrays.

- `pmesh.f` regroups the macros that are used in the set of geometrical and material data.

- `pmacr.f` regroups the macros that constitute the set of instructions. This is the biggest part. If you want to add a functionality, i.e., a macro to the program, you need to edit `pmacr.f`. It contains our beloved macros TANG, FORM, SOLV, CEQS, STRE.

- `pform.f` builds the global arrays from the element arrays. Most of the operations just mentioned before involve building a global array. Global arrays are formed from the element arrays in the process called the assembly method. It is a basic finite element method. In one sense, `pform.f` makes the interface between `pmacr.f` and `pelmt05.f`.

- `pelmt05.f` takes care of creating the element arrays. It contains the element subroutines. In this subroutine, we have the nonlinear viscoelastic constitutive relations implemented.

- `pnonlinve.f` regroups most of the subroutines that are called by subroutine `elmt05` and that are particularly specific to the nonlinear viscoelastic theory.

To give some examples, CEQS usually calls the subroutine `elmt05`, that itself calls some of the following subroutines contained in `pnonlinve.f`:

- The function `timshf(a,epskk,ma)` calculates the time shift factor ψ (actually $1/\psi$) according to the formulas

$$\log \psi = B\left(\frac{1}{f(t)} - \frac{1}{f_0}\right) \tag{A.2}$$

$$f(t) = f_{init} + \triangle f(t) \tag{A.3}$$

- The function `delvf(frevol,ma)` calculates the proportionality coefficient $\delta(f(t))$ that is used in the way:

$$f(t) = f_{init} + \int_0^t \delta(f(\tau)) d\epsilon_{kk}(\tau) \tag{A.4}$$

$$\delta(f) = \frac{\beta_{f_0}\phi(f)}{1+\beta_{f_0}\phi(f)} \tag{A.5}$$

$$\phi(f) = \frac{1}{1-\frac{A}{2}\left(1-\frac{B}{f}\right)\exp\left(\frac{B}{f}-\frac{B}{f_0}\right)} \tag{A.6}$$

- The subroutine `strupd(...)` updates the stresses and entropy. It calculates the stress increment corresponding to the strain increment of this iteration by using the FE algorithm for viscoelasticity (see section Stresses in General Numerical Scheme). Understand that for each operation it calculates the stress increment on the basis of the stress values at the previous step. More precisely, it needs the values of each Prony component contribution to the isotropic and deviatoric stress $(\overline{\sigma}^{(p)}, s_1^{(p)}, s_2^{(p)}, s_3^{(p)})$. These values are stored as internal variables `ql(14+kk,iq)`. Now, it can be understood the reason why the number of internal variables has to be set to $\texttt{nqdm} = 14 + \texttt{numblk} + 3 \times \texttt{numshr} + 1$.

Another remark: The FE algorithm decomposes the stress and the stress increments into their isotropic and deviatoric parts; the parts are calculated separately and then reassembled.

A.3 Summary of the FE Algorithm

The basic leitmotiv of the FE method is to solve the equation

$$K\triangle U = f_{ext} - f_{int} \tag{A.7}$$

where K is the tangent stiffness matrix, $\triangle U$ is the displacement we are solving for, f_{ext} is the external load, and f_{int} is the internal forces.

The question is how to build the tangent stiffness matrix K and the internal forces

vector f_{int}. This FE algorithm always calculates two parts: the isotropic and deviatoric parts. Building K and f_{int} goes along that way. The separation into an isotropic and deviatoric part proves particularly useful when formulating the numerical scheme for nearly-incompressible materials (a 'sort' of $\bar{\mathrm{B}}$-method).

The stress σ is also decomposed and calculated into its volumetric σ^{vol} and deviatoric s part. The equation becomes:

$$\left(K^{vol} + K^{dev}\right) \triangle U = f_{ext} - f_{int}^{vol}(\sigma^{vol}) - f_{int}^{dev}(s) \tag{A.8}$$

$$\text{where} \quad \sigma = \sigma^{vol} + s \tag{A.9}$$

In this duality, the isotropic part is linked to the bulk modulus while the deviatoric part relates to the shear modulus. Remember that the bulk relaxation modulus $K(t)$ and shear relaxation modulus $G(t)$ are expressed in terms of the Prony series:

$$K(t) = K_\infty + \sum_{p=1}^{M} K_p \exp(-t/\tau_p'') \tag{A.10}$$

$$G(t) = G_\infty + \sum_{p=1}^{N} G_p \exp(-t/\tau_p') \tag{A.11}$$

The equations used to calculate the tangent stiffness matrix, the internal forces and the stresses are presented, below.

A.4 General Numerical Scheme

A.4.1 Tangent stiffness

$$K^{dev} = \sum_{q=1}^{Q} W_q B^{e^T}(g_q) \left(2\widetilde{G}(g_q)\,[R_2]\right) B^e(g_q) \tag{A.12}$$

$$K^{vol} = \sum_{q=1}^{Q} W_q B^{e^T}(g_q) \left(3\widetilde{K}(g_q)\,[R_1]\right) B^e(g_q) \tag{A.13}$$

where Q is the number of Gauss points, g_q is the natural position and W_g the weight of

each Gauss point, B^e is the matrix in the strain-displacement relation $\epsilon(x) = B^e(x)U$ and the pseudo shear and bulk modulus $\tilde{G}$ and $\tilde{K}$ can be calculated [Taylor, 1970] by

$$\tilde{G}(g_q) = G_\infty + \sum_{p=1}^{N} G_p \tau'_p \frac{1 - \exp\left(-\triangle\xi(g_q)/\tau'_p\right)}{\triangle\xi(g_q)} \tag{A.14}$$

$$\widetilde{K}(g_q) = \overline{K}_\infty + \sum_{p=1}^{M} K_p \tau''_p \frac{1 - \exp\left(-\triangle\xi(g_q)/\tau''_p\right)}{\triangle\xi(g_q)} \tag{A.15}$$

where $\triangle\xi(g_q) = \xi(t, g_q) - \xi(t - \triangle t, g_q)$ is the increment of internal time at the Gauss point.

