INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Comparison of Time-Domain Finite Element Modelling of Viscoelastic Structures Using an Efficient Fractional Voigt-Kelvin Model or Prony Series

by

Marie-Josée Potvin

Department of Mechanical Engineering
McGill University
Montréal

April, 2001

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

The thesis centers on time domain modelling of viscoelastic materials. Classical models are compared to models involving fractional derivatives, which are derivatives of an order between 0 and 1. Parameters for classical and fractional order models are found for two materials, polymethylmethacrylate and 3M ISD 112, an acrylic based material sold as a viscoelastic layer by 3M. In both cases, only Prony series with several parameters achieve a good representation over a large frequency range. In the case of 3M ISD 112, a fractional model with only two parameters gives a good representation over a frequency range of three decades, which is often sufficient.

An algorithm based on an approximated definition of the fractional derivative and a trapezoidal rule is described to solve constitutive equations with fractional derivatives. The algorithm is implemented in C and tested against a numerical Laplace inverse for the case of a material submitted to sinusoidal strains. The algorithm gives accurate results and does not require very small steps, which is usually the case for algorithms based on finite differences or Grünwald series.

The algorithm is adapted to the structure of a user subroutine of a commercial finite element package, *Samcef*, for a six component isotropic tensor. The model assumes a constant bulk modulus and has one fractional derivative of the deviatoric strain. The Jacobian of the constitutive equation with a fractional derivative is derived and implemented. The results from the subroutine are compared satisfactorily to results from the numerical Laplace inverse for a cubic element submitted to sinusoidal strains.

Finally, the different models are tested to represent the experimental behaviour of slewing beams made either of polymethylmethacrylate or steel covered by constrained viscoelastic layers. The classical models give generally a poor representation of the experimental behaviour, except for the Prony series. The fractional model give a representation as satisfactory as the ones obtained with the Prony series, but for a much higher CPU times due to the hereditary nature of the fractional derivative. It is therefore recommended to use Prony series models, unless the data to perform the parameter identification is limited. In that case, the fractional order model becomes interesting despite the higher demands on the CPU time.

Sommaire

Le but de cette thèse est d'étudier la modélisation dans le domaine temporel des matériaux viscoélastiques. Les modèles classiques sont comparés aux modèles comprenant des dérivées fractionnaires, ces dérivées étant d'un ordre entre 0 et 1. Les paramètres des modèles classiques et fractionnaires sont identifiés pour deux matériaux, le polyméthacrylate de méthyle et le 3M ISD 112, un matériau acrylique vendu par 3M comme couche viscoélastique. Pour les deux matériaux, seules les séries de Prony ayant plusieurs paramètres donnent une bonne représentation sur une grande plage de fréquence. Dans le cas du 3M ISD 112, un modèle fractionnaire à seulement deux paramètres donne une bonne représentation sur une plage de fréquence de trois décades, ce qui est souvent suffisant.

Un algorithme basé sur une définition approximative de la dérivée fractionnaire et sur la méthode trapézoïdale est écrit pour résoudre des lois de comportement avec dérivées fractionnaires. L'algorithme est écrit en C et vérifié contre les résultats d'un inverse de Laplace numérique pour le cas d'un matériau soumis à des déformations sinusoïdales. L'algorithme donne des résultats adéquats et n'exige pas de petits pas comme c'est habituellement le cas avec les algorithmes basés sur les différences finies ou les séries de Grünwald.

L'algorithme est ensuite adapté à la structure d'une sous-routine usager d'un programme d'éléments finis commercial, *Samcef*, pour le cas d'un tenseur isotropique de six composantes. Le modèle suppose un module de compressibilité volumique constant et possède une dérivée fractionnaire de la

déformation déviatorique. Le jacobien de la loi de comportement à une dérivée fractionnaire est obtenu et implanté. Les résultats de la sous-routine se comparent de façon satisfaisante aux résultats de l'inverse numérique de Laplace pour un élément cubique soumis à des déformations sinusoïdales.

Finalement, les différents modèles sont utilisés pour représenter le comportement expérimental de poutres de polyméthacrylate de méthyle ou d'acier recouvert de couches viscoélastiques contraintes en rotation dans le plan. Les modèles classiques donnent généralement une piètre représentation du comportement expérimental, à l'exception des séries de Prony. Le modèle fractionnaire donne une représentation aussi satisfaisante que les séries de Prony, mais requiert un temps CPU beaucoup plus élevé à cause de la nature héréditaire de la dérivée fractionnaire. Il est donc recommandé d'utiliser les séries de Prony, sauf lorsque les données nécessaires à l'identification des paramètres sont insuffisantes. Dans ce cas, un modèle d'ordre fractionnaire est une alternative intéressante en dépit des grandes exigences au niveau du temps CPU.

Acknowledgments

How difficult it is to write this page after so many busy years, so many people to remember, so many memories. Raising four children, working on contracts, and doing a thesis spread over six years was a challenge that often seemed overwhelming. Many things happened in those years: moving a whole family several times, looking for good babysitters, being pregnant, feeling nauseous while running simulations, breastfeeding while reading papers, changing offices and computers several times, and sadly, turning down nice job offers. Of course, I was often taken over by a sense of perplexity for doing something that would bring me mainly a feeling of a job done, and done as well as I could. Many had a little word for me here and there which kept me going. They are too numerous to name all, going from friends, to my mother, to my husband, but I will still mention Professor James Nemes and Doctor Virendra Jha, from the Canadian Space Agency, who always made clear they expected me to finish, therefore giving me some incentive to do so. Doctor Albert-Paul Gonze, from *Samtech*, was always very responsive to my questions and encouraging with my work on *Samcef*. His support has been crucial. To all those who gave some encouragements, at one time or another, I thank you deeply. You made a real difference.

Money was always a concern, of course. Quality daycare is expensive for the parent, although it is little money for the caregiver. Several organizations supported me: NSERC, FCAR, Zonta International, and McGill Alma Mater. They also made a huge and real difference.

Despite all the hardships, I must admit I had tremendous fun. I did some-

thing I enjoy thoroughly: research. Finally, I would like to dedicate this work to my children as a reminder that young people should not accept the dictates of society and confine themselves to established models. They should not accept to choose between being a career person or a dedicated parent, but they should look for other options. I wish them the strength to bend the unwritten rules of the world they inhabit and make it one where men and women strive to be well-balanced and happy human beings. Good luck, my little ones. I do not think I have found the perfect answer, but I hope you will.

Claim of Originality

To the best of the author's knowledge, the following studies presented in this thesis are original and are not found elsewhere.

- Development of an algorithm to solve ordinary differential equation with a fractional derivative based on an approximated definition of the fractional derivative and a trapezoidal rule for the resulting integral.

- Implementation of this algorithm in a commercial finite element package, *Samcef*, in an implicit scheme requiring the Jacobian of the three-dimensional fractional Voigt-Kelvin constitutive equation.

- Comparison of the performance of the fractional Voigt-Kelvin model with other models such as the Prony series, the Voigt-Kelvin model, the Maxwell model, and the Zener model in terms of accuracy to represent an experimental slewing homogeneous polymethylmethacrylate beam.

- Comparison of the performance of the fractional Voigt-Kelvin model with other models such as the Prony series, the Voigt-Kelvin model, the Maxwell model, and the Zener model in terms of accuracy to represent an experimental slewing steel beam covered with a constrained viscoelastic layer.

- Evaluation of the efficiency of the algorithm presented in this thesis by comparing the required CPU time with the CPU time needed by other models such as the Prony series, the Voigt-Kelvin model, the Maxwell model, and the Zener model.

Contents

List of Figures

List of Tables

Nomenclature

A description of the notation, variables, and abbreviations used in this thesis.

Notation

vectors and matrices are in bold type, e.g. $\boldsymbol{r}$

derivative with respect to time: $\partial e_{xx}/\partial t = \dot{e}_{xx}$

derivatives of higher order with respect to time: $\partial^n e_{xx}/\partial t^n = e_{xx}^{(n)}$

fractional derivative of order ξ: $D^{\xi}\varepsilon$

asterisk indicates a complex variable, e.g. G^*

vertical bars indicate the magnitude of a complex variable, e.g. $|G^*|$

Variables

A	amplitude of a sine function
a	damping coefficient
B	constant term of a sine function
b	damping coefficient
E	Young's modulus (Pa)
E_e	equilibrium Young's modulus (Pa)
E^*	complex Young's modulus
e''	volumetric strain tensor
e_{ij}	component ij of the deviatoric strain tensor
e_{ij}^0	constant value for the component ij of the deviatoric strain tensor
exp	exponential

$F(s_K)$	discrete Laplace transform
$\mathcal{F}_{k+1}$	discrete Laplace transform used by *Matlab*
f	frequency (Hz)
$f(t_M)$	discrete time domain function
f_{m+1}	discrete time domain function used by *Matlab*
G'	shear storage modulus (Pa)
G_e	equilibrium shear modulus (Pa)
G_g	glassy shear modulus (Pa)
G_n	coefficient of the Prony series for the shear modulus (Pa)
G_{relax}	relaxation shear modulus (Pa)
G^*	complex shear modulus
g_n	coefficient of the Prony series for the shear modulus (Pa)
h	time step (s)
h_1	time step of the preceding step(s)
i	imaginary number $(\sqrt{-1})$
K_e	equilibrium bulk modulus (Pa)
K_n	coefficient of the Prony series for the bulk modulus (Pa)
K^*	complex bulk modulus
k	constant of an exponential
k_i	constants in constitutive equations
$\mathcal{L}$	Laplace transform operator
l	length (m)
l_0	initial length (m)
$\mathcal{O}$	order of error
p_r	damping coefficient
q_r	damping coefficient
M	number of cycles between two measurements
N	number of points used for the Laplace transform
n	half the number of points used for the Laplace transform
S''	volumetric stress tensor (Pa)
S_{ij}	component ij of the deviatoric stress tensor (Pa)
$S_{ij,relax}$	component ij of the relaxation deviatoric stress tensor (Pa)
s	Laplace variable
t	time (s)
t_f	maximum time (s)

α	damping coefficient
$\Gamma(x)$	Gamma function applied to x
γ	damping coefficient
δ	logarithmic decremment
δ_{ij}	Kronecker delta applied to the component ij of the tensor
ϵ	small amount of time compared to t (s)
ε_{ij}	component ij of the strain tensor
ζ_n	dummy variable for time (s)
ζ_{s_n}	dummy variable for time (s)
η	loss factor
ν_e	equilibrium Poisson's ratio
ξ	order of the fractional derivative
σ_{ij}	component ij of the stress tensor (Pa)
τ	dummy variable for time (s)
τ_n	coefficient of the Prony series (s)
ϕ	phase of a sine function (rad)
ω_{max}	maximum frequency excited (rad)

Abbreviations

CPU	Computer Processing Unit
DS	evaluation of the local error
DSN	evaluation of the local error at the preceding step
DTIO	initial time step
FFT	Fast Fourier Transform
memcom, MEMCOM	number of data for which larger steps are taken
memmax, MEMMAX	maximum number of stored data
PMMA	polymethylmethacrylate
PRCO	local error tolerance for the constitutive equation
PRCV	local error tolerance for the dynamic computation
SREF	reference stress defined by the user

Chapter 1

Introduction

1.1 Use of Viscoelastic Materials

In recent years, some types of satellites have become more bulky as the need for more transmission capabilities has increased. The drive to reduce the weight is still an important fact. However, as structures become thinner, and hence more flexible, they are also more prone to vibrations. Several means of reducing these vibrations are used, but viscoelastic materials are proving useful. These materials are thin sheets of polymers, often covered by a thin sheet of a stiff material, and able to significantly reduce the level of vibrations without adding much weight to the structure. Their use has been investigated for satellites (Jha and Tremblay [1, 2]), space stations (Jones et al [3]), launchers (Poizat et al [4]), as well as space manipulators (Alberts et al [5]). Modelling of the behaviour of these materials through finite element analysis is often required in the case of space structures, since prototypes are too costly and the design must be perfect. In most cases, accurate modelling is needed in the frequency domain to obtain the natural frequencies and the amplification ratio of the structure. Several authors have investigated the precision of frequency domain finite element analyses for structures covered with viscoelastic materials and they have found satisfying agreements between experimental data and simulation results for simple test cases (Johnson and Kienholz [6], Johnson et al [7], Sun et al [8], Slanik et al [9]).

Time domain modelling is also needed for simulation of deployment situa-

tions and manipulator movements. The precision of the time domain analysis depends on the accuracy of the finite element formulation and the effectiveness of the numerical integration algorithms, but also on the constitutive equation used to represent the stress-strain relationship of the polymer. Moreover, use of a commercial finite element package is important for acceptance of the results by the aerospace community.

Classical viscoelasticity models allow structural analysts to describe damping phenomena in the time domain. A description of these models is found in Axelrad [10]. Some of these models are implemented in commercial finite element packages and simulation of complex damped structures undergoing nonlinear motion is possible. However, classical viscoelasticity models often require several parameters to correctly represent the material behaviour. Slanik et al [11] obtain good time domain finite element simulation results using a five term Prony series, which involves the identification of at least ten parameters. The experimental set-ups needed to identify those parameters are not always available and often only limited data is available. There is a need for accurate models involving as few parameters as possible, and therefore, possible identification with limited data. In that respect, constitutive equations using fractional derivatives, rather than integer derivatives, have the advantage of being representative of the material behaviour with fewer parameters. Fractional derivatives are derivatives of an order between 0 and 1 and are represented by a hereditary convolution integral. This thesis deals specifically with time domain finite element simulations with fractional order constitutive equations. The next section reviews the work done on this problem.

1.2 Modelling of Dynamic Equations with Fractional Derivatives

In the last few decades, several authors have started to examine the feasibility of using fractional derivatives to describe the dynamic behaviour of systems submitted to internal damping. Rossikhin and Shitikova [12] give a review

of the work done in solid mechanics involving fractional derivatives. Many authors write the dynamic equations in the frequency domain and obtain the time domain solution by using a numerical Laplace inversion or a numerical Fourier inversion (Bagley and Torvik [13], Cooke and Keltie [14], Padovan and Guo [15], Makroglou et al [16], Suarez and Shokooh [17, 18], and Baker et al [19]). Fractional derivatives have a simple corresponding expression in the frequency domain, and therefore, their inverse can be taken. However, this approach is not useful when nonlinear geometric terms are present in the dynamic equations since some linearisation is needed to take the Laplace inverse. Some authors started to attempt direct time-domain simulations, but a fractional derivative is a hereditary convolution integral and as such, some terms need to be stored at each step and used in future calculations. This slows down the computation and authors have found various ways of dealing with this.

1.2.1 Bagley et al

Bagley and Torvik were probably the first to attempt solving structural equations involving fractional derivatives. In Bagley and Torvik [13], a simply supported beam covered with a constrained viscoelastic layer and excited at midspan is modelled. A constrained viscoelastic layer is a thin layer of polymer covered by a thin layer of a stiff material. In this case, the polymer is modelled with a constitutive equation including a fractional derivative of the shear strain and no derivative of the shear stress. A value of one half is used for the order of the fractional derivative to simplify the solution of the equations. Finite element equations are written. The beam itself is modelled with triangular elements, the viscoelastic layer with rectangular elements, and the constraining layers with rods. A modal solution is found and the time-domain displacement for an impulsive loading is obtained from the modal solution. The finite element equations give results similar to results obtained from the sixth order theory for beams with constrained viscoelastic layers. The differ-

ences are attributed to the different assumptions made in the finite element equations and in the sixth order beam theory [20].

In Bagley and Calico [21], the goal is to write closed loop structural equations incorporating control schemes. This time, they use a constitutive equation with fractional derivatives of the same order on both stress and strain. Again, a modal solution is first obtained and the time-domain solution is reconstructed from it. An important innovation is to address non zero initial conditions. As the authors explain, the fractional derivative model assumes the material to be in its undeformed state at time zero. To start the simulation with non trivial initial conditions, they start a different clock. The first clock has its zero when the material is in its undeformed state, the second clock starts when the material has reached its initial conditions. As in the previous paper, the dynamic equations are not solved numerically, but rather using analytical approximations for simple test cases. However, this approach can not be expanded to complex structures for arbitrary levels of damping, since it would not be possible to use these analytical approximations in more complex cases.

1.2.2 Koh and Kelly

One of the first papers to deal with time-domain numerical solutions of dynamic equations is the one by Koh and Kelly [22]. They use a constitutive equation using a fractional derivative of the strain. The algorithm they use is the L1 algorithm described in Oldham and Spanier [23] based on finite differences. They first test the algorithm against the numerical Laplace inverse for a simple system consisting of a mass, a spring, and a damper. The step size is critical and needs to be quite small for accurate results. To cut the number of data to store and the number of computations required by the hereditary nature of the fractional derivative, they use a time window, meaning they keep only a few data near the actual computation time. The size of the window is not as critical as the step size. However, this concept of time window works

well for oscillations having a zero-mean, which is the case for their specific problem, but it might not work as well for all problems. Finally, they write the one-dimensional dynamic equation of a bridge deck mounted on multilayered natural rubber bearings. Steady-state harmonic tests allow identification of the parameters of the fractional derivative model representing the behaviour of the system. The bridge deck is then submitted to displacements similar to the El Centro 1940 earthquake. The experimental results are compared to the simulation results and the fractional model gives very good agreement with errors in the peak amplitudes of less than 1%. A simulation is also done with a classical viscoelasticity model and the errors in the peak amplitudes are about 6%.

1.2.3 Makris and Constantinou

In Makris and Constantinou [24], viscous fluid used as dampers for earthquake protection are modelled with fractional derivatives. The constitutive equation has one integer derivative of the shear stress and one fractional derivative of the shear strain. The dynamic equations for a building supported on viscous dampers are written and solved using a numerical algorithm based on finite differences. The algorithm is a modification of the G1 scheme described in Oldham and Spanier [23]. The numerical solution is compared to experimental data for a six storey, quarter scale, model building, and good agreement is found.

1.2.4 Eldred, Baker, and Palazotto

Eldred, Baker, and Palazotto examine a bar fixed at one end and submitted to a load at the other end. The model used for the material behaviour involves one fractional derivative of the strain. The bar is represented with a one-dimensional dynamic equation. In Eldred et al [25], the authors examine the Voigt-Kelvin model with respect to fractional models in their ability to reproduce the behaviour of materials. In Eldred et al [26], two schemes based

on finite differences are explored. These schemes are taken from Oldham and Spanier [23] and are called L1 and G1. The authors find these schemes to be very sensitive to the time step, especially in the presence of nonlinear terms in the dynamic equations. In Eldred et al [27], they compare the solutions obtained with the finite difference schemes to the numerical Laplace inverse and they find the L1 algorithm reproduces the amplitude of the deflection within 2% of the values given by the numerical Laplace inverse. The finite difference scheme requires very small steps to start up and a data storage scheme is devised. The first steps are very small to enable a good start of the solution, but as the solution progresses, only one out of two data points are kept in memory to compute the rest of the solution.

1.2.5 Padovan

Padovan [28] outlines various algorithms to solve finite element equations with fractional derivatives in the time domain. He looks at implicit, explicit, and predictor-corrector schemes for cases of a single fractional derivative of the strain, as well as cases with both a fractional derivative of the stress and the strain. He uses the Grünwald series to define the fractional derivative. As an example, he looks at a simple system consisting of a mass, a spring, and a damper.

1.2.6 Chern

In Chern [29], an in-house finite element code is written with plane strain and plane stress elements described by a fractional order constitutive equation. The constitutive equation has a fractional derivative of the strain and no derivative of the stress. A constant Poisson's ratio is assumed and first order finite differences are used to approximate the fractional derivative. As an example, a beam is fixed at one end and excited at the other end. The response obtained with the finite element formulation is not compared to a response obtained by another method.

1.2.7 Enelund et al

Enelund and various co-authors explore the solution of dynamic equations with fractional derivatives represented by a Grünwald series. The viscoelastic model involves one fractional derivative of the stress and another one of the strain of the same order. The representation they use enables initial conditions different from zero. In Enelund and Olsson [30], they look at a one dimensional equation solved with a Newmark algorithm using a constant step. The equation represents a simple system consisting of a mass, a spring, and a damper. The results are validated against the inverse Fourier transform. In Enelund et al [31], the same system is looked at, but this time the model is modified to work with fractional integrals rather than fractional derivatives. This technique is advantageous to deal with the initial conditions, but it involves taking the integral derivative of the load. As each fractional integral or fractional derivative needs all values at each time step to be stored to solve future steps, taking the integral derivative of the load increases the demands on the computing ressources.

The step to three-dimensional equations is taken by writing finite element equations of the system. In Enelund and Josefson [32], the constitutive equations involve a fractional derivative of the stress and one in the strain of the same order. Moreover, the same constitutive equation is used for both deviatoric and hydrostatic equations, resulting in a constant Poisson's ratio. The finite element equations are written by the authors to deal with their specific problems. They study a simple mass, spring, and damper system, as well as a bar made of five linear elements. The order of the fractional derivative is one half, allowing to solve the equations analytically for comparison. The finite element equations are solved with a Newmark algorithm using a constant step. The step has to be less than $2/\omega_{max}$ to ensure stability, where ω_{max} is the maximum frequency excited. In Enelund and Lesieutre [33], they solve the same bar, but this time, the constitutive equations are dealt with by using internal variables. In Enelund et al [34], the constitutive equations

are implemented in *Abaqus*, a commercial finite element software. An implicit scheme is used for quasi-static solutions, and an explicit one for dynamic solutions. The bar treated earlier is studied again with five linear elements, but also with five plane stress elements. They also look at the problem of a viscoelastic material under rails, the material being modelled by plane strain elements. Approximately 15000 steps are needed to simulate 25 oscillations, a sampling frequency of approximately 600 data points per cycle. To reduce the computing time, they attempt keeping only part of the history as Koh and Kelly [22] had done, but if the order of the fractional derivative is small, than much of the history is needed to produce accurate results.

1.2.8 Potvin et al

In a series of papers, Potvin et al simulate the dynamic behaviour of a homogeneous polymethylmethacrylate (PMMA) beam using a constitutive equation with a fractional derivative of the strain. In Potvin et al [35, 36], they compare experimental data and simulation results for a PMMA beam fixed at one end and oscillating under gravity after being dropped from an undeformed position. The dynamic equations are written and solved using a Runge-Kutta scheme. The fractional derivative is split into an integral term and an approximation for the integral in the vicinity of t, the current time. The results are good, but the time step is small due to the large difference in frequency between the two first modes. This problem due to the stiffness of the dynamic equations is solved in Potvin et al [37, 38]. The dynamic equations are solved using an implicit numerical scheme of the Newmark family, the Hilber-Hughes-Taylor algorithm [39]. As in the preceding papers, the fractional derivative is split into an integral term and an approximation for the integral in the vicinity of t, but the numerical algorithm used to solve the integral term is also based on the Hilber-Hughes-Taylor algorithm, rather than a Runge-Kutta algorithm. Again, the fit between experimental data and simulation results is good. To accelerate the computation time, larger time steps than what is used for the

computation of the dynamic equations are taken for the computation of the integral term of the fractional derivative.

1.2.9 Summary

Several authors compare simulation results with experimental data and they obtain good agreement between the two for models with small numbers of parameters (Koh and Kelly [22], Makris and Constantinou [24], Potvin et al [35, 36, 37, 38]).

Most authors have solved dynamic equations involving fractional derivatives with finite differences or Grünwald series. They found the time step needed to be small to keep a reasonable level of accuracy (Koh and Kelly [22], Eldred et al [26, 27], and Enelund et al [34]). The number of time steps needed even for simple structures implies a large amount of CPU time, which render the finite element simulations prohibitive for more complex structures.

1.3 Scope of the Investigation

As shown by the preceding literature review, although good results are obtained with constitutive models of fractional orders, there are problems with memory management and the CPU time required. The alternative is to use classical models such as the Prony series, but at the expense of identifying several parameters. Most commercial finite element packages offer Prony series models, but not fractional models. The goal of this thesis is to evaluate the practicality of fractional models compared to the Prony series in terms of precision and computation time.

A model of fractional order with one fractional derivative of the strain will be implemented in a commercial finite element package allowing user material behaviour subroutines. Given the small time steps required by models based on finite differences or Grünwald series, a new algorithm built on the previous work of Potvin et al [35, 36, 37, 38] will be elaborated. The fractional derivative will be split into an integral term and an approximation for the integral in the

vicinity of t, but this time, the integral term will be simply solved with a trapezoidal rule. The goal is to obtain good precision for large time steps.

Parameters for the viscoelasticity models available in the commercial finite element package and the ones for the fractional model will be identified for two materials, polymethylmethacrylate and 3M ISD 112. Although polymethylmethacrylate is not perfectly isotropic, it will be approximated as such for this case. These two materials will then be used in beam configurations and simulation results for all models will be compared to experimental results. No attempt will be made to fit the model parameters to the simulation results. The goal is to see how reliable the models can be when the parameters are identified a priori.

1.4 Organisation of the Thesis

The next chapter of this thesis defines the viscoelasticity models available in commercial finite element packages, as well as the fractional model. Typical curves for the magnitude and phase of the modulus of a polymer with respect to frequency are given and the fit to these curves obtained with each model is shown.

Chapter 3 identifies the parameters for all models for the two chosen materials.

Chapter 4 details the numerical algorithm devised for the fractional derivative. Simple one-dimensional test cases are used to assess the accuracy of the algorithm. The results obtained with the algorithm are compared to results obtained with the numerical inversion of the Laplace transform of the one-dimensional equations. The effect of the different parameters of the algorithm, such as the large time steps used for the integral term of the fractional derivative, is evaluated in terms of accuracy and computation time.

Chapter 5 shows how the algorithm discussed in Chapter 4 is implemented in a commercial finite element package, *Samcef*. Tests similar to the one-dimensional tests used in Chapter 4 are done, but this time with a three

dimensional formulation. Again, the accuracy of the results obtained for these test cases is compared to the numerical inversion of the Laplace transform of the equations representing the test cases.

Finally, Chapter 6 gives examples of homogeneous slewing polymethylmethacrylate beams and slewing steel beams covered with constrained layers of 3M ISD 112. The simulation results obtained with the various models are compared to the experimental data. The precision and the computation time are evaluated with the objective of determining the usefulness of the fractional order model compared to classical models, such as the Prony series.

Chapter 2

Models of Linear Viscoelasticity Behaviour

2.1 Introduction

Hooke's law predicts a strain proportional to the stress with no energy dissipation. However, if the material is viscoelastic, it dissipates energy. For cyclic loading, the strain is delayed with respect to the stress and a phase difference appears. Obviously, Hooke's law is not sufficient to describe this phenomenon and other models are needed.

This chapter describes classical models used to represent viscoelastic behaviour. These models are built around Hooke's law by adding to it derivatives of the stress and the strain. Finally, a new class of models is introduced where fractional derivatives are used, which are derivatives of an order between 0 and 1, to represent more accurately the observed behaviour of polymers.

2.2 Classical Constitutive Equations

A constitutive equation relates the stress and strain for all tensor components. According to Malvern [40], energy is dissipated in polymers undergoing a periodic shearing or a periodic hydrostatic compression. The shear response often exhibits more variation than the volumetric response. Two different constitutive equations are written for the deviatoric part and the hydrostatic

part to reflect this fact. The deviatoric part, S_{ij}, of the stress tensor σ_{ij} is:

$$S_{ij} = \sigma_{ij} - \frac{1}{3}\delta_{ij}\sigma_{kk} \tag{2.1}$$

Similarly, the deviatoric part, e_{ij}, of the strain tensor ε_{ij} is

$$e_{ij} = \varepsilon_{ij} - \frac{1}{3}\delta_{ij}\varepsilon_{kk} \tag{2.2}$$

The hydrostatic parts, S'' and e'' are defined as:

$$S'' = \frac{1}{3}\sigma_{kk} \tag{2.3}$$

$$e'' = \frac{1}{3}\varepsilon_{kk} \tag{2.4}$$

The hydrostatic response is often considered elastic for polymers [40]:

$$S'' = 3K_e e'' \tag{2.5}$$

where K_e is the equilibrium bulk modulus, which is the modulus obtained for a static load. This simple expression is used throughout this work to characterize the hydrostatic response. However, there are several possibilities for the shear response.

The next sections discuss the most common models by showing their typical behaviour. Values of the shear modulus as it varies with frequency are shown for the given model and the experimental data for an acrylic based polymer manufactured by 3M, 3M ISD 112. This material will be discussed in details in Chapter 3, but it is used in this chapter as an example to illustrate the viscoelastic models. The parameters of the models are chosen to reasonably cover the whole chosen frequency interval. It is done manually, without an optimization algorithm, and their choice is discussed in Chapter 3. The goal of this chapter is to gain an understanding of the models, by illustrating the typical behaviour inherent to each model, without commenting on the specific values used for different materials.

2.2.1 The Voigt-Kelvin Model

The simplest model for the shear response is called the Voigt-Kelvin model. The stress is simply proportional to the strain and the first derivative of the strain:

$$S_{ij} = 2G_e e_{ij} + 2aG_e \dot{e}_{ij} \tag{2.6}$$

where a is a damping coefficient and G_e, the equilibrium shear modulus.

Assuming zero initial conditions, the Laplace transform of Equation 2.6 is taken to obtain the frequency-varying shear modulus:

$$\frac{S_{ij}(s)}{2e_{ij}(s)} = G^* = G_e + aG_e s \tag{2.7}$$

where $s = i2\pi f$ and is the Laplace domain variable. The factor 2 in front of e_{ij} comes from the use of elasticity strains. The star used as a superscript for G indicates a complex modulus, a modulus varying with frequency and characterized by a magnitude and a phase.

Typical experimental behaviour for polymers shows the magnitude of the shear modulus increasing with frequency up to an asymptotic value, whereas the phase is shaped as a bell. At very low frequencies, the phase difference is zero and the material behaves elastically. At very high frequencies, the phase difference is again zero, and the material again behaves elastically, but with a modulus presenting a higher magnitude than at low frequencies. A Voigt-Kelvin model, however, is characterized by a monotonically increasing magnitude and a phase increasing up to an asymptotic value of 90°. Figure 2.1 shows the experimental data for 3M ISD 112. A typical fit is obtained for a Voigt-Kelvin model and is also shown in Figure 2.1. Since the phase rises very sharply from 0° to 90°, the model cannot accomodate a broad range of data for the phase. The model also predicts a magnitude rising at a faster rate than is observed physically and the model either underestimates or overestimates the data for the magnitude.

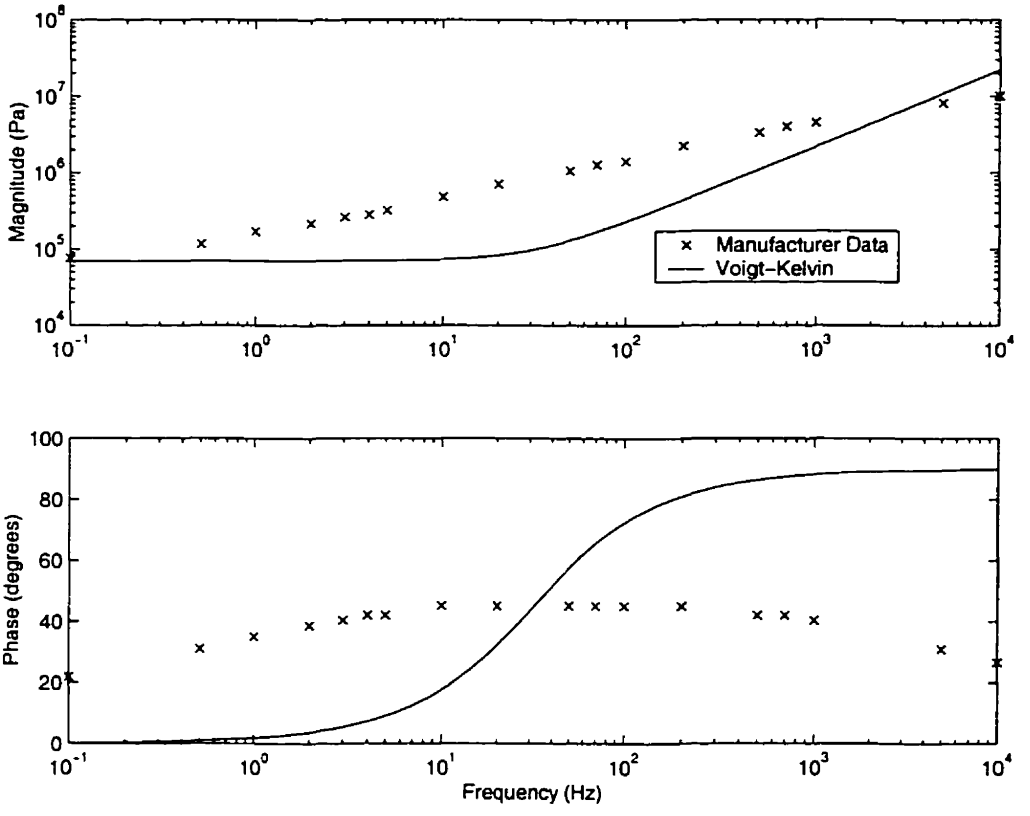

Figure 2.1: Complex Shear Modulus for the Voigt-Kelvin Model ($G_e = 7.00 \times 10^4\ Pa$, $a = 0.005$)

2.2.2 The Maxwell Model

Another simple model for the shear response is the Maxwell model. The stress added to the first derivative of the stress are proportional to the first derivative of the strain. In the Voigt-Kelvin model, there was no derivative of the stress, in Maxwell model, the strain itself is not involved:

$$S_{ij} + a\dot{S}_{ij} = 2aG_e\dot{e}_{ij} \tag{2.8}$$

where a is a damping coefficient and G_e, the equilibrium shear modulus.

Assuming zero initial conditions, the Laplace transform of Equation 2.8 is taken to obtain the frequency-varying shear modulus:

$$\frac{S_{ij}(s)}{2e_{ij}(s)} = G^* = \frac{aG_e s}{1 + as} \tag{2.9}$$

The Maxwell model is characterized by an amplitude increasing up to an asymptotical value. L'Hospital's rule shows the limit when the frequency becomes infinite to be G_e:

$$\lim_{s\to\infty} \frac{aG_e s}{1 + as} = \lim_{s\to\infty} \frac{aG_e}{a} = G_e \tag{2.10}$$

Physically, the material exhibits a modulus close to G_e at low frequencies, which then rises to an asymptotic value called the glassy modulus, G_g [40]. The Maxwell model, however, only goes up to G_e from a value of zero for the modulus. As for the phase, the model predicts an asymptotic value of 90° then sharply decreasing to zero. This does not approximate well the physical behaviour showing a phase starting at zero, rising to a maximum, and then decreasing to zero. Figure 2.2 shows the experimental data for 3M ISD 112 and an appropriate fit obtained for a Maxwell model.

2.2.3 The Standard Linear Solid Model

A better model than the preceding two would be obtained by combining properties of the two models. The magnitude should start at the equilibrium modulus, as predicted by the Voigt-Kelvin model, but it should not increase

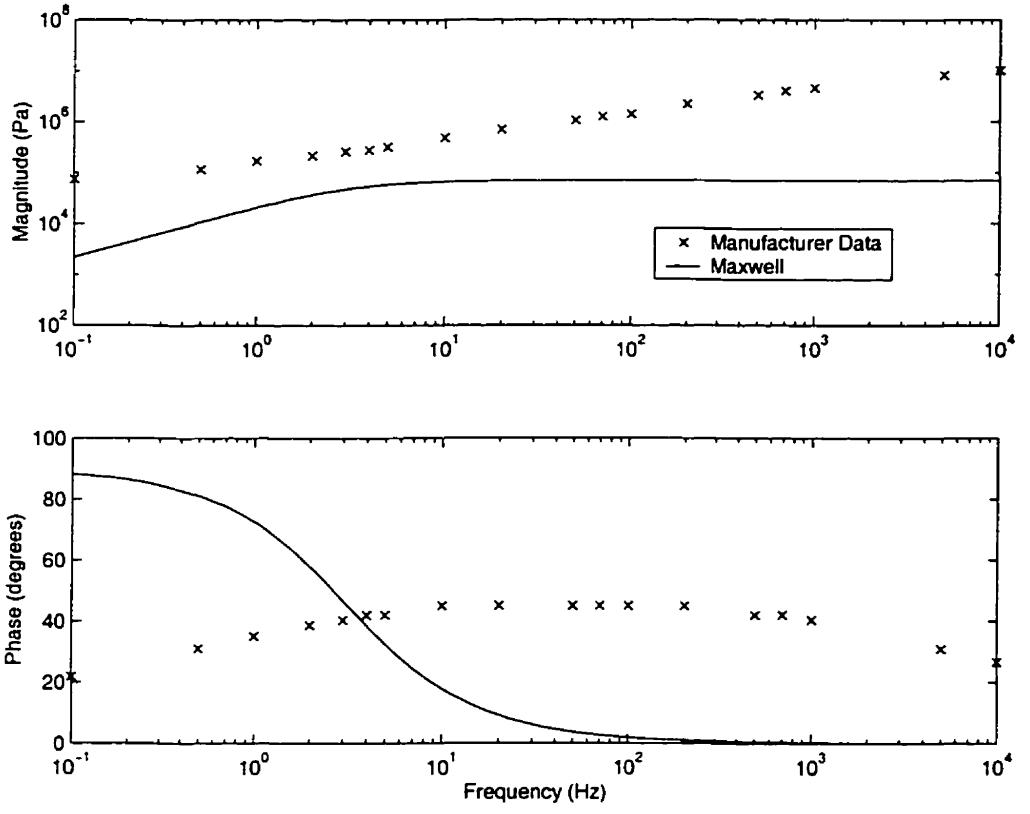

Figure 2.2: Complex Shear Modulus for the Maxwell Model ($G_e = 7.00 \times 10^4\ Pa$, $a = 0.05$)

infinitely. It should reach an asymptotic value at high frequencies, as it does for the Maxwell model, but this value should be the glassy modulus, and not the equilibrium modulus. A model combining these properties exhibits the same number of derivatives of stress and strain. The simplest of these models has one derivative of the stress added to the stress equal to one derivative of the strain added to the strain:

$$S_{ij} + b\dot{S}_{ij} = 2G_e e_{ij} + 2aG_e \dot{e}_{ij} \tag{2.11}$$

Assuming zero initial conditions, the Laplace transform of Equation 2.11 is taken to obtain the frequency-varying shear modulus:

$$\frac{S_{ij}(s)}{2e_{ij}(s)} = G^* = \frac{G_e + aG_e s}{1 + bs} \tag{2.12}$$

The limit as the frequency nears zero gives G_e, which agrees with physical observations:

$$\lim_{s \to 0} \frac{G_e + aG_e s}{1 + bs} = G_e \tag{2.13}$$

L'Hospital's rule is used to obtain the limit when the frequency becomes infinite:

$$\lim_{s \to \infty} \frac{G_e + aG_e s}{1 + bs} = \lim_{s \to \infty} \frac{aG_e}{b} = \frac{aG_e}{b} \tag{2.14}$$

The aymptotic value of G_g can be obtained if a and b are chosen accordingly. For both limits, the modulus no longer depends on s at very low or very high frequencies, and therefore, the phase is zero at these extremes, again satisfying physical observations.

Figure 2.3 shows some experimental data and an appropriate fit obtained for the standard linear solid model. Despite the model behaving appropriately at very low and very high frequencies, it still does not exhibit the gentle rise in both magnitude and phase observed experimentally. The magnitude rises too sharply from G_e to G_g, and the phase goes to higher values than seen experimentally. To obtain a better fit, more derivatives of the stress and strain are needed. As derivatives are added, the fit gets better, but the complexity

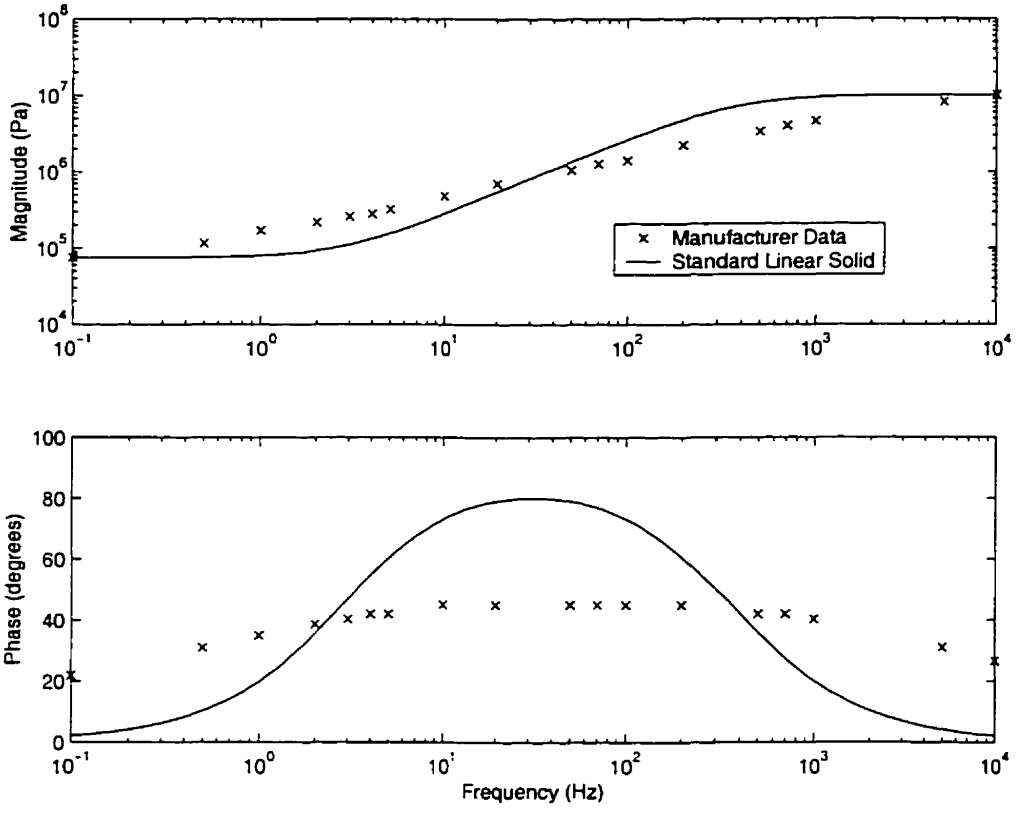

Figure 2.3: Complex Shear Modulus for the Standard Linear Solid Model ($G_e = 7.00 \times 10^4\ Pa$, $a = 0.05$, and $b = 3.5 \times 10^{-4}$)

increases and there are more parameters to identify. A general form of classical constitutive equations is:

$$\sum_{r=0}^{n} p_r \frac{d^r}{dt^r} S_{ij} = \sum_{r=0}^{m} q_r \frac{d^r}{dt^r} e_{ij} \tag{2.15}$$

If m and n are set to 1, then this form reduces to the standard linear solid model with the following parameters:

$$\begin{aligned} p_0 &= 1 \\ p_1 &= b \\ q_0 &= 1 \\ q_1 &= a \end{aligned} \tag{2.16}$$

2.3 Implementation of Classical Constitutive Equations in Commercial Finite Element Packages

Many commercial finite element packages offer some viscoelastic constitutive equations to represent material behaviour. Very often, the constitutive equation is expressed as a Prony series. This form of modulus is in fact a relaxation modulus written as:

$$G_{relax}(t) = G_e + \sum_{n=1}^{N} G_n \exp^{-t/\tau_n} \tag{2.17}$$

The relaxation modulus is defined as the modulus when the material is submitted to a step of strain of magnitude e^0_{xy} at $t = 0$ [41]. The complex modulus derived from Equation 2.17 is obtained by studying the relaxation response when a constant strain, e^0_{xy}, is applied:

$$S_{xy.relax}(t) = 2e^0_{xy} G_{relax}(t) \tag{2.18}$$

Since e^0_{12} is a step function applied at $t = 0$, the Laplace transform of Equation 2.18 is:

$$\frac{S_{xy.relax}(s)}{\frac{2e^0_{xy}}{s}} = G_{relax}(s) \tag{2.19}$$

The complex shear modulus is defined as the ratio of the stress over the strain when using the engineering strain. Rogers [42] and Tschoegl [41] used that definition to manipulate Equation 2.19 and obtain the complex modulus:

$$G^* = sG_{relax}(s) \tag{2.20}$$

Equation 2.17 can now be used with Equation 2.20 to get the complex modulus. The Laplace transform of Equation 2.17 is taken:

$$G^* = s(\frac{G_e}{s} + \sum_{n=1}^{N} \frac{G_n \tau_n}{1 + \tau_n s}) \tag{2.21}$$

By simplifying, a form for the complex shear modulus represented by the Prony series is obtained:

$$G^* = G_e + \sum_{n=1}^{N} \frac{G_n \tau_n s}{1 + \tau_n s} \tag{2.22}$$

The shear modulus is sometimes written using the glassy modulus rather than the equilibrium modulus [41]:

$$G_e = G_g - \sum_{n=1}^{N} G_g g_n \tag{2.23}$$

where

$$G_n = G_g g_n \tag{2.24}$$

Equation 2.22 becomes:

$$G^* = G_g - \sum_{n=1}^{N} G_g g_n + \sum_{n=1}^{N} \frac{G_g g_n \tau_n s}{1 + \tau_n s} \tag{2.25}$$

This last form is the one most often encountered in commercial finite element packages.

Figure 2.4 shows Prony series with one term, two terms, and five terms to model the behaviour of a typical polymer. With five terms, the Prony series is following quite closely the experimental data and more terms would give an even smoother curve. The numerical values of the parameters for the three curves are given in Chapter 3. The good fit obtained with five terms is related to the specific behaviour of the material. Other materials would require less or more terms for a suitable fit.

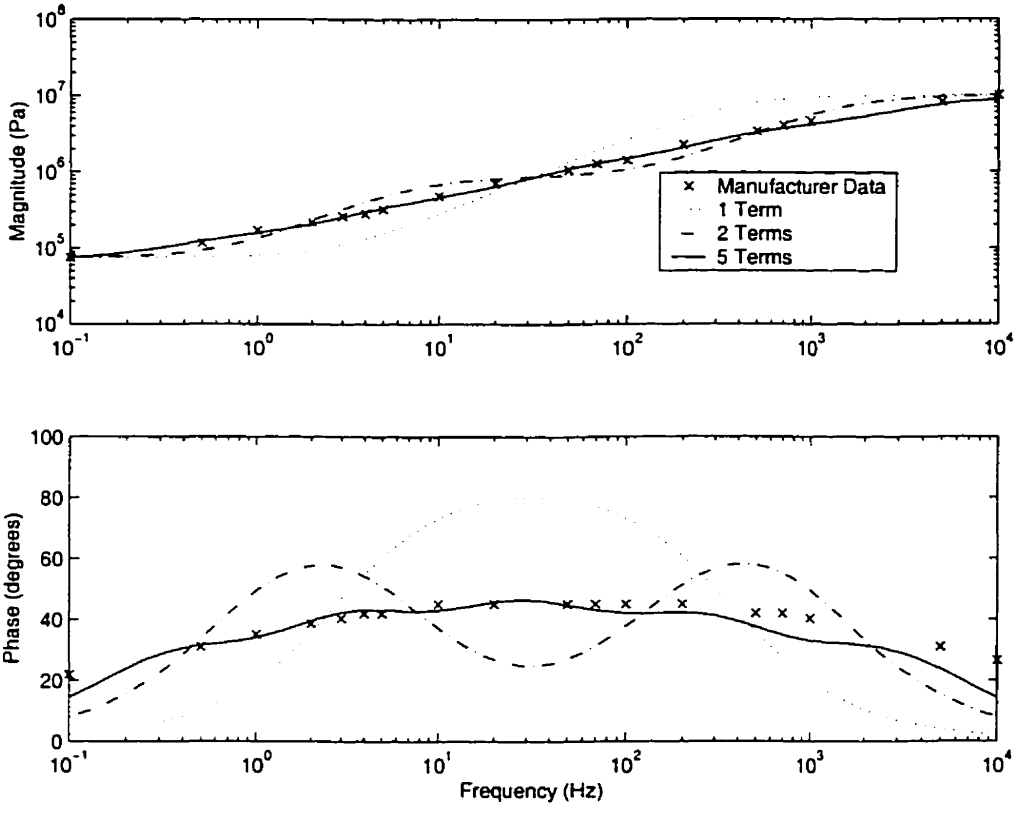

Figure 2.4: Complex Shear Modulus for Prony Series

2.3.1 Equivalence of the One-Term Prony Series and the Standard Linear Solid Model

The Prony series gives identical results to a classical constitutive equation with the same number of derivatives for the stress and strain. Therefore, a one-term Prony series is equivalent to a standard linear solid model. The one-term Prony series is written:

$$G^* = G_e + \frac{G_1 \tau_1 s}{1 + \tau_1 s} \tag{2.26}$$

This is compared to the complex modulus representing a standard linear solid model:

$$G^* = \frac{G_e + a G_e s}{1 + bs} \tag{2.27}$$

An equivalence for the parameters is found:

$$\begin{aligned} a &= \tau_1 + \frac{G_1 \tau_1}{G_e} \\ b &= \tau_1 \end{aligned} \tag{2.28}$$

The same can be done for higher order classical constitutive equations and higher order Prony series.