A.4.2 Internal forces

$$f_{int}^{vol}(\sigma^{vol}) = \sum_{q=1}^{Q} W_q \, B^{e^T}(g_q) \, \sigma^{vol}(t, g_q) \tag{A.16}$$

$$f_{int}^{dev}(s) = \sum_{q=1}^{Q} W_q \, B^{e^T}(g_q) \, s(t, g_q) \tag{A.17}$$

where the current stress $\sigma = \sigma^{vol} + s$ has been computed in a previous call to the macro CEQS.

A.4.3 Stresses

The current stress is evaluated on the basis of the contributions of each Prony component and their values at the previous time step.

$$\bar{\sigma}(t, g_q) = \sum_{p=1}^{M} \bar{\sigma}^{(p)}(t, g_q) + d\epsilon_{kk}\hat{K}_\infty + \epsilon_{kk}(t - \triangle t)\hat{K}_\infty \tag{A.18}$$

$$- \alpha_{l_v} K_\infty \triangle T(t) \tag{A.19}$$

$$s_i(t, g_q) = \sum_{p=1}^{N} s_i^{(p)}(t, g_q) + 2\epsilon_i^{dev} G_\infty \tag{A.20}$$

Each Prony component contribution to the stress can be calculated from the values

at the previous time step. These stress components $\bar{\sigma}^{(p)}, s_1^{(p)}, s_2^{(p)}, s_3^{(p)}$ of the previous time step $t - \triangle t$ are stored as internal variables `ql(14+kk,iq)` for each Gauss point `iq`. The internal variables are updated to the current time values by the macro UPDT.

$$\begin{aligned} \bar{\sigma}^{(p)}(t, g_q) &= \bar{\sigma}^{(p)}(t - \triangle t, g_q) \exp\left(-\frac{\triangle\xi(g_q)}{\tau_p''}\right) \\ &\quad + d\epsilon_{kk}\, K_p\, \tau_p'' \frac{1 - \exp\left(-\triangle\xi(g_q)/\tau_p''\right)}{\triangle\xi(g_q)} \end{aligned} \tag{A.21}$$

$$\begin{aligned} s_i^{(p)}(t, g_q) &= s_i^{(p)}(t - \triangle t, g_q) \exp\left(-\frac{\triangle\xi(g_q)}{\tau_p'}\right) \\ &\quad + 2d\epsilon_i^{dev}\, G_p\, \tau_p' \frac{1 - \exp\left(-\triangle\xi(g_q)/\tau_p'\right)}{\triangle\xi(g_q)} \end{aligned} \tag{A.22}$$

A.5 Numerical Scheme for Nearly-Incompressible Materials

A.5.1 Tangent stiffness

In TANG, we see this formation of the tangent stiffness matrix from the values $\tilde{G}$ and $\tilde{K}$. These pseudo shear and bulk modulus are calculated by the subroutine `moduli`.

In the case of a nearly incompressible material [Hughes, 1980], the numerical scheme is a little modified: the deviatoric part K^{dev} is evaluated by a 2x2 reduced integration, while the volumetric part K^{vol} is evaluated by 1x1 full integration.

$$K^{dev} = \sum_{q=1}^{4} W_q B^{e^T}(g_q) D^{dev}(centroid) B^e(g_q) \tag{A.23}$$

$$K^{vol} = W_{cent} B^{e^T}(cent) D^{vol}(centroid) B^e(cent) \tag{A.24}$$

$$\text{with} \quad D^{vol} = K \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad D^{dev} = 2G \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 0 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.25}$$

The parameter *spar* called the Belytchko stabilization parameter is also used to refine the integration [Belytschko, 1981]. It acts like a 'weight' between the volumetric and deviatoric contributions. In two loops, the algorithm is

1st loop

$$D_1 = D^{dev} + spar \times D^{vol} \tag{A.26}$$

$$K_1 = \sum_{iq=1}^{4} W_{iq} B^{e^T}(iq) D_1(centroid) B^e(iq) \tag{A.27}$$

2nd loop

$$D_2 = (1 - spar) \times D^{vol} \tag{A.28}$$

$$K_2 = W_{cent} B^{e^T}(cent) D_2(cent) B^e(cent) \tag{A.29}$$

and

$$K = K_1 + K_2 \tag{A.30}$$

A.5.2 Internal forces

In FORM, for the case of a nearly-incompressible material, the reduced integration method is similar to that of TANG

$$f_{int}^{dev} = \sum_{q=1}^{4} W_q B^{e^T}(g_q) \sigma^{dev}(g_q) \tag{A.31}$$

$$f_{int}^{vol} = W_{cent} B^{e^T}(cent) \sigma^{vol}(cent) \tag{A.32}$$

$$\text{with} \quad \sigma^{vol} = \begin{pmatrix} \bar{\sigma} \\ \bar{\sigma} \\ \bar{\sigma} \\ 0 \end{pmatrix}, \quad \sigma^{dev} = \begin{pmatrix} \sigma_1 - \bar{\sigma} \\ \sigma_2 - \bar{\sigma} \\ \sigma_3 - \bar{\sigma} \\ \sigma_4 \end{pmatrix} \tag{A.33}$$

With the Belytchko stabilization parameter *spar*,

1st loop

$$\sigma^{(1)} = \sigma^{dev} + spar \times \sigma^{vol}$$

$$= \begin{pmatrix} \sigma_1 - (1-spar)\bar{\sigma} \\ \sigma_2 - (1-spar)\bar{\sigma} \\ \sigma_3 - (1-spar)\bar{\sigma} \\ \sigma_4 \end{pmatrix} \tag{A.34}$$

$$f_{int}^{(1)} = \sum_{iq=1}^{4} W_{iq} B^{e^T}(iq)\sigma^{(1)}(centroid) \tag{A.35}$$

2nd loop

$$\sigma^{(2)} = (1-spar) \times \sigma^{vol} \tag{A.36}$$

$$= \begin{pmatrix} (1-spar)\bar{\sigma} \\ (1-spar)\bar{\sigma} \\ (1-spar)\bar{\sigma} \\ 0 \end{pmatrix} \tag{A.37}$$

$$f_{int}^{(2)} = W_{cent} B^{e^T}(cent)\sigma^{(2)}(cent) \tag{A.38}$$

and $$f_{int} = f_{int}^{(1)} + f_{int}^{(2)} \tag{A.39}$$

The parameter *spar* allows to vary our result between those of the full normal integration ($spar = 1$) and the reduced integration ($spar = 0$). This numerical scheme prevents phenomena such as 'mesh locking' for a nearly-incompressible material. It is incorporated in the code used.

Appendix B Input File Example

```
FEAP -my problem POLYMER RECTANGLE UNDER UNIAXIAL TENSION
   85   64    2    2    2    4    4  119    5
    0  100

COOR
    1    1    0   0.00000   0.00000
   17    0    0   8.00000   0.00000
   18    1    0   0.00000   1.00000
   34    0    0   8.00000   1.00000
   35    1    0   0.00000   2.00000
   51    0    0   8.00000   2.00000
   52    1    0   0.00000   3.00000
   68    0    0   8.00000   3.00000
   69    1    0   0.00000   4.00000
   85    0    0   8.00000   4.00000

ELEM
    1    1    1    2   19   18    1
   16    1   16   17   34   33    0
   17    1   18   19   36   35    1
   32    1   33   34   51   50    0
   33    1   35   36   53   52    1
   48    1   50   51   68   67    0
   49    1   52   53   70   69    1
   64    1   67   68   85   84    0


MATE
    1    5
       0.0       0.0         5       1.0       0.5
       0.1    0.0095    0.0095
MODU
   5.98e-4   25245.0        26
   5.056310      7.7426616E-09
   75.78130      1.8957456E-08
   144.8100      4.6416229E-08
   192.2820      1.1364756E-07
   245.6900      2.7825303E-07
   341.9550      6.8128708E-07
   370.6730      1.6680918E-06
   558.5750      4.0842301E-06
   582.8580      9.9999997E-06
   745.2520      2.4484416E-05
   937.5130      5.9948739E-05
   1043.720      1.4678108E-04
   1253.400      3.5938513E-04
   1425.040      8.7991334E-04
   1668.970      2.1544178E-03
   1445.760      5.2749696E-03
   1622.430      1.2915463E-02
   1664.520      3.1622775E-02
   1061.410      7.7426568E-02
   442.8020      0.1895737
   206.1010      0.4641593
   72.57840       1.136464
   57.20400       2.782557
   57.13970       6.812916
   4.580130       16.68092
   23.66940       40.84228
SHEA
       0.  3.16227766        26
  0.1101795E+01       0.2763996E-02
  0.2182498E+04       0.1190686E-01
  0.1024474E+04       0.5129290E-01
  0.2480911E+04       0.2209618E+00
  0.3623970E+04       0.9518692E+00
  0.1338277E+04       0.4100504E+01
  0.5720489E+03       0.1766433E+02
  0.3336653E+03       0.7609518E+02
  0.1938180E+03       0.3278062E+03
  0.1798677E+02       0.1412138E+04
  0.6957788E+01       0.6083269E+04
  0.3148028E+01       0.2620577E+05
  0.1889946E+01       0.1128904E+06
  0.1470057E+01       0.4863146E+06
  0.1083488E+01       0.2094968E+07
  0.8003443E+00       0.9024802E+07
  0.5977423E+00       0.3887746E+08
  0.4078681E+00       0.1674781E+09
  0.2850480E+00       0.7214699E+09
  0.2019787E+00       0.3107981E+10
  0.1230532E+00       0.1338670E+11
  0.8832651E-01       0.5767648E+11
  0.5256542E-01       0.2484614E+12
  0.3242005E-01       0.1070333E+13
  0.3305381E-02       0.4610831E+13
  0.1414650E-01       0.1986275E+14
PHI
     1.e-3     1.e-3         0
   2.09e+6     1.29e+6    6.509e-3   40.00  .19
    2    3
       0.0       0.0         1       1.0
  21.3E-06  237.11   0.25920E+06   0.67596E+09 0.24174E+07

BOUN
    1    1    1   -1
   17    0    0    1
   69    1    0   -1
   85    0    0    1

FORC
    1    0       0.0       0.0
    2    1     -50.0       0.0
   16    0     -50.0       0.0
   17    0     -25.0       0.0
   69    0      25.0       0.0
   70    1      50.0       0.0
   84    0      50.0       0.0
   85    0      25.0       0.0

END

MACR
MA01
AUTO
TOL          1.0E-05
TTAB
HIST STRA          1        20

DT            1.d-03
LOOP              10
TIME
TDIS
CEQS
LOOP              25
TANG
FORM
SOLV
CEQS
NEXT
```

```
UPDT
MA01
HIST STRA           1        20
NEXT

DT            1.d-02
LOOP              49
TIME
TDIS
CEQS
LOOP              25
TANG
FORM
SOLV
CEQS
NEXT
UPDT
MA01
HIST STRA           1        20
NEXT

DT            1.d-01
LOOP               5
TIME
TDIS
CEQS
LOOP              25
TANG
FORM
SOLV
CEQS
NEXT
UPDT
MA01
HIST STRA           1        20
NEXT


DISP
END
    2
    0.00000E+00    0.00000E+00
    0.20000E+05    0.00000E+00
```