2.3.2 The Zener Model Implemented in *Samcef*

Samcef, a commercial finite element package, offers a Zener model in addition to the Prony series, the Voigt-Kelvin model, and the Maxwell model. The Zener model is similar to a standard linear solid model, but the definition of the parameters as implemented in *Samcef* gives it some peculiar properties. It is written:

$$S_{ij} + \frac{\alpha}{1+\gamma}\dot{S}_{ij} = 2\frac{\gamma}{1+\gamma}G_e e_{ij} + 2\frac{\alpha}{1+\gamma}G_e \dot{e}_{ij} \tag{2.29}$$

where α and γ are damping parameters. Assuming zero initial conditions, the Laplace transform of Equation 2.29 is taken to obtain the frequency-varying shear modulus:

$$\frac{S_{ij}(s)}{2e_{ij}(s)} = G^* = \frac{\gamma G_e + \alpha G_e s}{(1+\gamma) + \alpha s} \tag{2.30}$$

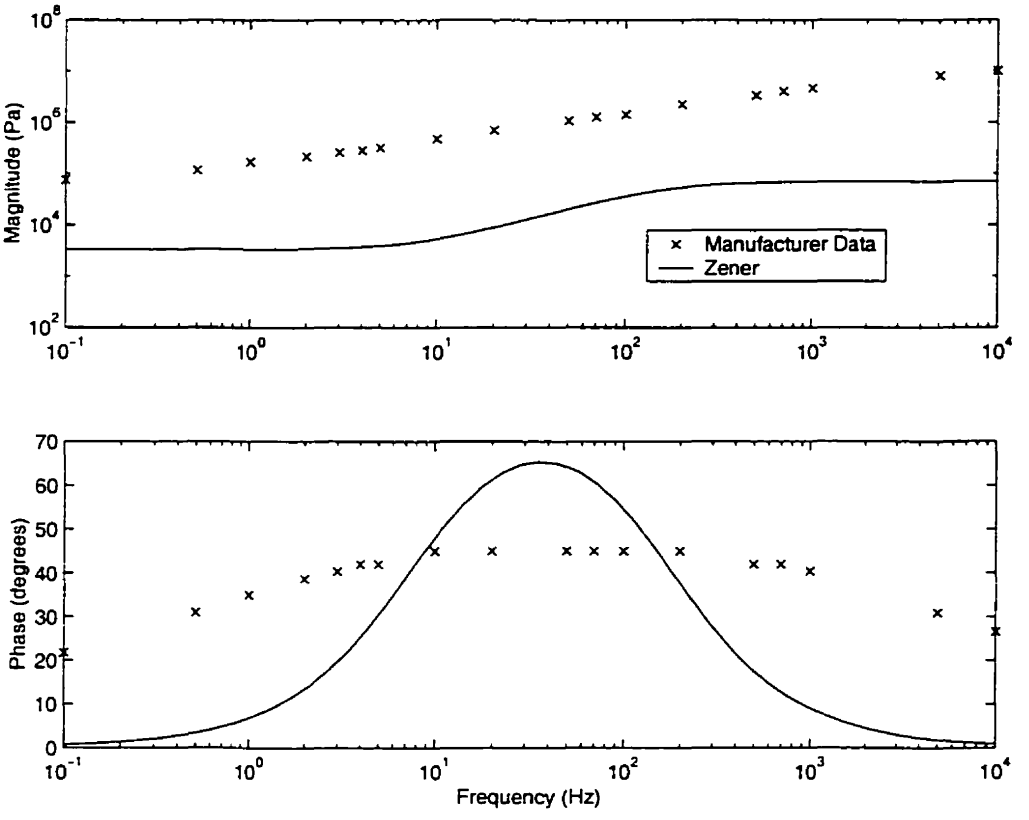

Figure 2.5: Complex Shear Modulus for the Zener Model ($G_e = 7.00 \times 10^4\ Pa$, $\gamma = 0.05$, and $\alpha = 0.001$)

The limit as the frequency nears zero is taken:

$$\lim_{s \to 0} \frac{\gamma G_e + \alpha G_e s}{(1+\gamma) + \alpha s} = \frac{\gamma}{1+\gamma} G_e \tag{2.31}$$

This result does not satisfy physical observations for most polymers. At very low frequencies, the modulus should approach G_e, whereas this model predicts an asymptotic value equal to a fraction of G_e.

L'Hospital's rule is used to obtain the limit when the frequency becomes infinite:

$$\lim_{s \to \infty} \frac{\gamma G_e + \alpha G_e s}{(1+\gamma) + \alpha s} = \lim_{s \to \infty} \frac{\alpha G_e}{\alpha} = G_e \tag{2.32}$$

At very high frequencies, the model tends to G_e rather than G_g. This model does not satisfy physical observations of most solid polymers. Figure 2.5 shows some experimental data and an appropriate fit obtained for the Zener model. The fit for the phase is similar to the fit obtained with the standard linear

solid model, but because the magnitude tends to G_e at high frequencies, it remains well below the experimental values. The Zener model would satisfy physical observations if the parameter G_e in Equation 2.29 was replaced by G_g.

2.4 Models Based on Fractional Derivatives

All models with few parameters reviewed in the preceding sections had serious shortcomings in representing the behaviour of polymers over a broad frequency range. The only models achieving this goal were the Prony series with several terms, or alternately, classical constitutive equations with several derivatives of the stress and strain. These can be used when the complex modulus is known over a wide frequency range. Usually, it is the case for materials sold commercially as damping layers, but it is not often the case for polymeric materials used as components of mechanical systems. Very often, the equipment to obtain the complex modulus is not available and limited data is known. Faced with that prospect, many engineers choose to model the damping behaviour using as few parameters as possible.

The main problem with classical constitutive equations with few parameters is the sharp rise in the magnitude of the modulus. This is governed by the first derivative of the strain which imposes an increase in magnitude of one decade per decade on a log log plot of the magnitude versus the frequency. To illustrate this for the Voigt-Kelvin model, the complex modulus is written as:

$$G^* = G_e + aG_e s \tag{2.33}$$

where s is equal to $i2\pi f$. The magnitude of G^* is:

$$|G^*| = \sqrt{G_e^2 + (aG_e 2\pi f)^2} \tag{2.34}$$

The logarithm is taken on both sides of the preceding equation:

$$\begin{aligned} \log|G^*| &= \log\sqrt{G_e^2 + (aG_e 2\pi f)^2} \\ &= \frac{1}{2}\log\left(G_e^2 + (aG_e 2\pi f)^2\right) \end{aligned} \tag{2.35}$$

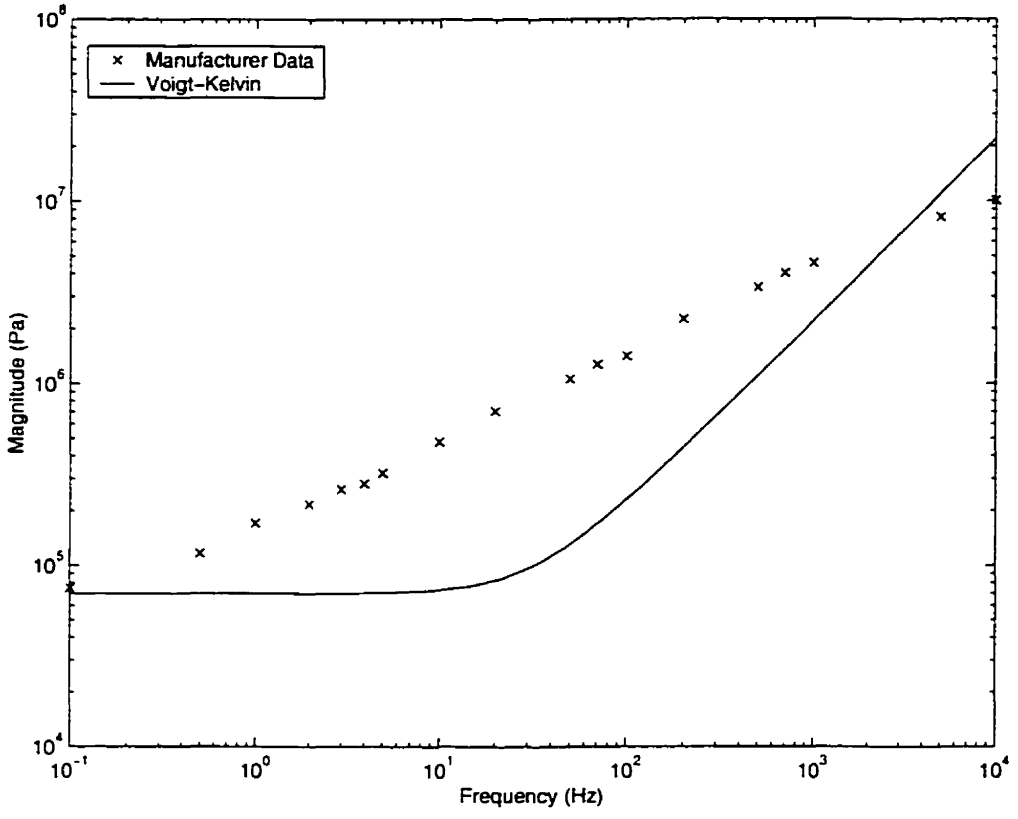

Figure 2.6: Magnitude of the Shear Modulus for The Voigt-Kelvin Model

For high frequencies, the first term can be neglected compared to the second and the logarithm of the magnitude becomes:

$$\begin{aligned} \log|G^*| &= \frac{1}{2}\log(aG_e 2\pi f)^2 \\ &= \log(aG_e 2\pi f) \end{aligned} \tag{2.36}$$

The magnitude of the modulus increases one decade as the frequency increases one decade for high values of frequency. Figure 2.6 shows the magnitude of the shear modulus shown previously as part of Figure 2.1 expressed on a log log scale for the magnitude. The slope of most polymers is much gentler than a one decade increase of magnitude per decade of frequency. Bagley and Torvik [43] were among the first to suggest using fractional derivatives instead of integer derivatives in the modelling of viscoelastic behaviour. Fractional derivatives are derivatives of an order between 0 and 1. It is defined as [23] :

$$D^{\xi}e_{ij}(t) = \frac{1}{\Gamma(1-\xi)}\frac{d}{dt}\int_0^t \frac{e_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau \tag{2.37}$$

The order of the fractional derivative is smaller than a first order derivative, and therefore, the slope of the magnitude with respect to frequency should be gentler. To carry on with this analysis, the Laplace transform of the fractional derivative is needed.

2.4.1 The Fractional Derivative Laplace Transform

The transform of $\mathcal{L}\left[D^{\xi}e_{ij}(t)\right]$ is sought for zero initial condition problems. The definition of the fractional derivative used in Equation 2.37 implies the material is in its undeformed state at $t = 0$. To obtain the Laplace transform of the fractional derivative, a new function h is defined:

$$\begin{aligned} h(t) &= \frac{1}{\Gamma(1-\xi)} \int_0^t \frac{e_{ij}(\tau)}{(t-\tau)^{\xi}} d\tau \\ \Rightarrow D^{\xi}e_{ij}(t) &= \frac{d}{dt}h(t) \end{aligned} \tag{2.38}$$

Use is made of the following property of the Laplace transforms [44]:

$$\mathcal{L}\left[\frac{d}{dt}f(t)\right] = sF(s) - f(0) \tag{2.39}$$

Substituting f by h yields:

$$\begin{aligned} \mathcal{L}\left[D^{\xi}e_{ij}(t)\right] &= \mathcal{L}\left[\frac{d}{dt}h(t)\right] \\ &= sH(s) - h(0) \\ &= s\mathcal{L}\left[\frac{1}{\Gamma(1-\xi)} \int_0^t \frac{e_{ij}(\tau)}{(t-\tau)^{\xi}} d\tau\right] \\ &\quad - \frac{1}{\Gamma(1-\xi)} \int_0^t \frac{e_{ij}(\tau)}{(t-\tau)^{\xi}} d\tau \Big|_{t=0} \end{aligned} \tag{2.40}$$

Because the last term of Equation 2.40 is an integral from 0 to 0, it is always equal to 0, and consequently:

$$\mathcal{L}\left[D^{\xi}e_{ij}(t)\right] = s\mathcal{L}\left[\frac{1}{\Gamma(1-\xi)} \int_0^t \frac{e_{ij}(\tau)}{(t-\tau)^{\xi}} d\tau\right] \tag{2.41}$$

The integral on the right hand side of Equation 2.41 is a convolution product and can be solved using the following property of Laplace transforms [44]:

$$\begin{aligned} \mathcal{L}\left[\int_0^t f(u)g(t-u)du\right] = F(s)G(s) \quad \text{where} \quad G(s) &= \mathcal{L}\left[g(t)\right] \\ \text{and} \quad F(s) &= \mathcal{L}\left[f(t)\right] \end{aligned} \tag{2.42}$$

Some simple substitutions are needed:

$$\begin{aligned} \tau &= u \\ f(u) &= e_{ij}(\tau) \\ g(t-u) &= \frac{1}{(t-\tau)^{\xi}\Gamma(1-\xi)} = \frac{(t-\tau)^{-\xi}}{\Gamma(1-\xi)} \end{aligned} \tag{2.43}$$

Using Equation 2.43 in Equation 2.41 yields:

$$\begin{aligned} \mathcal{L}\left[D^{\xi}e_{ij}(t)\right] &= s\mathcal{L}\left[\frac{1}{\Gamma(1-\xi)}\int_0^t \frac{e_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau\right] \\ &= se_{ij}(s)\mathcal{L}\left[\frac{t^{-\xi}}{\Gamma(1-\xi)}\right] \end{aligned} \tag{2.44}$$

To proceed, the following transformation is needed [44]:

$$\mathcal{L}\left[\frac{t^{n-1}}{\Gamma(n)}\right] = \frac{1}{s^n} \quad \text{where} \quad n > 0 \tag{2.45}$$

Letting $n = 1 - \xi$, Equation 2.44 becomes:

$$\begin{aligned} \mathcal{L}\left[D^{\xi}e_{ij}(t)\right] &= se_{ij}(s)\mathcal{L}\left[\frac{t^{-\xi}}{\Gamma(1-\xi)}\right] \\ &= se_{ij}(s)\frac{1}{s^{1-\xi}} \\ &= s^{\xi}e_{ij}(s) \end{aligned} \tag{2.46}$$

This is a nice and compact result easy to use. It also illustrates the fact that when ξ tends towards 1, the same result is obtained as for the first derivative of e_{ij} for zero initial conditions, whereas when ξ tends towards 0, the same Laplace transform as the undifferentiated function e_{ij} is reached.

2.4.2 The Fractional Voigt-Kelvin Model

The most basic model making use of a fractional derivative is the Voigt-Kelvin model in which the first order derivative is replaced with a fractional derivative:

$$S_{ij} = 2G_e e_{ij} + 2aG_e D^{\xi}e_{ij} \tag{2.47}$$

The Laplace transform for zero initial conditions brings out the complex modulus:

$$\frac{S_{ij}(s)}{2e_{ij}(s)} = G^* = G_e + aG_e s^{\xi} \tag{2.48}$$

To observe the impact of the fractional order of the derivative, the magnitude of G^* is needed. First, the Laplace variable, s, is written as $i2\pi f$:

$$G^* = G_e + aG_e(i2\pi f)^\xi \tag{2.49}$$

The identity $i^\xi = \cos(\frac{\pi\xi}{2}) + i\sin(\frac{\pi\xi}{2})$ is used:

$$G^* = G_e + aG_e(2\pi f)^\xi(\cos(\frac{\pi\xi}{2}) + i\sin(\frac{\pi\xi}{2})) \tag{2.50}$$

The magnitude of G^* is written:

$$|G^*| = \sqrt{(G_e + aG_e(2\pi f)^\xi\cos(\frac{\pi\xi}{2}))^2 + (aG_e(2\pi f)^\xi\sin(\frac{\pi\xi}{2}))^2} \tag{2.51}$$

Use is made of the identity $\cos^2(\frac{\pi\xi}{2}) + \sin^2(\frac{\pi\xi}{2}) = 1$:

$$|G^*| = \sqrt{G_e^2 + 2aG_e^2(2\pi f)^\xi\cos(\frac{\pi\xi}{2}) + a^2G_e^2(2\pi f)^{2\xi}} \tag{2.52}$$

The logarithm is taken on both sides of the preceding equation:

$$\log|G^*| = \frac{1}{2}\log\left(G_e^2 + 2aG_e^2(2\pi f)^\xi\cos(\frac{\pi\xi}{2}) + a^2G_e^2(2\pi f)^{2\xi}\right) \tag{2.53}$$

For high frequencies and a value of ξ of approximately 0.5, the first term can be neglected compared to the other two, and the remaining terms can then be reorganized:

$$\begin{aligned}
\log|G^*| &= \frac{1}{2}\log\left(2aG_e^2(2\pi f)^\xi\cos(\frac{\pi\xi}{2}) + a^2G_e^2(2\pi f)^{2\xi}\right) \\
&= \frac{1}{2}\log\left(aG_e^2(2\pi f)^\xi\right)\left(2\cos(\frac{\pi\xi}{2}) + a(2\pi f)^\xi\right) \\
&= \frac{1}{2}\left(\log\left(aG_e^2(2\pi f)^\xi\right) + \log\left(2\cos(\frac{\pi\xi}{2}) + a(2\pi f)^\xi\right)\right)
\end{aligned} \tag{2.54}$$

Again, the term $2\cos(\frac{\pi\xi}{2})$ can be neglected in front of the remaining term for high frequencies for a value of ξ of approximately 0.5. The terms left are split to show the effect of ξ:

$$\begin{aligned}
\log|G^*| &= \frac{1}{2}\left(\log\left(aG_e^2(2\pi f)^\xi\right) + \log\left(a(2\pi f)^\xi\right)\right) \\
&= \frac{1}{2}\left(\log a + \log G_e^2 + \log(2\pi f)^\xi + \log a + \log(2\pi f)^\xi\right) \\
&= \log a + \log G_e + \xi\log(2\pi f)
\end{aligned} \tag{2.55}$$

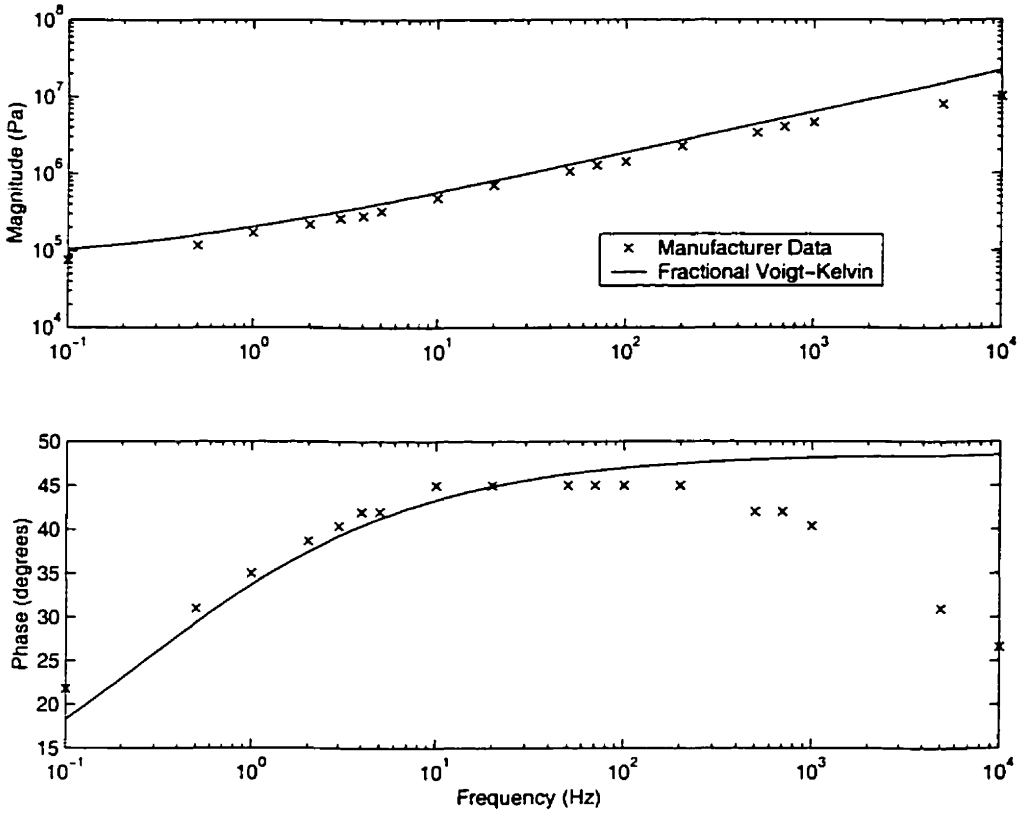

Figure 2.7: Complex Shear Modulus for The Fractional Voigt-Kelvin Model ($G_e = 7.00 \times 10^4\ Pa$, $a = 0.8$, and $\xi = 0.54$)

At high frequencies, when the frequency increases by one decade, the magnitude of the modulus increases by ξ decade, and ξ is between 0 and 1. For the Voigt-Kelvin model, the increase was of a full decade. The fractional Voigt-Kelvin model provides a gentler slope. This conclusion also holds for small values of ξ. However, if ξ is too low, the material behaves almost elastically and the slope approaches zero. Figure 2.7 shows experimental data for 3M ISD 112 and the fractional Voigt-Kelvin model for a value of 0.54 for ξ. When the frequency increases by one decade, the magnitude increases by approximately half a decade. This gentler slope is much closer to the actual behaviour of the polymer than what is obtained with the Voigt-Kelvin model. The rise in the phase value is also more representative of the experimental data. It reaches an asymptotic value closer to physical observations. A perfect model, however, would have a phase decreasing rather than reaching an aymptotic

value once it is passed its maximum value. A model with these properties is the standard linear solid model. The main difference between the standard linear solid model and the Voigt-Kelvin model is the derivative of the stress. A fractional derivative of the stress would probably allow a gentler negative slope for the phase than obtained with a first order derivative. This idea was also explored by Bagley and Torvik [43].

2.4.3 The Fractional Standard Linear Solid Model

Bagley [45] used a model with a fractional derivative of the stress and one of the strain of the same order. This model is:

$$S_{ij} + bD^{\xi}S_{ij} = 2G_e e_{ij} + 2aG_e D^{\xi} e_{ij} \tag{2.56}$$

Assuming zero initial conditions, the Laplace transform of Equation 2.56 is taken to obtain the frequency-varying shear modulus:

$$\frac{S_{ij}(s)}{2e_{ij}(s)} = G^* = \frac{G_e + aG_e s^{\xi}}{1 + bs^{\xi}} \tag{2.57}$$

Figure 2.8 shows an appropriate fit of the fractional standard linear solid model for a typical polymer. This model has the best fit for all the models presented with few parameters. This is a marked advantage when limited data is available to identify the parameters as is often the case in the engineering practice.

2.5 The Assumption of a Constant Poisson's Ratio

So far, a constant bulk modulus has always been assumed. It is an assumption that is often used because it is quite close to physical observations [41], and more often than not, the full behaviour with respect to frequency for all elasticity constants is not known.

Another assumption very often used is to assume constant Poisson's ratio. It does simplify the equations in some cases, and for low frequencies, it does

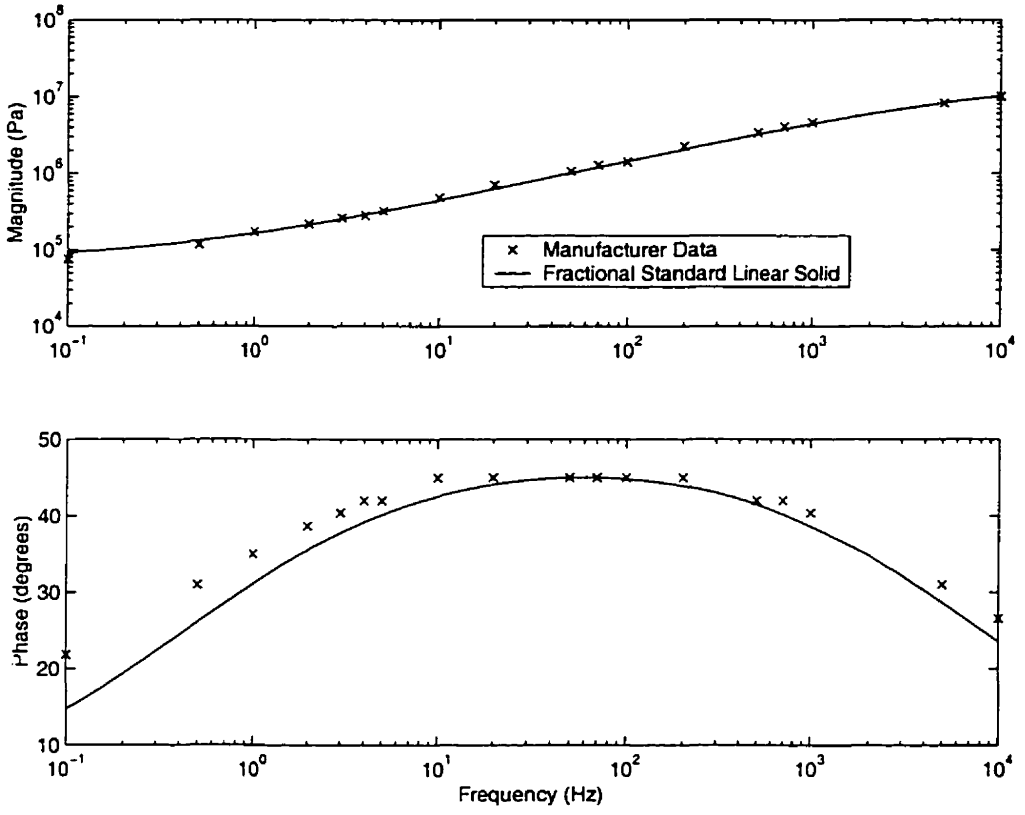

Figure 2.8: Complex Shear Modulus for The Fractional Standard Linear Solid Model ($G_e = 7.00 \times 10^4\ Pa$, $\xi = 0.56$, $a = 0.57$, and $b = 0.0023$)

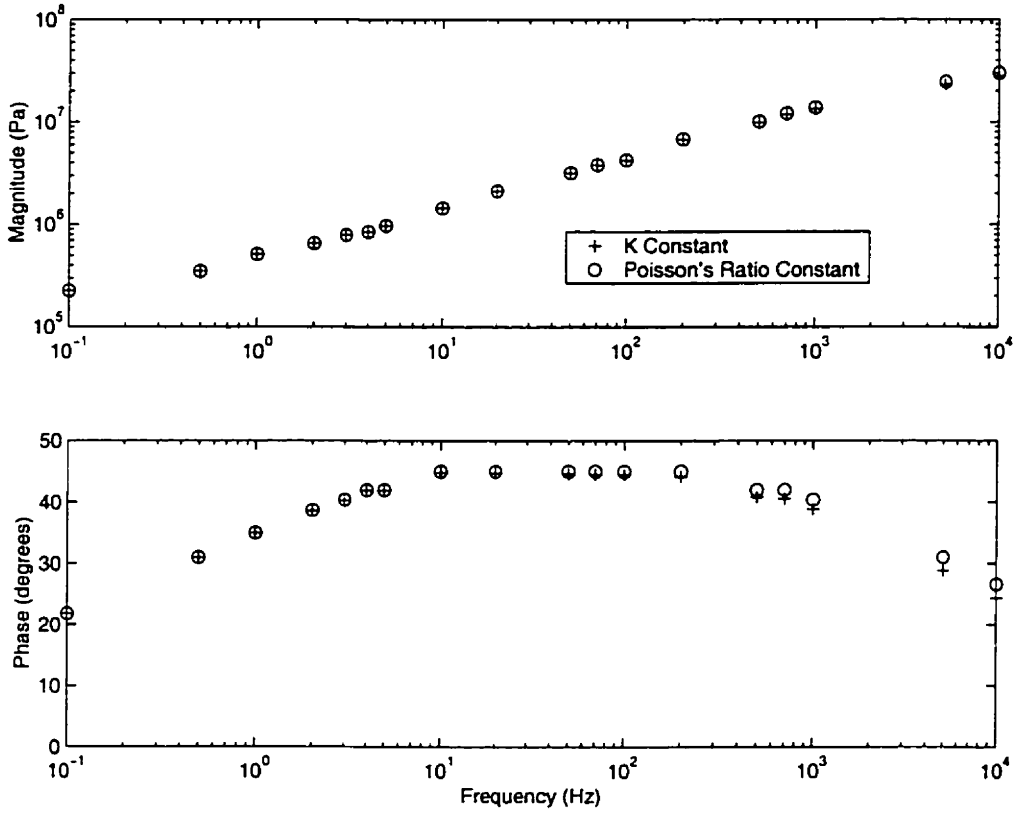

Figure 2.9: Young's Modulus for a Constant Poisson's Ratio or a Constant Bulk Modulus

not diverge too much from the results obtained with a constant bulk modulus assumption, especially in a nearly incompressible case. Figure 2.9 shows the calculated Young's modulus when both assumptions are applied to the shear modulus data of a typical polymer. The equilibrium Poisson's ratio is taken to be 0.499, representing a nearly incompressible material, which is typical of many elastomers. If Poisson's ratio is assumed constant, Young's modulus is calculated as:

$$E^* = 2G^*(1 + \nu_e) \tag{2.58}$$

where E^* is the complex Young's modulus and ν_e is the equilibrium Poisson's ratio.

If the bulk modulus is assumed constant, then, Young's modulus is obtained with:

$$E^* = \frac{9K_eG^*}{3K_e + G^*} \tag{2.59}$$

where the equilibrium bulk modulus, K_e, is

$$K_e = \frac{2G_e(1+\nu_e)}{3(1-2\nu_e)} \tag{2.60}$$

with a value of 70000 Pa for G_e, which is the value obtained for the real part of the modulus at 0.1 Hz. This frequency is the lowest frequency for which data is given by the manufacturer for 3M ISD 112. The results of Figure 2.9 show values of Young's modulus to be nearly the same under both assumptions. The values shown on the graph are calculated from the shear modulus values read from the chart provided by the manufacturer. The chart being difficult to read, it results in a curve not being entirely smooth. Therefore, the results are shown as discrete points representing the points read from the chart. For a nearly incompressible case, the effect of either assumption is felt mainly on the bulk modulus and Poisson's ratio themselves. This conclusion does not hold for lower values of Poisson's ratio. Figure 2.10 shows the complex Young's modulus for an equilibrium Poisson's ratio of 0.3. For such a value, the constant bulk modulus assumption and the constant Poisson's ratio assumption give different results.

For the nearly incompressible case, Figure 2.11 shows the calculated bulk modulus if Poisson's ratio is assumed constant, and Figure 2.12 shows the calculated Poisson's ratio if the bulk modulus is assumed constant. Poisson's ratio decreases slightly with frequency for a constant bulk modulus assumption, but the bulk modulus increases significantly at high frequencies for a constant Poisson's ratio assumption. At low frequencies, however, both assumptions should give similar results.

2.6 Models Implemented in *Samcef*

One of the goal of this thesis is to implement a fractional order constitutive equation in a commercial finite element package with the objective to compare the efficiency of this model to other available models. Ideally, a fractional order model would be more accurate and compute faster. Even if it does

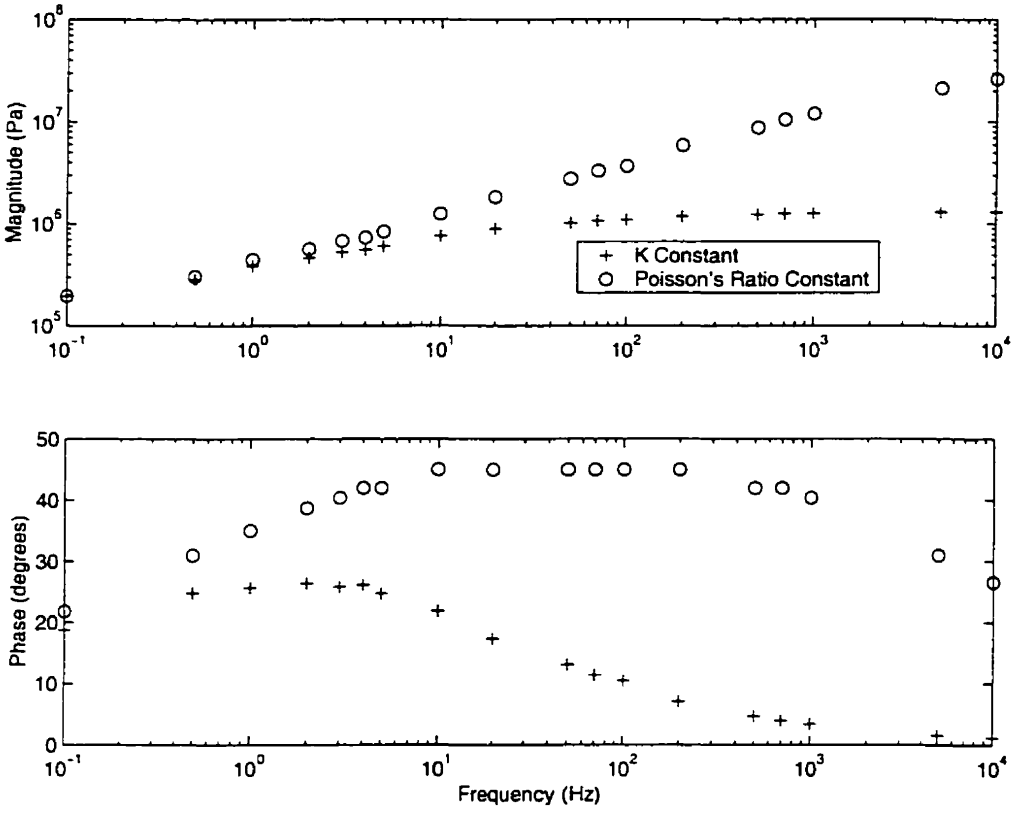

Figure 2.10: Young's Modulus for a Constant Poisson's Ratio or a Constant Bulk Modulus for an Equilibrium Poisson's Ratio of 0.3

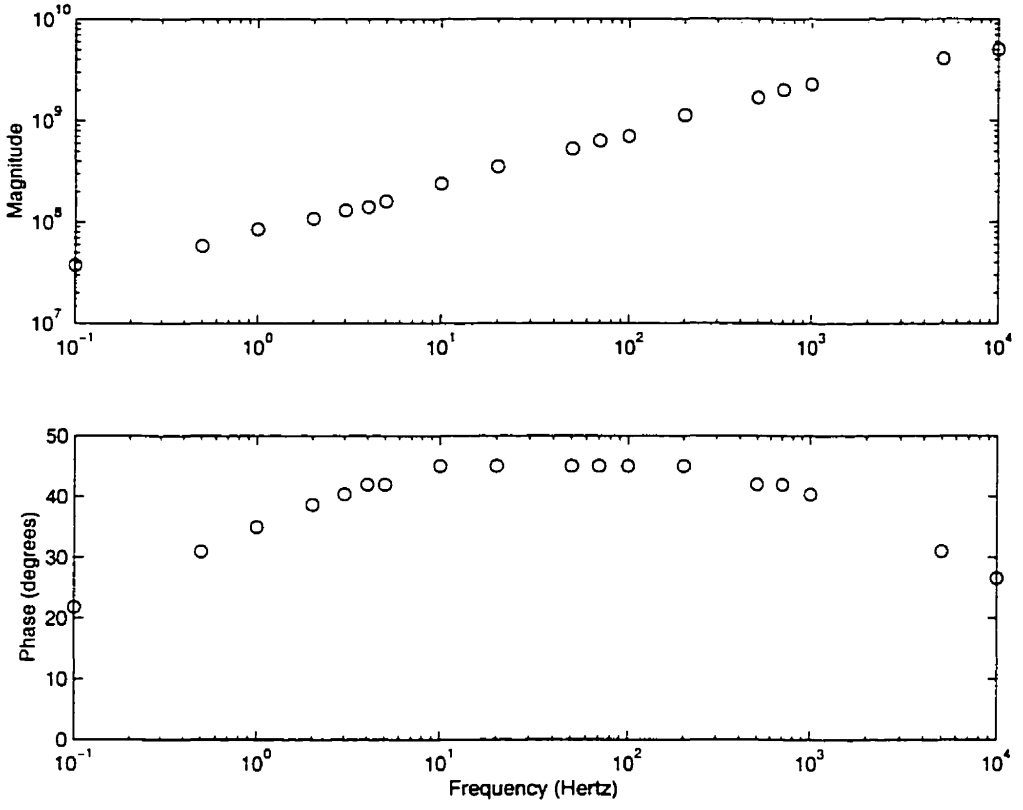

Figure 2.11: The Bulk Modulus for a Constant Poisson's Ratio

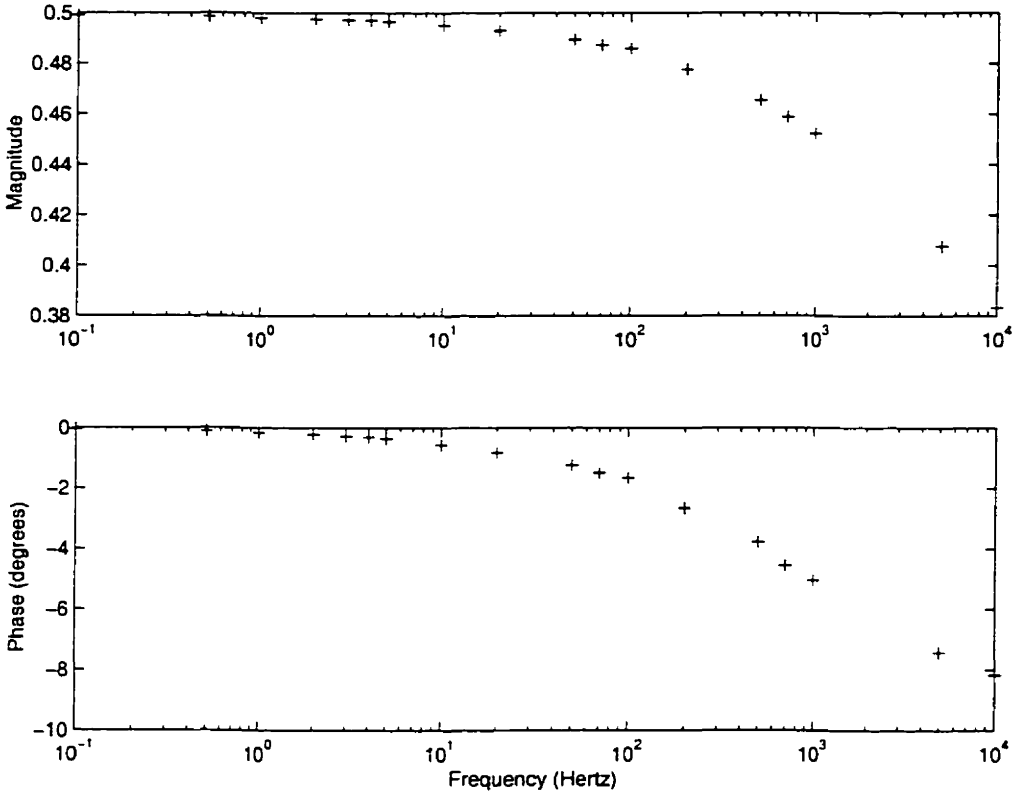

Figure 2.12: Poisson's Ratio for a Constant Bulk Modulus

not compute faster, it could still be of interest because of the few parameters needed to represent the viscoelastic behaviour. In practical applications, often only partial data is available, such as a frequency response, and it is easier to fit a model with few parameters.

Samcef version 8.1 [46] is the finite element package used for this work. The nonlinear dynamic module of *Samcef* used to model the viscoelastic behaviour is also the nonlinear dynamic module of *Nastran*. *Samcef* allows a user subroutine for the constitutive equation and a fractional Voigt-Kelvin model is implemented. A fractional standard linear solid model would be even better, but it involves a fractional derivative of the stress and one of the strain and therefore, it is more demanding on computer ressources. This work concentrates on the fractional Voigt-Kelvin model as a first evaluation of the efficiency of a fractional order model. The assumption of a constant bulk modulus is used.

Samcef also offers the Prony series. The fractional Voigt-Kelvin model will be compared to Prony series of various orders and for both assumptions, a constant Poisson's ratio, and then a constant bulk modulus. The three other models available in *Samcef* all use the constant Poisson's ratio assumption. They are the Voigt-Kelvin model, the Maxwell model, and the Zener model. Table 2.1 summarizes the various models compared in this study.

2.7 Conclusion

Models representing the viscoelastic behaviour are variations of Hooke's law where derivatives of the stress and the strain are added. For a model to represent the physical observations, the same number of derivatives for the stress and the strain are needed. A model with only one derivative of the stress and one derivative of the strain would have the general characteristic of the observed behaviour of a polymer, but such a modulus implies a variation with frequency much more acute than the physical behaviour. Several derivatives are needed for a good fit between the model and the experimental data for the

Model	Complex Modulus	Assumption
Voigt-Kelvin	$G^* = G_e + aG_e s$	Constant Poisson's Ratio
Maxwell	$G^* = \frac{aG_e s}{1+as}$	Constant Poisson's Ratio
Zener	$G^* = \frac{\gamma G_e + aG_e s}{(1+\gamma)+as}$	Constant Poisson's Ratio
Prony Series	$G^* = G_e + \sum_{n=1}^{N} \frac{G_n \tau_n s}{1+\tau_n s}$	Constant Poisson's Ratio
Prony Series	$G^* = G_e + \sum_{n=1}^{N} \frac{G_n \tau_n s}{1+\tau_n s}$	Constant Bulk Modulus
Fractional Voigt-Kelvin	$G^* = G_e + aG_e s^{\xi}$	Constant Bulk Modulus

Table 2.1: Models Used in this Study

complex modulus.

An alternative to these classical models is a model with fractional derivatives, derivatives of an order between 0 and 1, rather than integer derivatives. These types of models can result in representative behaviour with only one fractional derivative of the stress and one fractional derivative of the strain. However, they are more complex to handle mathematically.

In this thesis, a fractional model with one derivative of the strain will be implemented in a commercial finite element package. It will then be compared to classical models available in the chosen package, *Samcef*. The next chapter deals with the identification of the parameters of the various viscoelastic models for two materials, polymethylmethacrylate and 3M ISD 112, an acrylic based polymer.

Chapter 3

Viscoelastic Behaviour of Polymethylmethacrylate and an Acrylic Based Polymer

3.1 Introduction

The models developed in the previous chapter are applied to two polymeric materials: polymethylmethacrylate (PMMA) and an acrylic based polymer, 3M ISD 112. The goal is to define the parameters of the various models for these two materials and then, compare simulation results with experimental results for structures made of these materials and submitted to some dynamic loads.

PMMA is chosen because of its availability, low cost, and ease of machining to given dimensions. The other material, 3M ISD 112, is an acrylic based polymer manufactured by 3M and used as a damping layer. It will be sandwiched between a steel beam and an aluminium constraining layer and submitted to large displacements.

3.2 Behaviour of Polymethylmethacrylate

Testing of PMMA to determine its frequency-dependent modulus is necessary. First, the equilibrium modulus is found. Then, the modulus under various frequencies is found and these data points serve as the basis to identify the

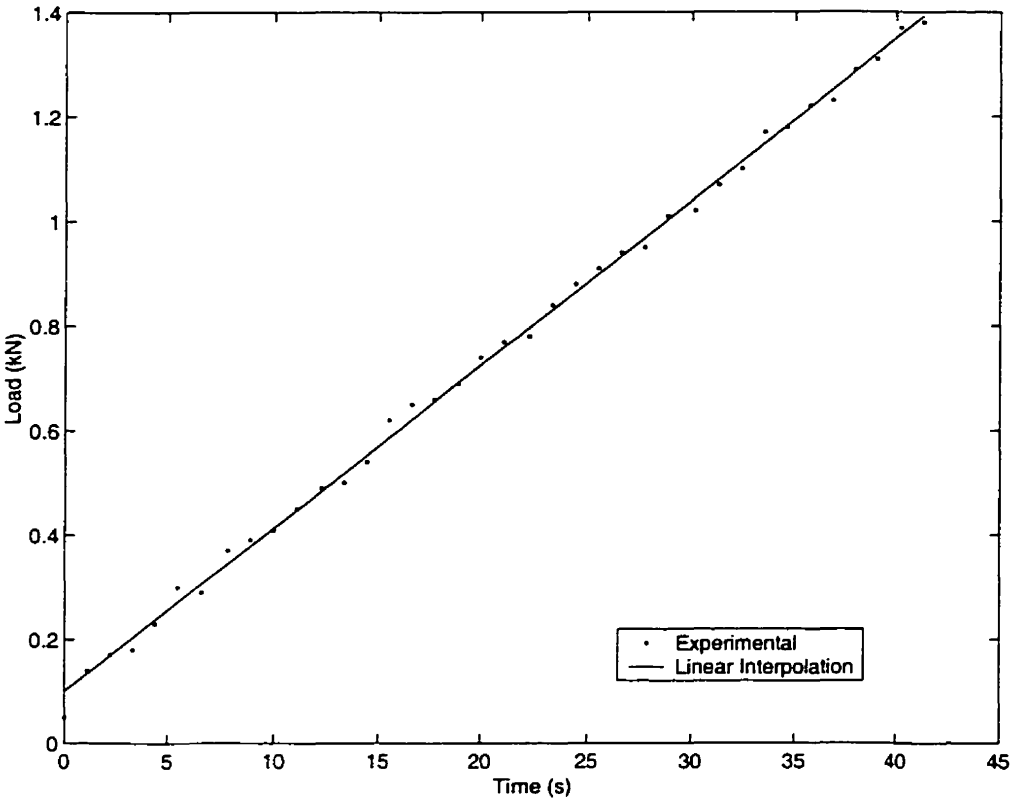

Figure 3.1: Loading History for the Tensile Test Done on PMMA

parameters of the models defined in the previous chapter.