Appendix C Program Listing

```
      PROGRAM GIANCARLO
      REAL*8 RELAX(500)                 ! RELAXATION MODULUS
      REAL*8 CREEP(500)                 ! CREEP COMPLIANCE
      INTEGER NPOINT                    ! NUMBER OF EXPERIMENTAL POINTS
      REAL*8 TIME(500)                  ! TIME INTERVALS
      REAL*8 CTIME(500)                 ! TIME AT WHICH CREEP(K) IS FOUND
      CHARACTER*24 FILIN                ! INPUT FILE NAME
      CHARACTER*24 FILOUT               ! OUTPUT FILE NAME
C
C
C
C
      TYPE *,' THIS IS A PROGRAM THAT INVERTS VISCOELASTIC FUNCTIONS'
      TYPE *,' SUCH AS THE RELAXATION MODULUS OR THE CREEP COMPLIANCE,'
      TYPE *,' USING THE HOPHAM ALGORITHM'
      TYPE *,' '
      TYPE *,' THE DATA HAVE TO BE GIVEN IN THE FORMAT (2E14.6)'
      TYPE *,' EACH LINE OF DATA FILE CONTAINING, IN THE FOLLOWING'
      TYPE *,' ORDER, LOG(tk) AND F(tK), WHERE F IS THE INPUT'
      TYPE *,' VISCOELASTIC FUNCTION'
      TYPE *,' '
      type *,' also note that the log is assumed to be in base 10'
      type *,' '
      TYPE *,' IN ORDER TO AVOID NUMERICAL PROBLEMS, THE PROGRAM'
      TYPE *,' GENERATES A FICTITIOUS DATA POINT AT TIME t=0, WITH'
      TYPE *,' VALUE EQUAL TO THE ONE OF THE FIRST DATA POINT'
      TYPE *,' AS A CONSEQUENCE, IF YOU WANT TO BE CONSISTENT WITH'
      TYPE *,' WHAT THE PROGRAM DOES, YOUR FIRST DATA POINTS HAVE TO'
      TYPE *,' BE QUITE CLOSE TO THE TIME t=0'
      TYPE *,' '
      TYPE *,' THE OUTPUT OF THE PROGRAM HAS THE SAME FORMAT AS '
      TYPE *,' THE INPUT'
      TYPE *,' INPUT FILE: '
      ACCEPT 100,FILIN
      TYPE *, 'INPUT FILE = ',FILIN
      TYPE *, ' OUTPUT FILE: '
      ACCEPT 100,FILOUT
      TYPE *, 'OUTPUT FILE = ',FILOUT
      OPEN (UNIT=1,FILE=FILIN,ACCESS='SEQUENTIAL',STATUS='OLD')
      OPEN (UNIT=5,FILE=FILOUT,ACCESS='SEQUENTIAL',STATUS='NEW')
      DO 10 K=1,500
      READ(1,*,END=20) TIME(K),RELAX(K)
   10 CONTINUE
   20 NPOINT =K-1
C     Go from log10(time) to (time)
      DO 30 K=1,NPOINT
      TIME(K)=10**(TIME(K))
   30 CONTINUE
      CALL HOPHAM(RELAX,TIME,CREEP,CTIME,NPOINT,IERR)
      IF(IERR.EQ.1) GOTO 300
      do i=1,npoint-1
        write(5,101) ctime(i), creep(i)
        end do
C      WRITE(5,101)((CTIME(K),CREEP(K)),K=1,NPOINT-1)
      STOP
100   FORMAT(1A24)
101   FORMAT(1x,2E14.6)
  300 WRITE(5,301)
  301 FORMAT(' INTERPOLATION ERROR; PROGRAM ABORTED')
      STOP
      END
C
C
C
      SUBROUTINE INTERP(C,E,F,LL,KK,RINT)
C.......
C.......THIS SUBROUTINE LINEARLY INTERPOLATES ANY GIVEN FUNCTION.
C.......
C.......LIST OF SYMBOLS :
C.......        C = Y-VALUES BETWEEN WHICH INTERPOLATION OCCURS
C.......        E = X-VALUES TO BE INTERPOLATED
C.......        F = X-VALUES BETWEEN WHICH INTERPOLATION OCCURS
C.......        LL = NUMBER OF VALUES TO BE CHECKED DURING THE INTER-
C.......             POLATION
C.......        KK = INDEX OF THE VALUE THAT HAS TO BE INTERPOLATED
C.......        RINT = INTERPOLATED VALUE (RESULT)
C.......
C
C
      REAL*8    C(1)
      REAL*8    E(1)
      REAL*8    F(1)
      REAL*8    RINT(1)
C
C
      DO 630 IF=1,LL
      IF(F(IF)-E(KK))631,632,630
 631  IF(F(IF+1).GT.E(KK))GO TO 633
      GO TO 630
 633  RINT(KK)=((C(IF)-C(IF+1))*E(KK)-(C(IF)*F(IF+1)-C(IF+1)*F(IF)))/(F(
     SIF)-F(IF+1))
      GO TO 630
 632  RINT(KK)=C(IF)
 630  CONTINUE
      RETURN
      END
C
C
C
C
      SUBROUTINE HOPHAM(PSI,XIMT,RPHI,X,K,IERR)
C.......THIS SUBROUTINE INVERTS FUNCTIONS THROUGH A CONVOLUTION INTE-
C.......GRAL.
C.......
C.......REF. ALGORITHM BY HOPKINS AND HAMMINGS
C.......
C
C
      REAL*8    DELT(500)               !
      REAL*8    PHI(500)                ! UNKNOWN FUNCTION
      REAL*8    PSI(1)                  ! INPUT FUNCTION
      REAL*8    RF(500)                 ! INTEGRAL OF PSI FROM t0 TO tK
      REAL*8    RFF(500)                ! INTEGRAL OF PSI IN THE SUMMATION
      REAL*8    RPHI(1)                 !
      REAL*8    SUM1                    ! INTERNAL VARIABLE
      REAL*8    SUM2                    ! INTERNAL VARIABLE
      REAL*8    X(1)                    ! VALUES OF T AT WHICH PHI IS COMPUTED
      REAL*8    XIMT(1)                 ! VALUES OF TIME AT WHICH PSI IS KNOWN
      LOGICAL   H
      SUM1=0.0D00
      IERR=0
C.......
C.......CALCULATE RF (INTEGRAL OF PSI)
C.......
C     COMPUTES NOW THE FUNCTION F(T)=INTEGRAL OF PSI(T),FROM t1 TO tK
C      OBVIOUSLY, F(0)=0
       RF(1)=0.0
C
C      THE FOLLOWING IS A TRICK TO AVOID THE PROBLEMS
C      THAT OCCUR WHEN THE FIRST POINT IS NOT SUFFICIENTLY
```

```
C       CLOSE TO THE TIME t=0; WHAT THE PROGRAM DOES
C       IS ADDING ANOTHER DATA POINT AT t=0, WITH THE VALUE OF PSI
C       EQUAL TO THE ONE OF THE FIRST DATA POINT
        DO 32 N=K,1,-1
        XIMT(N+1)=XIMT(N)
        PSI(N+1)=PSI(N)
   32   CONTINUE
        XIMT(1)=0.0
        K=K+1
C
        DO 11 I=2,K
C       INTEGRAL COMPUTED USING LINEAR INTERPOLATION
   12   SUM1=SUM1+(1.0/2.0)*(PSI(I)+PSI(I-1))*(XIMT(I)-XIMT(I-1))
        RF(I)=SUM1
   11 CONTINUE
      K1=K-1
C
C     THE FIRST VALUE OF PHI IS GIVEN BY t2/F(t2)
      PHI(1)=XIMT(2)/RF(2)
      RPHI(1)=PHI(1)
      X(1)=DLOG10((XIMT(2)+XIMT(1))/2.0)
C
C     FIND NOW THE OTHER VALUES OF PHI
      DO 36 I=2,K1
       SUM2=0.0D00
C
   30  DO 35 L=1,I
C
C
C       COMPUTES NOW THE VALUE t(I+1)-t(L), WHERE L IS THE INNER LOOP INDEX
        DELT(L)=XIMT(I+1)-XIMT(L)
        IF(DELT(L).LT.XIMT(1)) ierr=1
        IF(DELT(L).LT.XIMT(1)) GOTO 40
        DO 31 M=1,K
C
C        LOOKS NOW IN WHICH TIME INTERVAL THE VALUE t(I+1)-t(L) FALLS
C
         H=((XIMT(M)-DELT(L)).EQ.0.0)
         IF(H) RFF(L)=RF(M)
         IF(H) GOTO 31
         H=(((XIMT(M)-DELT(L)).LT.0.0).AND.((XIMT(M+1)-DELT(L)).GT.0.0))
         IF(.NOT.H) GOTO 31
C
C
C
C
C        THIS INSTRUCTION IS ACCESSED ONLY WHEN THE INTERVAL
C        [t(M),t(M+1)] IS SUCH THAT DELT(L) FALLS IN IT; CONSEQUENTLY,
C        A LINEAR INTERPOLATION IS PERFORMED USING THE VALUES AT THE
C        INTERVAL ENDPOINTS, IN ORDER TO FIND RF(DELT(L))
C
         RFF(L)=((RF(M)-RF(M+1))*DELT(L)-(RF(M)*XIMT(M+1)-RF(M+1)*
     #   XIMT(M)))/(XIMT(M)-XIMT(M+1))
C
C
C
C
   31    CONTINUE
   35   CONTINUE
        II=I-1
C
C       COMPUTES NOW PHI(I)
        DO 37 IE=1,II
         SUM2=SUM2-PHI(IE)*(RFF(IE)-RFF(IE+1))
   37    CONTINUE

  100   FORMAT (1I5,2E15.5)
        PHI(I)=(XIMT(I+1)+SUM2)/RFF(I)
          RPHI(I)=PHI(I)
        X(I)=DLOG10((XIMT(I)+XIMT(I+1))/2.0D00)
   36 CONTINUE
   40 CONTINUE
      RETURN
      END
```

```
      program smooth
      implicit double precision (a-h,o-z)
      CHARACTER*24 FILIN                ! INPUT FILE NAME
      CHARACTER*24 FILOUT               ! OUTPUT FILE NAME
      real*8 mat, imat
      dimension dlogt(500),dlogy(500)
      dimension mat(3,3),vec(3),fac(3),imat(3,3)

      write(6,*) 'Smooth my data set before decomposition '
      write(6,*) '   into Prony-Dirichlet series'
      write(6,*) 'The data input should be entered in log scale: '
      write(6,*) '   Log(time) and Log(relax).'
      write(6,*) 'Run this before prony.d '
      TYPE *,' INPUT FILE: '
      ACCEPT 100,FILIN
      TYPE *, 'INPUT FILE = ',FILIN
      TYPE *, ' OUTPUT FILE: '
      ACCEPT 100,FILOUT
      TYPE *, 'OUTPUT FILE = ',FILOUT
      open(unit=7,file=filin, status='old')
      open(unit=8,file=filout, status='unknown')

      DO 10 k=1,500
      READ(7,*,END=20) dlogt(k), dlogy(k)
   10 CONTINUE
   20 numdata= k-1
      write(6,*) 'numdata(input data)= ',numdata

      write(6,*) 'Enter the total number of interp. points (500 max)'
      read(5,*)  numtot
c      write(6,*) 'Enter min. and max. time (Log(tmin),Log(tmax))'
c      read(5,*)  tmin, tmax
      tmin= dlogt(1)
      tmax= dlogt(numdata)
      write(6,*) 'tmin(input data)= ',tmin
      write(6,*) 'tmax(input data)= ',tmax

      step= (tmax-tmin)/numtot
c     First point
      t= tmin
      write(8,'(2e14.6)') t, dlogy(1)