3.2.1 Equilibrium Modulus

One parameter used to model the behaviour of the material which is easily found experimentally is the equilibrium Young's modulus. To determine it for PMMA, a tensile test is performed with an MTS hydraulic testing machine. The PMMA sample is a rectangular prism with a section of 22.5 mm by 5.88 mm. The strain is measured using an extensometer. Figure 3.1 shows the load as a function of time. The loading rate is 31.3 N/s. This value is found with the linear interpolation done using the *polyfit* function of *Matlab*, which is based on least squares. Figure 3.2 shows the stress-strain relationship. The slope of the linear interpolation represents the equilibrium Young's modulus:

$$E_e = 3.43 \times 10^9 \ Pa \tag{3.1}$$

The equilibrium shear modulus and the equilibrium bulk modulus can be

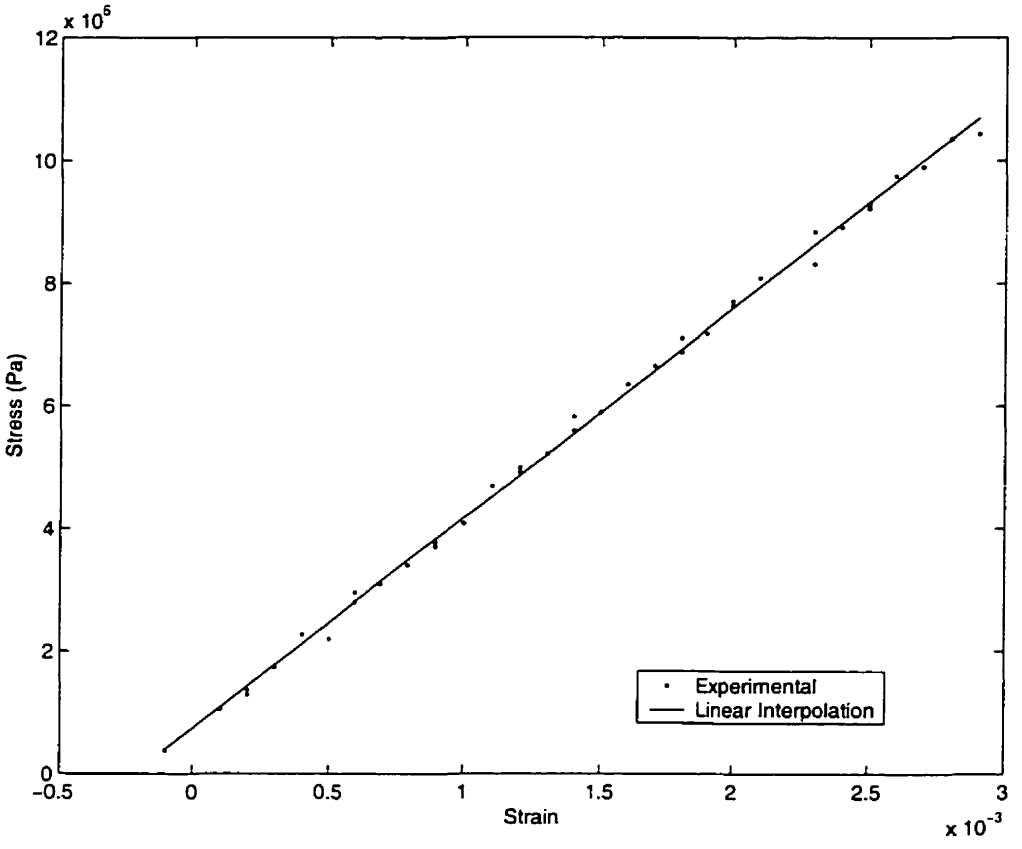

Figure 3.2: Tensile Test for PMMA

calculated with the value of the equilibrium Poisson's ratio. This value is taken from Van Krevelen [47] and is equal to 0.4 for PMMA. The equilibrium shear modulus, G_e, and the equilibrium bulk modulus, K_e are:

$$G_e = \frac{E_e}{2(1+\nu_e)} = 1.23 \times 10^9 \ Pa \tag{3.2}$$

$$K_e = \frac{E_e}{3(1-2\nu_e)} = 5.72 \times 10^9 \ Pa \tag{3.3}$$

3.2.2 Frequency Dependent Behaviour

The frequency response of PMMA is established by imposing a cyclic sinusoidal load on the material for various frequencies. The load is kept low enough for the strain to remain below 0.003, which is the strain region of interest for the type of applications studied in Chapter 6.

The magnitude of the complex Young's modulus is obtained by dividing the amplitude of the sinusoidal stress signal by the amplitude of the sinusoidal strain signal. The phase of the complex Young's modulus is positive and represents the delay between the strain and the stress signals. The first step is to analyse the stress and strain signals and obtain their amplitudes and phases.

Analyses of the Stress and Strain Signals

The dynamic tests are load control experiments where the load is a sinusoidal signal of a given frequency and amplitude. However, the control is not perfect and the resulting signal is not a perfect sine wave. Dividing the load by the section, the stress signal is obtained. The resulting strain signal is not a perfect sine wave either, but rather something mirroring the stress signal. Representive values of the amplitude and the phase of each signals are obtained by fitting a sine wave through the experimental data. Figure 3.3 and Figure 3.4 show the experimental data at 1 Hz and the sine wave fitted to each. It can be seen that the actual amplitude and phase would be difficult to read directly from the experimental data, since the data points do not form a perfect sine

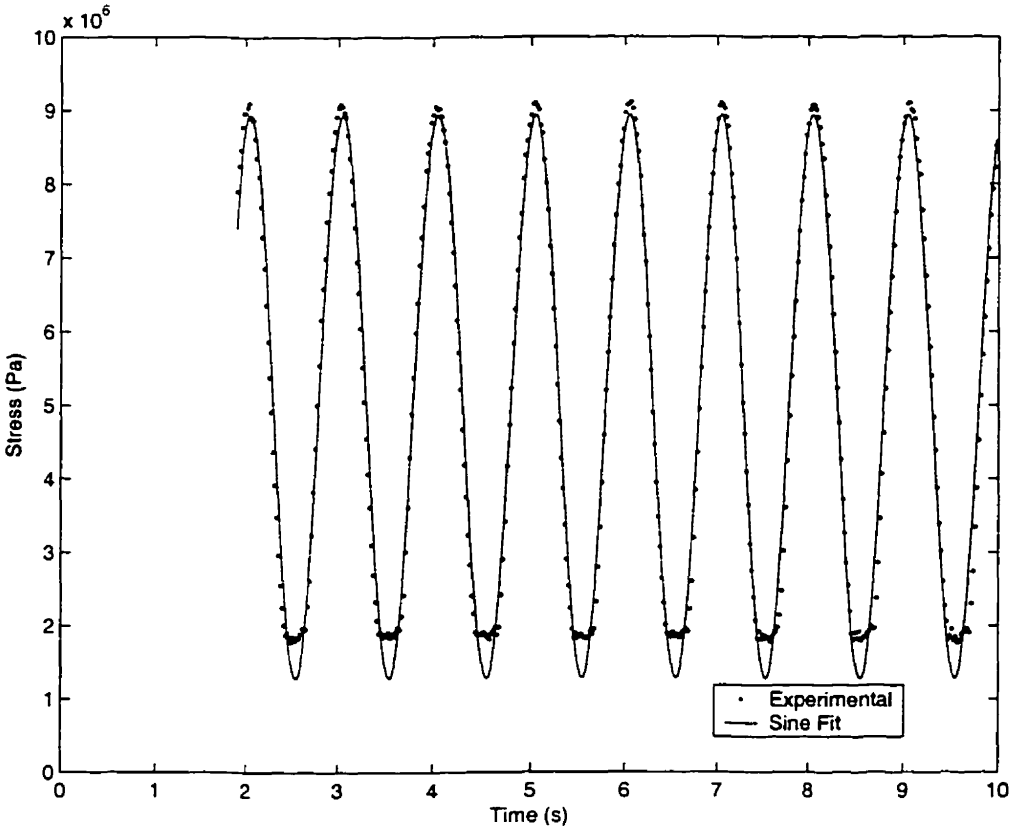

Figure 3.3: Stress Signal of PMMA at 1 Hz

wave. At low frequencies, the controller performs better. Figure 3.5 shows the stress signal at 0.1 Hz and its sine fit. The signal is closer to a sinusoidal curve than the stress data at 1 Hz. As the frequency increases, the load controller has difficulty maintaining a sinusoidal signal. At 50 Hz, as shown in Figure 3.6, the load signal is not very regular and no attempt was made to collect data at higher frequencies.

The general form of the stress wave is:

$$\begin{aligned} S_{xx} &= A\sin(2\pi ft + \phi) + B \\ &= A\sin(2\pi ft)\cos\phi + A\cos(2\pi ft)\sin\phi + B \end{aligned} \tag{3.4}$$

The data consists of the experimental vector of time, $\mathbf{t}$, and the corresponding stress values, $\mathbf{S_{xx}}$, for each frequency. Equation 3.4 is fitted to these data by

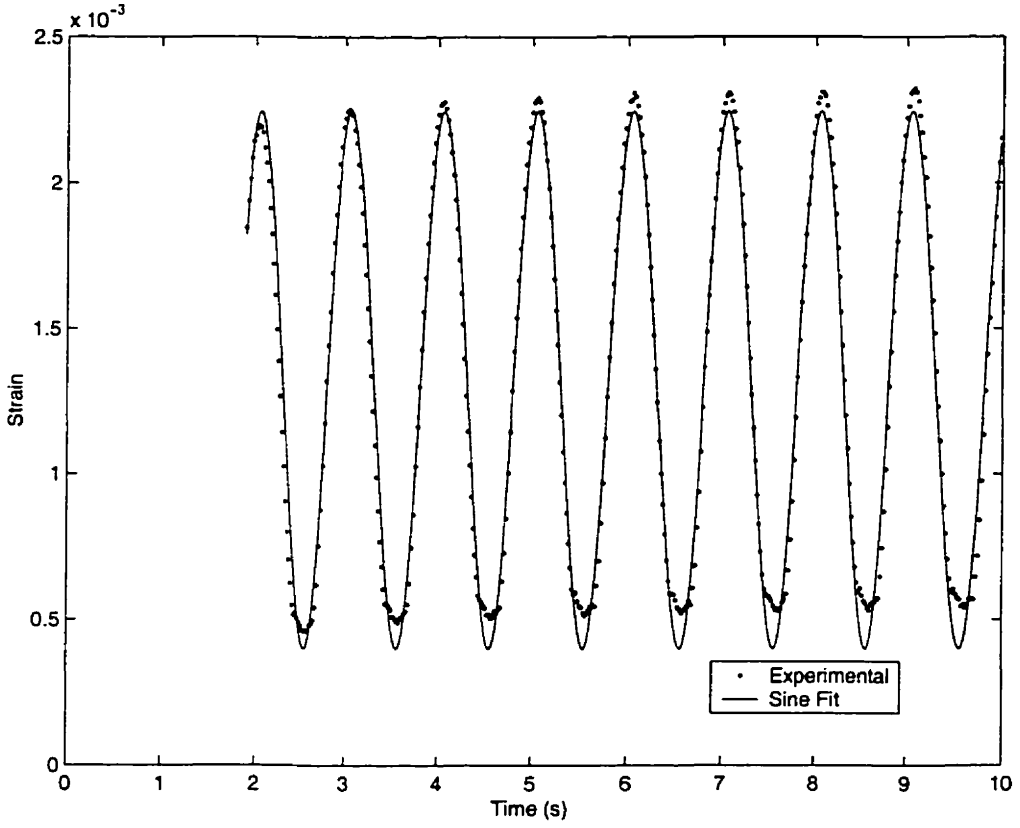

Figure 3.4: Strain Signal of PMMA at 1 Hz

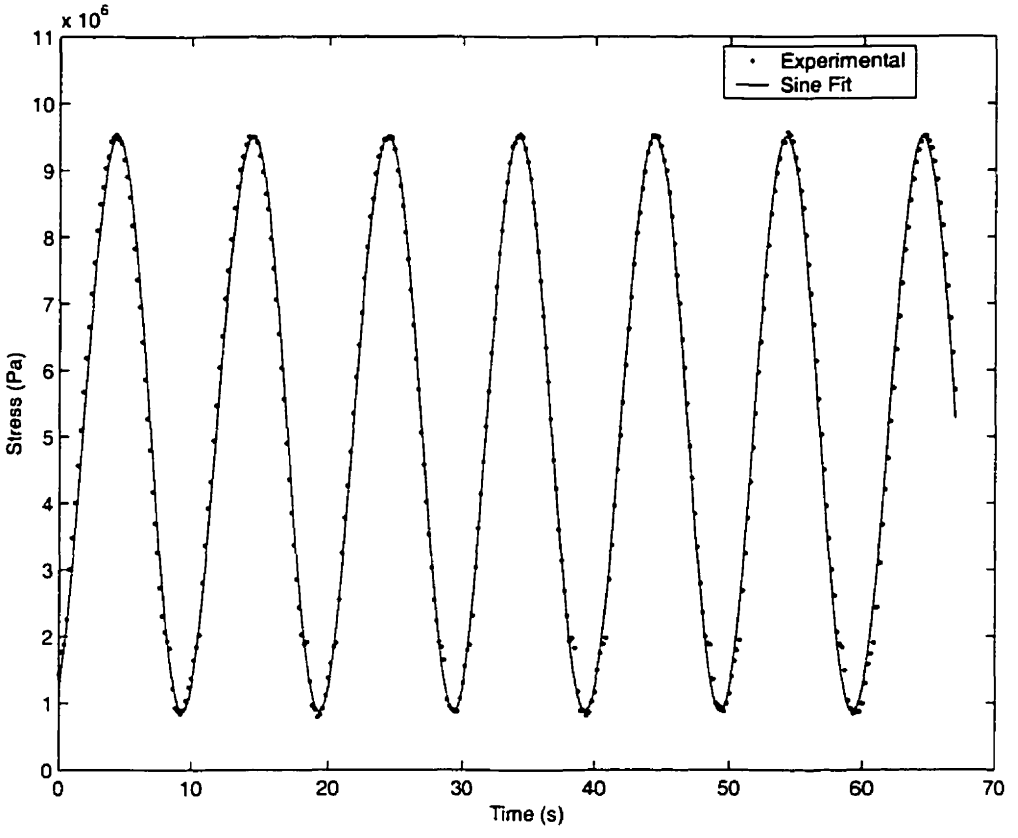

Figure 3.5: Stress Signal of PMMA at 0.1 Hz

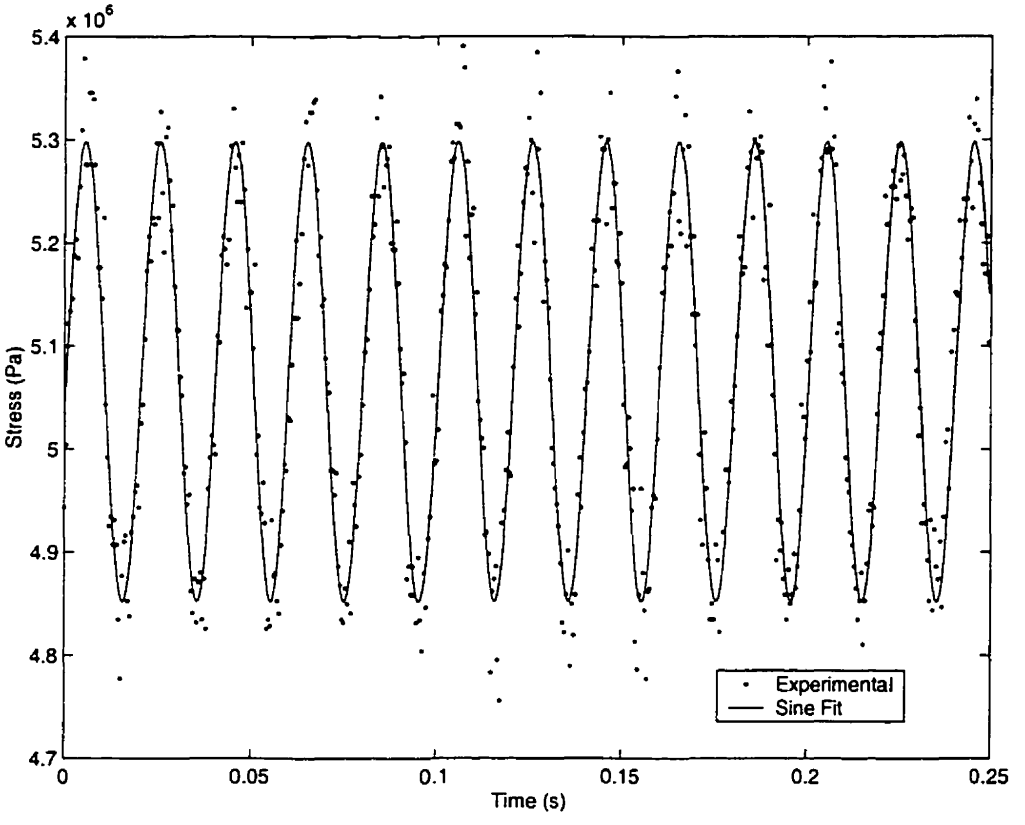

Figure 3.6: Stress Signal of PMMA at 50 Hz

Frequency (Hz)	Stress Signal		Strain Signal	
	Amplitude (Pa)	Phase (degrees)	Amplitude	Phase (degrees)
0.1	4.3209×10^6	-65.7	1.1×10^{-3}	-68.9
0.2	4.2675×10^6	-60.9	1.1×10^{-3}	-64.6
0.5	4.1058×10^6	-29.9	1.0×10^{-3}	-33.9
1	3.8341×10^6	72.6	9.2376×10^{-4}	68.7
2	3.3068×10^6	54.2	7.7334×10^{-4}	50.2
5	2.0249×10^6	-50.4	4.5422×10^{-4}	-54.5
10	1.1562×10^6	-26.1	2.4933×10^{-4}	-29.9
20	6.3220×10^5	-63.3	1.3048×10^{-4}	-66.5
30	3.8114×10^5	30.1	8.1095×10^{-5}	27.7
50	2.2348×10^5	-10.9	4.3909×10^{-5}	-13.6

Table 3.1: Amplitudes and Phases of the Stress and Strain Signals of PMMA

using:

$$\mathbf{S}_{xx} = \begin{bmatrix} \sin(2\pi f \mathbf{t}) & \cos(2\pi f \mathbf{t}) & \mathbf{1} \end{bmatrix} \begin{bmatrix} A\cos\phi \\ A\sin\phi \\ B \end{bmatrix} \tag{3.5}$$

Using matrix division in *Matlab*, the best coefficients in a least square sense, A, B, and ϕ, are found. A similar fit is done using the strain signals. Table 3.1 gives the amplitude and phase found for each signal. As the frequency increases, the amplitude decreases, indicating the difficulty the controller has in producing a sinusoidal load.

Young's Modulus

By dividing the amplitude of the stress signal by the amplitude of the strain signal, the magnitude of Young's modulus is found. By subtracting the phase of the strain signal from the phase of the stress signal, the phase of Young's modulus is obtained. Table 3.2 gives the magnitude and the phase of Young's modulus for each frequency tested. In the next section, the different models developed in the preceding chapter will be fitted to these experimental data.

3.2.3 Parameters of the Models

Table 2.1 gave six models used in this study to be compared. Parameters for each of these models are found for PMMA. The data collected for PMMA

Frequency (Hz)	Magnitude (Pa)	Phase (degrees)
0.1	3.76×10^9	3.2
0.2	3.87×10^9	3.8
0.5	4.01×10^9	4.0
1	4.15×10^9	4.0
2	4.28×10^9	4.0
5	4.46×10^9	4.0
10	4.64×10^9	3.8
20	4.85×10^9	3.2
30	4.70×10^9	2.4
50	5.09×10^9	2.7

Table 3.2: Magnitude and Phase of Young's Modulus for PMMA

yields Young's modulus, but the models are expressed in terms of the shear modulus. The assumption used for each model, either a constant Poisson's ratio or a constant bulk modulus, is specified in relating Young's modulus to the shear modulus.

The Voigt-Kelvin Model

The first model is a Voigt-Kelvin model with a constant Poisson's ratio. As stated by Equation 2.58, the complex shear modulus is obtained with:

$$G^* = \frac{E^*}{2(1+\nu_e)} \tag{3.6}$$

The fit is done using the values of the complex shear modulus for a value of Poisson's ratio of 0.4. The Voigt-Kelvin model is characterized by a magnitude increasing rapidly. A fit where the magnitude at 50 Hz, which is the last data point, is approximately the same as the experimental data is chosen. Such a fit is obtained with:

$$G^* = 1.23 \times 10^9 (1 + 0.003s)\ Pa \tag{3.7}$$

or

$$\begin{aligned} G_e &= 1.23 \times 10^9\ Pa \\ a &= 0.003 \end{aligned} \tag{3.8}$$

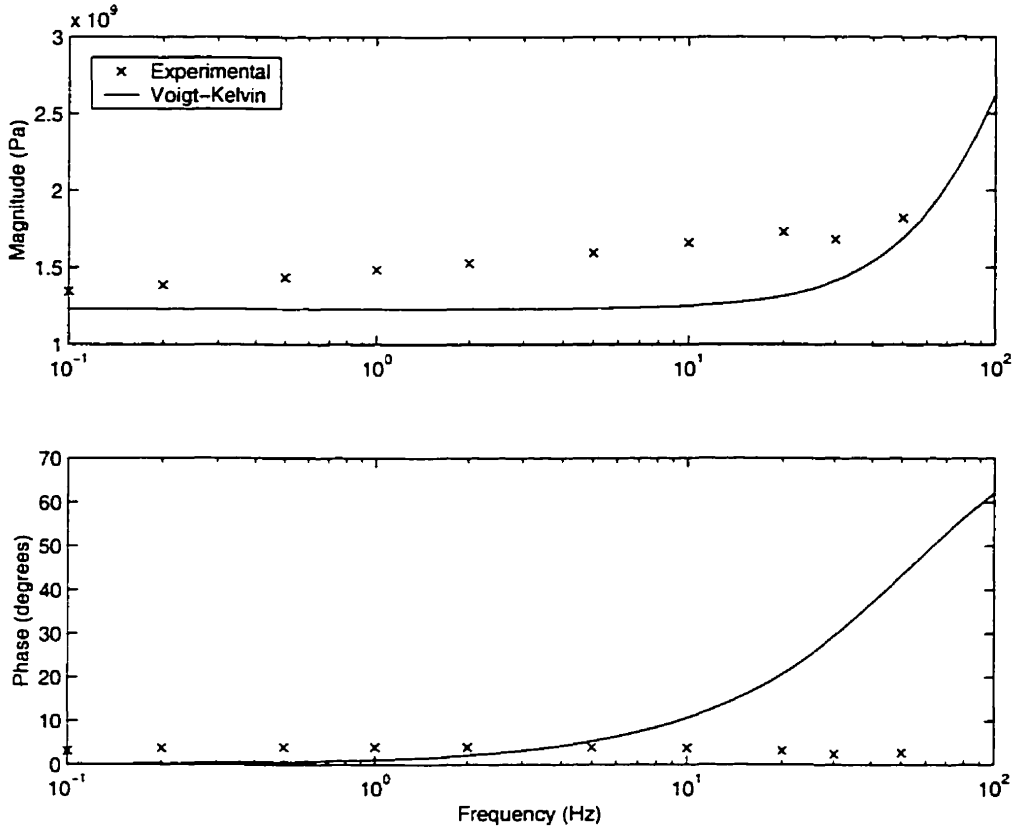

Figure 3.7: Voigt-Kelvin Model with a Constant Poisson's Ratio for the Shear Modulus of PMMA

If the experimental and modelled magnitude are equal for a lower frequency, then the model is totally unrepresentative at 50 Hz. Figure 3.7 gives the experimental data for the shear modulus calculated with a constant Poisson's ratio and the Voigt-Kelvin model.

The Maxwell Model

The Maxwell model also uses a constant Poisson's ratio assumption. Equation 3.6 is used to obtain the calculated experimental shear modulus data with a value of 0.4 for the equilibrium Poisson's ratio. The fit is done using the values of the complex shear modulus. The Maxwell model is characterized by a phase decreasing rapidly. The chosen fit has a modelled phase close to the experimental phase at a frequency of 1 Hz. This frequency is the lower bound of the frequency range of interest for slewing beams. This choice of fit

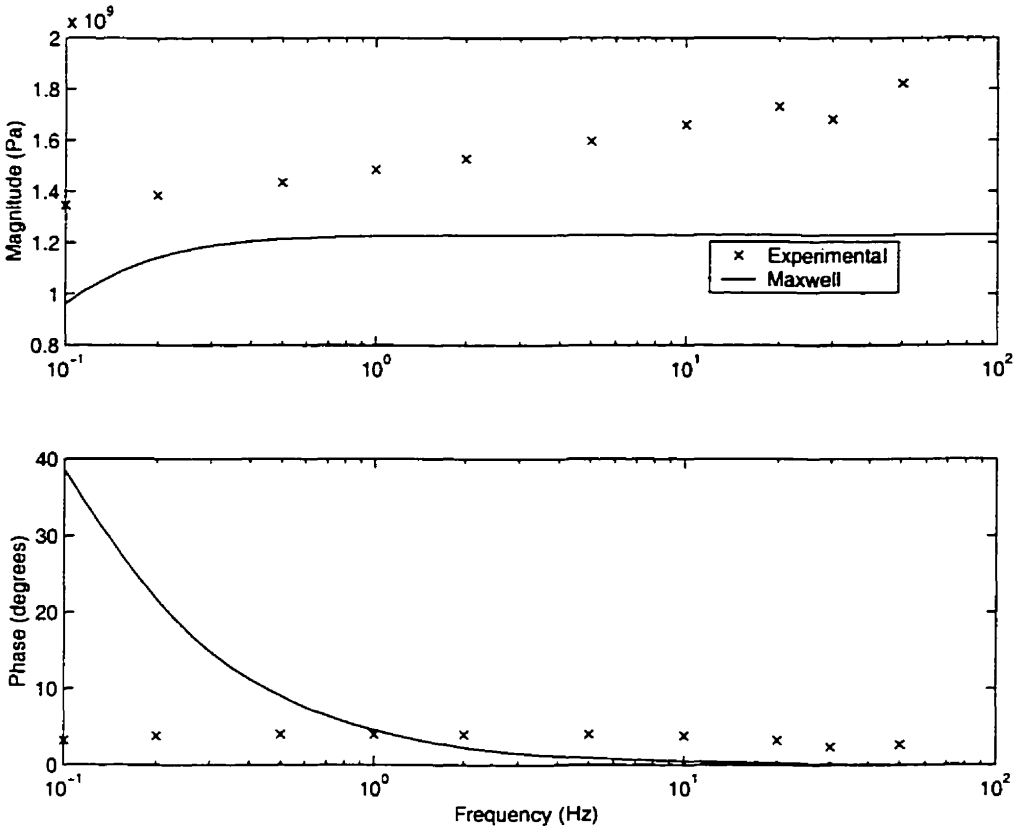

Figure 3.8: Maxwell Model with a Constant Poisson's Ratio for the Shear Modulus of PMMA

avoids a very high phase in the frequency range of interest which would not be representative of observed behaviour. This fit is obtained with:

$$G^* = \frac{2.0 \times 1.23 \times 10^9 s}{(1 + 2.0s)} \; Pa \tag{3.9}$$

or

$$\begin{aligned} G_e &= 1.23 \times 10^9 \; Pa \\ a &= 2.0 \end{aligned} \tag{3.10}$$

Figure 3.8 gives the experimental data for the shear modulus calculated with a constant Poisson's ratio and the results using the Maxwell model.

The Zener Model

The Zener model also uses a constant Poisson's ratio assumption, and again Equation 3.6 is used to obtain the calculated experimental shear modulus data

with a value of 0.4 for the equilibrium Poisson's ratio. The fit is done on the values of the complex shear modulus. As in the case of the Maxwell model, this model is characterized by a rapidly decreasing phase. A good compromise is reached by having the modelled phase close to the experimental phase at 2 Hz, while not being too high at 1 Hz. This fit is obtained with:

$$G^* = \frac{0.05 \times 1.23 \times 10^9 + 1.0 \times 1.23 \times 10^9 s}{((1 + 0.05) + 1.0s)} \; Pa \tag{3.11}$$

or

$$\begin{aligned} G_e &= 1.23 \times 10^9 \; Pa \\ \gamma &= 0.05 \\ \alpha &= 1.0 \end{aligned} \tag{3.12}$$

Figure 3.9 gives the experimental data for the shear modulus calculated with a constant Poisson's ratio and the Zener model.

The Prony Series with a Constant Poisson's Ratio Assumption

If Poisson's ratio is not constant, the parameters of the Prony series are needed for both the shear modulus and the bulk modulus. With a constant Poisson's ratio, the shear modulus and the bulk modulus exhibit the same variation with frequency, resulting in the same τ_n and proportional G_g and K_g. The experimental values for the shear modulus are obtained using Equation 3.6 and a value of 0.4 for the equilibrium Poisson's ratio. The best fit is found with an algorithm devised by Rogers [42], and implemented in *Matlab* by Slanik [48]. Table 3.3 gives the parameters of the Prony series up to three terms for a constant Poisson's ratio. The data used to build the Prony series is limited to the interval $[0.1 \; Hz, 20 \; Hz]$. The algorithm centers the phase bell shaped curve in the middle of the data points. Therefore, an even number of data points on each side of the phase peak value are kept.

The data points between 0.1 Hz and 20 Hz are neither low enough in frequency to point towards the value of the equilibrium shear modulus, or

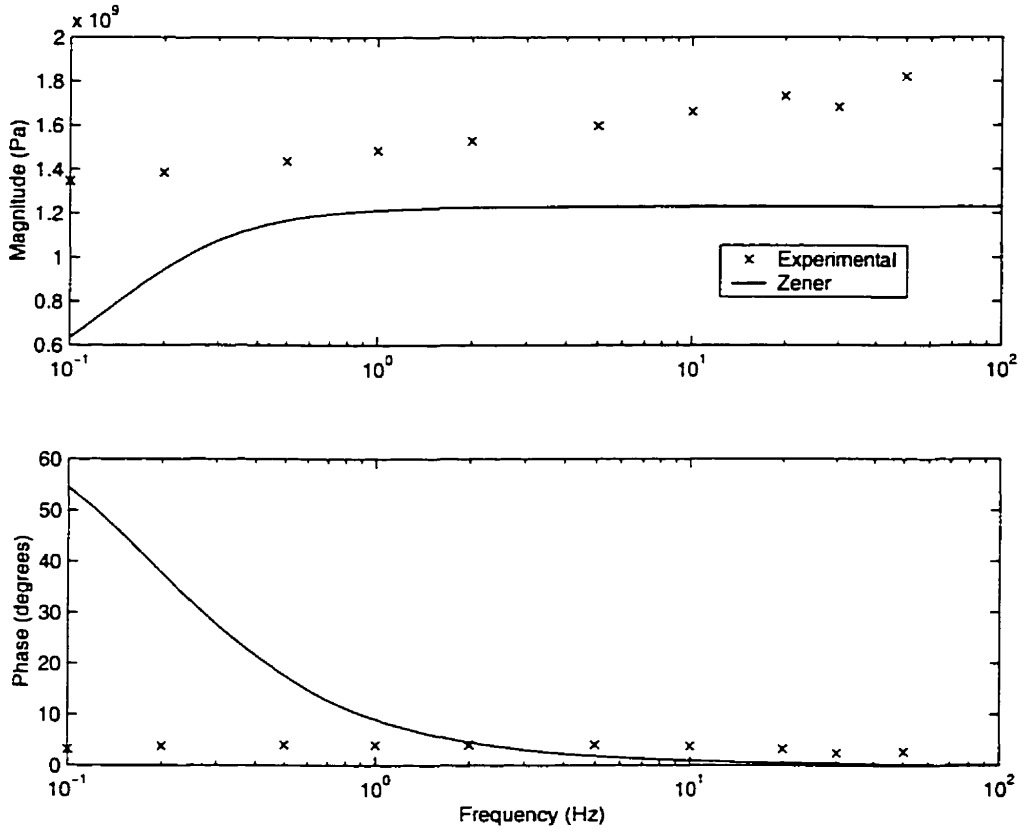

Figure 3.9: Zener Model with a Constant Poisson's Ratio for the Shear Modulus of PMMA

Number of Terms	G_g (Pa)	g_n (Pa)	τ_n (s)
1	1.7300×10^9	2.2474×10^{-1}	9.6851×10^{-2}
2	1.7192×10^9	9.2382×10^{-2}	4.0067×10^{-1}
		1.3236×10^{-1}	2.7647×10^{-2}
3	1.7126×10^9	5.9271×10^{-2}	6.3937×10^{-1}
		7.0638×10^{-2}	1.0906×10^{-1}
		9.4834×10^{-2}	1.8398×10^{-2}

Table 3.3: Parameters of the Prony Series with a Constant Poisson's Ratio for the Shear Modulus of PMMA

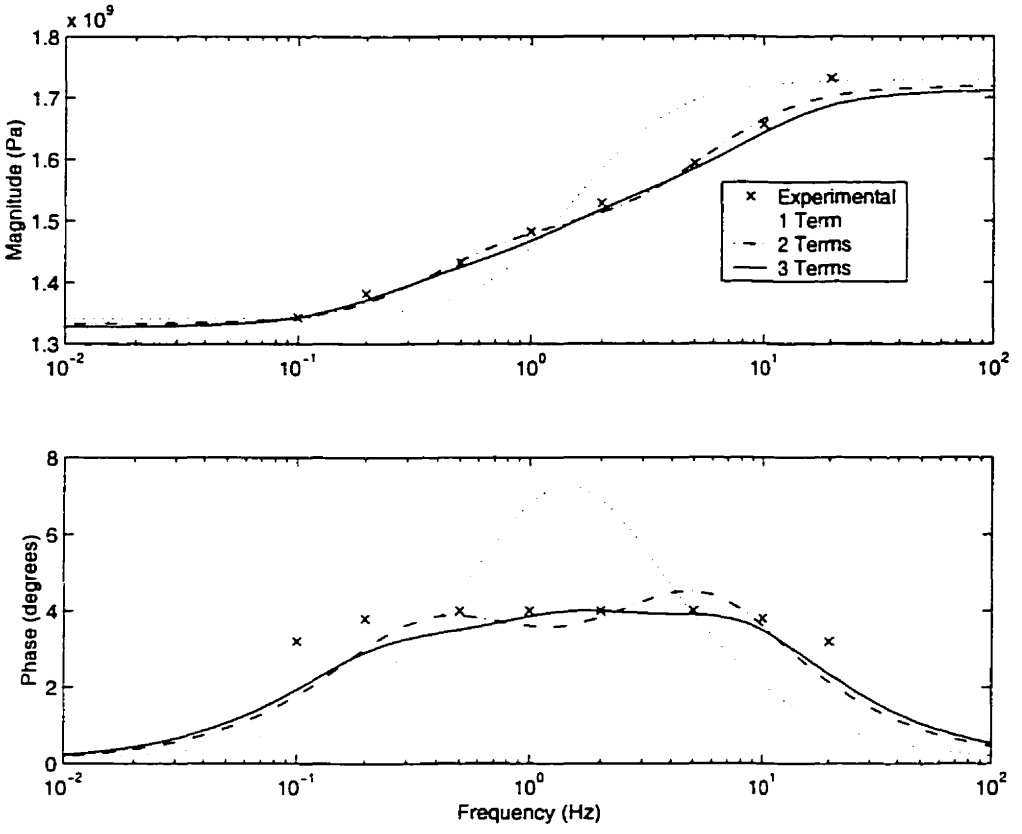

Figure 3.10: Prony Series with a Constant Poisson's Ratio for the Shear Modulus of PMMA

high enough in frequency to point towards the value of the instantaneous or glassy modulus. The Prony series found give the best fit for the data in the range $[0.1\ Hz, 20\ Hz]$ and not outside this range.

Figure 3.10 gives the three Prony series with the experimental data. With three parameters, a reasonable agreement with the experimental magnitude is found. The phase is also relatively smooth, but the model can not reproduce closely the slowly varying bell shaped curve. Adding terms does not improve the fit, since the very low and slowly varying phase exhibited by this material is difficult to approximate with the Prony series. In Chapter 2, a good fit is obtained with a Prony series of five terms for the material shown, but that material has a phase rising much higher and more smoothly than PMMA.

Number of Terms	G_g (Pa)	g_n (Pa)	τ_n (s)
1	1.7821×10^9	2.4242×10^{-1}	9.5405×10^{-2}
2	1.7700×10^9	9.8286×10^{-2}	3.9863×10^{-1}
		1.4413×10^{-1}	2.7417×10^{-2}
3	1.7627×10^9	6.2743×10^{-2}	6.3772×10^{-1}
		7.5870×10^{-2}	1.0871×10^{-1}
		1.0380×10^{-1}	1.8307×10^{-2}

Table 3.4: Parameters of the Prony Series with a Constant Bulk Modulus for the Shear Modulus of PMMA

The Prony Series with a Constant Bulk Modulus Assumption

The experimental shear modulus data is calculated from the experimental Young's modulus data using the following equation in the case of a constant bulk modulus [41]:

$$G^* = \frac{3K_e E^*}{9K_e - E^*} \tag{3.13}$$

with a value of K_e equal to $5.72 \times 10^9\ Pa$ as found in Equation 3.3. Again, the best fit is found with the algorithm devised by Rogers [42]. Table 3.4 gives the parameters of the Prony series up to three terms for a constant bulk modulus. The parameters in Table 3.4 are close to the ones found in Table 3.3. The two assumptions give similar values for the shear modulus. Figure 3.11 gives the three Prony series with the experimental data.

The corresponding bulk modulus Prony series is taken as a constant bulk modulus simply written as:

$$K^* = K_e = 5.72 \times 10^9\ Pa \tag{3.14}$$

The Fractional Voigt-Kelvin Model

The fractional Voigt-Kelvin model does not exhibit a bell shape for the phase, but rather a slowly increasing phase. A fit where the modelled phase is close to the experimental phase at 10 Hz is chosen. This frequency is the upper limit of the first mode natural frequency for slewing motions of PMMA beams of the dimensions used. A constant bulk modulus is assumed and Equation 3.13

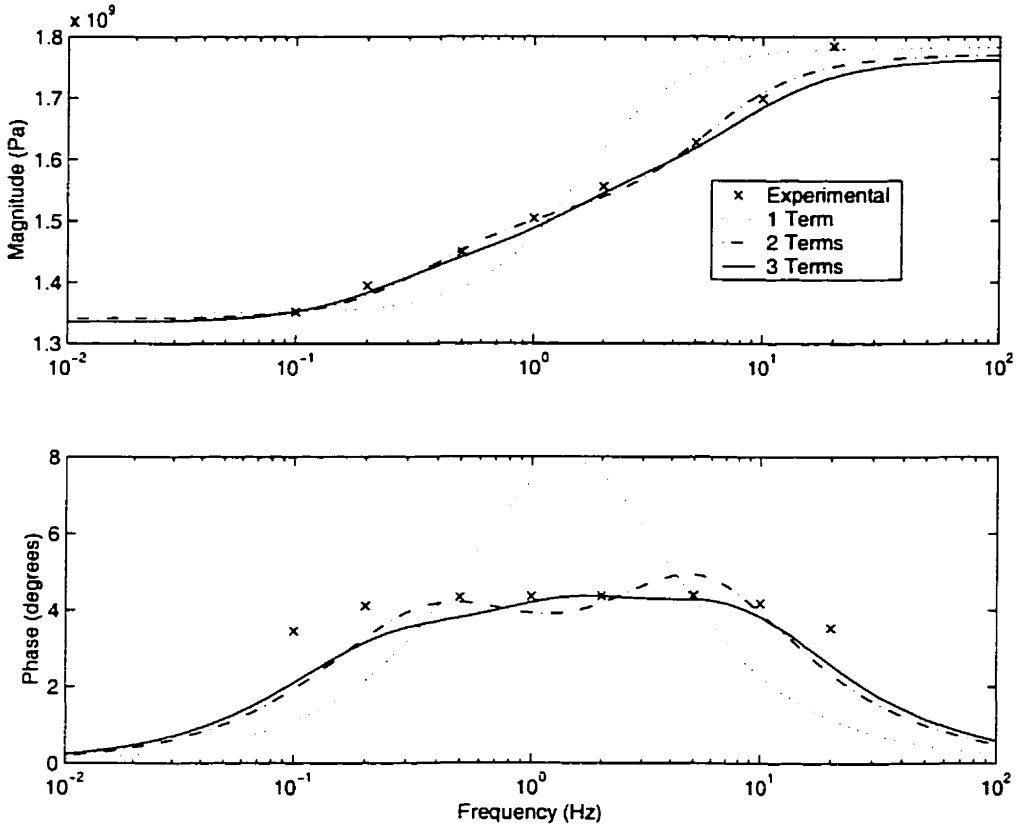

Figure 3.11: Prony Series with a Constant Bulk Modulus for the Shear Modulus of PMMA

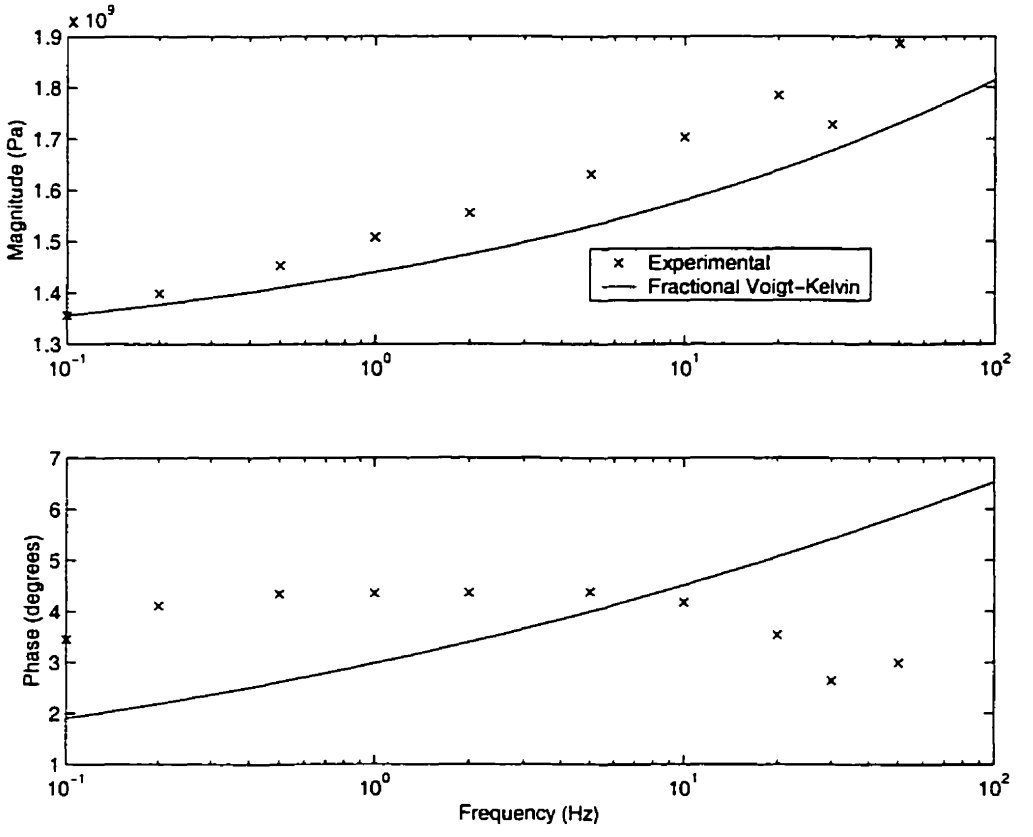

Figure 3.12: Fractional Voigt-Kelvin Model for the Shear Modulus of PMMA

is used to obtain the shear modulus data from the Young's modulus data. An equilibrium bulk modulus equal to $5.72 \times 10^9\ Pa$ is also used. The complex shear modulus with the fractional Voigt-Kelvin model is:

$$G^* = 1.23 \times 10^9 (1 + 0.12 s^{0.22})\ Pa \tag{3.15}$$

or

$$\begin{aligned} G_e &= 1.23 \times 10^9\ Pa \\ a &= 0.12 \\ \xi &= 0.22 \end{aligned} \tag{3.16}$$

The fit of the model to the values of the shear modulus determined experimentally is shown in Figure 3.12.

Figure 3.13 gives the fractional Voigt-Kelvin model for a constant bulk

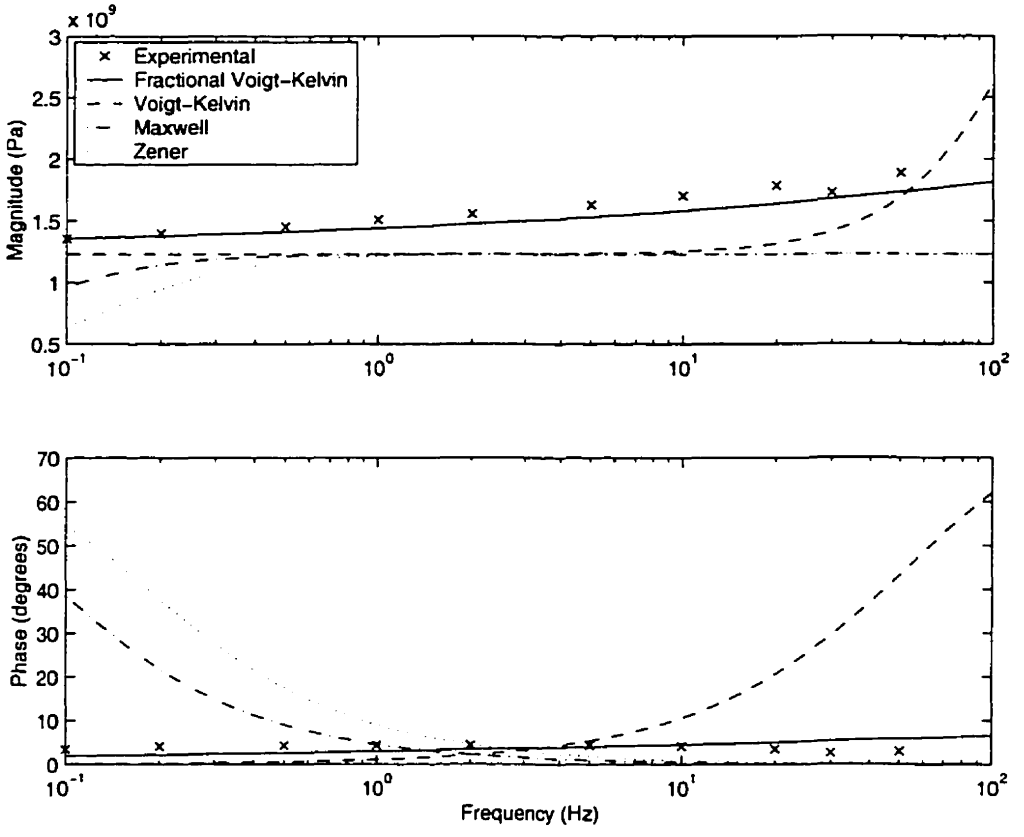

Figure 3.13: Fractional Voigt-Kelvin, Voigt-Kelvin, Maxwell, and Zener Models for the Shear Modulus of PMMA

modulus assumption compared to the Voigt-Kelvin, Maxwell, and Zener models for a constant Poisson's ratio assumption. Despite not being as close to the experimental data as the Prony series, the fractional Voigt-Kelvin is obviously the second best one after the Prony series for its fit.

3.3 Behaviour of the Viscoelastic Layer 3M ISD 112

A viscoelastic layer is a thin polymeric sheet used to damp vibrations. As any polymer, its elastic constants vary with frequency. Users select the viscoelastic layer according to the magnitude and phase of its shear modulus at the frequency they wish to damp. Manufacturers establish charts of the modulus as a function of frequency usually using a DMTA (Digital Mechanical-Thermal

Frequency (Hz)	Storage Modulus (Pa)	Loss Factor
0.1	7.00×10^4	0.4
0.5	1.00×10^5	0.6
1	1.40×10^5	0.7
2	1.70×10^5	0.8
3	2.00×10^5	0.85
4	2.10×10^5	0.9
5	2.40×10^5	0.9
10	3.40×10^5	1.0
20	5.00×10^5	1.0
50	7.50×10^5	1.0
70	9.00×10^5	1.0
100	1.00×10^6	1.0
200	1.60×10^6	1.0
500	2.50×10^6	0.9
700	3.00×10^6	0.9
1000	3.50×10^6	0.85
5000	7.00×10^6	0.6
10000	9.00×10^6	0.5

Table 3.5: Shear Storage Modulus and Loss Factor for 3M ISD 112

Analyzer). Viscoelastic layer 3M ISD 112 is used in this study because it is mostly efficient at room temperature and it has good potential for the aerospace industry.