      do i=1,numtot
         t= tmin + step*i

c     Find the window(3 pts) framing the current point'
         do j=1,numdata
            if( dlogt(j).eq.t) then
               y= dlogt(j)
               goto 40
            endif
c            if( dlogt(j).gt.t .and. dlogt(j-1).lt.t) then
            if( dlogt(j).gt.t ) then
               ind= j
               goto 30
c            else
c               write(6,*) 'Error in finding the 3points'
c               stop
            endif
         enddo

c     Have the window, build matrix and vector
   30    continue
         t1= dlogt(ind-1)
         t2= dlogt(ind)
         t3= dlogt(ind+1)
         mat(1,1)= t1**2
         mat(2,1)= t1
         mat(3,1)= 1.
         mat(1,2)= t2**2
         mat(2,2)= t2
         mat(3,2)= 1.
         mat(1,3)= t3**2
         mat(2,3)= t3
         mat(3,3)= 1.

         vec(1)= dlogy(ind-1)
         vec(2)= dlogy(ind  )
         vec(3)= dlogy(ind+1)

c     Calculate factors for the quadratic interpolation
         call inv33(mat,imat)
c         do k=1,3
c            fac(k)= 0.
c            do l=1,3
c               fac(k)= fac(k) + imat(l,k)*vec(l)
c            enddo
c         enddo
         a= imat(1,1)*vec(1) + imat(2,1)*vec(2) + imat(3,1)*vec(3)
         b= imat(1,2)*vec(1) + imat(2,2)*vec(2) + imat(3,2)*vec(3)
         c= imat(1,3)*vec(1) + imat(2,3)*vec(2) + imat(3,3)*vec(3)
         write(6,*) 'a,b,c= ', a, b, c

c     get interpolated value and output data
c         y= fac(1)*t**2 + fac(2)*t + fac(3)
         y= a*t**2 + b*t + c
   40    write(8,'(2e14.6)') t, y

      enddo


  100 FORMAT(1A24)
      END

c-----------------------------------------------------------------------
      double precision function det33(a)
      implicit double precision (a-h,o-z)
c
      dimension a(3,3)
c
      det33 = a(1,1)*a(2,2)*a(3,3) + a(2,1)*a(3,2)*a(1,3)
     1      + a(1,2)*a(2,3)*a(3,1) - a(3,1)*a(2,2)*a(1,3)
     2      - a(1,2)*a(2,1)*a(3,3) - a(3,2)*a(2,3)*a(1,1)
      return
      end
c-----------------------------------------------------------------------
      subroutine inv33(a,ainv)
      implicit double precision (a-h,o-z)
      dimension a(3,3),ainv(3,3)

      det = det33(a)
      if(det.eq.0.) then
         write(6,*) '** error det=0  subroutine inv33'
         stop
      endif
      ainv(1,1)= (a(2,2)*a(3,3) - a(2,3)*a(3,2)) / det
      ainv(1,2)= (a(1,3)*a(3,2) - a(1,2)*a(3,3)) / det
      ainv(1,3)= (a(1,2)*a(2,3) - a(1,3)*a(2,2)) / det

      ainv(2,1)= (a(2,3)*a(3,1) - a(2,1)*a(3,3)) / det
```

```
      ainv(2,2)= (a(1,1)*a(3,3) - a(1,3)*a(3,1)) / det
      ainv(2,3)= (a(2,1)*a(1,3) - a(2,3)*a(1,1)) / det

      ainv(3,1)= (a(3,2)*a(2,1) - a(3,1)*a(2,2)) / det
      ainv(3,2)= (a(3,1)*a(1,2) - a(3,2)*a(1,1)) / det
      ainv(3,3)= (a(1,1)*a(2,2) - a(1,2)*a(2,1)) / det
      return
      end
```

```
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      !!!!!!!!!!!!!!                                          !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!   ANALYTICAL DESCRIPTION OF THE          !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!     RELAXATION FUNCTIONS  BY             !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!      PRONY-DIRICHLET SERIES              !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!   written by: I. Emri                    !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!   modified by: L.C. Brinson              !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!   modified to adapt Sun: H.B. Lu         !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!        FILE = relax.for                  !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!                                          !!!!!!!!!!!!!!
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      PROGRAM     relax

      REAL*4      LZ, Lt,Ltmin, Ltmax, Ltstep,
     C            It_lim, LZi, lzmax, LZo
      REAL*4      Zo, Zi, Tau, dZ, Time, Zap, Zip1, Sum,
     C            LTau, Znew, dZpr,
     C            Zip, SSum
      INTEGER*4   I, K, m, N, G, J1, J2, U1,
     C            r, l, N1, Max_pre
      DIMENSION   LZ(5000), Lt(5000), Zi(500), Tau(500), LTau(500),
     C            J1(500), J2(500), dZ(500), LZi(500)




C  READ INPUT DATA
c  "irelax.dat" is input data file containing log time and log modulus
c  arrays "Lt" and "Lz" are the log time and log modulus arrays
c  "m" = number of points in input data
c  FIRST LINE in input file should contain:,
c              A) no. of prony elements ("N", integer)
c              B) iteration limit ("It_lim", real)
c              C) selection indicator ("ISLECT", 1=relaxation modulus data
c                    is given; 2=retardation modulus is given
c


      OPEN ( UNIT=10, FILE = 'irelax.dat', STATUS = 'OLD')
      READ (10,*) N, It_lim, ISLECT, LZo
      Zo=10.**LZo
      if(LZo.eq.0.0) Zo=0.0
      Max_pre=1
      do I = 1,1000,1
      READ (10,*,end=737) Lt(I), LZ(I)
c     TYPE*,Lt(I), LZ(I)
      END DO
      m=i
      goto 738
  737 m=i-1
  738 continue
      close(10)
C
C If creep compliance (retardation modulus) is read in instead of
C relaxation modulus, create a new LZ array from the compliance
C that looks like a relaxation modulus (so this program may be run
C as is, then the result converted back to compliance form)
C
      IF (ISLECT.EQ.2) THEN
       lzmax=LZ(1)
       m1=m-1
       DO 211 I=1,m1
         lzmax=AMAX1(lzmax,LZ(i+1))
211    CONTINUE
        EX=lzmax+(lzmax/100000)
       ENDIF
c        type*, n,it_lim,max_pre,lt(1),lz(1)




c-----------------------------------------------Approximation parameters
C  CALCULATE LOWER AND UPPER TIME LIMITS

       LTMIN=LT(1)
       LTMAX=LT(M)

c  "LTMIN", "LTMAX" are the log time maximum and minimum values
c  "LTSTEP" is the stepsize between the prony elements in logtime

       Ltstep = ABS(Ltmax - Ltmin)/N



c-------------------------------------------------------N_th decade
c  "LTAU" is array containing the log values of the times of the prony
c          elements.  Ends with LTAU(N+1)=LTMAX.
c       LTAU(K) is calculated by beginning with LTMAX and successively
c          subtracting LTSTEP(=(ltmax-ltmin)/N).  Thus, the logtime
c          period is divided up into N equal elements.
c  "TAU" is array containing the real-time values of the prony element times.
c  "Zo" is value of Z at 'infinity': Zo=Z(m)
c  "To" is value of time at 'infinity': To=T(m)

       K = N
C      Zo = 10.**LZ(m)
       To = 10.**Lt(m)
       LTau(N) = Ltmax - Ltstep
       Tau(N) = 10.**LTau(N)
       LTau(N+1) = Ltmax
       LTau(N-1) = LTau(N) - Ltstep
       Tau(N-1) = 10.**LTau(N-1)
       G = 1
c -- no need to calc the N-1 values above?



c----------------------------Find experimental data in time intervals
c  J1(K) has the value of the subscript of Lt (logtime) which is just above
c     the value of LTAU(K) (the kth element's logtime value)
c  J2(K) has the value of the subscript of Lt (logtime) which is just below
c     the value of LTAU(K+1)
c
       I = m + 1
       K = N

1      I = I - 1
       IF (I .EQ. 0) GOTO 3
       IF (Lt(I) .GT. LTau(K+1)) GOTO 1
       J2(K) = I

       I = I + 1

2      I = I - 1
       IF (I .EQ. 0) GOTO 3
       IF (Lt(I) .GE. LTau(K)) GOTO 2
       J1(K) = I + 1
```

```
        IF (K .EQ. 1) GOTO 3

C       TYPE*,'K=', K, 'J1=',J1(K), '  J2=', J2(K)
        K = K-1
        I = I+2
c -- why calc LTAU again here?
        LTau(K) = LTau(K+1) - Ltstep
        Tau(K) = 10.**LTau(K)
        GO TO 1


3       CONTINUE
        j1(1)=1
        K = N




c-------------------------------------------------------Compute Zi(N)
5       CONTINUE
        IF (K .LT. N) GOTO 4
        Time = 10.**Lt(J1(N))
        Zap = EXP(-Time/Tau(N))
        Zi(N) = (10.**LZ(J1(N)) - Zo)/Zap
        IF (Zi(N) .LE. 0.) Zi(N) = 0.
        GO TO 110
4       CONTINUE