3.3.1 Frequency Dependent Behaviour

The shear modulus is given as the storage modulus, G', being in fact the real part of the complex modulus, and the loss factor, η, representing the imaginary part. The storage modulus and loss factor constitute the complex modulus when written as:

$$G^* = G'(1 + i\eta) \tag{3.17}$$

Table 3.5 gives the data read from the manufacturer charts at $20°C$.

Figure 3.14 shows the data from the manufacturer at $20°C$ in the form of the magnitude and the phase of the shear modulus. The same data is shown in Table 3.6. The equilibrium modulus is not given by the manufacturer, but

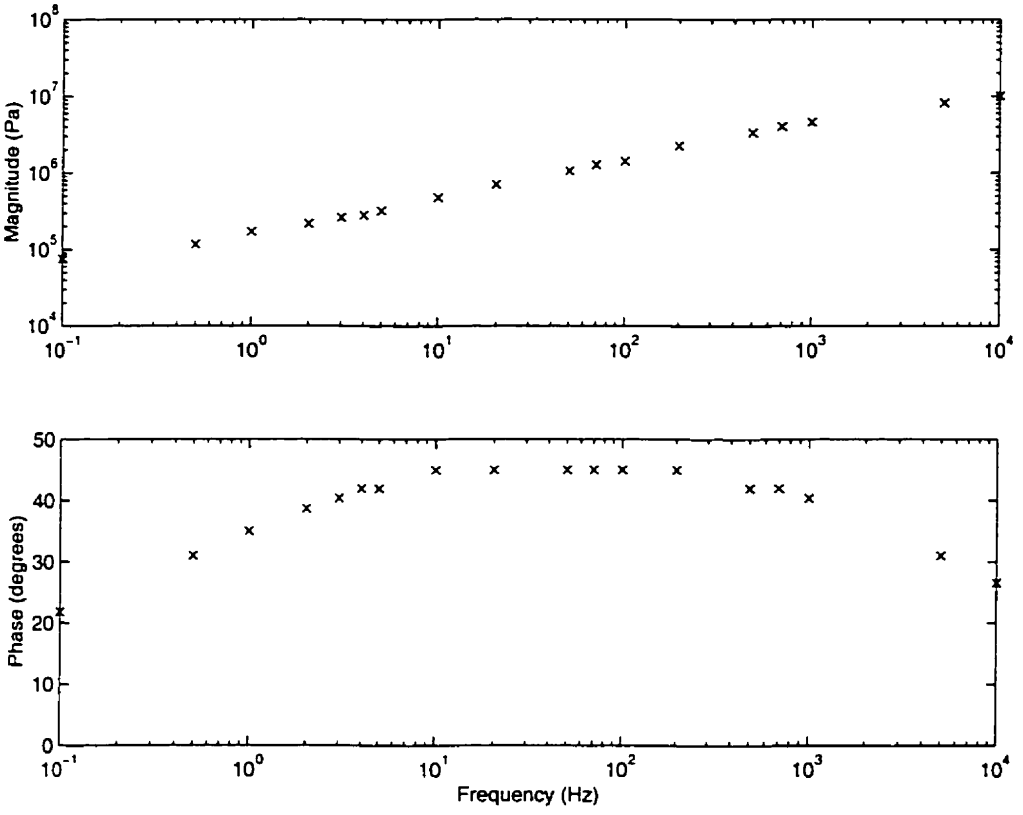

Figure 3.14: Shear Modulus of 3M ISD 112

Frequency (Hz)	Magnitude (Pa)	Phase (degrees)
0.1	7.54×10^4	21.8
0.5	1.17×10^5	31.0
1	1.71×10^5	35.0
2	2.18×10^5	38.7
3	2.62×10^5	40.4
4	2.83×10^5	42.0
5	3.23×10^5	42.0
10	4.81×10^5	45.0
20	7.07×10^5	45.0
50	1.06×10^6	45.0
70	1.27×10^6	45.0
100	1.41×10^6	45.0
200	2.26×10^6	45.0
500	3.36×10^6	42.0
700	4.04×10^6	42.0
1000	4.59×10^6	40.4
5000	8.16×10^6	31.0
10000	1.01×10^7	26.6

Table 3.6: Magnitude and Phase of Shear Modulus for 3M ISD 112

Table 3.6 gives a value of $7.54 \times 10^4\ Pa$ at $0.1\ Hz$. The equilibrium value is arbitrarily set at $7.00 \times 10^4\ Pa$, a value slightly lower than the value at $0.1\ Hz$.

3.3.2 Parameters of the Models

Parameters for the six studied models are also found for 3M ISD 112. In the case of PMMA, the experimental data was obtained for Young's modulus. The assumption used, either a constant Poisson's ratio or a constant bulk modulus, was important in obtaining the corresponding shear modulus data. In the case of 3M ISD 112, the manufacturer gives directly the shear modulus.

The Voigt-Kelvin Model

The first model is the Voigt-Kelvin model. The phase of this model increases rapidly from 0° to 90°. Since the experimental phase of 3M ISD 112 peaks at 45°, a fit where the transition from 0° to 90° occurs approximately in the middle of the bell shape is chosen:

$$G^* = 7.00 \times 10^4 (1 + 0.005s)\ Pa \tag{3.18}$$

or

$$\begin{aligned} G_e &= 7.00 \times 10^4\ Pa \\ a &= 0.005 \end{aligned} \tag{3.19}$$

Figure 3.15 gives the manufacturer data for the shear modulus and the Voigt-Kelvin model.

The Maxwell Model

The Maxwell model is characterized by a phase decreasing rapidly from 90° to 0°. A fit is chosen for which the phase is close to the experimental phase between 3 Hz and 4 Hz, the natural frequency of the experimental structure which will be tested. This fit is obtained with:

$$G^* = \frac{0.05 \times 7.00 \times 10^4 s}{(1 + 0.05s)}\ Pa \tag{3.20}$$

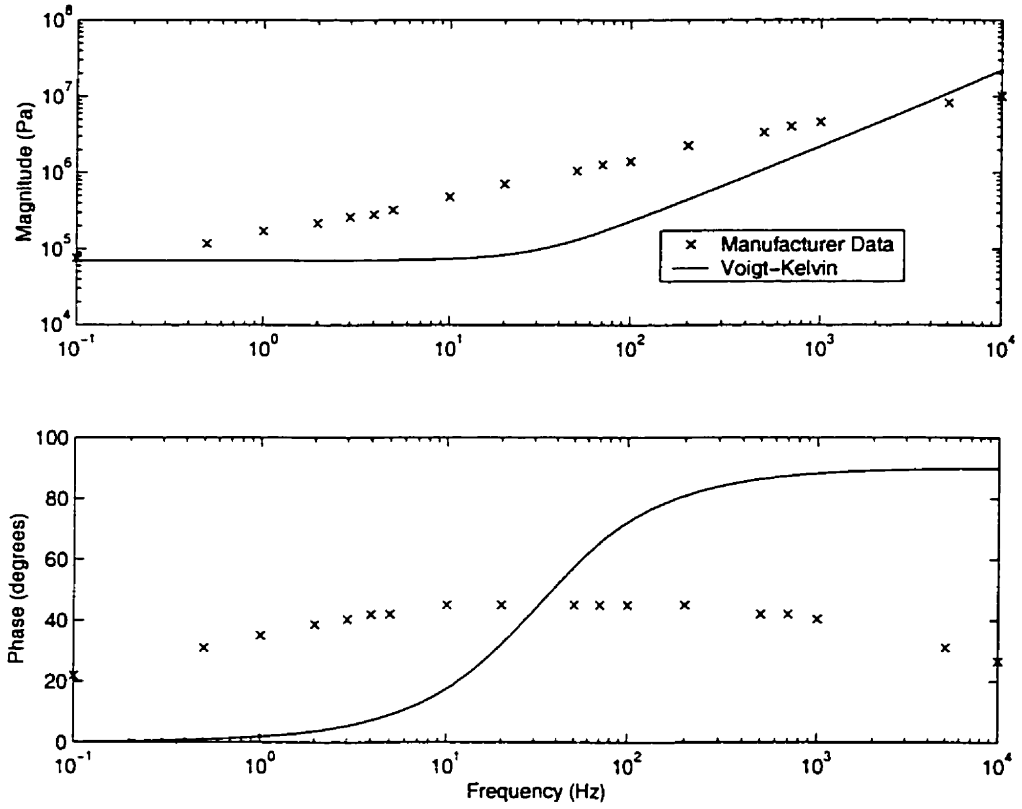

Figure 3.15: Voigt-Kelvin Model for the Shear Modulus of 3M ISD 112

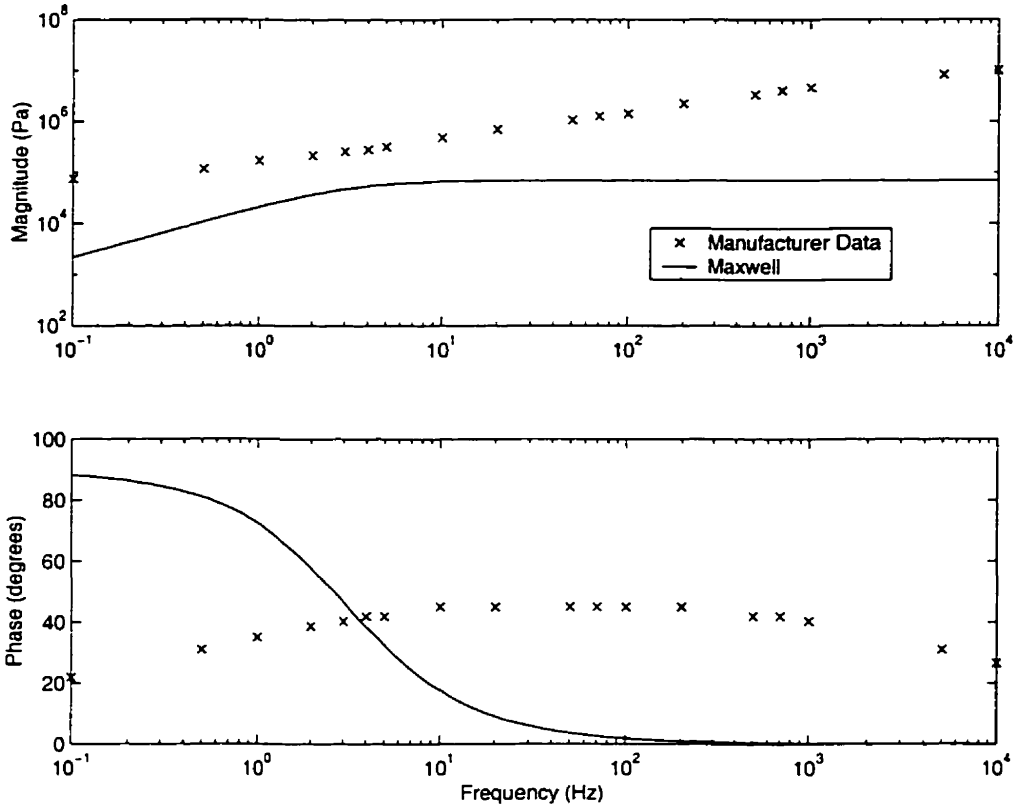

Figure 3.16: Maxwell Model for the Shear Modulus of 3M ISD 112

or

$$\begin{aligned} G_e &= 7.00 \times 10^4\ Pa \\ a &= 0.05 \end{aligned} \tag{3.21}$$

Figure 3.16 gives the manufacturer data and the Maxwell model.

The Zener Model

A reasonable fit for the Zener model for which the phase peaks in the area where the manufacturer data peaks is obtained with:

$$G^* = \frac{0.05 \times 7.00 \times 10^4 + 0.001 \times 7.00 \times 10^4 s}{((1 + 0.05) + 0.001s)}\ Pa \tag{3.22}$$

or

$$\begin{aligned} G_e &= 7.00 \times 10^4\ Pa \\ \gamma &= 0.05 \\ \alpha &= 0.001 \end{aligned} \tag{3.23}$$

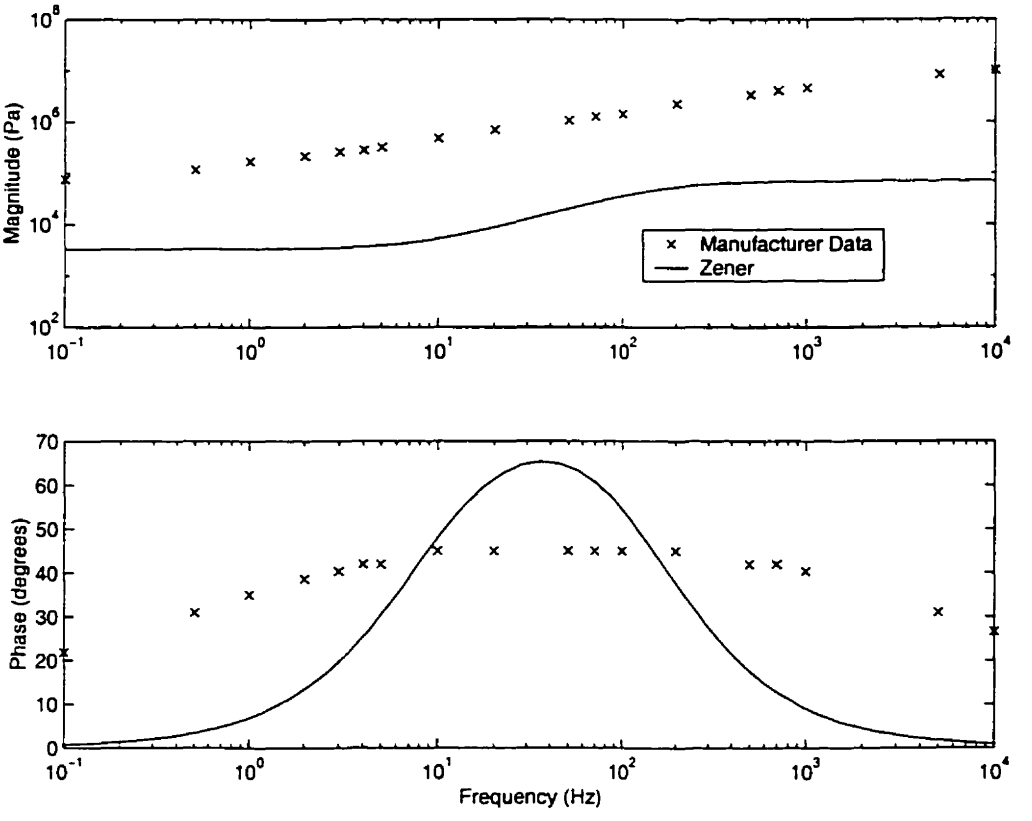

Figure 3.17: Zener Model for the Shear Modulus of 3M ISD 112

Figure 3.17 shows the manufacturer data and the Zener model.

The Prony Series

The Prony series for the shear modulus will be the same with both assumptions, a constant Poisson's ratio and a constant bulk modulus, since the shear data is taken directly. The chosen assumption will affect the bulk modulus, but not the shear modulus. The best fit is also found using the algorithm devised by Rogers [42]. Table 3.7 gives the parameters G_g, g_n, and τ_n of the Prony series up to seven terms for the shear modulus. These parameters are the ones input in the finite element package. The value of G_e is equal to $6.9042 \times 10^4\ Pa$ for a seven term Prony series, which is in agreement with the choice of a G_e of $7.00 \times 10^4\ Pa$ made in Equations 3.18 to 3.22.

Figure 3.18 gives the four first Prony series with the manufacturer data. Figure 3.19 gives the three last Prony series with the manufacturer data. With

Number of Terms	G_g (Pa)	g_n (Pa)	τ_n (s)
1	1.0056×10^7	9.9251×10^{-1}	4.3342×10^{-4}
2	9.9433×10^6	7.4316×10^{-2}	2.1162×10^{-2}
		9.1819×10^{-1}	1.0535×10^{-4}
3	9.7334×10^6	2.3516×10^{-2}	9.1238×10^{-2}
		1.6072×10^{-1}	2.1237×10^{-3}
		8.0827×10^{-1}	6.1213×10^{-5}
4	9.5689×10^6	1.1170×10^{-2}	1.9318×10^{-1}
		5.1392×10^{-2}	1.0731×10^{-2}
		2.1573×10^{-1}	6.7327×10^{-4}
		7.1421×10^{-1}	4.5697×10^{-5}
5	9.4104×10^6	7.0184×10^{-3}	3.1445×10^{-1}
		2.3852×10^{-2}	2.8951×10^{-2}
		8.5601×10^{-2}	3.0332×10^{-3}
		2.5886×10^{-1}	3.5349×10^{-4}
		6.1717×10^{-1}	3.8288×10^{-5}
6	9.2988×10^6	5.0212×10^{-3}	4.2442×10^{-1}
		1.4201×10^{-2}	5.8737×10^{-2}
		4.0498×10^{-2}	9.0394×10^{-3}
		1.0242×10^{-1}	1.2610×10^{-3}
		2.8052×10^{-1}	2.2384×10^{-4}
		5.4985×10^{-1}	3.4307×10^{-5}
7	9.2147×10^6	3.9026×10^{-3}	5.1558×10^{-1}
		9.7895×10^{-3}	1.0322×10^{-1}
		2.0646×10^{-2}	1.7741×10^{-2}
		5.7318×10^{-2}	3.6691×10^{-3}
		1.3123×10^{-1}	7.2364×10^{-4}
		2.7891×10^{-1}	1.5945×10^{-4}
		4.9071×10^{-1}	3.1760×10^{-5}

Table 3.7: Parameters of the Prony Series for the Shear Modulus of 3M ISD 112

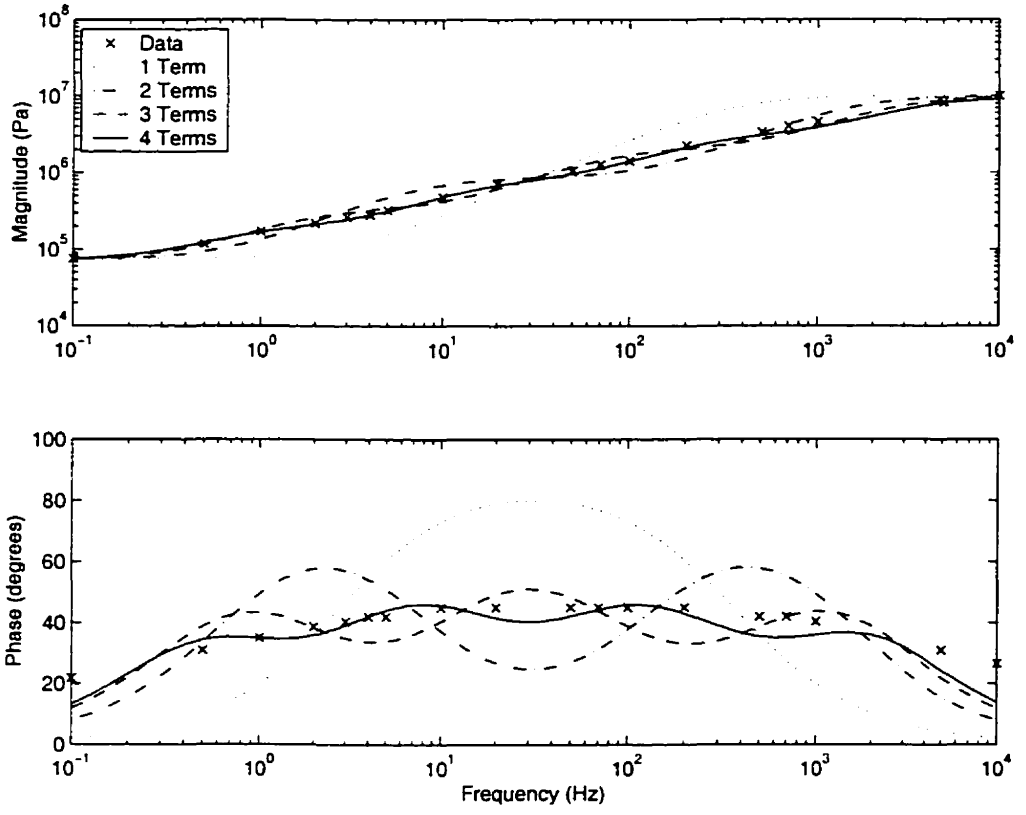

Figure 3.18: Prony Series up to Four Terms for the Shear Modulus of 3M ISD 112

five terms in the Prony series, the fit for the magnitude is excellent, but the phase is still not smooth. With seven terms in the series, the phase is smooth.

For a constant Poisson's ratio assumption, the corresponding bulk modulus Prony series are obtained using:

$$K^* = \frac{2G^*(1+\nu_e)}{3(1-2\nu_e)} \tag{3.24}$$

The manufacturer suggests an equilibrium Poisson's ratio in the range $[0.49, 0.5]$. A value of 0.499 is arbitrarily chosen.

If a constant bulk modulus is assumed, then the following equation is used:

$$K_e = \frac{2G_e(1+\nu_e)}{3(1-2\nu_e)} \tag{3.25}$$

A value of 7.00×10^4 Pa is assumed for G_e, and a value of 0.499 for ν_e. In

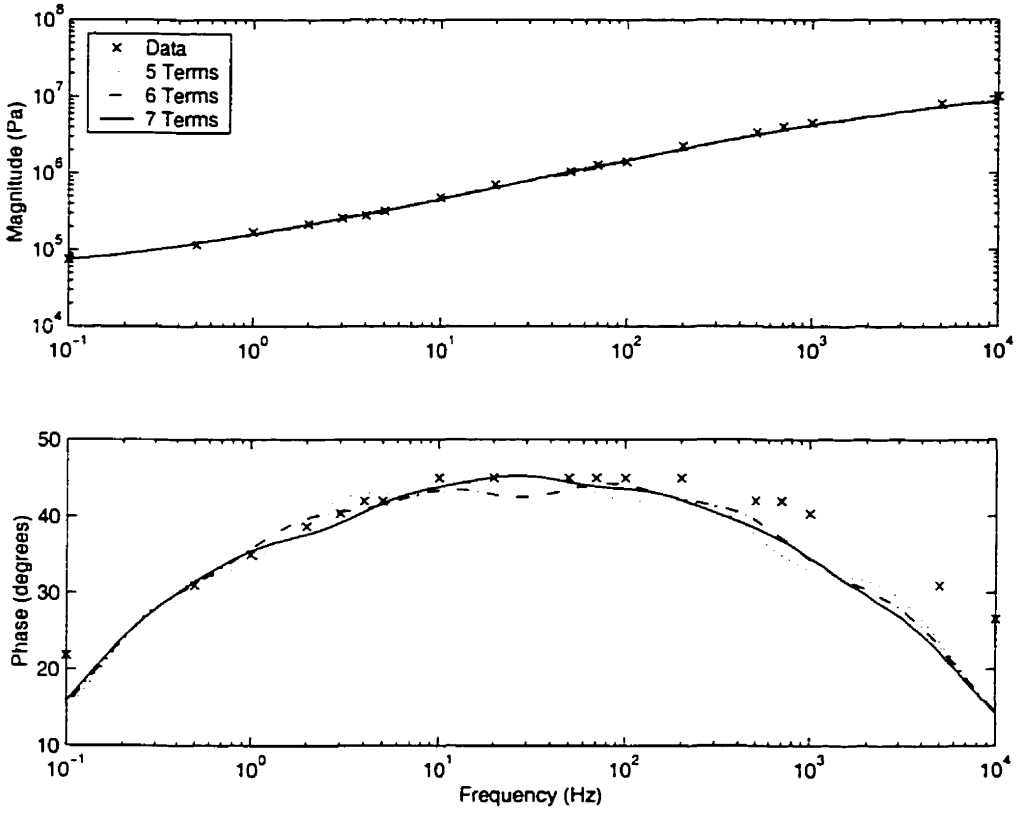

Figure 3.19: Prony Series from Five to Seven Terms for the Shear Modulus of 3M ISD 112

that case, the bulk modulus is equal to:

$$K^* = K_e = 3.50 \times 10^7 \ Pa \tag{3.26}$$

The Fractional Voigt-Kelvin Model

The fractional Voigt-Kelvin model has one derivative of the strain and no derivative of the stress. Therefore, the phase does not form a bell, but half a bell reaching an asymptotic value. A fit is chosen for the model to correctly represent the first half of the bell:

$$G^* = 7.00 \times 10^4 (1 + 0.8 s^{0.54}) \ Pa \tag{3.27}$$

or

$$\begin{aligned} G_e &= 7.00 \times 10^4 \ Pa \\ a &= 0.8 \\ \xi &= 0.54 \end{aligned} \tag{3.28}$$

The fit of the model to the manufacturer values is shown in Figure 3.20.

Figure 3.21 gives the fractional Voigt-Kelvin model for a constant bulk modulus assumption compared to the Voigt-Kelvin, Maxwell, and Zener models for a constant Poisson's ratio assumption. As in the case of PMMA, the fractional Voigt-Kelvin model is not as close to the experimental data as the Prony series, but it obviously gives a better fit than the other classical models.

3.4 Conclusion

The parameters of the chosen models were found for both PMMA and 3M ISD 112. Most models exhibit a rather poor fit. The Prony series manage to approximate the experimental data, but several parameters are needed. The fractional Voigt-Kelvin model does not give a perfect fit for PMMA, but the phase remains in the same range, although it does not have a bell shape. For 3M ISD 112, the fractional Voigt-Kelvin model gives a good fit of the

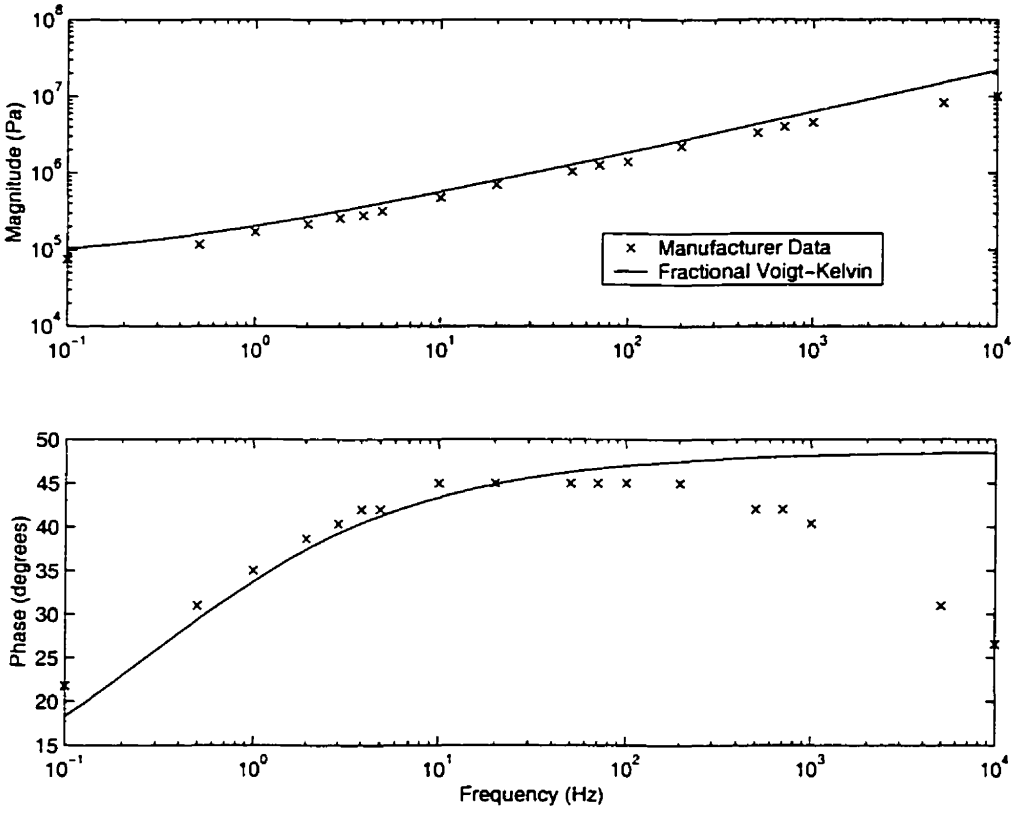

Figure 3.20: Fractional Voigt-Kelvin Model for the Shear Modulus of 3M ISD 112

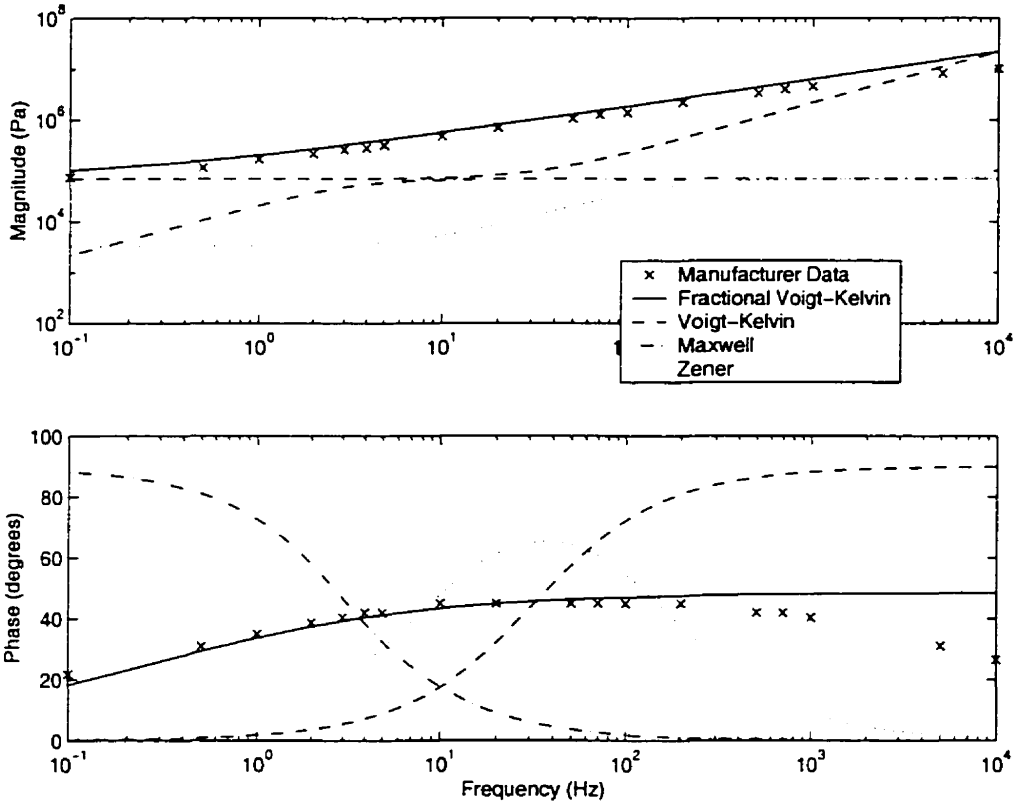

Figure 3.21: Fractional Voigt-Kelvin, Voigt-Kelvin, Maxwell, and Zener Models for the Shear Modulus of 3M ISD 112

magnitude and also a good fit of the phase up to 100 Hz. The Prony series and the fractional Voigt-Kelvin model are the two most promising models for both materials. In the case of PMMA, the Prony series with three terms gives a superior fit than the fractional Voigt-Kelvin model. In the case of 3M ISD 112, seven terms are needed for a good fit with the Prony series. The fractional Voigt-Kelvin gives a fit just as good up to 100 Hz with fewer parameters. In the following chapters, the various models for the two materials studied will be used to attempt reproducing experimental behaviour of several structures. The objective will be to obtain accurate simulations with the material parameters identified a priori, rather than adjusting the material parameters to obtain an accurate simulation.

Chapter 4

Numerical Modelling of the Fractional Constitutive Equation

4.1 Introduction

This chapter develops an algorithm, which is an original contribution of this thesis, based on a simple trapezoidal rule to solve an equation with a fractional derivative. A one-dimensional constitutive equation containing a fractional derivative is written. A strain input resulting in a linear constitutive equation is solved through the frequency domain. The results obtained from the frequency domain are compared to the results obtained from the time domain algorithm described in this chapter. The fractional derivative requires storing all data from the beginning of the computation. A solution to this problem is offered and its limits are explored. Solving such a simple one-dimensional constitutive equation gives insight into problems that could arise when solving the three-dimensional constitutive equation.

4.2 The Fractional Derivative

A differential equation with a fractional derivative has a term defined as [23]:

$$D^{\xi}\varepsilon_{ij}(t) = \frac{1}{\Gamma(1-\xi)}\frac{d}{dt}\int_0^t \frac{\varepsilon_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau \tag{4.1}$$

An alternate form of Equation 4.1 is:

$$D^{\xi}\varepsilon_{ij}(t) = \frac{t^{-\xi}\varepsilon_{ij}(0)}{\Gamma(1-\xi)} + \frac{1}{\Gamma(1-\xi)}\int_0^t \frac{\dot{\varepsilon}_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau \tag{4.2}$$

For zero initial conditions, $\varepsilon_{ij}(0)$ is equal to 0 and Equation 4.2 reduces to:

$$D^{\xi}\varepsilon_{ij}(t) = \frac{1}{\Gamma(1-\xi)}\int_0^t \frac{\dot{\varepsilon}_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau \tag{4.3}$$

A singularity arises as the denominator of the integrand becomes 0 when $t = \tau$, τ being a dummy variable for time. Moreover, the integral is hereditary since it starts at 0.

To remove the singularity, the integration is done from 0 to $(t-\epsilon)$ instead of 0 to t, where ϵ is a small amount of time compared to t, and an approximation is used for the integration from $(t-\epsilon)$ to t. The integral term of the fractional derivative is divided into two parts:

$$D^{\xi}\varepsilon_{ij}(t) = \frac{1}{\Gamma(1-\xi)}\left[\int_0^{t-\epsilon} \frac{\dot{\varepsilon}_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau + \int_{t-\epsilon}^t \frac{\dot{\varepsilon}_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau\right] \tag{4.4}$$

For the integral between $(t-\epsilon)$ and t, $\dot{\varepsilon}_{ij}(\tau)$ is approximated using the two first terms of Taylor's expansion around t, t being in this case a fixed value and τ, the variable:

$$\dot{\varepsilon}_{ij}(\tau) = \dot{\varepsilon}_{ij}(t) - (t-\tau)\ddot{\varepsilon}_{ij}(t) + \mathcal{O}(h^2) \tag{4.5}$$

The last term is the truncation error and refers to the error associated to the approximation of $\dot{\varepsilon}_{ij}(\tau)$. This second order error term due to the truncation is:

$$\frac{(t-\tau)^2}{2}\varepsilon_{ij}^{(3)}(\zeta_1) \tag{4.6}$$

with ζ_1 taking a value in the interval $[t-\epsilon, t]$.

Equation 4.5 is used in the last part of Equation 4.4 to obtain:

$$\int_{t-\epsilon}^t \frac{\dot{\varepsilon}_{ij}(\tau)}{(t-\tau)^{\xi}}d\tau = \int_{t-\epsilon}^t \frac{\dot{\varepsilon}_{ij}(t) - (t-\tau)\ddot{\varepsilon}_{ij}(t)}{(t-\tau)^{\xi}}d\tau \tag{4.7}$$

The terms $\dot{\varepsilon}_{ij}$ and $\ddot{\varepsilon}_{ij}$ on the right side of Equation 4.7 do not depend on τ; they are independent of the integration variable. The integral of Equation 4.7

is solved by applying a simple substitution:

$$u = t - \tau \Rightarrow du = -d\tau \tag{4.8}$$

The integral becomes:

$$\int_0^{\epsilon} \frac{\dot{\varepsilon}_{ij}(t) - u\ddot{\varepsilon}_{ij}(t)}{u^{\xi}} du = \frac{\dot{\varepsilon}_{ij}(t)\epsilon^{1-\xi}}{1-\xi} - \frac{\ddot{\varepsilon}_{ij}(t)\epsilon^{2-\xi}}{2-\xi} \tag{4.9}$$

Integration of the truncation error term expressed in Equation 4.6 is also done:

$$\frac{1}{2}\int_{t-\epsilon}^{t} \frac{(t-\tau)^2 \varepsilon_{ij}^{(3)}(\zeta_1)}{(t-\tau)^{\xi}} d\tau = \frac{\varepsilon_{ij}^{(3)}(\zeta_1)\epsilon^{3-\xi}}{2(3-\xi)} \tag{4.10}$$

where ζ_1 is in the interval $[t-\epsilon, t]$. Finally, Equation 4.3 is rewritten without any singularity:

$$D^{\xi}\varepsilon_{ij}(t) = \frac{1}{\Gamma(1-\xi)}\left(\frac{\dot{\varepsilon}_{ij}(t)\epsilon^{1-\xi}}{1-\xi} - \frac{\ddot{\varepsilon}_{ij}(t)\epsilon^{2-\xi}}{2-\xi} + \int_0^{t-\epsilon} \frac{\dot{\varepsilon}_{ij}(\tau)}{(t-\tau)^{\xi}} d\tau\right) \tag{4.11}$$

with the error term being:

$$\frac{\varepsilon_{ij}^{(3)}(\zeta_1)\epsilon^{3-\xi}}{2(3-\xi)} \tag{4.12}$$

Here, the parameter ϵ is chosen equal to h, the stepsize. If the next step being calculated is at t_{n+1}, then the upper limit of the integral is $t_{n+1} - h = t_n$. Equation 4.11 at t_{n+1} becomes:

$$\begin{aligned} D^{\xi}\varepsilon_{ij}(t_{n+1}) &= \frac{1}{\Gamma(1-\xi)}\left(\frac{\dot{\varepsilon}_{ij}(t_{n+1})h^{1-\xi}}{1-\xi} - \frac{\ddot{\varepsilon}_{ij}(t_{n+1})h^{2-\xi}}{2-\xi}\right. \\ &+ \left.\int_0^{t_n} \frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^{\xi}} d\tau\right) \end{aligned} \tag{4.13}$$

with the error term becoming:

$$\frac{\varepsilon_{ij}^{(3)}(\zeta_1)h^{3-\xi}}{2(3-\xi)} \tag{4.14}$$

with ζ_1 in the interval $[t_n, t_{n+1}]$. Depending on the value of ξ, which is in the interval $]0, 1[$, the error due to this simplification will be of order 2 or 3.

The next step is to compute the integral of Equation 4.13:

$$\int_0^{t_n} \frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^{\xi}} d\tau \tag{4.15}$$

4.3 A Numerical Algorithm for the Integral of the Fractional Derivative

A numerical integrator for Equation 4.15 must satisfy some constraints. Solving this integral is a sub-process in solving the complete constitutive equation of which the fractional derivative is just one term. A numerical algorithm takes care of the constitutive equation and sets the value of h according to the precision of the results. Therefore, when solving Equation 4.15, the values of the different steps have already been set by the numerical algorithm solving the constitutive equation and cannot be modified by the algorithm solving the integral. The integrator of Equation 4.15 is using a vector of data containing values of τ and corresponding values of $\dot{\varepsilon}_{ij}(\tau)$. These values of τ are not equally spaced, and for simplicity, the integrator will ideally be a one-step method. A trapezoidal rule is the simplest integrator meeting this condition [49]:

$$\int_{t_0}^{t_1} f(t)dt = \frac{h}{2}\left(f(t_0) + f(t_1)\right) + \mathcal{O}(h^3) \tag{4.16}$$

with the error term being:

$$-\frac{h^3}{12}\ddot{f}(\zeta) \tag{4.17}$$

where ζ is in the interval $[t_0, t_1]$. Applying this piecewise to the integral part of the fractional derivative gives:

$$\begin{aligned}\int_0^{t_n} \frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^\xi}d\tau \;&=\; \sum_{t_k=0}^{t_{n-1}} \frac{t_{k+1}-t_k}{2}\left(\frac{\dot{\varepsilon}_{ij}(t_k)}{(t_{n+1}-t_k)^\xi}\right. \\ &\quad \left. + \frac{\dot{\varepsilon}_{ij}(t_{k+1})}{(t_{n+1}-t_{k+1})^\xi}\right)\end{aligned} \tag{4.18}$$

This sub-process carries the index k, whereas the integration of the whole constitutive equation carries the index n. The error term is:

$$\sum_{t_k=0}^{t_{n-1}} \frac{(t_{k+1}-t_k)^3}{12}\frac{d^2}{d\tau^2}\left(\frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^\xi}\right)\Bigg|_{\zeta_k} \tag{4.19}$$

The variable ζ_k is a value in the interval $[t_k, t_{k+1}]$ used to evaluate the error.

The fractional derivative expressed by Equation 4.13 becomes:

$$D^{\xi}\varepsilon_{ij}(t_{n+1}) = \frac{1}{\Gamma(1-\xi)}\left[\frac{\dot{\varepsilon}_{ij}(t_{n+1})h^{1-\xi}}{1-\xi} - \frac{\ddot{\varepsilon}_{ij}(t_{n+1})h^{2-\xi}}{2-\xi} + \sum_{t_k=0}^{t_{n-1}} \frac{t_{k+1}-t_k}{2}\left(\frac{\dot{\varepsilon}_{ij}(t_k)}{(t_{n+1}-t_k)^{\xi}} + \frac{\dot{\varepsilon}_{ij}(t_{k+1})}{(t_{n+1}-t_{k+1})^{\xi}}\right)\right] \quad (4.20)$$

with the error term being:

$$\frac{\varepsilon_{ij}^{(3)}(\zeta_1)h^{3-\xi}}{2(3-\xi)} + \sum_{t_k=0}^{t_{n-1}} \frac{(t_{k+1}-t_k)^3}{12}\frac{d^2}{d\tau^2}\left(\frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^{\xi}}\right)\Bigg|_{\zeta_k} \quad (4.21)$$

with ζ_1 in the interval $[t_n, t_{n+1}]$ and ζ_k in the intervals $[t_k, t_{k+1}]$.

The values of $\dot{\varepsilon}_{ij}(t_k)$ and $\dot{\varepsilon}_{ij}(t_{k+1})$ are stored values retrieved by the integrator, but an algorithm for Equation 4.20 still needs to calculate $\dot{\varepsilon}_{ij}(t_{n+1})$ and $\ddot{\varepsilon}_{ij}(t_{n+1})$. This is the topic of the next section.

4.4 The First and Second Derivatives

Typically, commercial finite element packages allowing user material subroutine give the values of the strain at t_{n+1} to the subroutine and expect values for the stress at t_{n+1} to be calculated. The first and second derivatives of the strain must be calculated within the subroutine.

For the first derivative, finite differences are used. A first order Taylor polynomial is expanded about t_{n+1}:

$$\varepsilon_{ij}(t_n) = \varepsilon_{ij}(t_{n+1}) - h\dot{\varepsilon}_{ij}(t_{n+1}) + \mathcal{O}(h^2) \quad (4.22)$$

with the error term being:

$$\frac{h^2}{2}\ddot{\varepsilon}_{ij}(\zeta_2) \quad (4.23)$$

where ζ_2 is in the interval $[t_n, t_{n+1}]$. The next step is to isolate $\dot{\varepsilon}_{ij}(t_{n+1})$:

$$\dot{\varepsilon}_{ij}(t_{n+1}) = \frac{\varepsilon_{ij}(t_{n+1}) - \varepsilon_{ij}(t_n)}{h} \quad (4.24)$$

The error term is also divided by h, just as is done to the other terms of the equation when isolating $\dot{\varepsilon}_{ij}(t_{n+1})$, and becomes an error term of order one:

$$-\frac{h}{2}\ddot{\varepsilon}_{ij}(\zeta_2) \quad (4.25)$$

The value of $\varepsilon_{ij}(t_{n+1})$ is given to the user, as well as t_{n+1}. The value of $\varepsilon_{ij}(t_n)$ is stored at the end of the step going from t_{n-1} to t_n. If it is the first step, then $\varepsilon_{ij}(t_n)$ is equal to 0, since zero initial conditions are assumed.

For the second derivative, an expression is derived from the two following Taylor's expansions:

$$\varepsilon_{ij}(t_n) = \varepsilon_{ij}(t_{n+1}) - h\dot{\varepsilon}_{ij}(t_{n+1}) + \frac{h^2}{2}\ddot{\varepsilon}_{ij}(t_{n+1}) + \mathcal{O}(h^3) \tag{4.26}$$

$$\varepsilon_{ij}(t_{n-1}) = \varepsilon_{ij}(t_{n+1}) - (h+h_1)\dot{\varepsilon}_{ij}(t_{n+1}) + \frac{(h+h_1)^2}{2}\ddot{\varepsilon}_{ij}(t_{n+1}) + \mathcal{O}(h^3) \tag{4.27}$$

where h_1 is $(t_n - t_{n-1})$. The error terms associated to Equations 4.26 and 4.27 are respectively:

$$-\frac{h^3}{3}\varepsilon_{ij}^{(3)}(\zeta_3) \tag{4.28}$$

$$-\frac{(h+h_1)^3}{3}\varepsilon_{ij}^{(3)}(\zeta_4) \tag{4.29}$$

where ζ_3 is in the interval $[t_n, t_{n+1}]$ and ζ_4 is in the interval $[t_{n-1}, t_{n+1}]$. Equation 4.27 is multiplied by a factor of $h/(h+h_1)$, and it is subtracted from Equation 4.26 in order to eliminate the term $\dot{\varepsilon}_{ij}(t_{n+1})$. This yields:

$$\ddot{\varepsilon}_{ij}(t_{n+1}) = 2\frac{h(\varepsilon_{ij}(t_{n-1}) - \varepsilon_{ij}(t_n)) + h_1(\varepsilon_{ij}(t_{n+1}) - \varepsilon_{ij}(t_n))}{hh_1(h+h_1)} \tag{4.30}$$

The same manipulation is applied to the error terms given in Equations 4.28 and 4.29 to obtain:

$$\frac{2}{3}\left(\frac{(h+h_1)^2}{h}\varepsilon_{ij}^{(3)}(\zeta_4) - \frac{h^2}{h_1}\varepsilon_{ij}^{(4)}(\zeta_3)\right) \tag{4.31}$$

As for the first derivative, the error is of order 1.

For the first step, the information at t_{n-1} is not available. The only available data are the strains at t_0 and t_1. With only two data points, only a linear function can be assumed and it results in a null second derivative. Therefore, $\ddot{\varepsilon}_{ij}(t_{n+1})$ is assumed equal to 0 on the first step.

The expressions for the first and second derivatives are now incorporated into Equation 4.20:

$$\begin{aligned} D^{\xi}\varepsilon_{ij}(t_{n+1}) &= \frac{1}{\Gamma(1-\xi)}\left[\frac{\varepsilon_{ij}(t_{n+1})-\varepsilon_{ij}(t_n)}{h}\frac{h^{1-\xi}}{1-\xi}\right. \\ &- 2\frac{h(\varepsilon_{ij}(t_{n-1})-\varepsilon_{ij}(t_n))+h_1(\varepsilon_{ij}(t_{n+1})-\varepsilon_{ij}(t_n))}{hh_1(h+h_1)}\frac{h^{2-\xi}}{2-\xi} \\ &+ \left.\sum_{t_k=0}^{t_{n-1}}\frac{t_{k+1}-t_k}{2}\left(\frac{\dot{\varepsilon}_{ij}(t_k)}{(t_{n+1}-t_k)^{\xi}}+\frac{\dot{\varepsilon}_{ij}(t_{k+1})}{(t_{n+1}-t_{k+1})^{\xi}}\right)\right] \end{aligned} \tag{4.32}$$

The error terms specified in Equations 4.21, 4.25, and 4.31 are combined together to give:

$$\begin{aligned} &\frac{1}{\Gamma(1-\xi)}\left[\left(-\frac{h}{2}\ddot{\varepsilon}_{ij}(\zeta_2)\right)\frac{h^{1-\xi}}{1-\xi}\right. \\ &+ \frac{2}{3}\left(\frac{(h+h_1)^2}{h}\varepsilon_{ij}^{(3)}(\zeta_4)-\frac{h^2}{h_1}\varepsilon_{ij}^{(4)}(\zeta_3)\right)\frac{h^{2-\xi}}{2-\xi} \\ &+ \left.\sum_{t_k=0}^{t_{n-1}}\frac{(t_{k+1}-t_k)^3}{12}\frac{d^2}{d\tau^2}\left(\frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^{\xi}}\right)\Bigg|_{\zeta_k}\right] \end{aligned} \tag{4.33}$$

The lowest order of error is $(2-\xi)$ and Equation 4.32 is written:

$$\begin{aligned} D^{\xi}\varepsilon_{ij}(t_{n+1}) &= \frac{1}{\Gamma(1-\xi)}\left[\frac{\varepsilon_{ij}(t_{n+1})-\varepsilon_{ij}(t_n)}{h}\frac{h^{1-\xi}}{1-\xi}\right. \\ &- 2\frac{h(\varepsilon_{ij}(t_{n-1})-\varepsilon_{ij}(t_n))+h_1(\varepsilon_{ij}(t_{n+1})-\varepsilon_{ij}(t_n))}{hh_1(h+h_1)}\frac{h^{2-\xi}}{2-\xi} \\ &+ \sum_{t_k=0}^{t_{n-1}}\frac{t_{k+1}-t_k}{2}\left(\frac{\dot{\varepsilon}_{ij}(t_k)}{(t_{n+1}-t_k)^{\xi}}+\frac{\dot{\varepsilon}_{ij}(t_{k+1})}{(t_{n+1}-t_{k+1})^{\xi}}\right) \\ &\left.+ \mathcal{O}(h^{2-\xi})\right] \end{aligned} \tag{4.34}$$

This expression for the fractional derivative is now ready for implementation.

4.5 Test Equations for the One-Dimensional Case

The algorithm is implemented in language C and tested for a one-dimensional constitutive equation:

$$\sigma_{xx} = E_e\varepsilon_{xx} + aE_eD^{\xi}\varepsilon_{xx} \tag{4.35}$$

where E_e is the equilibrium Young's modulus. A user subroutine for a finite element package typically passes to the subroutine values for the strain and the subroutine returns values for the stress according to the simulated constitutive equation. To emulate this, values for ε_{xx} are passed to the subroutine written in C, and values for σ_{xx} are returned. Equation 4.35 is transformed in the frequency domain, and inverted back to the time domain numerically to obtain some results against which to test the performance of the algorithm programmed in C. This approach is adopted since analytical results for Equation 4.35 cannot be obtained for an arbitrary value of ξ.

The Laplace transform of Equation 4.35 is:

$$\sigma_{xx}(s) = E_e\varepsilon_{xx}(s) + aE_e s^{\xi}\varepsilon_{xx}(s) \tag{4.36}$$

where s is the Laplace variable. Two different strain inputs are considered.

4.5.1 Sinusoidal Strain Input

The first strain input is a sine function:

$$\varepsilon_{xx}(t) = 1 \times 10^{-3}\frac{\sin(2\pi ft)}{2\pi f} \tag{4.37}$$

where f is the frequency of the sine wave. The factor in front of the sine function is chosen to reflect the order of magnitude typical for small strain theory. The Laplace transform of Equation 4.37 is:

$$\varepsilon_{xx}(s) = 1 \times 10^{-3}\frac{1}{s^2 + (2\pi f)^2} \tag{4.38}$$

4.5.2 Decaying Sinusoidal Strain Input

Another strain input tested is an exponentially decreasing sine function:

$$\varepsilon_{xx}(t) = 1 \times 10^{-3}\frac{\exp^{kt}\sin(2\pi ft)}{2\pi f} \tag{4.39}$$

where k is negative to ensure the decay of the sine function. The Laplace transform of Equation 4.39 is:

$$\varepsilon_{xx}(s) = 1 \times 10^{-3}\frac{1}{(s-k)^2 + (2\pi f)^2} \tag{4.40}$$

Frequency f	$1.59\ Hz$
Equilibrium Young's Modulus E_e	$3.43 \times 10^9\ Pa$
Damping Coefficient a	0.12
Order of the Fractional Derivative ξ	0.22
Parameter of the Exponential k	-0.5

Table 4.1: Parameters of the One-Dimensional Test Equation

4.5.3 Constitutive Equations for the Test Cases

Since no time-domain closed form solution exists for arbitrary values of ξ, the constitutive equations with Equations 4.37 and 4.39 as strain inputs are solved in the frequency domain to provide a comparative solution to the ones obtained with the time domain algorithm written in C.

Using the first strain input, Equation 4.38, the constitutive equation becomes:

$$\sigma_{xx}(s) = 1 \times 10^{-3} \left(E_e \frac{1}{s^2 + (2\pi f)^2} + a E_e s^{\xi} \frac{1}{s^2 + (2\pi f)^2} \right) \tag{4.41}$$

With the second strain input, Equation 4.40, the constitutive equation becomes:

$$\sigma_{xx}(s) = 1 \times 10^{-3} \left(E_e \frac{1}{(s-k)^2 + (2\pi f)^2} + a E_e s^{\xi} \frac{1}{(s-k)^2 + (2\pi f)^2} \right) \tag{4.42}$$

Equations 4.41 and 4.42 are solved in the frequency domain with a numerical scheme described by Wilcox [50]. The algorithm is given in Appendix A. In Potvin [51], it is tested against another algorithm developed by Wilcox and Gibson [52] and it gives comparable results. Solutions for Equations 4.41 and 4.42 are shown in Figure 4.1. The parameters used in both equations are given in Table 4.1.

4.6 Accuracy of the Algorithm

The numerical Laplace inverse gives results of varying accuracy according to the number of data points used to obtain the results. The numerical Laplace inverse used [50] is based on a Fast Fourier Transform and needs a number of

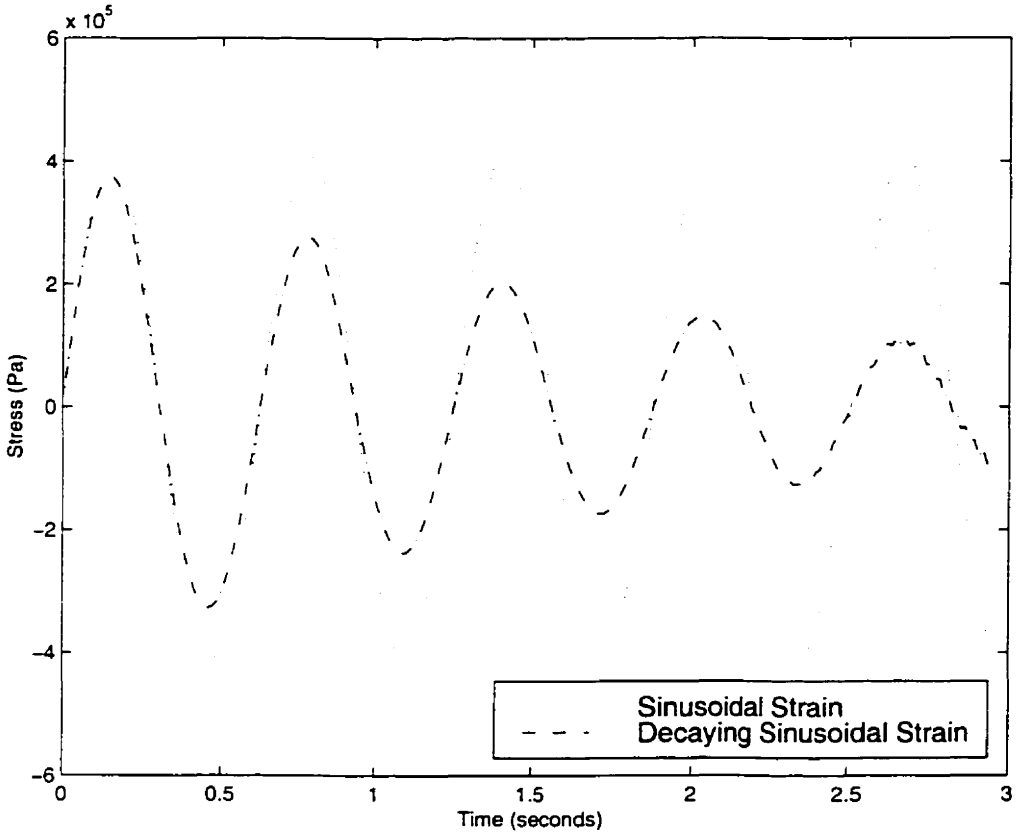

Figure 4.1: Two One-Dimensional Test Equations

data points equal to 2^n. However, the time solution produced with the numerical Laplace inverse is not accurate at the end of the time scale. Therefore, 85% of the time scale is kept and the rest of the data points are discarded.

The chosen test cases exhibit an oscillating frequency of 1.59 Hz. The simulation is done for 2.975 s, which implies 4.73 cycles. If the number of data points is chosen equal to 2^{10}, 870 data points are kept, resulting in 184 data points per cycle. This number is high enough to ensure precision.

For the algorithm designed in this chapter, varying numbers of data points are chosen to evaluate its sensitiveness to step size. The results from the numerical Laplace inverse are interpolated at the same data points used for the algorithm and the difference between the results from the algorithm and from the interpolated numerical Laplace inverse is computed. The number of data points for the algorithm is reduced until a difference of 1.5×10^4 Pa is obtained for any data point between the results from the algorithm and the results from the numerical Laplace inverse. By trial and error, this value was found to be the one for which the error would always lead to some divergence between the two methods for the studied cases.

4.6.1 Results for a Sinusoidal Strain Input

Table 4.2 compares the results from the algorithm based on the trapezoidal rule and from the numerical Laplace inverse in the case of the constitutive equation for a sinusoidal strain input. The time in the simulation for which a difference of 1.5×10^4 Pa is reached between the two methods is indicated in the table as being the critical time. The maximum difference between the two methods for any data point is indicated as the maximum error. The tests are done on a Pentium personal computer. Figures 4.2 and 4.3 show the fits between the two curves in the best case and the worst case. The agreement between the two curves in Figure 4.2 is excellent. In Figure 4.3, the agreement is still very good, but only 14 data points per cycle are used for the curve obtained with the algorithm and it is not perfectly smooth. Moreover, the level of error is

Number of Data Points	Critical Time (seconds)	Maximum Error (Pa)	CPU Time for the Laplace Inverse (seconds)	CPU Time for the Algorithm (seconds)
800		2.1196×10^3	2.14	3.46
600		2.6587×10^3	1.54	1.32
400		3.5206×10^3	1.54	0.61
200		5.7624×10^3	1.48	0.17
150		6.9912×10^3	1.48	0.11
125		8.2962×10^3	1.48	0.06
110		9.1526×10^3	1.60	0.05
100		1.0016×10^4	1.65	0.05
90		1.1237×10^4	1.65	0.01
80		1.2555×10^4	1.59	0.01
70		1.4174×10^4	1.60	0.05
66	2.1818	1.5465×10^4	1.54	0.01

Table 4.2: Accuracy of the Algorithm for a Sinusoidal Strain Input

slowly increasing. Still, the algorithm is very stable and predicts accurate results even for very few data points per cycle.

Table 4.2 also compares the CPU time, the time used by the processor to do the calculations, for both the numerical Laplace inverse and the algorithm. The numerical Laplace inverse is run for each case with 1024 data points. The variations in the CPU time for the Laplace inverse are due to the Windows operating system since the same equation is run everytime. The CPU time for the Laplace inverse is indicated to give a comparative value of the CPU time for the algorithm. There are instances when the algorithm requires more CPU time, but the Laplace inverse also does, indicating a variation due to the operating system, rather than the algorithm itself. The CPU time required by the algorithm based on the trapezoidal rule decreases significantly as the number of data points is reduced.

4.6.2 Results for a Decaying Sinusoidal Strain Input

A second test is done with a decaying sinusoidal strain input represented by Equation 4.39. Table 4.3 gives the results. The same trend as with the

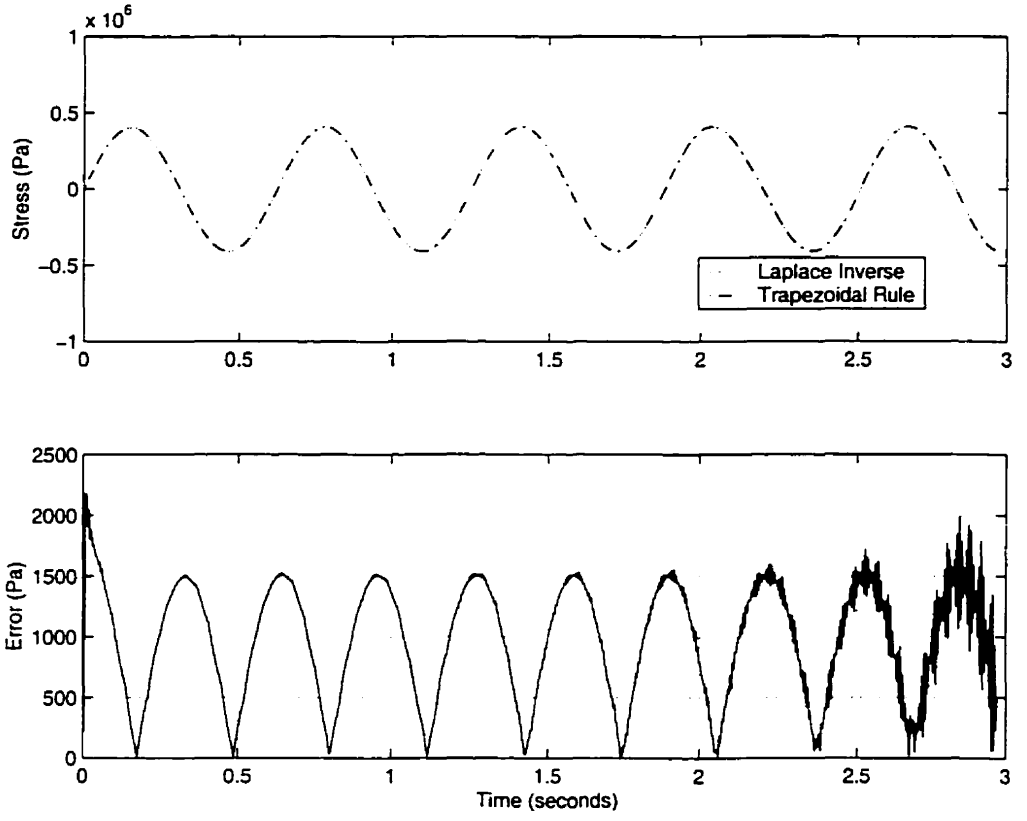

Figure 4.2: Results with 800 Points for a Sinusoidal Strain Input

Number of Data Points	Critical Time (seconds)	Maximum Error (Pa)	CPU Time for the Laplace Inverse (seconds)	CPU Time for the Algorithm (seconds)
800		1.8160×10^3	1.59	2.36
600		2.2723×10^3	1.48	1.32
400		3.0872×10^3	1.48	0.60
200		5.1406×10^3	1.54	0.17
180		5.5731×10^3	1.48	0.11
160		5.9441×10^3	1.48	0.11
140		6.4098×10^3	1.60	0.11
120		6.9980×10^3	1.60	0.06
100		8.3365×10^3	1.49	0.01
80		1.0405×10^4	1.48	0.01
60		1.3943×10^4	1.48	0.01
54	0.2778	1.5307×10^4	1.48	0.01

Table 4.3: Accuracy of the Algorithm for a Decaying Sinusoidal Strain Input

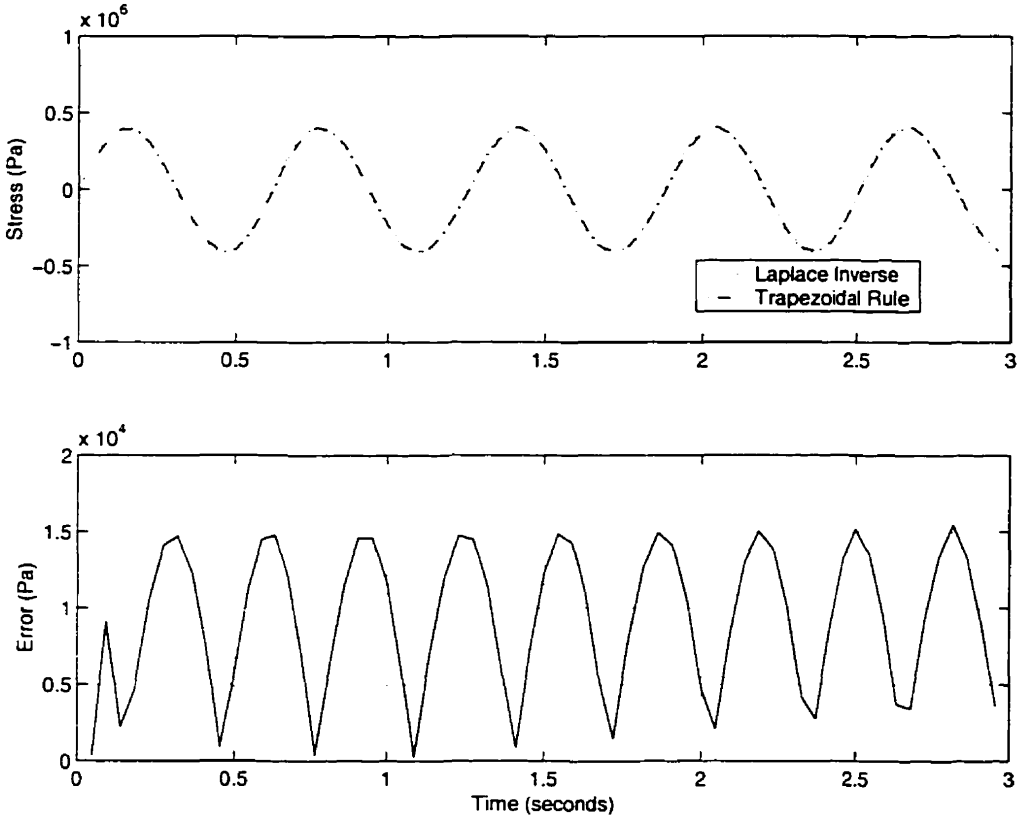

Figure 4.3: Results with 66 Points for a Sinusoidal Strain Input

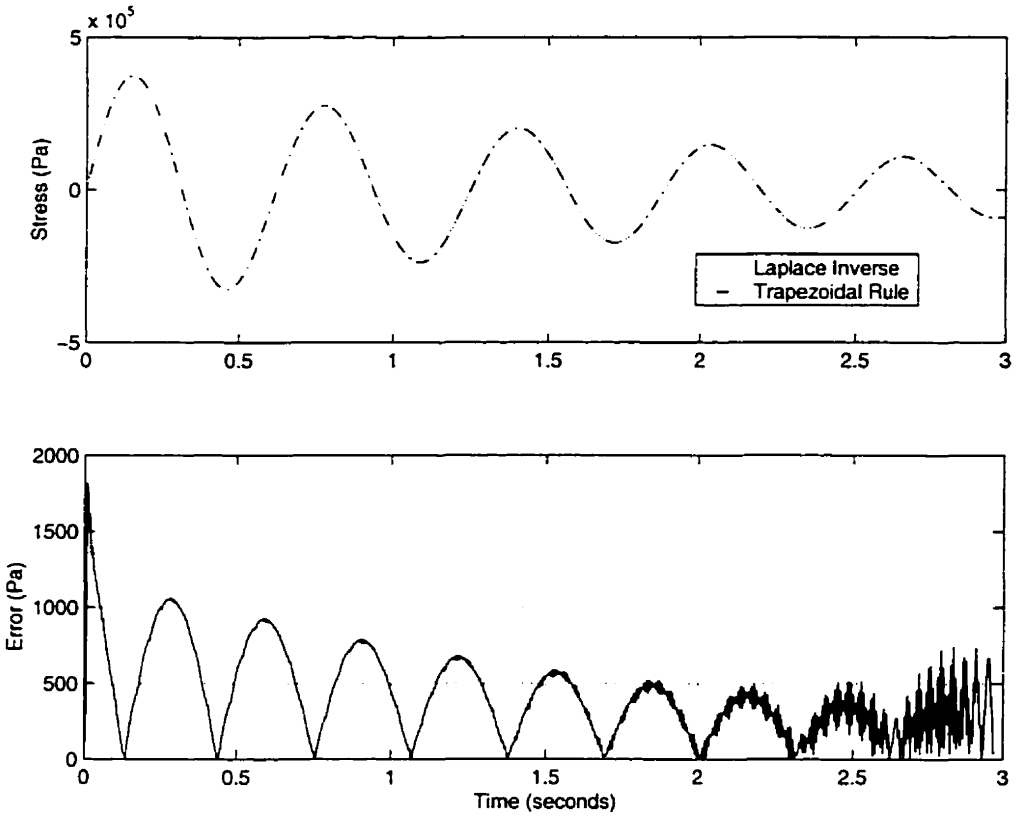

Figure 4.4: Results with 800 Points for a Decaying Sinusoidal Strain Input

sinusoidal strain input is seen. The CPU time required for the algorithm based on the trapezoidal rule decreases significantly as the number of data points decreases. Figures 4.4 and 4.5 show the fits between the two curves in the best case and the worst case. The agreement between the two curves in Figure 4.4 is excellent. In Figure 4.5, the agreement between the two curves is still very good, but small differences can be seen due to the low number of data points for the algorithm based on the trapezoidal rule. Only 11 points per cycle are used which is very low to accurately reproduce a sinusoidal wave. Again, the algorithm does not need very small steps despite computing a differential equation with a fractional derivative.

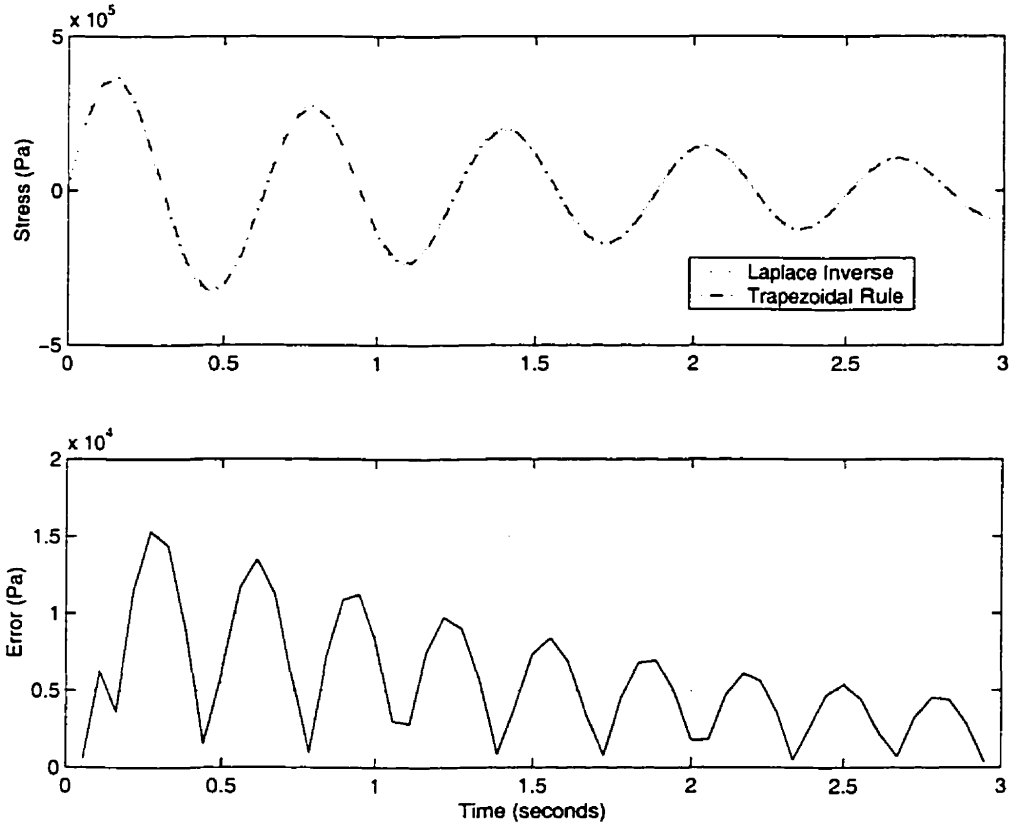

Figure 4.5: Results with 54 Points for a Decaying Sinusoidal Strain Input

4.7 Selection of Stored Data

The integral part of the fractional derivative, as written in Equation 4.13, has 0 as an inferior limit, implying the integral has to be started from zero for each step of the integration of the dynamic system. The integral depends explicitly on t_{n+1}, which is different at every step, preventing use of results obtained for the integral on previous steps. As the integration process gets away from zero, the number of calculations to be carried out increases significantly, slowing down the numerical computation. To accelerate the algorithm, Koh and Kelly [22] used a time window. Only the values for a determined amount of time were considered. What happened previous to that was neglected. However, as stated by Koh and Kelly themselves, this works only for functions oscillating around a zero mean. Dynamic systems do not necessarily oscillate around zero and a more general approach is needed. In the numerical algorithm described here, larger steps are taken for events happening far from the current time. Hence, the whole history is taken into account, but with less precision for faraway events.

A maximum number of data to be stored is determined. This number is called *memmax*. A smaller number, also chosen by the user and called *memcom*, is the number of data for which larger steps will be taken. At the end of each step, the algorithm stores in a list the time at which the stress is computed and the corresponding strain. A counter keeps track of the number of data stored in the list. When the maximum number of data allowed is reached, the algorithm takes the first group of data, of size *memcom*, and keeps only one data out of two. The remaining data, between the *memcom* and *memmax* limits are displaced in lower positions in the list. A number of places equal to half the size of *memcom* are now available in the list. The computation goes on and new data fills the list until the *memmax* limit is reached. At this point, another compression and displacement of data takes place.

The first time such a compression of data is carried out, the steps in the *memcom* group become approximately twice as large as the initial steps if it is assumed the steps have similar lengths. When a compression is done for the second time, the last *memcom* group now has steps equivalent to approximately four times the initial step size. Thus, as more compressions are carried out, the first few steps become larger.

The last section showed that the algorithm is accurate when all data points are stored if a sufficient number of data points is used. This section looks into the limits of the compression scheme in terms of accuracy of the solution.

The same test cases defined in Sections 4.5 are used to study the acceptable level of data compression.

4.7.1 Data Compression for a Sinusoidal Strain Input

Table 4.4 gives the accuracy of various solutions for the case where 800 data points are used with the algorithm based on the trapezoidal rule. As for the numerical Laplace inverse, it is calculated using 1024 data points. Because the last data points generated by the numerical Laplace inverse are not accurate, only 85% of the data points are kept and this gives a maximum of 870 data points. With the algorithm based on the trapezoidal rule, the whole vector of data points is compressed each time it is full, therefore *memmax* is equal to *memcom*. For a simulation of 800 data points, the smallest storage vector which can be used to keep an acceptable level of precision is 264. This represents 33% of the total data points in a case where there are 169 data points per cycle.

The computing time of the algorithm based on the trapezoidal rule is quite high compared to the numerical Laplace inverse if all data points are kept in memory and used in calculations of the subsequent steps. However, if the minimum allowable storage vector is used, then the computing time required by the algorithm based on the trapezoidal rule becomes competitive. The algorithm based on the trapezoidal rule also has the advantage over the

memmax	*memcom*	Critical Time (seconds)	Maximum Error (Pa)	CPU Time Laplace Inverse (seconds)	CPU Time Algorithm (seconds)
800	800		2.1789×10^3	1.92	2.80
600	600		2.2445×10^3	1.43	1.93
400	400		3.3383×10^3	1.43	1.48
300	300		6.8059×10^3	1.54	1.15
270	270		9.3128×10^3	1.54	1.10
264	264		9.6437×10^3	1.43	1.10
263	263	2.9587	1.5977×10^4	1.48	1.10

Table 4.4: Accuracy of the Algorithm for a Sinusoidal Strain Input of 800 Data Points According to the Size of the Storage Vector

Number of Data Points	*memmax*	Ratio of *memmax* to Data Points	Number of Data Points per Cycle	CPU Time Laplace Inverse (seconds)	CPU Time Algorithm (seconds)
800	264	0.33	169	1.48	1.10
400	140	0.35	85	1.48	0.33
200	100	0.50	42	1.48	0.33
100	75	0.75	21	1.43	0.01

Table 4.5: Minimum Allowable Storage Vectors According to the Number of Data Points per Cycle for a Sinusoidal Strain Input

numerical Laplace inverse of being able to process strain inputs for which no theoretical Laplace transform exists.

Table 4.5 compares the smallest allowable storage vectors for different number of data points per cycle. As the number of data points per cycle decreases, the storage vector must increase in size. If the simulation is less precise, compression of data decreases the precision further. However, if a large number of data points is used, then it is possible to use a quite small storage vector.

Number of Data Points	*memmax*	Ratio of *memmax* to Data Points	Number of Data Points per Cycle	CPU Time Laplace Inverse (seconds)	CPU Time Algorithm (seconds)
800	264	0.33	169	1.43	1.10
400	140	0.35	85	1.43	0.27
200	100	0.50	42	1.37	0.11
100	57	0.57	21	1.43	0.01

Table 4.6: Minimum Allowable Storage Vectors According to the Number of Data Points per Cycle for a Decaying Sinusoidal Strain Input

4.7.2 Data Compression for a Decaying Sinusoidal Strain Input

The same tests are carried out on the decaying sinusoidal strain input described in Section 4.5. Table 4.6 gives the results. Very similar results to the case of a sinususoidal strain input are obtained. Improvement is seen in the case where there are 100 data points. For the decaying sinusoidal strain input. the smallest storage vector is 57, whereas it is 75 for the sinusoidal strain input.

These two examples show it is useful to compress data, rather than use all data in the computation of the fractional derivative. Although the precision is decreased by compressing the data, acceptable levels of precision can be achieved for a fraction of the CPU time.

4.7.3 Optimal Size of the Parameter *memcom*

So far, all cases examined had values of *memcom* equal to *memmax*. The case of a sinusoidal strain input for 400 data points is examined with varying values of *memmax* and *memcom*. For a given *memmax*, in this case, 200, the smallest allowable *memcom* is sought, again with 1.5×10^4 Pa as the limit of difference between the numerical Laplace inverse and the algorithm on any given data point. The numerical Laplace inverse is done with 1024 data points out of which the 870 first data points are kept. Table 4.7 shows the results.

memmax	*memcom*	Critical Time (seconds)	Maximum Error (Pa)	CPU Time Algorithm (seconds)
200	200		6.1174×10^3	0.77
200	150		6.9450×10^3	0.39
200	140		8.9762×10^3	0.38
200	130		1.0346×10^4	0.38
200	129	2.9625	1.5523×10^4	0.39

Table 4.7: Effect of *memcom* for a Sinusoidal Strain Input with 400 Data Points

There is no obvious gain from using a *memcom* different from *memmax*. The error increases when *memcom* decreases. As for the CPU time, there is no obvious advantage. No trend is seen as the value of *memcom* is reduced. The simulation with a *memcom* of 200 takes longer, but the corresponding numerical Laplace inverse takes also longer, 1.81 s, rather than the average 1.50 s. It can be assumed that the longer CPU time in that case is due to the operating system, rather than the solution itself. Both the numerical Laplace inverse and the algorithm for the case with *memcom* equal to 200 use approximately 40 s in excess to the other cases, probably to initialize arrays.

When *memcom* is smaller than *memmax*, less data get compressed when the storage vector is full. Thus, less space is freed in the storage vector. It will fill up more quickly and another compression happens earlier. This reduces the precision of the first steps and deteriorates the whole solution.

Table 4.8 shows the minimum *memcom* that can be used for given values of *memmax* to respect the maximum level of error allowed, again for a case with 400 data points. When *memmax* is low compared to the number of data points, it is not possible to use a *memcom* much lower than *memmax*. As *memmax* increases, it is possible to decrease significantly the value of *memcom*, but the computing time depends mainly on *memmax* and there is nothing to be gained with a value of *memcom* different from *memmax*. The best solution is still the one with the lowest allowable *memmax* and a *memcom* equal to *memmax*, since for similar CPU time, it is more precise.

memmax	*memcom*	CPU Time Algorithm (seconds)
200	130	0.38
190	134	0.38
180	134	0.38
170	134	0.39
160	134	0.33
150	138	0.33

Table 4.8: Values of *memmax* and *memcom* for a Sinusoidal Strain Input

4.8 Conclusion

A one-dimensional constitutive equation with a fractional derivative was solved numerically. The algorithm used an approximation for the fractional derivative and solved this approximation with finite differences and the trapezoidal rule. The algorithm generated accurate results when compared to a numerical Laplace inverse, using a competitive amount of computer time and it did not require small time steps as it has been observed for methods based on finite differences or Grünwald series. Moreover, the algorithm can handle strain inputs for which the Laplace inverse does not exist.

For the algorithm to be competitive in terms of computer time, a storage vector must be used. Rather than storing all the data points and using them all in the calculation of the fractional derivative, the data points far from the actual data point are kept with less precision. When the storage vector is full, one data out of two is kept, and this frees space for new data. The minimum efficient storage vector size depends on the complexity of the equation, but for the cases studied, it was about 33%. There is no gain in compressing only part of the storage vector and moving down the pile the remaining data. The storage vector will fill up more quickly, imposing another compression earlier. The CPU time is not improved and the precision is lowered.

The next chapter will address the implementation of this algorithm for three-dimensional cases in a finite element package.

Chapter 5

Finite Element Formulation of the Fractional Constitutive Equation

5.1 Introduction

The algorithm presented in Chapter 4 is implemented in *Samcef*, a commercial finite element package. The storage vector assumes the fixed format imposed by the finite element package and the Jacobian of the fractional constitutive equation must be derived and implemented. To test the efficiency of the subroutine, a cubic element is submitted to a uni-directional displacement. Two cases are explored, one with a sinusoidal displacement and one with a decaying sinusoidal displacement. For both cases, the impact of the various computing parameters are explored. The results from the subroutine are compared to the results obtained with the Laplace inverse developed in the last chapter. Comparing the results gives a measure of the precision of the results computed by the subroutine. The goal of this chapter is to verify the validity of the implementation of the fractional model in *Samcef*. The efficiency of the fractional model compared to other viscoelasticity models will be studied in the next chapter.

5.2 Numerical Implementation

In Chapter 4, an algorithm to solve the following one-dimensional equation was developed:

$$\sigma_{xx} = E_e \varepsilon_{xx} + a E_e D^{\xi} \varepsilon_{xx} \tag{5.1}$$

Three-dimensional constitutive equations are now considered. Therefore, some adjustments to the algorithm developed in Chapter 4 are necessary. At each step, the strain tensor at the end of the interval is given to the user, and the corresponding stress tensor must be returned to the main program, as well as the Jacobian of the stress-strain relationship. As defined in Section 2.4.2, the fractional Voigt-Kelvin model is:

$$S_{ij} = 2G_e e_{ij} + 2aG_e D^{\xi} e_{ij} \tag{5.2}$$

$$S'' = 3K_e e'' \tag{5.3}$$

Whereas the algorithm developed in Chapter 4 dealt with one fractional equation, for Equation 5.2, six fractional equations need to be calculated. This does not introduce additional difficulties, but the storage requirements are multiplied by six. Therefore, the two main differences with the algorithm for the one-dimensional equation are the storage structure and the need for the Jacobian of the stress-strain equation.

5.2.1 The Storage Structure

To solve the one-dimensional equation, a single storage vector containing the time and the corresponding derivative of the strain was developed. Within a finite element package, the storage structure is more rigidly defined and varies from package to package. In the case of *Samcef*, different types of vectors can be defined. Vectors are identified as containing scalars, vectors, or tensors, where a vector of scalars is a vector of one-dimensional data, a vector of vectors is a vector of three-dimensional data, and a vector of tensors is a vector of data comprising six or nine components. For this algorithm,

three vectors are defined. The first one is a vector of three tensors having six components each. The first tensor is the variable DS, a variable used in the estimation of the so-called local error, which is rather the difference between the stress computed by the constitutive equation and the stress obtained with Hooke's law. The second tensor is the strain tensor at the beginning of the current step and the third tensor is the strain tensor at the beginning of the preceding step.

The second storage vector is a vector of (MEMMAX + 1) scalars. The parameter MEMMAX has the same significance as the parameter *memmax* defined in Chapter 4. It is the maximum number of data stored. The first scalar of this storage vector is the number of steps stored so far, and the remaining scalars are the times at each stored step, up to a maximum of MEMMAX, after which a compression of data occurs as described in Chapter 4.

The last storage vector is a vector of MEMMAX tensors containing the time derivatives of the strain corresponding to the stored times in the second storage vector. Figure 5.1 illustrates the data stored within the subroutine. The finite element package initializes the storage vectors by filling all values with zeroes. As the time solution proceeds, the vectors fill themselves. Once the vectors are full, one data out of two is kept, therefore freeing half the storage vectors for new data. For example, the data at Step 1 would be eliminated, but the data at Step 2 would be kept. The data at Step 3 would be eliminated, but the data at Step 4 would be kept, and so on. Zero initial conditions are assumed and the values at $t = 0$ are not kept, since they are known and are equal to zero.

5.2.2 The Jacobian of the Fractional Constitutive Equation

Samcef, as most finite element packages, requires the user to give the Jacobian of the constitutive equation, $\partial\sigma_{ij}(t_{n+1})/\partial\varepsilon_{kl}(t_{n+1})$. This Jacobian defines the change in the stress component caused by a perturbation in the strain

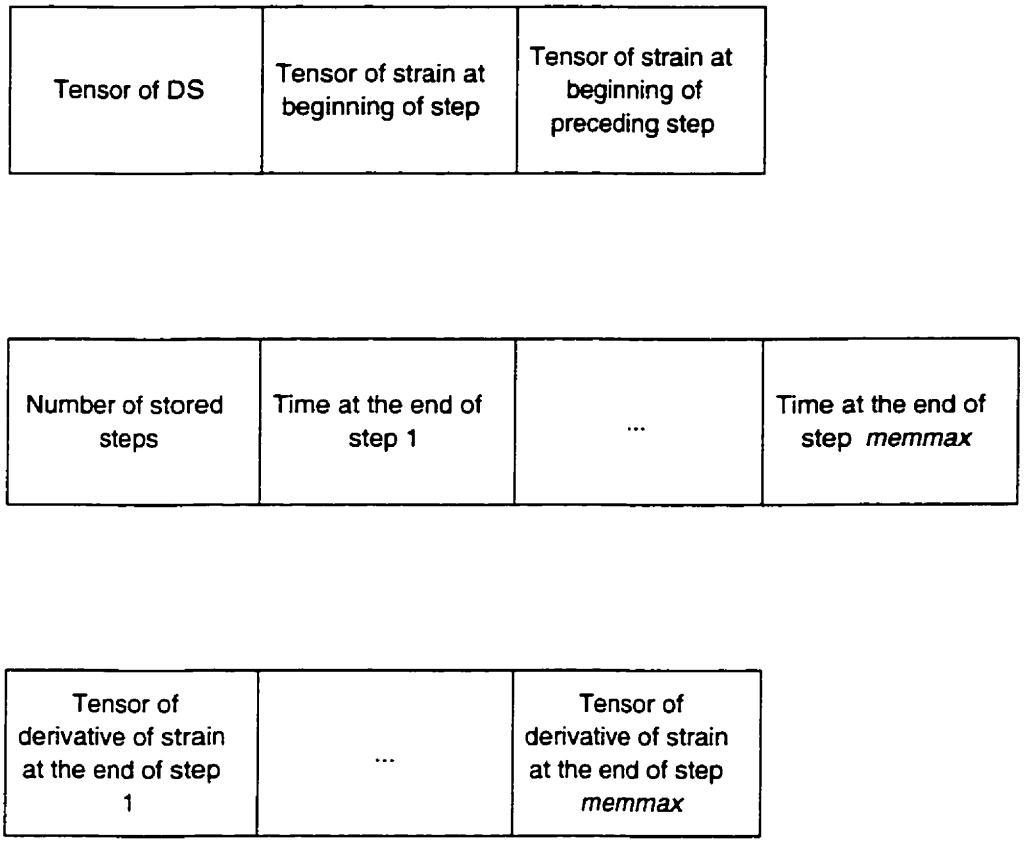

Figure 5.1: Data Storage for the *Samcef* Subroutine

component at the end of that given step.

In Chapter 4, one form of the constitutive equation is derived for the first step, for which the second derivative of the strain is zero, and another expression for the subsequent steps. Therefore, there is one Jacobian expression for the first step, and another Jacobian for the subsequent steps.

The expressions for $\sigma_{ij}(t_{n+1})$ and $\varepsilon_{ij}(t_{n+1})$ are needed, but the algorithm calculates the deviatoric and volumetric stresses. Some conversions are necessary. The following steps are taken:

1. Calculate $\dot{\varepsilon}_{ij}(t_{n+1})$.

2. Calculate $\ddot{\varepsilon}_{ij}(t_{n+1})$.

3. Calculate the integral part of the fractional derivative for the six components of the strain tensor.

4. Calculate $D^{\xi}\varepsilon_{ij}(t_{n+1})$.

5. Calculate $e''(t_{n+1})$.

6. Calculate $e_{ij}(t_1)$.

7. Calculate $D^{\xi}e_{ij}(t_{n+1})$.

8. Calculate $S_{ij}(t_{n+1})$ using the fractional constitutive equation.

9. Calculate $S''(t_{n+1})$ using the elastic constitutive equation.

10. Calculate $\sigma_{ij}(t_{n+1})$.

11. Take the partial derivatives $\partial\sigma_{ij}(t_{n+1})/\partial\varepsilon_{kl}(t_{n+1})$.

As an example, these steps are applied to $\partial\sigma_{xx}(t_1)/\partial\varepsilon_{xx}(t_1)$.

Example of Calculations for the Jacobian

In this case, the values at t_0 are all zero: there is no stress and no strain at t_0. Moreover, to obtain the first component of the deviatoric strain tensor, only the three first components of the strain tensor are needed. It is a simplified case, but it is used to understand the process. The steps described previously are applied sequentially.

Step 1 *Calculate* $\dot{\varepsilon}_{ij}(t_{n+1})$.

The first derivatives of the first three components of the strain tensor are calculated using Equation 4.24:

$$\dot{\varepsilon}_{xx}(t_1) = \frac{\varepsilon_{xx}(t_1) - \varepsilon_{xx}(t_0)}{h} \tag{5.4}$$

$$\dot{\varepsilon}_{yy}(t_1) = \frac{\varepsilon_{yy}(t_1) - \varepsilon_{yy}(t_0)}{h} \tag{5.5}$$

$$\dot{\varepsilon}_{zz}(t_1) = \frac{\varepsilon_{zz}(t_1) - \varepsilon_{zz}(t_0)}{h} \tag{5.6}$$

Since all strains at t_0 are equal to 0, Equations 5.4 to 5.6 reduce to:

$$\dot{\varepsilon}_{xx}(t_1) = \frac{\varepsilon_{xx}(t_1)}{h} \tag{5.7}$$

$$\dot{\varepsilon}_{yy}(t_1) = \frac{\varepsilon_{yy}(t_1)}{h} \tag{5.8}$$

$$\dot{\varepsilon}_{zz}(t_1) = \frac{\varepsilon_{zz}(t_1)}{h} \tag{5.9}$$

Step 2 *Calculate* $\ddot{\varepsilon}_{ij}(t_{n+1})$.

As explained in Section 4.4, the second derivatives at t_1 are zero, since a linear approximation is made between t_0 and t_1. Therefore:

$$\ddot{\varepsilon}_{xx}(t_1) = \ddot{\varepsilon}_{yy}(t_1) = \ddot{\varepsilon}_{zz}(t_1) = 0 \tag{5.10}$$

Step 3 *Calculate the integral part of the fractional derivative for the six components of the strain tensor.*

Since only the first partial derivative of the Jacobian is needed for this example, only the fractional derivatives of the first three components of the

strain tensor are needed to compute the corresponding deviatoric stress. The expression for the integral part of the fractional derivative is given by Equation 4.18:

$$\int_0^{t_n} \frac{\dot{\varepsilon}_{ij}(\tau)}{(t_{n+1}-\tau)^{\xi}} d\tau = \sum_{t_k=0}^{t_{n-1}} \frac{t_{k+1}-t_k}{2} \left(\frac{\dot{\varepsilon}_{ij}(t_k)}{(t_{n+1}-t_k)^{\xi}} + \frac{\dot{\varepsilon}_{ij}(t_{k+1})}{(t_{n+1}-t_{k+1})^{\xi}} \right) \quad (5.11)$$

In this case, $t_{n+1} = t_1$ and $t_n = t_0$. The integral becomes an integral between zero and zero and is equal to zero. Moreover, this integral does not depend on $\varepsilon_{ij}(t_{n+1})$ and as such will not contribute any term to the Jacobian, $\partial\sigma_{ij}(t_{n+1})/\partial\varepsilon_{kl}(t_{n+1})$.

Step 4 *Calculate* $D^{\xi}\varepsilon_{ij}(t_{n+1})$.

Equation 4.20 gives the expression for the fractional derivative:

$$\begin{aligned} D^{\xi}\varepsilon_{ij}(t_{n+1}) &= \frac{1}{\Gamma(1-\xi)} \left[\frac{\dot{\varepsilon}_{ij}(t_{n+1})h^{1-\xi}}{1-\xi} - \frac{\ddot{\varepsilon}_{ij}(t_{n+1})h^{2-\xi}}{2-\xi} \right. \\ &+ \left. \sum_{t_k=0}^{t_{n-1}} \frac{t_{k+1}-t_k}{2} \left(\frac{\dot{\varepsilon}_{ij}(t_k)}{(t_{n+1}-t_k)^{\xi}} + \frac{\dot{\varepsilon}_{ij}(t_{k+1})}{(t_{n+1}-t_{k+1})^{\xi}} \right) \right] \quad (5.12) \end{aligned}$$

Using the values found in Steps 1 to 3, the fractional derivatives for the first three components of the strain tensors are:

$$D^{\xi}\varepsilon_{xx}(t_1) = \frac{1}{\Gamma(1-\xi)} \left(\frac{\varepsilon_{xx}(t_1)}{(1-\xi)h^{\xi}} \right) \quad (5.13)$$

$$D^{\xi}\varepsilon_{yy}(t_1) = \frac{1}{\Gamma(1-\xi)} \left(\frac{\varepsilon_{yy}(t_1)}{(1-\xi)h^{\xi}} \right) \quad (5.14)$$

$$D^{\xi}\varepsilon_{zz}(t_1) = \frac{1}{\Gamma(1-\xi)} \left(\frac{\varepsilon_{zz}(t_1)}{(1-\xi)h^{\xi}} \right) \quad (5.15)$$

Step 5 *Calculate* $e''(t_{n+1})$.

The volumetric strain is taken as the sum of the first three components of the strain tensor divided by three:

$$e''(t_1) = \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right] \quad (5.16)$$

Step 6 *Calculate* $e_{ij}(t_{n+1})$.

To obtain $e_{xx}(t_1)$, the expression given by Equation 2.2 is used:

$$e_{xx}(t_1) = \varepsilon_{xx}(t_1) - \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right] \quad (5.17)$$

Step 7 *Calculate* $D^{\xi}e_i(t_{n+1})$.