c-------------------------------------------------------Compute Zi(K)
        Time = 10.**Lt(J1(K))
        Zip = 0.
        DO  I = K+1, N, 1
        Zip = Zip + Zi(I)*EXP(-Time/Tau(I))
        END DO
        Zap = EXP(-Time/Tau(K))
        Zi(K) = (10**LZ(J1(K)) - Zo - Zip)/Zap
        IF (Zi(K) .LE. 0.) Zi(K) = 0.




c-----------------------------------------Recalculate the parameters
c -- note: dZpr is not used?
c
111     dZpr = 0.
        l = K
19      l = l+1

        Sum = 0.
c -- not sure of need for U1?
        U1 = l - 1
        IF (l .LE. K+1)  U1 = K
c  begin loop for summing for recalculation.
        DO  11, r = J1(U1), J2(l-1), 1
C       DO 11, r=J1(U1), m, 1
C       TYPE*, 'R=', R
        Time = 10.**Lt(r)

        Zip2 = 0.
        IF (l .EQ. N) GOTO 12
        DO  I = l+1, N, 1
        Zip2 = Zip2 + Zi(I)*EXP(-Time/Tau(I))
        END DO
12      CONTINUE

        Zip1 = 0.
        DO  I = K, l-1, 1
        Zip1 = Zip1 + Zi(I)*EXP(-Time/Tau(I))
        END DO
        Zap = EXP(-Time/Tau(l))
        SSum = (10.**LZ(r) - Zo - Zip1 - Zip2)/Zap
C       TYPE*,'SSUM= ', SSUM
        IF (SSum .LE. 0.) SSum = 0.
        Sum = Sum + SSum
11      CONTINUE
c  end loop for summing for recalculation.
        Znew = Sum/(J2(l-1) - J1(U1) + 1)
C       Znew = Sum/(m-J1(U1) + 1)
        dZ(l) = ABS((Zi(l) - Znew))
        Zi(l) = Znew
C       TYPE*,'********** Z',L,' =   ', ZNEW
        IF (Zi(l) .LE. 0.) GOTO 10
        dZ(l) = dZ(l)/Zi(l)
C       TYPE*, 'L, DZ(L)=', L, DZ(L)

c --  again dZpr never used?
        IF (dZ(l) .GE. It_lim) THEN
C         type*, 'l, dz(l)=', l,dz(l)
          GOTO 4
        END IF
        IF(dZ(l) .GE. dZpr) dZpr = dZ(l)

10      N1 = K + 4
        IF (N1 .GE. N) N1 = N
        IF (Max_pre .EQ. 2) N1 = N
        IF (l .LT. N1) GOTO 19



110     CONTINUE
        IF (K .EQ. 1) GOTO 120
        K = K - 1
c -- why calc LTAU again?
        LTau(K) = LTau(K+1) - Ltstep
        Tau(K) = 10**LTau(K)
        G = 1
        GO TO 4



120     CONTINUE
c-----------------------------------------Filing the Prony elements
c    "PRONY.DAT" is file name for output file to receive prony series
c          approximation for modulus.
c
        OPEN (UNIT=12, FILE = 'prony.dat', STATUS = 'unknown')
        OPEN (UNIT=13, FILE = 'origpr.dat', STATUS = 'unknown')

C        TYPE*, 'PRONY-DIRICHLET SERIES APPROXIMATION TERMS'
C        TYPE*, '          I       Ei'
        DO 1212 I=1,N
C          TYPE*,I, Zi(I)
          IF (Zi(I).EQ.0.) GOTO 1212
          LZi(I)=ALOG10(Zi(I))
          IF (LZi(I).LT.0) LZi(I)=0
1212    CONTINUE
```

```
      DO  I = 0, N, 1
c     IF ( I .EQ. 0) WRITE (5,*) To, Zo
      IF ( I .EQ. 0) GOTO  1129

      WRITE (12,1234) Zi(I), Tau(I)
c      WRITE (12,1234) Tau(I), Zi(I)
c      WRITE(5,1234) LTau(I), LZi(I)
      WRITE(13,1234) LTau(I), LZi(I)
1234  format(E15.7,5X,E15.7)
1129  END DO
      CLOSE (12)
      CLOSE (13)
      STOP
      END
```

```
SHEA
      0.          3.16227766    26
  0.1101795E+01        0.2763996E-02
  0.2182498E+04        0.1190686E-01
  0.1024474E+04        0.5129290E-01
  0.2480911E+04        0.2209618E+00
  0.3623970E+04        0.9518692E+00
  0.1338277E+04        0.4100504E+01
  0.5720489E+03        0.1766433E+02
  0.3336653E+03        0.7609518E+02
  0.1938180E+03        0.3278062E+03
  0.1798677E+02        0.1412138E+04
  0.6957788E+01        0.6083269E+04
  0.3148028E+01        0.2620577E+05
  0.1889946E+01        0.1128904E+06
  0.1470057E+01        0.4863146E+06
  0.1083488E+01        0.2094968E+07
  0.8003443E+00        0.9024802E+07
  0.5977423E+00        0.3887746E+08
  0.4078681E+00        0.1674781E+09
  0.2850480E+00        0.7214699E+09
  0.2019787E+00        0.3107981E+10
  0.1230532E+00        0.1338870E+11
  0.8832651E-01        0.5767648E+11
  0.5256542E-01        0.2484614E+12
  0.3242005E-01        0.1070333E+13
  0.3305381E-02        0.4610831E+13
  0.1414650E-01        0.1986275E+14
```