The fractional derivative of the deviatoric strain is calculated similarly to Equation 5.17 since the fractional derivative is a linear operator:

$$D^{\xi}e_{xx}(t_1) = D^{\xi}\varepsilon_{xx}(t_1) - \frac{1}{3}\left(D^{\xi}\varepsilon_{xx}(t_1) + D^{\xi}\varepsilon_{yy}(t_1) + D^{\xi}\varepsilon_{zz}(t_1)\right) \tag{5.18}$$

Using the values calculated in Step 4 in Equation 5.18 gives:

$$D^{\xi}e_{xx}(t_1) = \frac{1}{(1-\xi)h^{\xi}\Gamma(1-\xi)}\left(\varepsilon_{xx}(t_1) - \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right]\right) \tag{5.19}$$

Step 8 *Calculate* $S_{ij}(t_{n+1})$ *using the fractional constitutive equation.*

Equation 5.2 gives the fractional constitutive equation for the deviatoric part:

$$S_{xx}(t_1) = 2G_e e_{xx}(t_1) + 2aG_e D^{\xi}e_{xx}(t_1) \tag{5.20}$$

Using the values found in Steps 6 and 7, $S_{xx}(t_1)$ becomes:

$$\begin{aligned} S_{xx}(t_1) &= 2G_e\left(\varepsilon_{xx}(t_1) - \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right]\right) \\ &+ \frac{2aG_e}{(1-\xi)h^{\xi}\Gamma(1-\xi)}\left(\varepsilon_{xx}(t_1) - \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1)\right.\right. \\ &+ \left.\left.\varepsilon_{zz}(t_1)\right]\right) \end{aligned} \tag{5.21}$$

Step 9 *Calculate* $S''(t_{n+1})$ *using the elastic constitutive equation.*

The elastic volumetric constitutive equation is given by Equation 5.3:

$$S''(t_1) = 3K_e e''(t_1) \tag{5.22}$$

Using the values calculated in Step 5, Equation 5.22 becomes:

$$S''(t_1) = K_e\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right] \tag{5.23}$$

Step 10 *Calculate* $\sigma_{ij}(t_{n+1})$.

The stress is obtained by manipulating Equation 2.1:

$$\sigma_{xx}(t_1) = S_{xx}(t_1) + S''(t_1) \tag{5.24}$$

The results from Steps 8 and 9 are input in this equation:

$$\begin{aligned}\sigma_{xx}(t_1) &= 2G_e\left(\varepsilon_{xx}(t_1) - \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right]\right) \\ &+ \frac{2aG_e}{(1-\xi)h^{\xi}\Gamma(1-\xi)}\left(\varepsilon_{xx}(t_1) - \frac{1}{3}\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right]\right) \\ &+ K_e\left[\varepsilon_{xx}(t_1) + \varepsilon_{yy}(t_1) + \varepsilon_{zz}(t_1)\right] \qquad (5.25)\end{aligned}$$

Step 11 *Take the partial derivatives* $\partial\sigma_{ij}(t_{n+1})/\partial\varepsilon_{kl}(t_{n+1})$.

The partial derivative of Equation 5.25 is taken:

$$\frac{\partial\sigma_{xx}}{\partial\varepsilon_{xx}} = \frac{4G_e}{3} + \frac{4aG_e}{3(1-\xi)h^{\xi}\Gamma(1-\xi)} + K_e \qquad (5.26)$$

The Jacobian for the First Step

The preceding example is applied to all the components. Some symmetry applies given the isotropic nature of the model. The Jacobian for the first step, for which the second derivative is zero, is:

$$\frac{\partial\sigma_{xx}}{\partial\varepsilon_{xx}} = \frac{\partial\sigma_{yy}}{\partial\varepsilon_{yy}} = \frac{\partial\sigma_{zz}}{\partial\varepsilon_{zz}} = \frac{4G_e}{3} + \frac{4aG_e}{3(1-\xi)h^{\xi}\Gamma(1-\xi)} + K_e \qquad (5.27)$$

$$\frac{\partial\sigma_{xx}}{\partial\varepsilon_{yy}} = \frac{\partial\sigma_{xx}}{\partial\varepsilon_{zz}} = \frac{\partial\sigma_{yy}}{\partial\varepsilon_{xx}} = \frac{\partial\sigma_{yy}}{\partial\varepsilon_{zz}} = \frac{\partial\sigma_{zz}}{\partial\varepsilon_{xx}} = \frac{\partial\sigma_{zz}}{\partial\varepsilon_{yy}} = -\frac{2G_e}{3} - \frac{2aG_e}{3(1-\xi)h^{\xi}\Gamma(1-\xi)} + K_e \qquad (5.28)$$

$$\frac{\partial\sigma_{xy}}{\partial\varepsilon_{xy}} = \frac{\partial\sigma_{xz}}{\partial\varepsilon_{xz}} = \frac{\partial\sigma_{yz}}{\partial\varepsilon_{yz}} = G_e + \frac{aG_e}{(1-\xi)h^{\xi}\Gamma(1-\xi)} \qquad (5.29)$$

with all other partial derivatives equal to 0.

The Jacobian for the Subsequent Steps

For all other steps, the Jacobian is:

$$\begin{aligned}\frac{\partial\sigma_{xx}}{\partial\varepsilon_{xx}} &= \frac{\partial\sigma_{yy}}{\partial\varepsilon_{yy}} = \frac{\partial\sigma_{zz}}{\partial\varepsilon_{zz}} = \\ &\frac{4G_e}{3} + \frac{4aG_e}{3\Gamma(1-\xi)}\left(\frac{1}{(1-\xi)h^{\xi}} - 2\frac{h^{(1-\xi)}}{(2-\xi)(h+h_1)}\right) \\ &+K_e \qquad (5.30)\end{aligned}$$

$$\frac{\partial\sigma_{xx}}{\partial\varepsilon_{yy}} = \frac{\partial\sigma_{xx}}{\partial\varepsilon_{zz}} = \frac{\partial\sigma_{yy}}{\partial\varepsilon_{xx}} = \frac{\partial\sigma_{yy}}{\partial\varepsilon_{zz}} = \frac{\partial\sigma_{zz}}{\partial\varepsilon_{xx}} = \frac{\partial\sigma_{zz}}{\partial\varepsilon_{yy}} =$$

$$-\frac{2G_e}{3} - \frac{2aG_e}{3\Gamma(1-\xi)}\left(\frac{1}{(1-\xi)h^{\xi}} - 2\frac{h^{(1-\xi)}}{(2-\xi)(h+h_1)}\right)$$
$$+K_e \tag{5.31}$$

$$\frac{\partial\sigma_{xy}}{\partial\varepsilon_{xy}} = \frac{\partial\sigma_{xz}}{\partial\varepsilon_{xz}} = \frac{\partial\sigma_{yz}}{\partial\varepsilon_{yz}} =$$
$$G_e + \frac{aG_e}{\Gamma(1-\xi)}\left(\frac{1}{(1-\xi)h^{\xi}} - 2\frac{h^{(1-\xi)}}{(2-\xi)(h+h_1)}\right) \tag{5.32}$$

and again, with all other partial derivatives equal to 0. The variable h_1 is the step taken before the current step.

5.2.3 The Local Error

Finite element packages give the user the opportunity to use automatic time stepping. To use this option, an estimation of the local error produced by the user constitutive equation subroutine must be given back to the software, as well as an upper bound for the acceptable local error. Ideally, the local error of an equation is estimated by comparing the numerical solution with the analytical solution, but numerical solutions are precisely used because often the analytical solution is not available. Alternately, a higher order method is used to compare both numerical solutions and obtain an estimation of the local error. However, a higher order method is not available for this method. Another approach sometimes used for viscoelastic constitutive equations is to compare the stress obtained with the constitutive equation with the stress obtained with Hooke's law. Of course, Hooke's law is not the desired solution, but between two steps, the difference between the stress produced by the constitutive equation and Hooke's law should not vary excessively. If it does, it means the numerical solution produced by the constitutive equation is starting to diverge and a smaller time step should be taken. *Samcef* defines the local error for a user constitutive equation in the following manner:

$$\text{local error} = \frac{\max_{elements}(\text{local stress error})}{\max\left(\text{SREF}, \max_{elements}(\text{average local stress})\right)} \tag{5.33}$$

The local stress error is the variation in the difference between the computed stress and the one obtained with Hooke's law and SREF is a reference stress

defined by the user. The user also specifies a value called PRCV which is the upper limit tolerated for the local error.

To compute the local stress error, for each component of the stress tensor, the stress obtained by Hooke's law is subtracted from the computed stress. This variable is called DS(i), where the index represents the component of the new vector DS. For an isotropic tensor, there are six components in the vector DS. The value of DS(i) is then compared to DSN(i), where DSN(i) is the value of DS(i) which was computed at the last step. To get only one value for the local stress error, the magnitude of the vector resulting from the subtraction of DSN from DS is taken. The following steps summarize the evaluation of the local error for one element:

1. Add the six components of the stress tensor and divide the sum by six to obtain the average local stress.

2. Form the vector DS by subtracting the stress obtained from Hooke's law from the stress computed with the constitutive equation.

3. Form the local error vector by doing (DS − DSN) where DSN is the local error vector calculated at the last step.

4. Take the magnitude of the vector (DS − DSN) to obtain the local stress error.

5. Divide the local stress error by either the reference stress, SREF, or by the average local stress depending which one is the greatest. The result is the so-called local error.

This local error is then compared to PRCV, a value specified by the user. If the local error divided by PRCV is greater than 0.5×10^{-1}, then the size of the next step is decreased. If the ratio is equal to or greater than one, the step is rejected and a smaller time step is chosen.

5.3 Tests on a Single Cubic Element

The results from the subroutine are compared to the results from the inverse Laplace transform for the case of a cubic element submitted to a prescribed displacement on one of its face. The element is a first degree element with eight integration points, two along each direction. In the finite element package, the three-dimensional formulation is used:

$$\begin{aligned} S_{ij} &= 2G_e e_{ij} + 2aG_e D^{\xi} e_{ij} \\ S'' &= 3K_e e'' \end{aligned} \tag{5.34}$$

As in Chapter 4, the strain is input and the stress is calculated. From the finite element package point of view, it is a prescribed position which is specified. This prescribed position is applied to the whole face of the element. The strain is defined as [53]:

$$\varepsilon_{xx} = \frac{l - l_0}{l_0} \tag{5.35}$$

with l_0 the initial length and l the length after deformation. From this, the prescribed position becomes:

$$l = l_0(1 + \varepsilon_{xx}) \tag{5.36}$$

The two test cases are a sinusoidal strain input and a decaying sinusoidal strain input.

Case 1 *Sinusoidal Strain*

$$\varepsilon_{xx}(t) = 1 \times 10^{-3} \frac{\sin(2\pi f t)}{2\pi f} \tag{5.37}$$

where f is the frequency of the sine wave.

Case 2 *Decaying Sinusoidal Strain*

$$\varepsilon_{xx}(t) = 1 \times 10^{-3} \frac{\exp^{kt} \sin(2\pi f t)}{2\pi f} \tag{5.38}$$

where k is negative to ensure the decay of the sine function.

The cube simulated is made of PMMA and the material parameters are defined in Chapter 3. The parameter f of Equations 5.37 and 5.38 is taken equal to 1.59 Hz and the parameter k equal to -0.5.

5.3.1 One-Dimensional Frequency Domain Expression for Test Cases

The test case defined in the preceding section is compared to a solution obtained with the numerical Laplace inverse used in the last chapter. The test case of the preceding section is a three-dimensional case, but the numerical Laplace inverse algorithm is written for a one-dimensional equation. An expression for σ_{xx} in the case of a uni-directional prescribed displacement is sought to compare the finite element results with the numerical Laplace inverse results.

In the frequency domain, the one-dimensional equation is written:

$$\sigma_{xx}(s) = E^* \varepsilon_{xx}(s) \tag{5.39}$$

with s being the Laplace variable and E^* the complex Young's modulus.

In the case of the fractional Voigt-Kelvin model, the following complex shear modulus is defined in Section 2.4.2:

$$G^* = G_e + aG_e s^\xi \tag{5.40}$$

A constant bulk modulus is assumed resulting in a constant complex bulk modulus:

$$K^* = K_e \tag{5.41}$$

The complex Young's modulus is defined as [41]:

$$E^* = \frac{9K^*G^*}{3K^* + G^*} \tag{5.42}$$

Using Equations 5.40 and 5.41 in Equation 5.42 gives:

$$E^* = \frac{9K_e(G_e + aG_e s^\xi)}{3K_e + (G_e + aG_e s^\xi)} \tag{5.43}$$

This complex Young's modulus is used in Equation 5.39, the one-dimensional expression:

$$\sigma_{xx}(s) = \frac{9K_e(G_e + aG_e s^{\xi})}{3K_e + (G_e + aG_e s^{\xi})}\varepsilon_{xx}(s) \tag{5.44}$$

Equation 5.44 is used to evaluate the accuracy of the results generated by the subroutine. As in Chapter 4, it is solved using the numerical Laplace transform described in Wilcox [50]. The two test cases are the frequency domain expressions of Equations 5.37 and 5.38.

Case 1 *Sinusoidal Strain*

$$\varepsilon_{xx}(s) = 1 \times 10^{-3}\frac{1}{s^2 + (2\pi f)^2} \tag{5.45}$$

Case 2 *Decaying Sinusoidal Strain*

$$\varepsilon_{xx}(s) = 1 \times 10^{-3}\frac{1}{(s-k)^2 + (2\pi f)^2} \tag{5.46}$$

5.3.2 Parameters Affecting the Performance of the Subroutine

The performance of the subroutine is evaluated in terms of accuracy when compared to the results from the numerical Laplace inverse. In any finite element package, the user can modify some parameters affecting the convergence of the solution, especially if automatic time stepping is used. The initial time step, called DTIO in *Samcef*, can sometimes affect the solution. A parameter controls the level of convergence of the dynamic solution and modifies the time step accordingly. This parameter is called PRCO. The two parameters evaluating the convergence of the constitutive equation are PRCV and SREF and have been presented in Section 5.2.3. The last parameter affecting directly the solution in this case is MEMMAX. This parameter plays the same role as the *memmax* parameter defined in Chapter 4. It defines the length of the storage vector. Dynamic memory allocation is not possible yet within a

PRCO	1.0×10^{-2}
DTIO (*sec*)	1.0×10^{-3}
PRCV	5.0×10^{-2}
SREF (*Pa*)	4.0×10^{5}
MEMMAX	1000

Table 5.1: Reference Values for the Subroutine Parameters

user subroutine, and therefore, the size of the vector affects the computation efficiency of the subroutine if space is reserved uselessly. The influence of these five parameters, PRCO, DTIO, PRCV, SREF, and MEMMAX, is studied in the following sections. Table 5.1 gives the reference values chosen for the parameters. These values have been chosen to give convergent solutions for the two test equations defined in Section 5.3. They are not necessarily optimal. To evaluate the solution, it is compared to the numerical Laplace inverse obtained with 1024 data points of which 870 are kept. An interpolation is done on the data obtained with the numerical Laplace inverse to get information at the same points in time as the data obtained with the finite element analysis. The interpolation is done with *Matlab* using the *interp1* function which does a linear interpolation. A simple subtraction between the data obtained from the numerical Laplace inverse and the data obtained from the finite element package gives a measure of the error. The absolute value of this difference is taken. The maximum error obtained in the studied time interval is given and reflects the accuracy of the solution. The number of steps taken by the finite element package to cover the time interval is also given, as well as the CPU time, as a measure of the efficiency of the subroutine. In terms of CPU time, the numerical Laplace inverse takes between 0.28 *sec* and 0.35 *sec* for both cases outlined in Section 5.3. This is much less than the CPU time required for the finite element solutions, but the finite element analysis solves the constitutive equation for the six components of the constitutive equation, whereas the numerical Laplace inverse is done for a one-dimensional equation. The numerical Laplace inverse and the finite element solutions are both run

PRCO	Maximum Error (Pa)	Number of Steps	CPU Time (sec)
Case 1	*Sinusoidal Strain*		
1.0×10^{0}	2.3247×10^{4}	148	10.25
1.0×10^{-1}	2.3247×10^{4}	148	9.80
1.0×10^{-2}	2.3247×10^{4}	148	9.90
1.0×10^{-3}	2.0351×10^{4}	156	10.91
1.0×10^{-4}	2.1402×10^{4}	178	12.78
1.0×10^{-5}	Stopped at 0.1523599 sec		
Case 2	*Decaying Sinusoidal Strain*		
1.0×10^{-1}	1.7284×10^{4}	95	5.59
1.0×10^{-2}	1.7284×10^{4}	95	5.43
1.0×10^{-3}	1.2081×10^{4}	111	7.05
1.0×10^{-4}	1.1334×10^{4}	136	8.71
1.0×10^{-5}	Stopped at 0.1572033 sec		

Table 5.2: Effect of PRCO

on a Enterprise 450 with two processors UltraSPARC-II. In the case of the finite element package, automatic time stepping is used, but to limit the CPU time, the time step is not allowed to go below 1.0×10^{-4} sec. If conditions are such that a step below this value is required to obtain enough accuracy as defined by the PRCO and PRCV parameters, the computation is stopped. The computation would proceed without hindrance using smaller time steps, but the computation time would be prohibitive.

5.3.3 Effect of PRCO

Table 5.2 shows the effect of PRCO, the parameter controlling the accuracy of the dynamic solution. The maximum error is not following the same trend as PRCO. As a more stringent condition is put on PRCO, by decreasing its value, the maximum error is not necessarily decreasing. The automatic time stepping procedure will reduce the step when the error grows, correcting any tendency to diverge. Figure 5.2 shows the worst case obtained for Case 1 with an error of 2.3247×10^{4} Pa and a value for PRCO of 1.0×10^{-2}. The error is smaller near the peaks of the stress function and greater where the function nears zero. There is a small phase shift between the solution generated by the

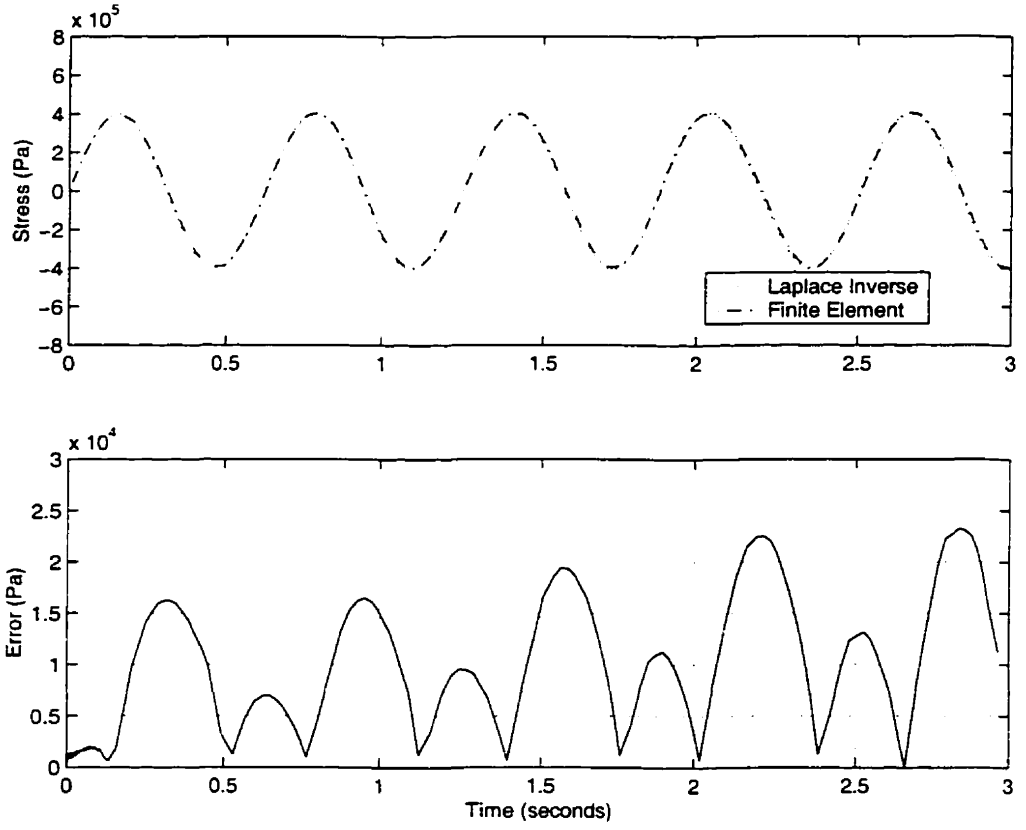

Figure 5.2: Case 1 with PRCO Equal to 1.0×10^{-2} and DTIO Equal to 1.0×10^{-3} *sec*

finite element analysis and the solution generated by the numerical Laplace inverse.

The main tendency to notice for both Case 1 and Case 2 is the increasing number of steps as a more stringent condition is put on PRCO, and the corresponding growing CPU time. As the value given to PRCO decreases, more steps are needed to meet the condition, and more steps mean more data stored needing to be retrieved for calculations, thus leading to a higher CPU time. There are slight variations in the CPU time for cases using the same number of time steps, but this is due to different number of iterations done by the software to converge to these steps.

For the test cases shown, if PRCO is increased above 1.0×10^{-2}, results remain almost the same. At this level, the condition on PRCO is not affecting the time step anymore, and any restriction on the time step is coming from PRCV. The condition on PRCV is such that the solution is still good.

For both Case 1 and Case 2, when PRCO reaches 1.0×10^{-5}, the step becomes very small to obtain that level of accuracy and the CPU time increases drastically. A requirement was put stopping the computation when the step would reach 1.0×10^{-4} *sec*. With Case 1, it happened after 0.1523599 *sec* of the solution and it took 43.12 *sec* of CPU time. With Case 2, it happened after 0.1572033 *sec* of the solution and it took 49.65 *sec* of CPU time. Figure 5.3 shows that solution for Case 1.

There is quite a large range of admissible values of PRCO for the two cases presented. If the value is too stringent, the requirement will force the time step to become too small, resulting in an excessive computing time. If the value is not stringent enough, some inaccuracies might appear, but overall, this parameter does not cause unwanted effect on the constitutive equation subroutine.

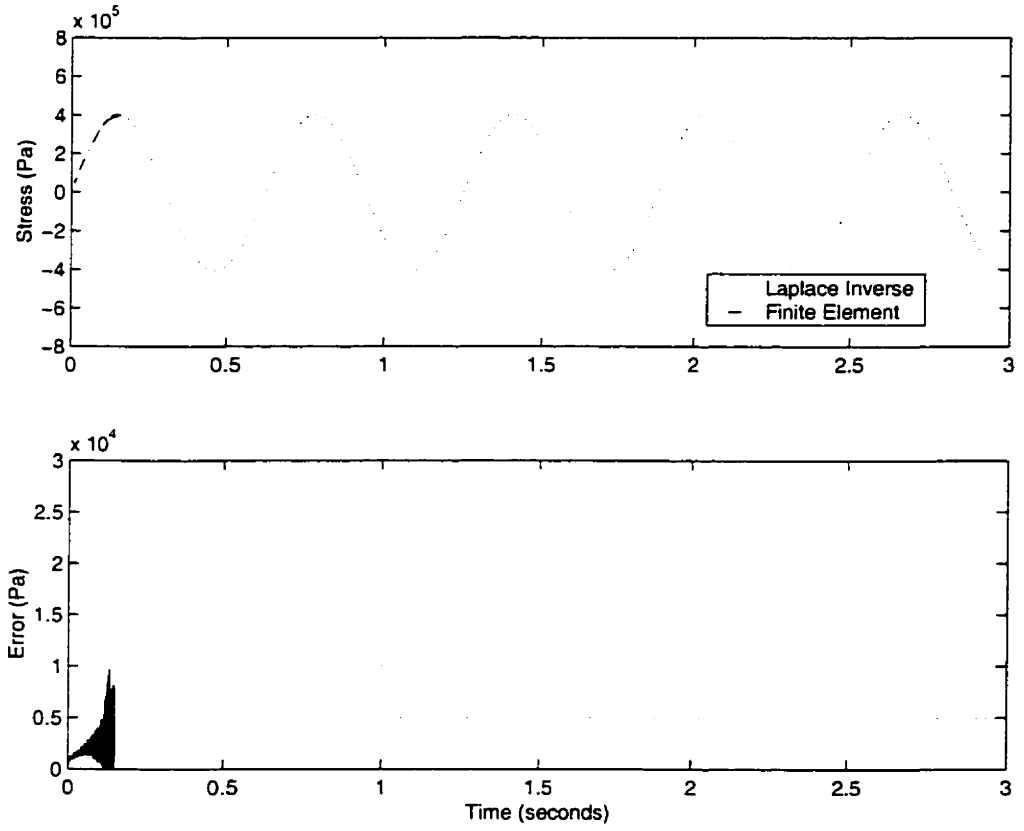

Figure 5.3: Case 1 with PRCO Equal to 1.0×10^{-5}

DTIO (*sec*)	Maximum Error (*Pa*)	Number of Steps	CPU Time (*sec*)
Case 1	*Sinusoidal Strain*		
1.0×10^{0}	The step is not small enough after being decreased three times		
5.0×10^{-1}	The step is not small enough after being decreased three times		
1.0×10^{-1}	2.1603×10^{4}	116	7.78
1.0×10^{-2}	1.9358×10^{4}	127	8.80
1.0×10^{-3}	2.3247×10^{4}	148	9.90
1.0×10^{-4}	2.0935×10^{4}	160	12.44
1.0×10^{-5}	2.3180×10^{4}	168	13.62
Case 2	*Decaying Sinusoidal Strain*		
1.0×10^{0}	2.6043×10^{4}	3	0.96
8.0×10^{-1}	2.0814×10^{4}	4	0.95
5.0×10^{-1}	2.1530×10^{4}	31	1.95
1.0×10^{-1}	1.2993×10^{4}	67	3.90
1.0×10^{-2}	2.0215×10^{4}	69	3.94
1.0×10^{-3}	1.7284×10^{4}	95	5.43
1.0×10^{-4}	1.3094×10^{4}	102	5.87
1.0×10^{-5}	1.2424×10^{4}	113	6.67

Table 5.3: Effect of DTIO

5.3.4 Effect of DTIO

DTIO specifies the initial time step. With the automatic time step procedure, the finite element package should reduce the time step if it is too large to satisfy PRCO and PRCV and it should increase the subsequent steps if the conditions set by PRCO and PRCV are easily met. Table 5.3 gives the performance of the subroutine as the value of DTIO changes. The parameters for the finite element computation include a parameter limiting the number of attempts at reducing the time step to three and this value is kept. The finite element package fails to start the computation due to this parameter in a few cases for which the initial step is too large. For the other cases, the time step is reduced to an acceptable level and the computation proceeds with a satisfying level of accuracy. When the initial time step is too small, the finite element package adds a few time steps to the solution until the time step reaches an acceptable size, and these added time steps translate into an increased CPU time.

An interesting phenomenon happens with Case 2 when the initial time

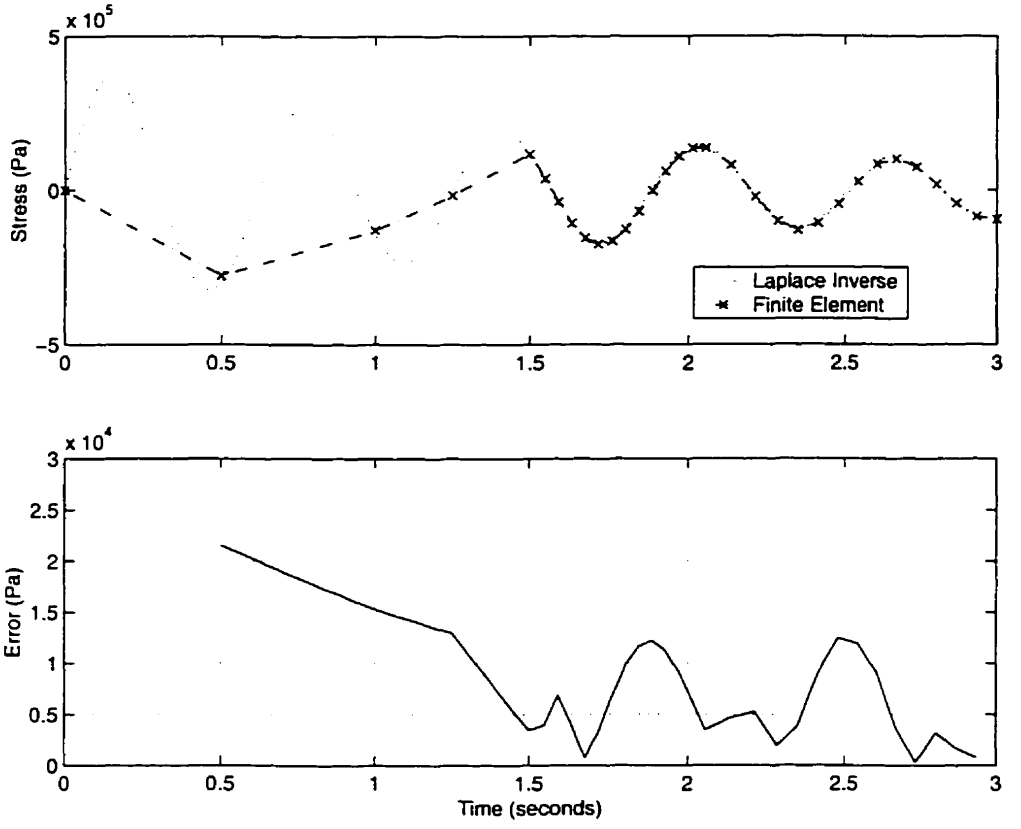

Figure 5.4: Case 2 with DTIO Equal to 5.0×10^{-1} *sec*

step is very large. In Figure 5.4, the steps are marked. The initial step is very large, but the algorithm is able to calculate a solution despite this. Of course, the first few data points do not describe the function accurately, but eventually, the finite element package reduces the time step drastically and starts to follow the function correctly. The interesting phenomenon is that few data points are stored for the computation of the fractional derivative, and yet, the subroutine still computes correctly. In Chapter 4, there were no automatic time stepping, and the ability to reduce the amount of stored data was limited, but in this case, because the finite element package adapts the time step when needed, less precision is necessary with the stored data.

The maximum error shown in Table 5.3 is not directly affected by the initial time step. It depends on the location of the individual data point and the step size taken in that area. For Case 2, with an initial time step of 1.0×10^{-4} *sec*, the solution is less precise at the end of the computing time,

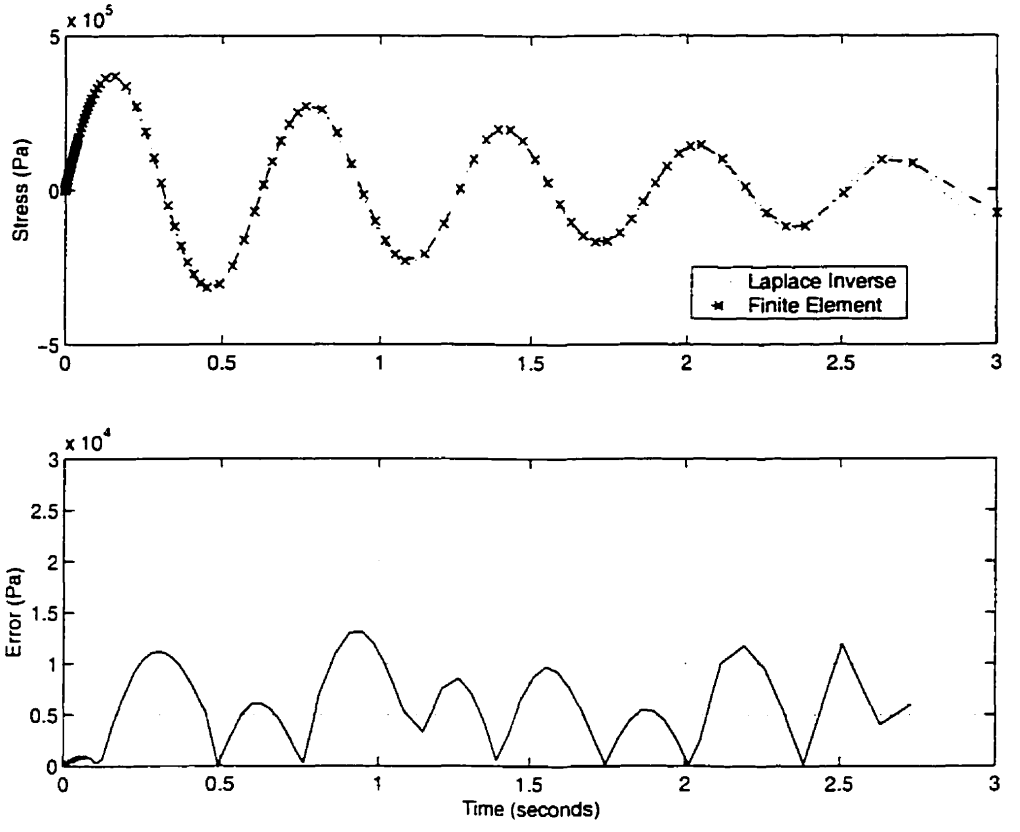

Figure 5.5: Case 2 with DTIO Equal to 1.0×10^{-4} *sec*

because too large steps are taken as shown in Figure 5.5. This does not happen with 1.0×10^{-3} *sec* or 1.0×10^{-5} *sec*. The automatic time stepping procedure can produce unexpected effects which are controlled by adjusting PRCO.

5.3.5 Effect of MEMMAX

To verify the amount of stored data which is necessary for good precision, MEMMAX is varied. Table 5.4 shows the effect of the variation of MEMMAX. The minimum step size allowable is fixed at 1.0×10^{-4} *sec*, and when the algorithm stops, it is because the step needed to keep a sufficient level of accuracy according to PRCO and PRCV would be smaller or equal to this value. For both cases, the algorithm cannot work efficiently with a storage vector equal to less than one third of the needed data to compute the whole solution. For Case 1, the computation is done in approximately 148 steps and it stops performing efficiently if the storage vector contains only 50 data

MEMMAX	Maximum Error (Pa)	Number of Steps	CPU Time (sec)
Case 1	*Sinusoidal Strain*		
700	2.3247×10^{4}	148	9.90
100	2.3247×10^{4}	147	7.22
60	2.4829×10^{4}	149	6.07
50	Stopped at 2.301079 *sec*		
Case 2	*Decaying Sinusoidal Strain*		
700	1.7284×10^{4}	95	5.43
60	2.1925×10^{4}	95	3.86
50	2.3120×10^{4}	95	3.55
40	2.5848×10^{4}	92	3.27
30	Stopped at 2.513808 *sec*		

Table 5.4: Effect of MEMMAX

points, which is approximately 34% of the total number of data points. For Case 2, the computation requires approximately 95 steps, but the finite element package is not efficient if the size of the storage vector is 30, which is about 33% of the total number of data points. In both cases when the algorithm fails to compute, it is right after a compression of data occurs. For Case 1, the algorithm stops after computing 125 steps, which is right after the fourth compression. The first compression occurs when 50 steps are computed. One step out of two is kept and 25 new steps can be stored. The following compressions occur at 75, 100, and 125 steps. At the 126^{th} step, the level of error brought by the fractional derivative requires a smaller step than 1.0×10^{-4} *sec*. For Case 2, the algorithm stops computing after 90 steps, compressions occuring at 30, 45, 60, 75, and 90 steps. However, for Case 2, even though the algorithm computes to the end of the simulation time with a storage vector of 40, the error is quite large. The simulation results show some divergence from the results obtained with the numerical Laplace inverse. Figure 5.6 shows the results for Case 2 with a storage vector of 40 data points.

As shown in Table 5.4, the computing time is shortened by using smaller storage vectors. Manipulating a large storage vector slows down the compu-

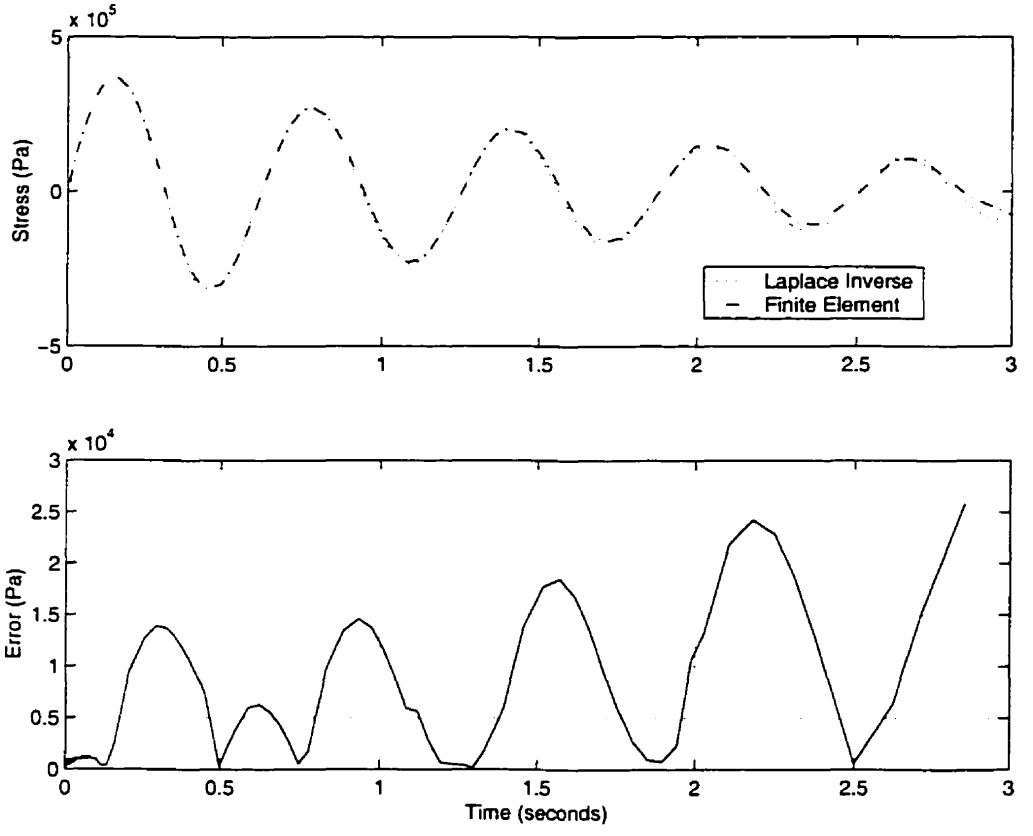

Figure 5.6: Case 2 with MEMMAX Equal to 40

SREF (Pa)	Maximum Error (Pa)	Number of Steps	CPU Time (sec)
Case 1	*Sinusoidal Strain*		
4.0×10^8	5.3380×10^4	44	3.11
4.0×10^7	5.3380×10^4	44	3.02
4.0×10^6	5.4132×10^4	46	3.20
4.0×10^5	2.3247×10^4	148	9.90
4.0×10^4	Stopped at 0.09272425 sec		
Case 2	*Decaying Sinusoidal Strain*		
4.0×10^8	2.7556×10^4	43	2.87
4.0×10^7	2.7556×10^4	43	2.92
4.0×10^6	2.6519×10^4	45	3.05
4.0×10^5	1.7284×10^4	95	5.43
4.0×10^4	Stopped at 0.1002489 sec		

Table 5.5: Effect of SREF

tation. Even if the total number of time steps is the same, a smaller storage vector significantly reduces the computing time. However, the error also grows if the storage vector is smaller than the total number of time steps. Ideally, the storage vector should not be large needlessly, but care is needed for a storage vector smaller than the total number of time steps.

5.3.6 Effect of SREF

The error on the constitutive equation is adimensionalized by dividing it either by the average stress, the sum of the six values of stress divided by six, or by a reference stress, SREF, as stated in Equation 5.33:

$$\text{local error} = \frac{\max_{elements}(\text{local stress error})}{\max\left(\text{SREF}, \max_{elements}(\text{average local stress})\right)} \tag{5.47}$$

Table 5.5 shows the effect of SREF. For both Case 1 and Case 2, the maximum amplitude of the stress is 4.0×10^5 Pa. This is the value chosen for SREF. If SREF is larger, the computed value of the error becomes quite small for the chosen PRCV and the criteria is ineffective. Figure 5.7 shows Case 1 with a value of SREF of 4.0×10^8 Pa. The finite element solution is not accurate.

If SREF is too small, the error is larger, and the step is constantly reduced to try to meet the criteria set by PRCV. This happens with SREF equal to

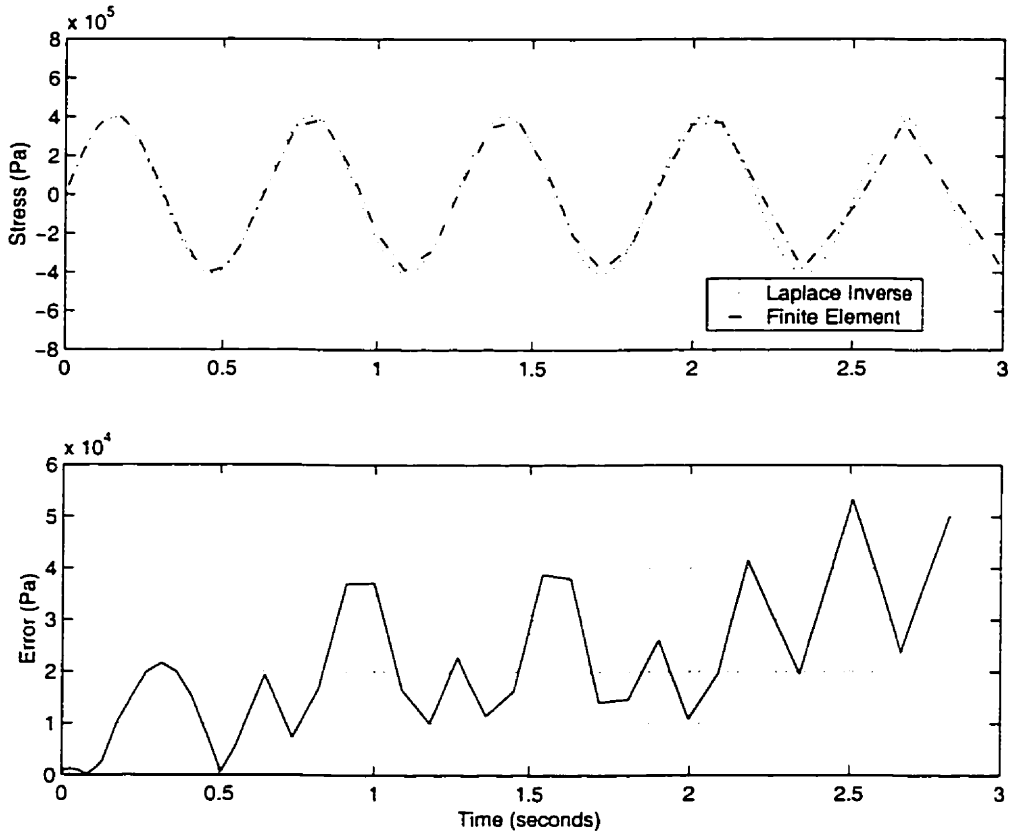

Figure 5.7: Case 1 with SREF Equal to 4.0×10^8 Pa

PRCV	Maximum Error (Pa)	Number of Steps	CPU Time (sec)
Case 1	*Sinusoidal Strain*		
5.0×10^{-1}	5.4132×10^{4}	46	3.08
1.0×10^{-1}	2.8776×10^{4}	82	6.81
5.0×10^{-2}	2.3247×10^{4}	148	9.90
1.0×10^{-2}	Stopped at 0.1464766 *sec*		
Case 2	*Decaying Sinusoidal Strain*		
5.0×10^{-1}	2.6519×10^{4}	45	2.50
1.0×10^{-1}	2.3143×10^{4}	61	4.11
5.0×10^{-2}	1.7284×10^{4}	95	5.43
1.0×10^{-2}	Stopped at 0.1423889 *sec*		

Table 5.6: Effect of PRCV

4.0×10^{4} Pa. Both simulations done with this value of SREF stop when the step becomes smaller or equal to 1.0×10^{-4} sec. For a given PRCO, there are not many admissible values of SREF which render the criteria effective without being too stringent.

5.3.7 Effect of PRCV

The parameter PRCV sets the admissible error on the computation of the constitutive equation. Table 5.6 shows the results. The effect of PRCV is similar to the effect of SREF. When the value is too high, the criteria is ineffective. The number of steps needed to complete the computation gets lower as the step is never reduced due to the error control on the computation of the constitutive equation. If PRCV is too small, the criteria is so stringent that the step is kept very small and the computation is not efficient anymore. For a given SREF, there are only few values of PRCV which give an efficient control of the error due to the constitutive equation computation. Clearly, SREF and PRCV must be chosen together.

5.3.8 Combined Effect of PRCV and SREF

The last two sections showed that for a given PRCV, there are only a few values of SREF producing good results, and vice versa. Starting with a value

PRCV	SREF (Pa)	Maximum Error (Pa)	Number of Steps	CPU Time (sec)
Case 1		*Sinusoidal Strain*		
5.0×10^{-3}	4.0×10^{6}	2.3247×10^{4}	148	10.18
5.0×10^{-2}	4.0×10^{5}	2.3247×10^{4}	148	9.90
5.0×10^{-1}	4.0×10^{4}	2.3733×10^{4}	147	10.34
5.0×10^{0}	4.0×10^{3}	3.6936×10^{4}	53	3.07
Case 2		*Decaying Sinusoidal Strain*		
5.0×10^{-3}	4.0×10^{6}	1.7284×10^{4}	95	5.27
5.0×10^{-2}	4.0×10^{5}	1.7284×10^{4}	95	5.43
5.0×10^{-1}	4.0×10^{4}	1.4894×10^{4}	91	5.29
5.0×10^{0}	4.0×10^{3}	1.8125×10^{4}	48	2.77

Table 5.7: Combined Effect of PRCV and SREF

of PRCV equal to 5.0×10^{-2} and a value of SREF equal to 4.0×10^{5} Pa, these values are decreased or increased by an order magnitude simultaneously to assess the limits of the acceptable choices for PRCV and SREF. Table 5.7 shows the results. Despite the fact that the ratio of PRCV to SREF remains the same, as SREF decreases, the error criteria has less and less impact. The definition of the error is stated again:

$$\text{local error} = \frac{\max_{elements}(\text{local stress error})}{\max\left(\text{SREF}, \max_{elements}(\text{average local stress})\right)} \tag{5.48}$$

The numerator is divided either by SREF or by the average local stress. If SREF is too low, very often the numerator will be divided by the average local stress which is higher. Since PRCV has been increased to compensate the decreasing value of SREF, the local error criteria becomes ineffective. This happens in both cases for a value of SREF equal to 4.0×10^{3} Pa. For this value of SREF, the maximum error increases significantly. At this point, only PRCO affects the step size and there is no control on the precision of the computation of the constitutive equation. If SREF is higher than the maximum stress, than the numerator is always divided by SREF, and decreasing accordingly PRCV gives the same results as lower values of SREF. This is seen for both cases with values of SREF equal to 4.0×10^{6} Pa. The results obtained are the same as with a value of SREF of 4.0×10^{5} Pa.

The two preceding sections showed that when one of the two parameters, PRCV or SREF, is fixed, there are few values for the other parameters which will enable the error criteria to be effective. However, even when the right ratio between PRCV and SREF is kept, if SREF is below the average local stress, the criteria is not as stringent as expected. It is better to choose SREF as the highest possible local stress, and then find an appropriate value of PRCV to ensure some control on the precision of the computation of the constitutive equation.

5.4 Conclusion

The implementation of the fractional constitutive equation within a commercial finite element package was described. The expression for the Jacobian of the constitutive equation was given. Two test cases were defined. The first one is a cubic element of PMMA submitted to a sinusoidal strain, and the second test case is the same cubic element of PMMA submitted to a decaying sinusoidal strain.

The influence of various computing parameters was studied, and the algorithm gave reliable, precise results in most cases. The dynamic precision, PRCO and the initial step, DTIO, affected the simulations in the same manner as they affect any time domain simulations. If PRCO is too small, the time step becomes prohibitively small. If DTIO is too large, the automatic time stepping procedure cannot reduce the step sufficiently to ensure a good precision. The size of the storage vector affected the simulation time and the accuracy. If MEMMAX, the size of the storage vector is larger than the total number of time steps in the simulation, than the CPU time increases with no gain in precision. If MEMMAX is much smaller than the total number of time steps, the precision decreases. When MEMMAX was chosen to be equal to approximately one third of the total number of data points, the algorithm had to compress the data too often and the step became very small to keep a reasonable level of accuracy. A certain level of precision in the history of the

time derivative of the strain is necessary for the fractional Voigt-Kelvin model to be effective. The most difficult parameters to select were the ones pertaining to the estimation of the error due to the computation of the constitutive equation. The two parameters, SREF and PRCV, are linked and cannot be chosen independently. Some a priori knowledge of the maximum stress seen in the given problem proved useful.

The goal of this chapter was to establish the accuracy of the finite element implementation of the fractional Voigt-Kelvin model. The results were compared to results from a numerical Laplace inverse. The algorithm for the fractional Voigt-Kelvin model did not require very small time steps. The number of time steps used was just enough to ensure a smooth representation of a sinusoidal signal. In the next chapter, some experimental test cases will be studied and the performance of the fractional Voigt-Kelvin model will also be compared to the performance of other classical constitutive equations.

Chapter 6

Examples of Experimental Slewing Beams

6.1 Introduction

In the preceding chapters, parameters have been found for various viscoelastic models and two materials, PMMA and 3M ISD 112. In this chapter, slewing beams made of PMMA or steel covered by constrained viscoelastic layers of 3M ISD 112 are studied. The experimental results are compared to the simulation results produced by the viscoelastic models to find which ones are efficient and which ones are accurate.

6.2 Steel Beam Covered by a Constrained Viscoelastic Layer

The first example is a steel beam completely covered by a constrained viscoelastic layer. The materials are those usually considered for use in space. The beam is simply slewing in the horizontal plane and a correlation between the recorded experimental data and the simulation results is attempted. The experiments were done in collaboration with Slanik [48] and Tremblay [54] and are both part of their respective master thesis, Slanik concentrating on the time domain finite element simulation with Prony series (Slanik et al [9, 11]), and Tremblay on the use of equivalent homogeneous parameters for the three layer beam (Piedboeuf et al [55]).

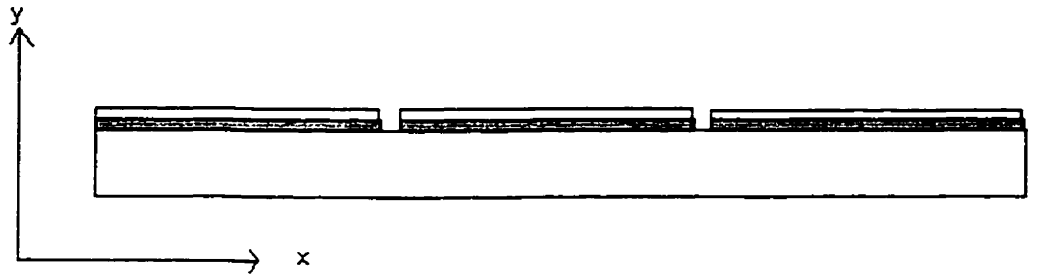

Figure 6.1: Beam with a Constrained Viscoelastic Layer

	Young's Modulus (GPa)	Poisson's Ratio	Density (kg/m^3)
Steel	210	0.3	7962.5
Aluminium	70	0.3	2710

Table 6.1: Mechanical Properties of Steel and Aluminium

6.2.1 Experimental Set-Up

The set-up is fully described in Slanik [48]. A slender steel beam is covered by a viscoelastic layer of 3M ISD 112 constrained by a thin layer of aluminium. The beam has a length of 600 mm, but the samples of 3M ISD 112 provided by 3M are shorter. Three strips of 3M ISD 112 of approximately 200 mm are used with spaces as small as is physically possible between them. Figure 6.1 illustrates the configuration of the beam. The steel beam has a width of 19.05 mm and an average thickness of 1.603 mm. This thickness shows a variation of approximately $\pm$ 2.5% along the length. This variation in thickness is not modelled and could affect the simulation results. The viscoelastic layer has a nominal thickness of 0.127 mm and the aluminium constraining layer, a nominal thickness of 0.254 mm. The mechanical properties of the viscoelastic layer 3M ISD 112 were investigated in Chapter 3, and the properties of the steel and the aluminium used are given in Table 6.1. Young's modulus and the density of steel are measured, whereas the aluminium is simply described as a standard soft aluminium by 3M, and usual values are taken. The density of the viscoelastic layer is given by 3M as 970 kg/m^3.

The beam is slewed in the $x - y$ plane, which is horizontal, as shown in

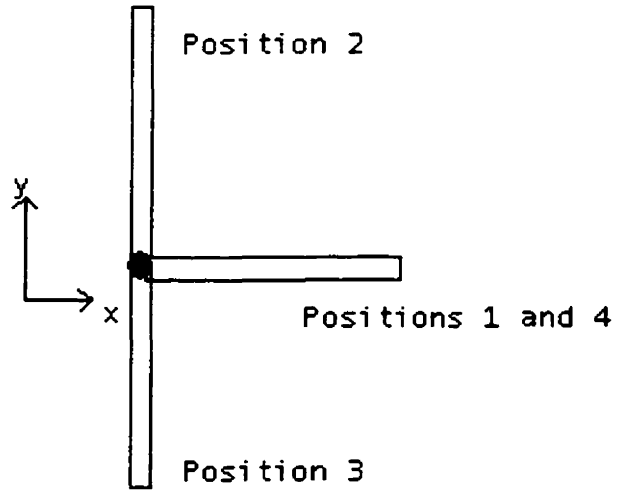

Figure 6.2: Slewing Input

Figure 6.2. The beam is first lying along the x axis with negligible deflection. It is then slewed 90° counterclockwise, then 180° clockwise, and finally, 90° counterclockwise. The resulting position of the base with respect to time is shown in Figure 6.3. Since the beam is rotating in the $x-y$ plane and bending is about the strong axis, gravity effects are neglected.

A strain gauge is used on the steel beam near the base to record the deformation. The gauge has a resistance of 350 Ω and a grid area of 8.08 mm^2. The excitation voltage is 9.0 V and an amplification factor of 1000 is used. The strain gauge is mounted in a quarter-bridge configuration with a gauge factor of 2.085.

6.2.2 Finite Element Model

The beam is modelled with plane strain elements lying in the $x-y$ plane as illustrated in Figure 6.1. Along the length, one layer of 99 elements is used to model the steel beam assuming an elastic behaviour. Another 99 plane strain elements model the viscoelastic layer. A last layer of 99 elements models the aluminium layer, also assuming an elastic behaviour. The beam length is 600 mm, therefore each element has a length of 6.061 mm. The thickness of the viscoelastic layer, which is the thinnest element, is 0.127 mm. The aspect ratio of the viscoelastic element is 48 and the results from the elements

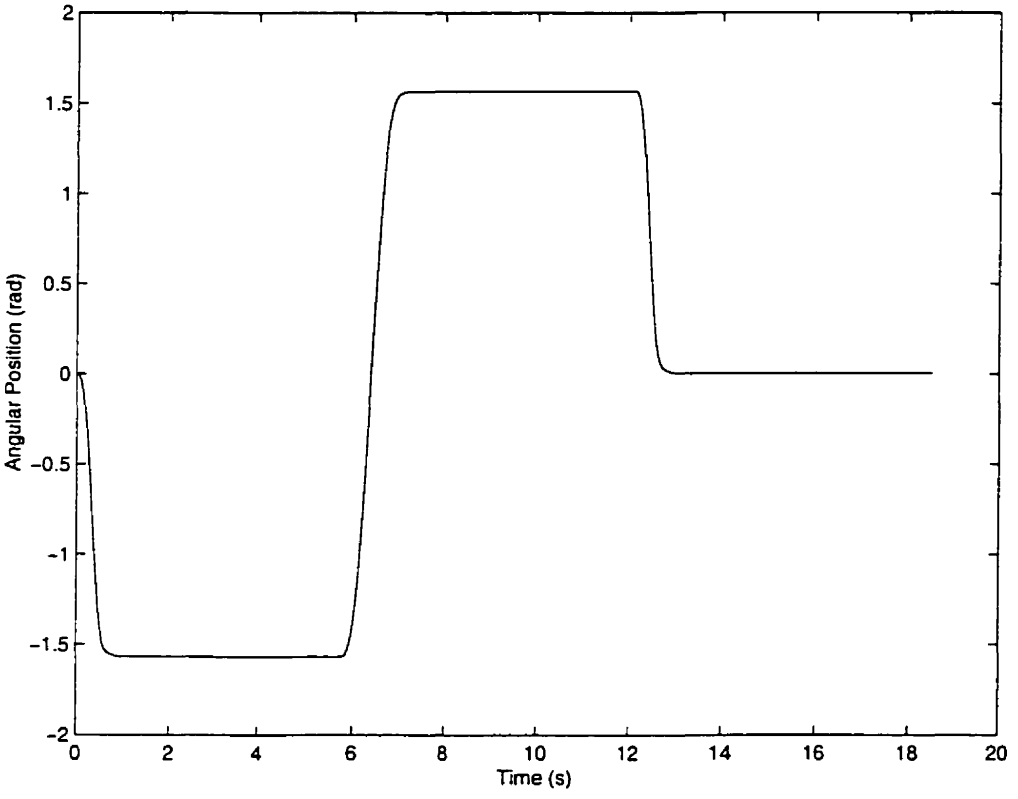

Figure 6.3: Position of the Base with Respect to Time

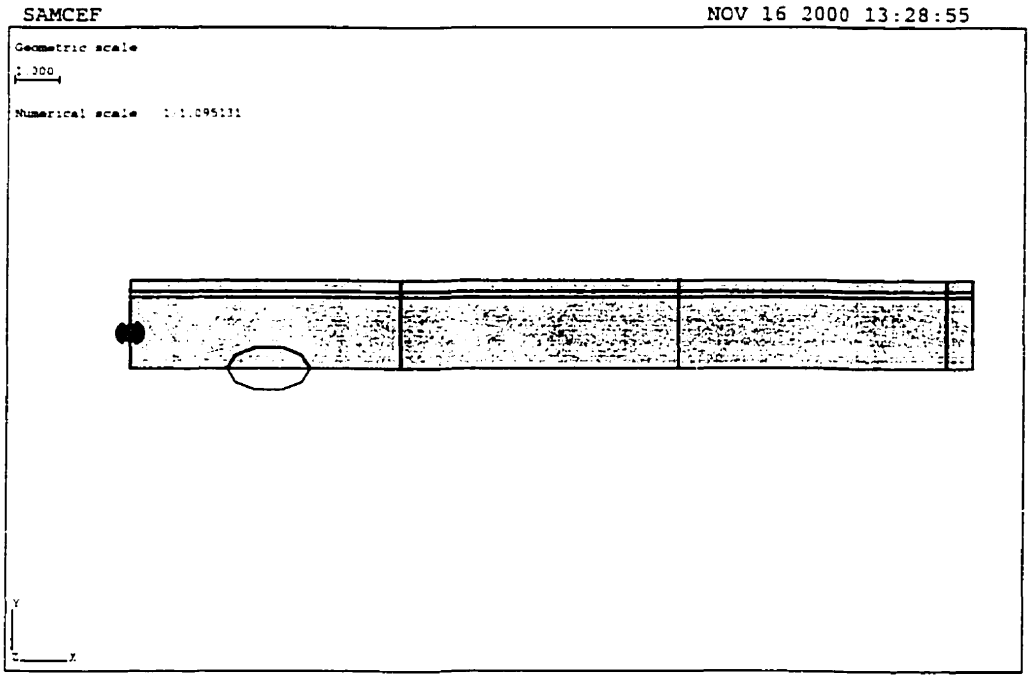

Figure 6.4: Portion of the Finite Element Model of the Beam with a Constrained Viscoelastic Layer

are considered accurate by *Samcef* up to an aspect ratio of 50. Figure 6.4 shows the finite element model. The strain of the steel beam near the base is measured using the element *DIST*, which is a distance indicator. It gives the length of the line marked by an ellipse on Figure 6.4. The initial length of the element is subtracted from that value. The difference is divided by the initial length to obtain a measure of the strain.

The constrained viscoelastic layer is not continuous: three sections of $200mm$ are placed end to end to cover the whole beam. These discontinuities in the layer are simply modelled by superposing supplementary nodes where one strip ends to create the beginning of the new strip, for a total of five nodes needed to model one slit. No distance between the strips is modelled. Figure 6.5 illustrates the model used for the slits. A certain number of parameters proper to the finite element package were described in Chapter 5. The values chosen for all simulations are listed in Table 6.2.

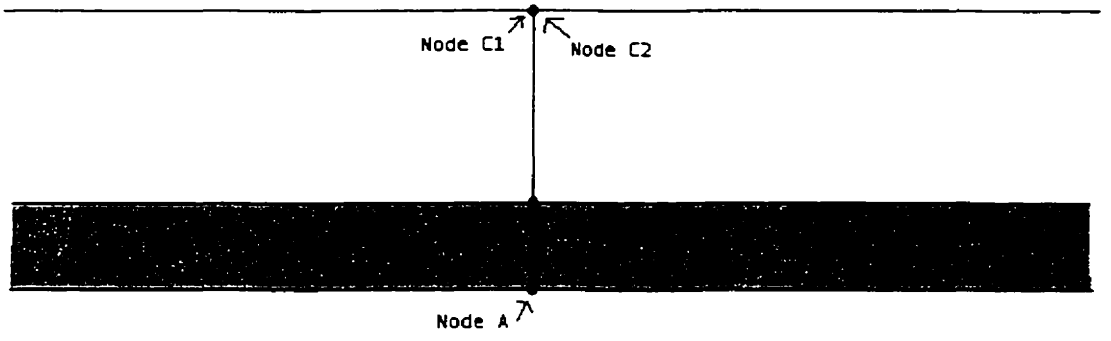

Figure 6.5: Modelling of the Slits

PRCV	0.1
PRCO	0.01
SREF	$5 \times 10^6\ Pa$
DTIO	$0.001\ s$

Table 6.2: Finite Element Simulation Parameters

6.2.3 Simulated Beam Response

The parameters found in Chapter 3 for the viscoelastic behaviour of 3M ISD 112 are used to simulate the beam response. Each model is compared to the experimental response. The simulated and experimental responses is evaluated for both the natural frequency exhibited and the logarithmic decrement. The logarithmic decrement is a measure of damping and it compares the respective amplitudes of two maxima. If M cycles separate the two maxima A_i and A_{i+M}, then the logarithmic decrement is:

$$\delta = \frac{1}{M} \ln \frac{A_i}{A_{i+M}} \tag{6.1}$$

In Figure 6.3. the angular position of the base changes rapidly three times. After each slew. the frequency and the logarithmic decrement is measured and compared to the experimental values. This is done rather than taking the average of the three natural frequencies and the three logarithmic decrements, since the amplitude values after the second slew are very small and more difficult to measure. A model can do well in predicting what is happening after the first and third slew, but not so well on what is happening after the second slew. Taking the average prevents the reader from fairly assessing the effectiveness of each model.

The Voigt-Kelvin Model

Figure 6.6 shows the experimental response compared to the simulated response with a Voigt-Kelvin model using the parameters found in Chapter 3. The first part of the figure is the response for the whole time interval, whereas the three other parts zoom on the response for intervals of six seconds. Table 6.3 compares the natural frequencies and logarithmic decrements after each slewing movements. The beam response obtained with a Voigt-Kelvin model shows on average a difference of 4.3% for the natural frequency, and a difference of 88% for the logarithmic decrement. Figure 3.15 showed the magnitude and the phase of the shear modulus of 3M ISD 112 to be underestimated by the

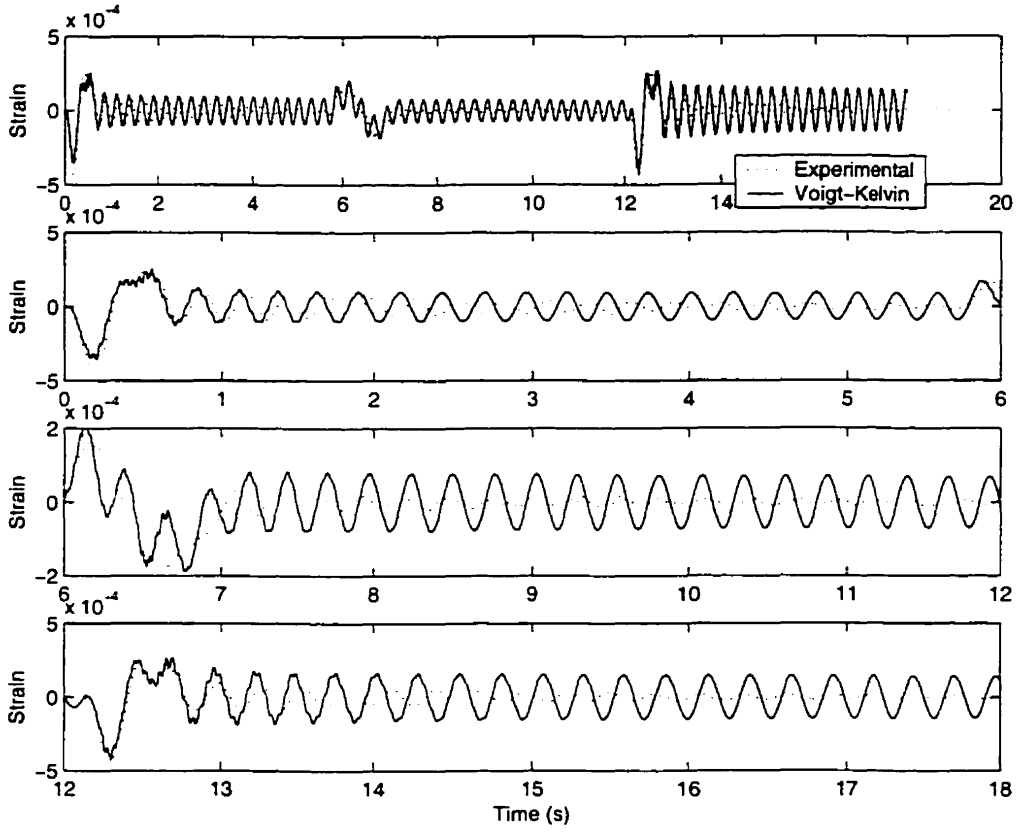

Figure 6.6: Beam Response Using a Voigt-Kelvin Model

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.77	3.64	3.4
Logarithmic Decrement	7.8×10^{-3}	1.1×10^{-1}	92
Second Slew			
Natural Frequency (Hz)	3.78	3.67	3.0
Logarithmic Decrement	9.7×10^{-3}	6.9×10^{-2}	86
Third Slew			
Natural Frequency (Hz)	3.79	3.56	6.5
Logarithmic Decrement	1.3×10^{-2}	9.9×10^{-2}	87

Table 6.3: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Voigt-Kelvin Model

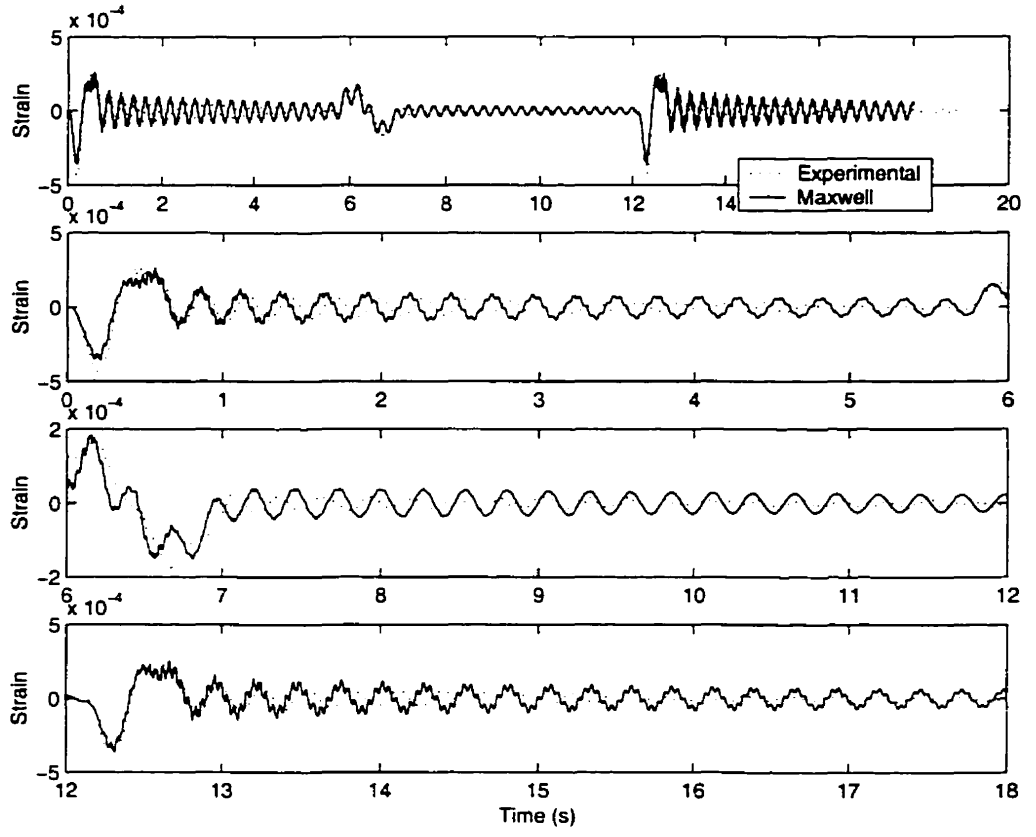

Figure 6.7: Beam Response Using a Maxwell Model

Voigt-Kelvin model. In this case, the natural frequency is driven by the properties of the steel beam, but the damping in the model is uniquely introduced through the material properties of the viscoelastic layer. The underestimation of the phase by the Voigt-Kelvin model around 3 Hz shows in the underestimated damping of the simulation. The simulation is accomplished in 4190 steps and required 0.683 h of CPU time.

The Maxwell Model

Figure 6.7 shows the experimental response compared to the simulated response with a Maxwell model. Table 6.4 compares the natural frequencies and logarithmic decrements after each slewing movements. The beam response obtained with a Maxwell model shows on average a difference of 4.0% for the natural frequency, and a difference of 59% for the logarithmic decrement. Figure 3.16 showed the magnitude of the shear modulus of 3M ISD 112 to be

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.75	3.64	3.0
Logarithmic Decrement	1.8×10^{-2}	1.1×10^{-1}	84
Second Slew			
Natural Frequency (Hz)	3.77	3.67	2.7
Logarithmic Decrement	4.2×10^{-2}	6.9×10^{-2}	39
Third Slew			
Natural Frequency (Hz)	3.78	3.56	6.2
Logarithmic Decrement	4.5×10^{-2}	9.9×10^{-2}	55

Table 6.4: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Maxwell Model

underestimated by the Maxwell model. However, the phase in the vicinity of 3 Hz, the natural frequency in this case, is very close to the manufacturer data. This good prediction of the phase accounts for the improvements in the simulation results over the ones obtained with a Voigt-Kelvin model. However, the results still exhibit discrepancies due to the underestimation of the magnitude of the shear modulus. The simulation is accomplished in 5543 steps and required 0.880 h of CPU time.

The Zener Model

Figure 6.8 shows the experimental response compared to the simulated response with a Zener model. Table 6.5 compares the natural frequencies and logarithmic decrements after each slewing movements. The beam response obtained with a Zener model shows on average, a difference of 3.8% for the natural frequency, and a difference of 88% for the logarithmic decrement. Figure 3.17 showed both the magnitude and the phase of the shear modulus of 3M ISD 112 to be underestimated by the Zener model. This results in an underestimation of the damping level. The simulation was accomplished in 5043 steps and required 0.867 h of CPU time.

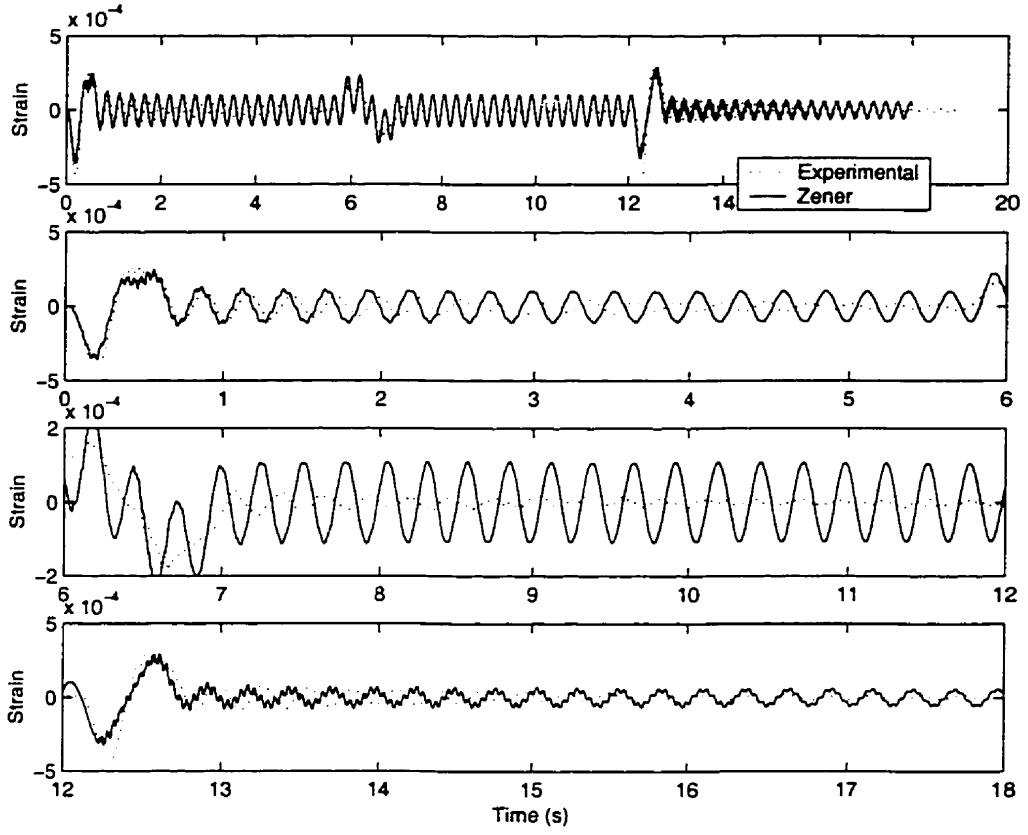

Figure 6.8: Beam Response Using a Zener Model

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.77	3.64	3.6
Logarithmic Decrement	8.3×10^{-3}	1.1×10^{-1}	92
Second Slew			
Natural Frequency (Hz)	3.74	3.67	1.9
Logarithmic Decrement	6.1×10^{-3}	6.9×10^{-2}	91
Third Slew			
Natural Frequency (Hz)	3.77	3.56	5.9
Logarithmic Decrement	1.9×10^{-2}	9.9×10^{-2}	81

Table 6.5: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Zener Model

Prony Series

The first three models presented, Voigt-Kelvin, Maxwell, and Zener, are all characterized by large differences between the model and the manufacturer data. In contrast, the Prony series show a better fit to the manufacturer data, even for the one-term Prony series as shown in Figure 3.18. Between 3 Hz and 4 Hz, slight differences appear in the predicted magnitude of the shear modulus for the four models, and somewhat larger differences in the phase of the shear modulus. However, as shown in Tables 6.6 to 6.9, the differences in the solutions are small. All the solutions overpredict the natural frequency, which is due to the finite element model rather than the viscoelastic model as discussed in Section 6.2.1.

The damping level is either overpredicted or underpredicted by the different models, but the prediction is closer to experimental results with a Prony series of four terms or more. Solutions for Prony series between five and seven terms are shown in Appendix B.

An interesting fact is the lack of obvious differences between the solutions using a constant Poisson's ratio or a constant bulk modulus. As explained in Section 2.5, at this low frequency and for this nearly incompressible material, either assumptions give similar values of Young's modulus.

Figure 6.9 shows the worst case among the first four Prony series against the best case when a constant Poisson's ratio is assumed. The worst case is the two-term Prony series with an appreciable level of overestimation of the damping, whereas the best case is the four-term Prony series with small differences between the predicted and the experimental logarithmic decrements. Despite this, both solutions are very close and differences are not obvious on the graph. Clearly, for this type of problem for which the viscoelastic material is only one component of the structure, a simple but realistic model such as a one-term Prony series is sufficient.

The number of steps and CPU time is similar for all solutions as shown in Table 6.10. Increasing the order of the Prony series does not increase the CPU

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.80	4.4	3.80	4.4
Log. Decrement	1.1×10^{-1}	8.0×10^{-2}	27	8.0×10^{-2}	27
Second Slew					
Natural Freq. (Hz)	3.67	3.83	4.4	3.76	2.5
Log. Decrement	6.9×10^{-2}	1.0×10^{-1}	45	1.1×10^{-1}	59
Third Slew					
Natural Freq. (Hz)	3.56	3.79	6.5	3.80	6.7
Log. Decrement	9.9×10^{-2}	7.7×10^{-2}	22	7.6×10^{-2}	23

Table 6.6: Natural Frequencies and Logarithmic Decrements of the Simulation Using a One Term Prony Series

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.88	6.6	3.89	6.9
Log. Decrement	1.1×10^{-1}	1.5×10^{-1}	36	1.5×10^{-1}	36
Second Slew					
Natural Freq. (Hz)	3.67	3.77	2.7	3.88	5.7
Log. Decrement	6.9×10^{-2}	1.4×10^{-1}	103	1.2×10^{-1}	74
Third Slew					
Natural Freq. (Hz)	3.56	3.87	8.7	3.87	8.7
Log. Decrement	9.9×10^{-2}	1.5×10^{-1}	52	1.5×10^{-1}	52

Table 6.7: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Two Term Prony Series

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.86	6.0	3.87	6.3
Log. Decrement	1.1×10^{-1}	8.0×10^{-2}	27	8.0×10^{-2}	27
Second Slew					
Natural Freq. (Hz)	3.67	3.84	4.6	3.82	4.1
Log. Decrement	6.9×10^{-2}	1.1×10^{-1}	59	1.1×10^{-1}	59
Third Slew					
Natural Freq. (Hz)	3.56	3.86	8.4	3.87	8.7
Log. Decrement	9.9×10^{-2}	8.0×10^{-2}	19	8.1×10^{-2}	18

Table 6.8: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Three Term Prony Series

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.87	6.3	3.86	6.0
Log. Decrement	1.1×10^{-1}	9.6×10^{-2}	13	9.5×10^{-2}	14
Second Slew					
Natural Freq. (Hz)	3.67	3.82	4.1	3.85	4.9
Log. Decrement	6.9×10^{-2}	6.7×10^{-2}	2.3	5.6×10^{-2}	19
Third Slew					
Natural Freq. (Hz)	3.56	3.84	7.9	3.84	7.9
Log. Decrement	9.9×10^{-2}	9.2×10^{-2}	7.1	9.2×10^{-2}	7.1

Table 6.9: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Four Term Prony Series

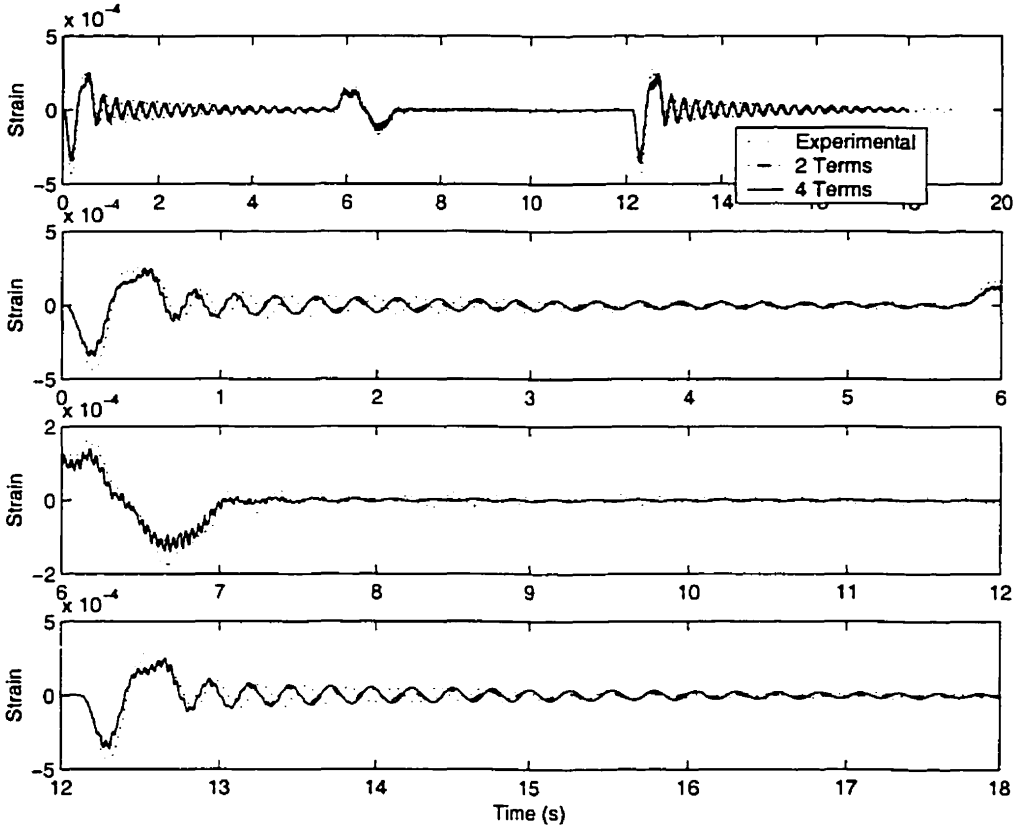

Figure 6.9: The Solution for a Two-Term Prony Series Compared to a Four-Term Prony Series

Prony Series	Constant Poisson's Ratio		Constant Bulk Modulus	
	Number of Time Steps	CPU Time (h)	Number of Time Steps	CPU Time (h)
1 Term	3074	0.5095	2989	0.5084
2 Terms	3247	0.5244	3206	0.5295
3 Terms	3046	0.5138	3176	0.5412
4 Terms	3130	0.5200	3085	0.5181
5 Terms	3023	0.5152	2954	0.5078
6 Terms	3073	0.4953	3123	0.5444
7 Terms	3074	0.5013	3155	0.5555

Table 6.10: Number of Time Steps and CPU Time for the Simulations Using Prony Series

time. The variations from solution to solution seem due to the system rather than the complexity of the Prony series. There is no obvious difference between the solutions using a constant Poisson's ratio and the ones using a constant bulk modulus. However, on average, the CPU time for the solutions using a constant Poisson's ratio is 0.5113 h, whereas it is 0.5293 h for the solutions using a constant bulk modulus. This is not necessarily significant since for the same number of terms, the solution with a constant bulk modulus does not always require more CPU time than the solution with a constant Poisson's ratio.

The Fractional Voigt-Kelvin Model

Figure 6.10 shows the experimental response compared to the simulated response with a fractional Voigt-Kelvin model. Table 6.11 compares the natural frequencies and logarithmic decrements after each slewing movements. The beam response obtained with a fractional Voigt-Kelvin model overpredict the natural frequency in a similar fashion to the Prony series. As for the Prony series, the logarithmic decrement values are very close to the experimental ones except for the second slew for which the damping is overestimated. The simulation is accomplished in 3298 steps and required 45.567 h of CPU time. In Chapter 5, the computation time was improved by reducing the value of

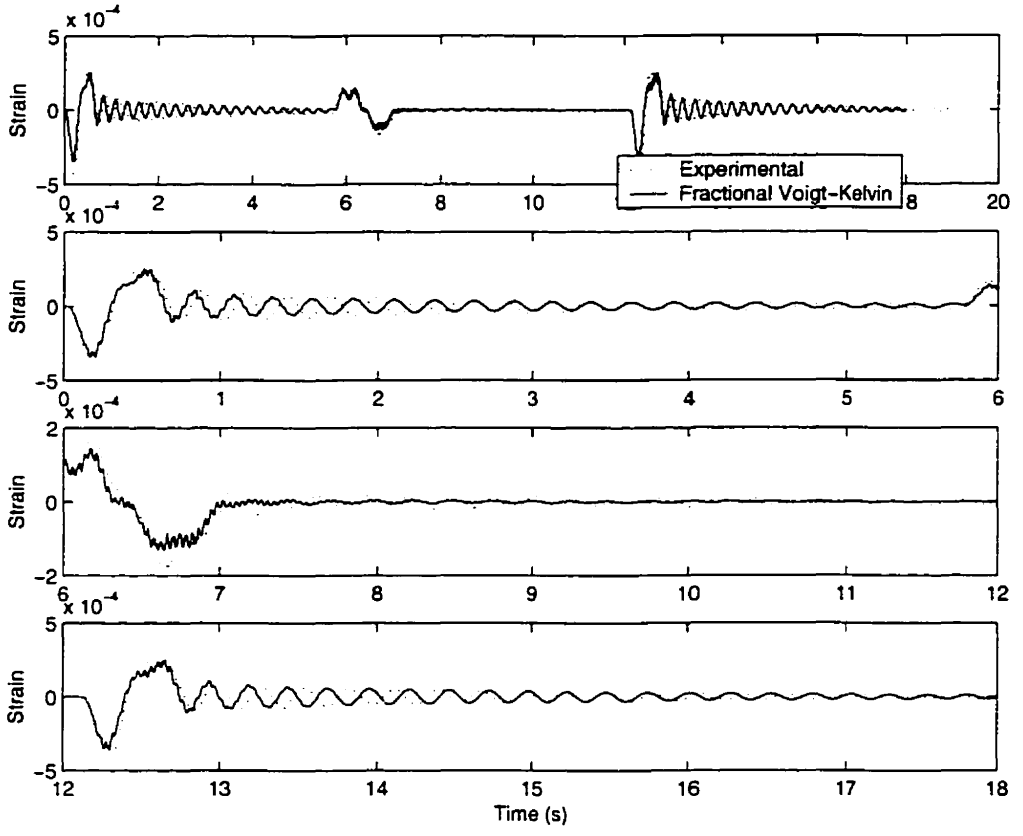

Figure 6.10: Beam Response Using a Fractional Voigt-Kelvin Model

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.90	3.64	7.1
Logarithmic Decrement	9.5×10^{-2}	1.1×10^{-1}	14
Second Slew			
Natural Frequency (Hz)	3.97	3.67	8.2
Logarithmic Decrement	1.4×10^{-1}	6.9×10^{-2}	103
Third Slew			
Natural Frequency (Hz)	3.88	3.56	9.0
Logarithmic Decrement	9.1×10^{-2}	9.9×10^{-2}	8.1

Table 6.11: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Fractional Voigt-Kelvin Model

MEMMAX	Number of Time Steps	CPU Time (h)
5000	3298	48.477
3300	3298	45.567
1700	3236	27.996
1100	3273	19.982
660	3291	12.705

Table 6.12: Number of Time Steps and CPU Time for the Simulations According to the Value of MEMMAX

MEMMAX, the amount of data points being stored for use in calculations. Table 6.12 gives the number of steps taken to simulate the time interval and the CPU time for simulations using different values of MEMMAX. For a value of MEMMAX of 660, the CPU time is reduced to 12.705 h without any loss of precision. Tables B.4 to B.6 in Appendix B show the differences between these cases and the one with a MEMMAX of 3300 to be negligible. Figure 6.11 compares the solutions for a MEMMAX of 3300 and one of 660. The differences are very small. The value of MEMMAX can not be reduced indefinitely, however. Figure 6.12 shows the simulation for a value of MEMMAX of 300. The results are very noisy and do not show a proper level of damping. Moreover, the CPU time is 11.622 h, only a small improvement over the 12.705 h obtained with a MEMMAX of 660.

6.2.4 Conclusion

As expected, the Voigt-Kelvin, Maxwell, and Zener models are not appropriate to represent the damping level obtained with a viscoelastic layer. On the other hand, the Prony series and the fractional Voigt-Kelvin model are accurate. A four-term Prony series predicted the logarithmic decrements within a few percents of the experimental values in approximately half an hour of CPU time. This is about the CPU time required to run a simulation of this beam with only elastic properties. At this frequency, there is no significant difference between using a constant Poisson's ratio or a constant bulk modulus as an

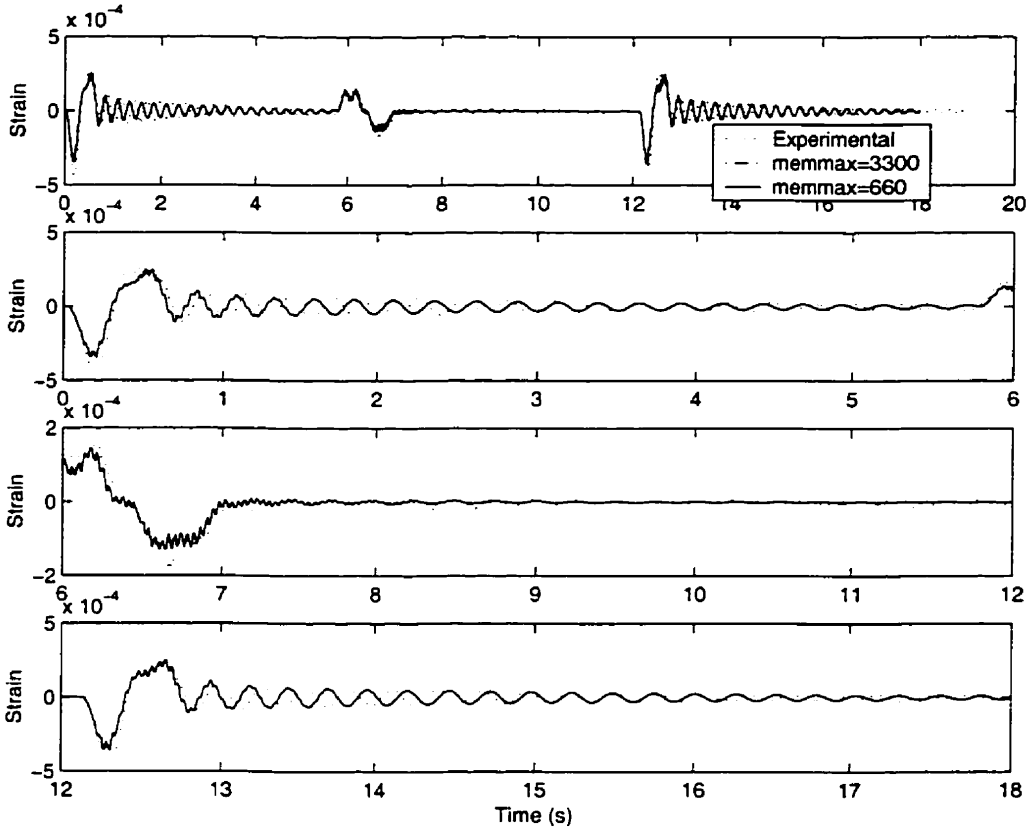

Figure 6.11: Beam Response Using a Fractional Voigt-Kelvin Model with a MEMMAX of 3300 and 660

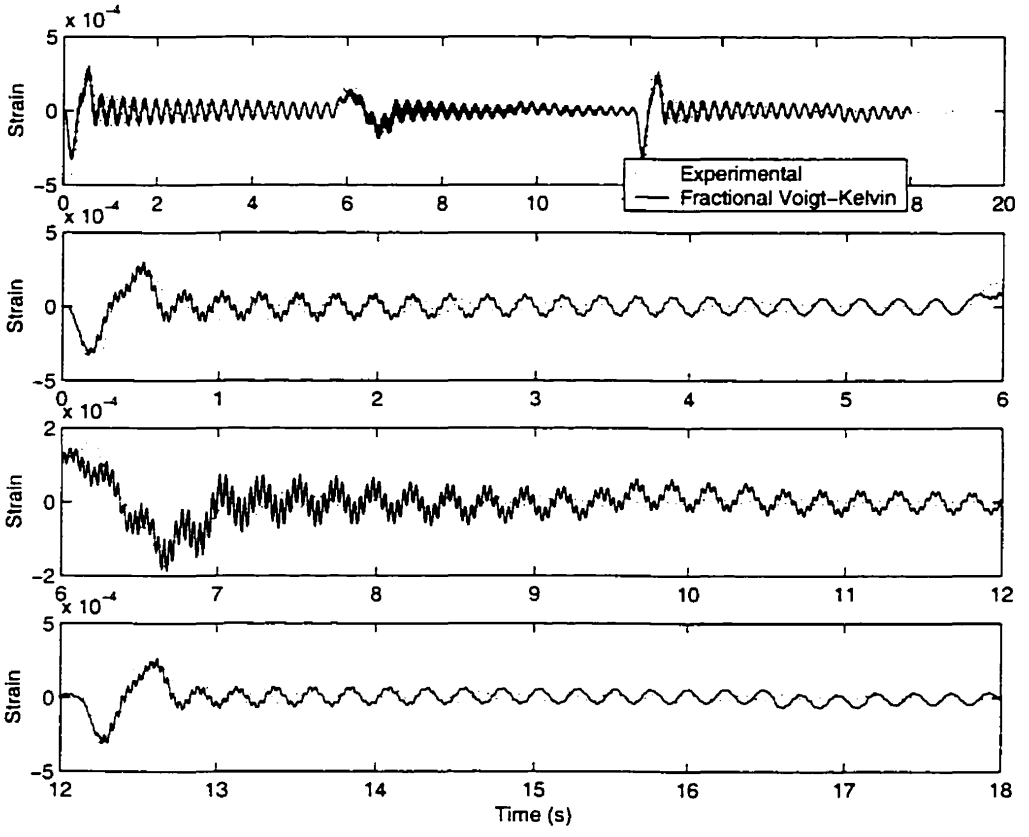

Figure 6.12: Beam Response Using a Fractional Voigt-Kelvin Model with a MEMMAX of 300

assumption to build the Prony series.

The fractional Voigt-Kelvin model gives similar results to the four-term Prony series, but it is much less efficient in terms of CPU time. Even by storing as few values as possible, the simulation requires 12.705 h, which is much more than the half hour needed for the Prony series. The fractional Voigt-Kelvin model still retains some use for cases in which little data is available for parameter identification. A four-term Prony series requires nine parameters to be identified, whereas the fractional Voigt-Kelvin model needs only three.

The Prony series required on average 3097 steps and the fractional Voigt-Kelvin model on average, for different values of MEMMAX, 3279 steps, a 5.9% increase over the Prony series. This is very reasonable and shows the approach used in this thesis to solve numerically the fractional derivative does not require very small time steps. The difference in CPU time between the Prony series and the fractional Voigt-Kelvin model is due to the hereditary nature of the fractional model, and not to the number of time steps taken.

6.3 Homogeneous PMMA Beams

In the preceding example, the viscoelastic material is just one component of the structure. In this section, homogeneous polymer beams are examined. The beams are made of polymethylmethacrylate (PMMA). The material is not perfectly isotropic [56] or perfectly linear [57, 58], but it is considered as such as an approximation. Another important difference with the preceding example is the level of damping. The viscoelastic layer, 3M ISD 112, is a high damping material with a modulus phase describing a bell and reaching a maximum of 45°. In contrast, PMMA is also characterized by a modulus phase describing a bell, but the maximum is 4°. In Chapter 3, the fractional model with only one derivative of the strain is shown to be less representative of the PMMA behaviour than the behaviour of 3M ISD 112. This could affect the simulation results. Two slewing tests are done. Both are in the horizontal plane, but one beam has a payload at the end and the other does

not. As a result, the natural frequencies are different, giving insights into the performance of the models in different situations.

6.3.1 Experimental Set-Up

The beams are clamped and are rotating in the horizontal plane. The motor hub radius is 0.0375 m. The beams have a section of 26 mm by 12.3 mm. The density of the PMMA used is 1225 kg/m^3. The strain near the base is read with two strain gauges set in a half-bridge configuration. The gauges have a resistance of 350 Ω and a grid area of 14.5 mm^2. Given the poor heat sink capacity of PMMA, a low excitation voltage is used.

Beam with no Payload

The beam with no payload has a length of 0.836 m. It is slewed in the $x-y$ plane from $-90°$ to $90°$, and then back. Figure 6.13 gives the position of the base with respect to time.

Beam with a Payload

The beam with a payload has a length of 0.7852 m. The payload center of mass is located at 0.031417 m from the link extremity. The payload mass is 0.36036 kg and the centroidal moment of inertia about the axis of rotations of the beam is $5.238 \times 10^{-4}\ kg \cdot m^2$. The beam is slewed in the $x-y$ plane from $-90°$ to $90°$. Figure 6.14 gives the position of the base with respect to time.

6.3.2 Finite Element Models

Different types of elements are used for the two beams to accomodate the payload. In both cases, an element acting as a distance indicator (*DIST* type element) is placed near the base of the beam to evaluate the strain. The initial length of the element is subtracted from the value given by the *DIST* element, and this difference is divided by the initial length of the element to obtain the strain.

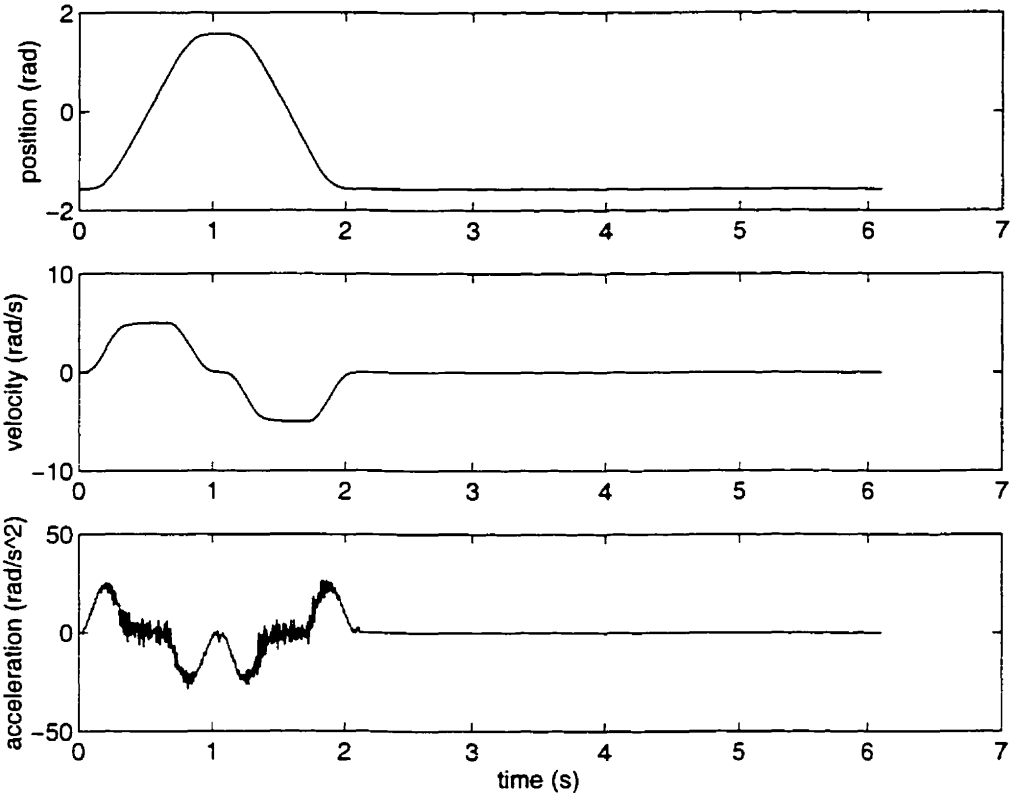

Figure 6.13: Position of the Base with Respect to Time for the PMMA Beam with no Payload

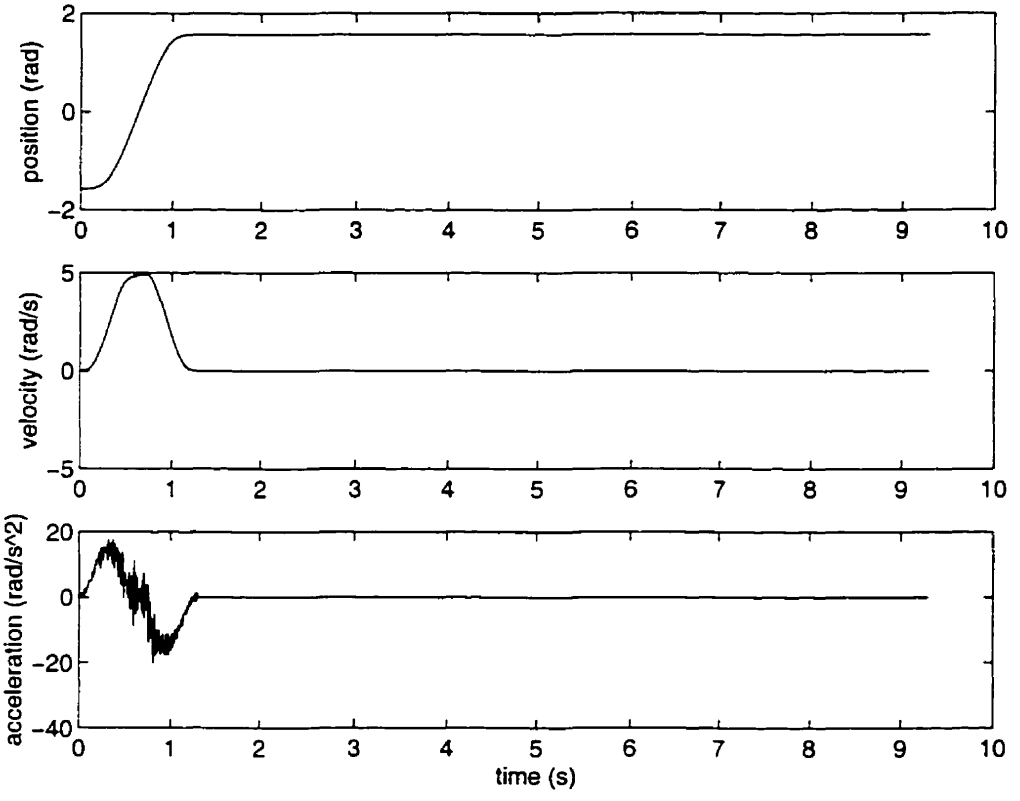

Figure 6.14: Position of the Base with Respect to Time for the PMMA Beam with a Payload

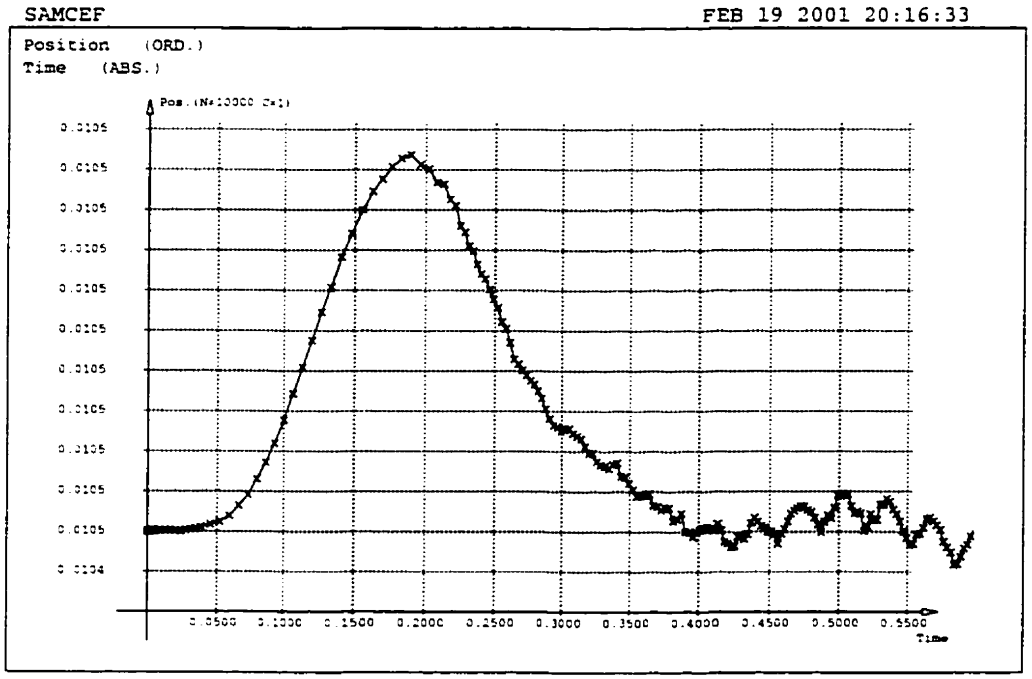

Figure 6.15: Length of the Element Near the Base without Numerical Damping

In both cases, some numerical damping is used. Figure 6.15 shows the length of the element near the base of the beam without a payload when an elastic behaviour is assumed. Soon after the beginning of the simulation, some instabilities appear in the length of the element. Convergence is difficult to obtain due to these instabilities.

Figure 6.16 shows again the length of the element near the base of the beam without a payload when an elastic behaviour is assumed, but numerical damping is added to the solution by setting the parameter α of the Hilber-Hughes-Taylor algorithm to -0.3 [39]. The solution is more stable and can now converge to the end of the simulation time.

The concern with numerical damping is not to distort the results. Figure 6.17 gives the strain near the base of the beam without a payload in the case for which the behaviour is assumed elastic. After the base stops, the beam oscillates and no substantial level of damping is seen. The numerical damping introduced has stabilized the solution without distorting the results. The other simulation parameters are set as in the case of the beam with a

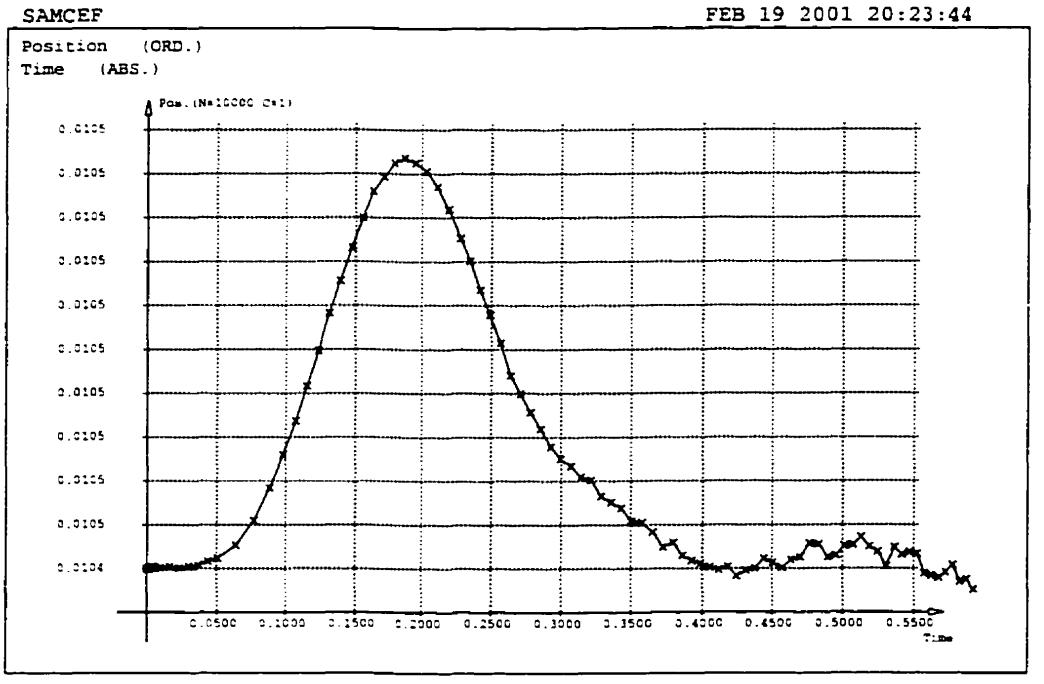

Figure 6.16: Length of the Element Near the Base with Numerical Damping

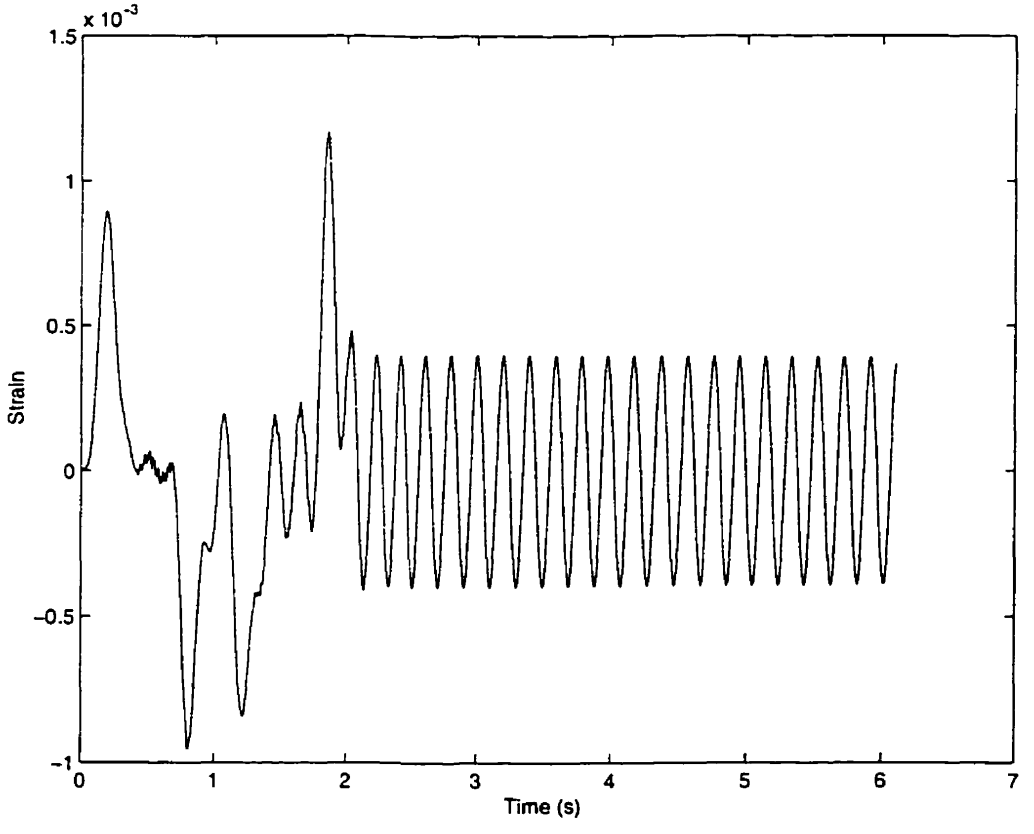

Figure 6.17: Strain Near the Base with Numerical Damping and Assuming an Elastic Behaviour

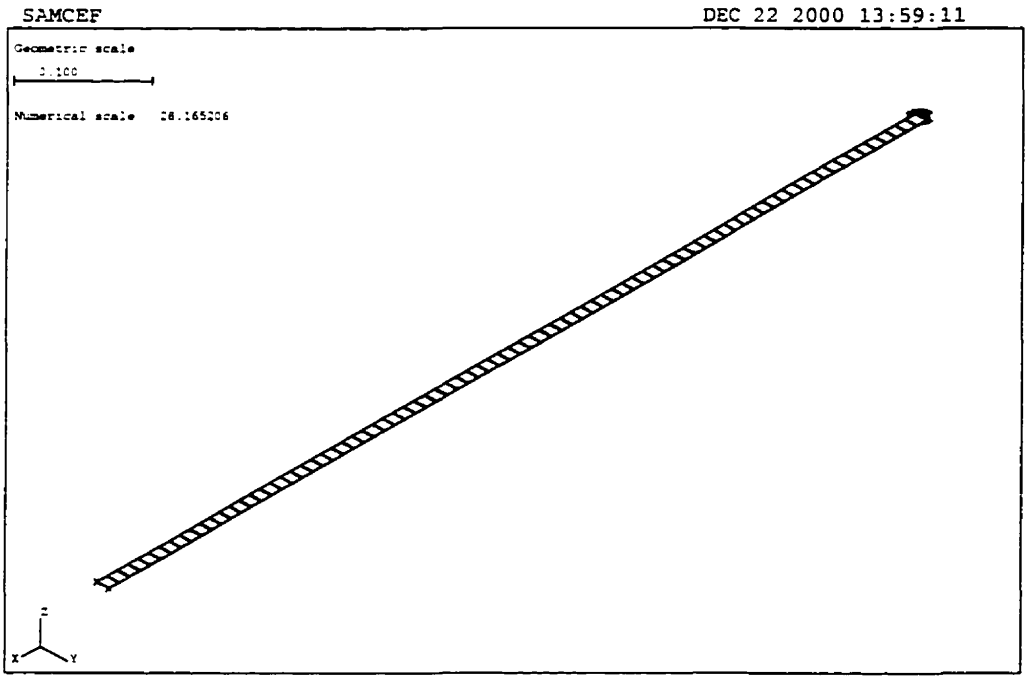

Figure 6.18: Finite Element Model of the PMMA Beam with no Payload

viscoelastic layer and the values of the parameters are given in Table 6.2.

Modelling of the Beam with no Payload

The beam without a payload is modelled with 80 plane strain elements lying in the $x - y$ plane as illustrated by Figure 6.18.

Modelling of the Beam with a Payload

The beam with a payload is also slewing in the $x - y$ plane about the z axis. The plane strain elements in *Samcef* assume a unitary thickness and only the density is input. The payload is represented as a point mass with the proper centroidal moment of inertia. Using plane strain elements would require scaling the payload mass to a unitary thickness for the beam. To avoid this, three-dimensional brick elements are used to model the beam. Figure 6.19 represents the finite element model of the beam with a payload. Along the length, 100 elements are used, and four are used along the thickness.

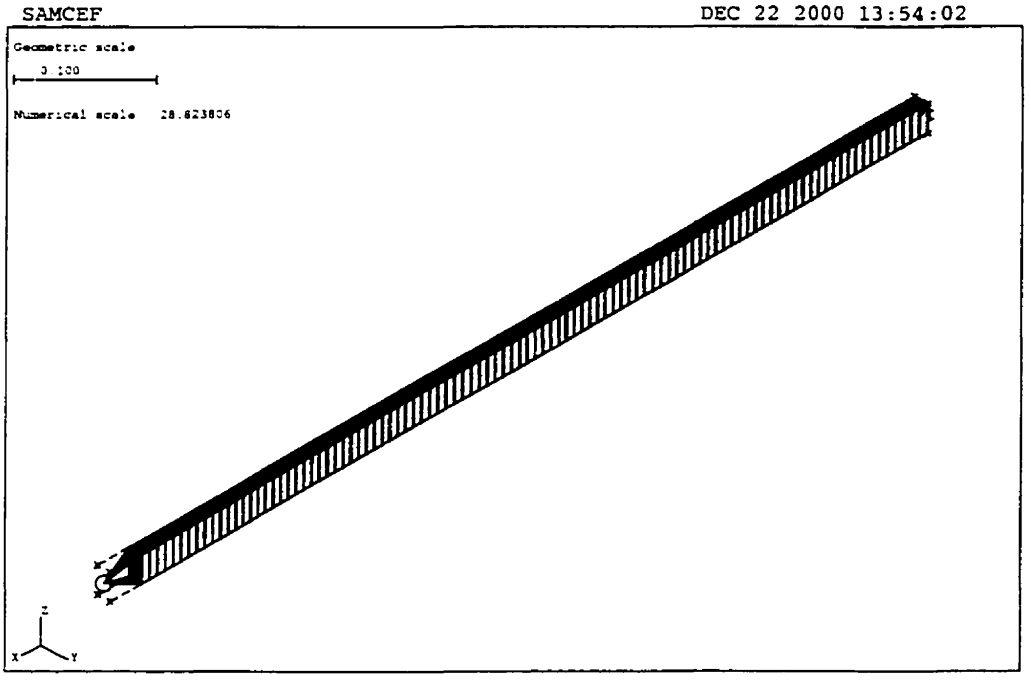

Figure 6.19: Finite Element Model of the PMMA Beam with a Payload

6.3.3 Simulated Response of the Beam with no Payload

The parameters found for PMMA in Chapter 3 are used to simulate the behaviour of the beam with no payload. Again, the goal is not to adjust the simulation parameters to reproduce the experimental results, but rather to obtain simulation results as accurate as possible using the identified material parameters. The experimental position of the motor with respect to time shown in Section 6.3.1 is input in the finite element data file. As in Section 6.2.3, the natural frequency and the logarithmic decrement are used to compare the simulation results and the experimental data. The natural frequency and the logarithmic decrement are measured on the first five cycles occurring after the forced oscillation.

The Voigt-Kelvin Model

Figure 6.20 shows the experimental response compared to the simulated response for a Voigt-Kelvin model. The dots appearing below the curve are noise generated by the strain gauge and the amplification system. The Voigt-Kelvin

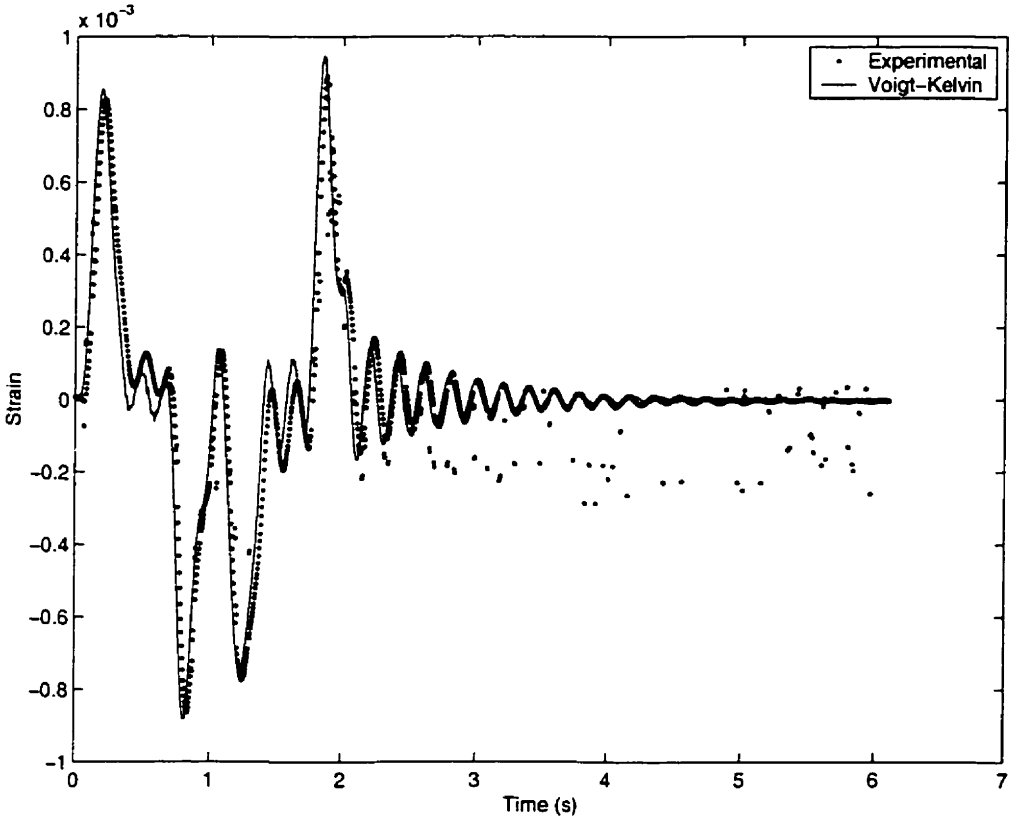

Figure 6.20: Simulated Response Using a Voigt-Kelvin Model for the PMMA Beam with no Payload

Model	Natural Frequency (Hz)	Difference with Experimental (%)	Logarithmic Decrement	Difference with Experimental (%)
Experimental	5.15		0.297	
Voigt-Klevin	5.09	1.2	0.297	0.0
Maxwell	5.15	0.0	0.051	83
Zener	5.06	1.7	0.104	65
Prony Series for a Constant Poisson's Ratio				
1 Term	5.98	16	0.225	24
2 Terms	5.78	12	0.253	15
3 Terms	5.84	13	0.224	25
Prony Series for a Constant Bulk Modulus				
1 Term	5.94	15	0.192	35
2 Terms	5.70	11	0.227	24
3 Terms	5.78	12	0.203	32
Fractional Voigt-Kelvin	5.62	9.1	0.161	46

Table 6.13: Natural Frequency and Logarithmic Decrement for the PMMA Beam with no Payload

model gives an almost perfect fit to the experimental data. Figure 3.7 showed the magnitude of the shear modulus to be underestimated by the Voigt-Kelvin model, but close to 5 Hz, the phase was just slightly overestimated. This produced the right combination to obtain very good results. Table 6.13 gives the natural frequency and the logarithmic decrement obtained with the Voigt-Kelvin model and the differences in percentage with the experimental data are very small. The simulation with the Voigt-Kelvin model requires 0.0285 h of CPU time and is accomplished in 527 time steps.

The Maxwell Model

Figure 6.21 shows the simulation with a Maxwell model. The predicted natural frequency is very good, but the damping level is underestimated. Figure 3.8 showed both the magnitude and the phase of the shear modulus to be underestimated by the Maxwell model, leading to the underestimation of the damping level in simulation. The CPU time required is 0.0433 h for 812 time steps.

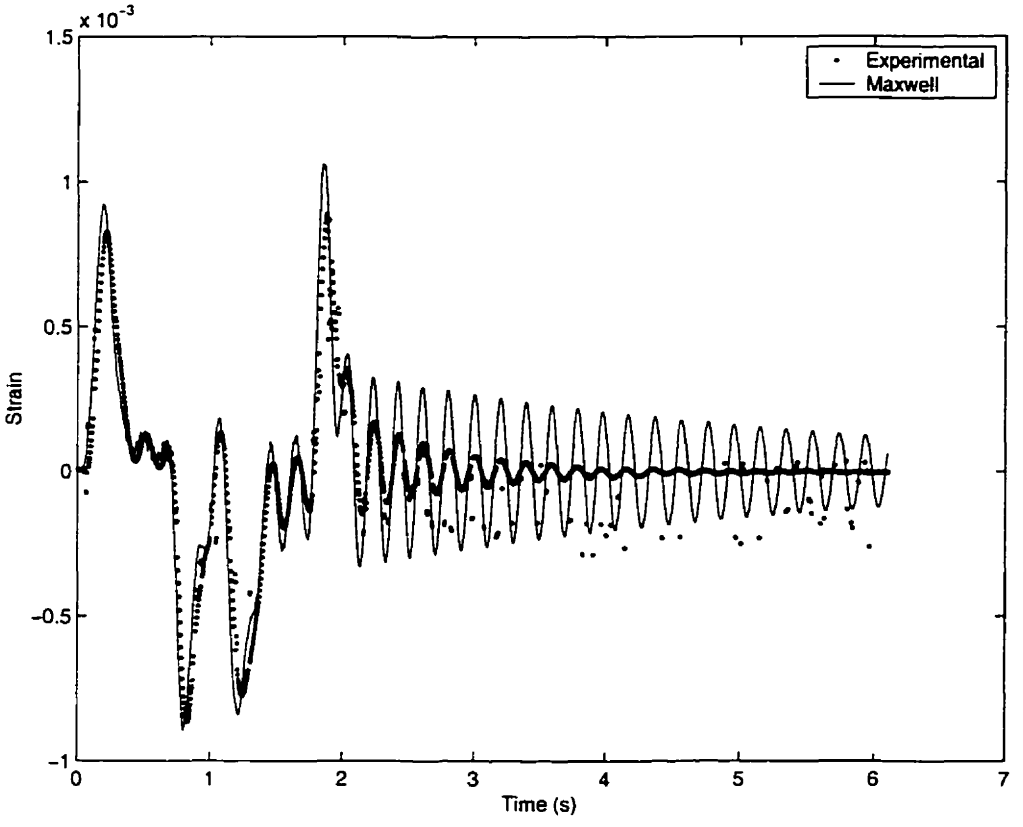

Figure 6.21: Simulated Response Using a Maxwell Model for the PMMA Beam with no Payload

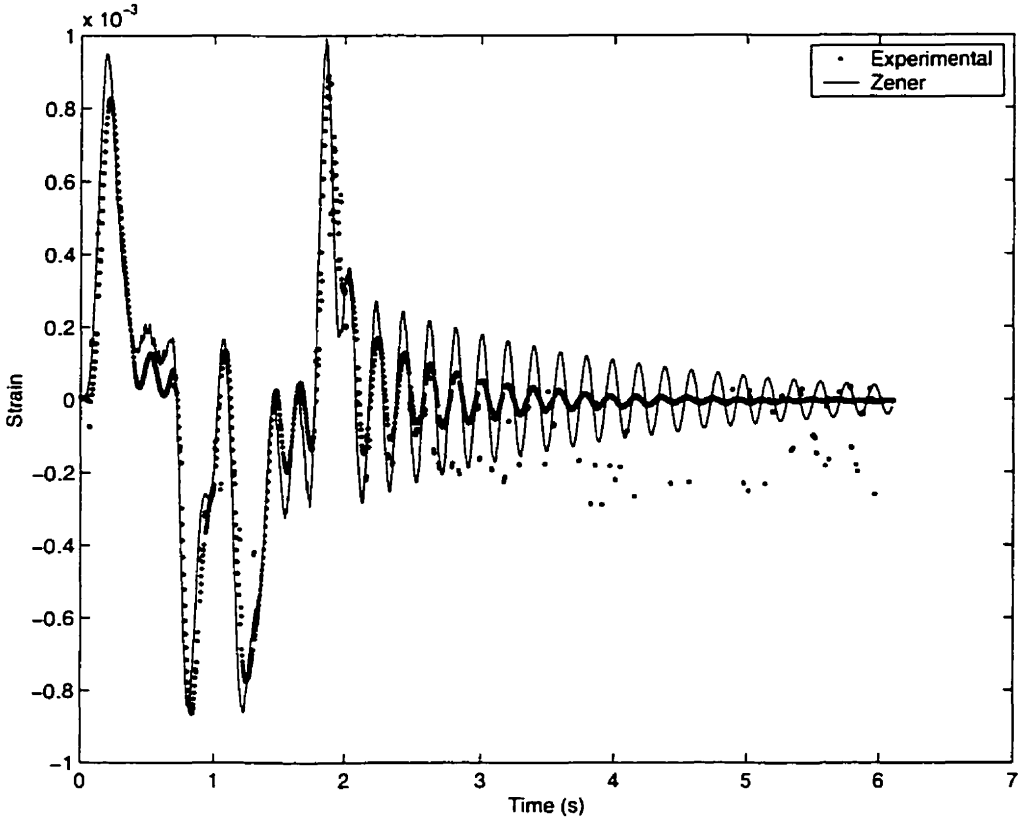

Figure 6.22: Simulated Response Using a Zener Model for the PMMA Beam with no Payload

The Zener Model

Figure 6.22 shows the comparison between the experimental data and the simulation results obtained with a Zener model. As in the case of the Maxwell model, the predicted natural frequency is very close to the experimental one, but the damping level is underestimated. Figure 3.9 also showed both the magnitude and the phase of the shear modulus to be underestimated by the Zener model in the vicinity of 5 Hz. The simulation is carried out in 729 time steps and requires 0.0405 h of CPU time.

The Prony Series

Table 6.13 shows the natural frequency and the logarithmic decrements predicted for all Prony series models. The two term Prony series give a better prediction of the logarithmic decrements than the one term or three term

Prony Series	Constant Poisson's Ratio		Constant Bulk Modulus	
	Number of Time Steps	CPU Time (h)	Number of Time Steps	CPU Time (h)
1 Term	644	0.0362	685	0.0391
2 Terms	592	0.0349	578	0.0332
3 Terms	624	0.0366	633	0.0369

Table 6.14: Number of Time Steps and CPU Time for the Simulations Using Prony Series for the PMMA Beam with no Payload

Prony series. Figures 3.10 and 3.11 showed the phase of the shear modulus to be well predicted by the one term and three term Prony series, and slightly overpredicted by the two term Prony series. The phase was probably slightly underestimated in the experimental tensile tests. The parameter identification tests were carried out at an ambient temperature of $18°C$, whereas the slewing tests were carried out at an ambient temperature of $22°C$. PMMA is very sensitive to temperature changes even at room temperature [59], and the slightly colder temperature for the parameter identification tests could explain why the models seems to predict a more rigid behaviour than what is seen in the slewing tests, but the difference could also be due to other experimental factors.

In this case, the constant Poisson's ratio assumption gives slightly better results than the constant bulk modulus assumption. Figure 6.23 shows the best case, which is the two term Prony series with a constant Poisson's ratio, and the worst case, which is the one term Prony series with a constant bulk modulus. The differences between the two models are quite small. The one term Prony series underestimates the damping level more than the two term Prony series, but both models give a satisfactory fit.

Table 6.14 gives the number of steps and the CPU time required by each simulation. There is no tendency to be observed in the CPU time as a function of the number of terms in the Prony series. There is no consistent difference either between the constant Poisson's ratio assumption or the constant bulk

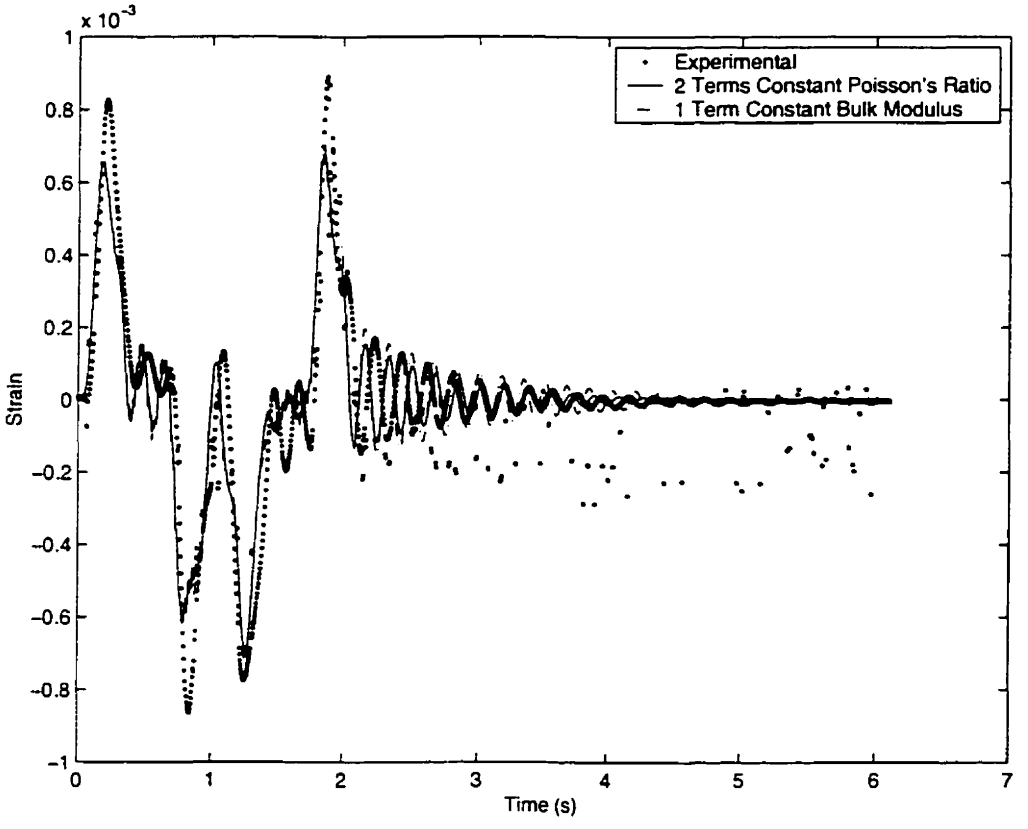

Figure 6.23: Simulated Response Using Prony Series for the PMMA Beam with no Payload

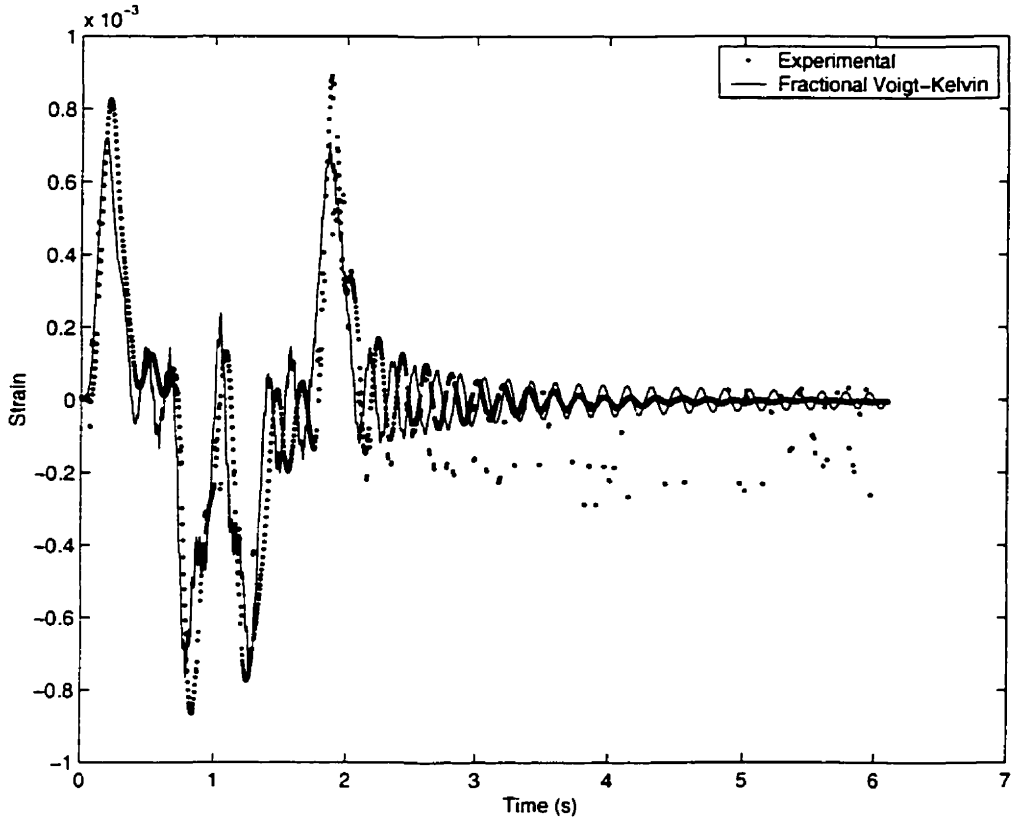

Figure 6.24: Simulated Response Using a Fractional Voigt-Kelvin Model for the PMMA Beam with no Payload

modulus assumption. However, the most precise solution is the two terms Prony series for a constant Poisson's ratio assumption, and this is the solution requiring the least time steps and the least amount of CPU time. The worst solution is the one term Prony series for a constant bulk modulus assumption, and it is also the solution requiring the most time steps and the most CPU time.

The Fractional Voigt-Kelvin Model

Figure 6.24 shows the simulation obtained with a fractional Voigt-Kelvin model against the experimental data. The simulated solution is underdamped, although not as severely as the solutions obtained with the Maxwell or the Zener model.

The results shown in Figure 6.24 are obtained with a MEMMAX value of

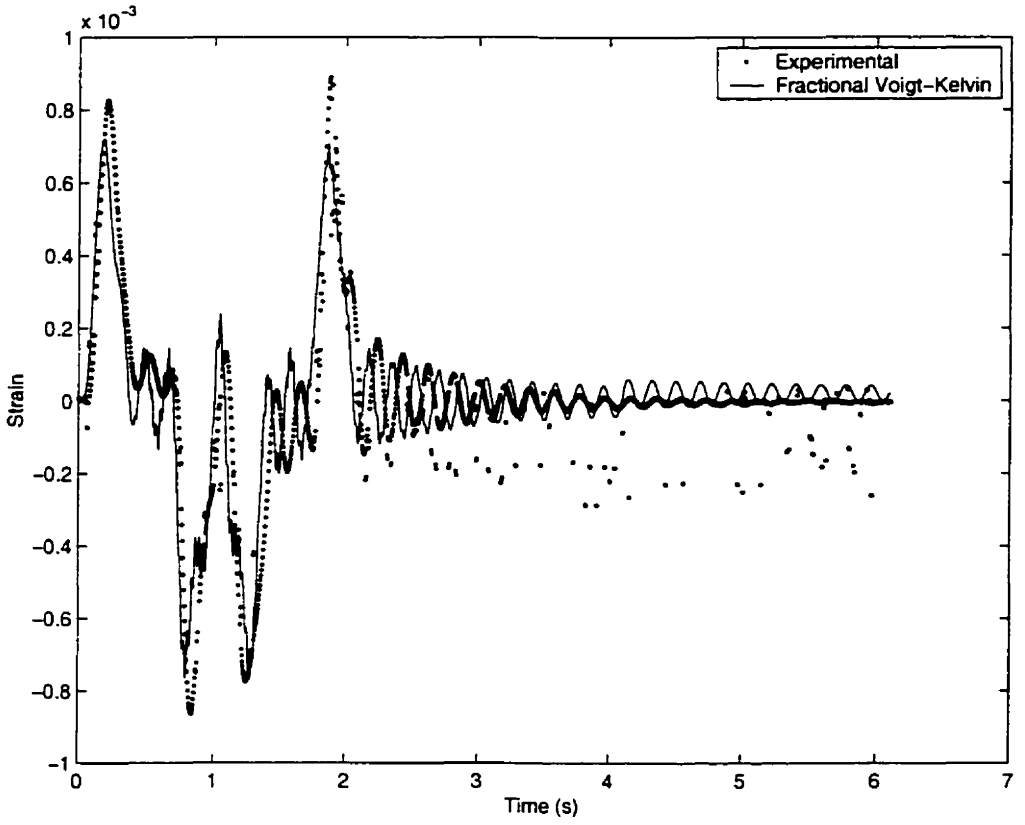

Figure 6.25: Simulated Response Using a Fractional Voigt-Kelvin Model for the PMMA Beam with no Payload and a MEMMAX of 700

1000. when the total number of time steps is 835 and the CPU time is 2.4680 h, which means no compression of the data has been done in the computation of the integral term of the fractional derivative. Figure 6.25 shows the results for a value of MEMMAX of 700. Compression occurs only near the end of the simulation time at 4.09 *sec*, but despite this, errors are introduced as soon as the data compression is done. In this case, the forced simulation is a substantial part of the total simulation time, and compressing this history of forced motion affects the results significantly.

Conclusion

Unexpectedly, for this case, the Voigt-Kelvin model gives a very good fit to the experimental data due to the right combination of underestimation of the magnitude of the shear modulus and overestimation of the phase of the shear

modulus. The Maxwell and Zener models both severely underestimate the damping level of the slewing beam.

The Prony series all give an acceptable fit, but the predicted natural frequencies are all overestimated and the predicted damping level is underestimated by all series. This could be a result of the colder ambient temperature when the parameter identification was done or of other experimental factors in measuring the damping properties. There is no significant difference between using a constant Poisson's ratio assumption or a constant bulk modulus assumption.

The fractional Voigt-Kelvin model does not overestimate the natural frequency as much as the Prony series, but it underestimates more severely the damping level. In Figure 3.12, the model underestimates the magnitude and the phase of the shear modulus. The overestimation of the natural frequency is unexpected and could come from the colder ambient temperature when the parameter identification tests were done or of other experimental factors in measuring the damping properties.

The Prony series required on average 626 steps and the fractional Voigt-Kelvin model, 835 steps, a significant 33% increase over the Prony series. However, this represents a sampling frequency of approximately 25 data point per cycle, which is not excessive. The Maxwell model also required 812 time steps, a number close to what is obtained with the fractional Voigt-Kelvin model. The Prony series required on average 0.0362 h compared to 2.4680 h for the fractional Voigt-Kelvin model, a large difference.

6.3.4 Simulated Response of the Beam with a Payload

This example takes the same beam used in the preceding section, but the beam is shorter and there is a payload at the extremity. The payload decreases the natural frequency and it is an interesting way of evaluating the performance of the models for the same material, but at a different frequency. Again, the goal is to predict the experimental behaviour from the material data,

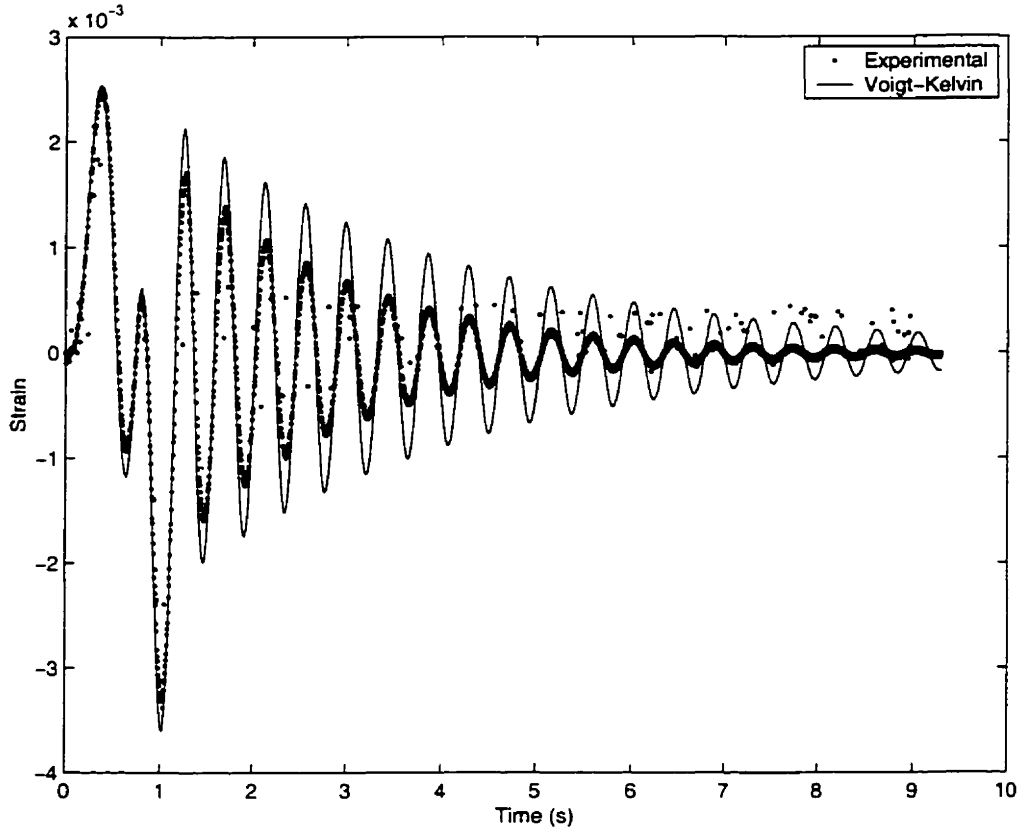

Figure 6.26: Simulated Response Using a Voigt-Kelvin Model for the PMMA Beam with a Payload

without adjusting the simulation parameters to obtain a better fit between the simulation results and the experimental data.

The experimental position of the motor with respect to time shown in Section 6.3.1 is input in the finite element data file. The natural frequency and the logarithmic decrement are used to compare the simulation results and the experimental data. The natural frequency and the logarithmic decrement are measured over the first five cycles occurring after the forced oscillation.

The Voigt-Kelvin Model

Figure 6.26 shows the experimental response compared to the simulated response for a Voigt-Kelvin model. The Voigt-Kelvin model predicts a natural frequency very close to the experimental data, but the model underestimates the damping level. Figure 3.7 showed the magnitude and the phase of the

Model	Natural Frequency (Hz)	Difference with Experimental (%)	Logarithmic Decrement	Difference with Experimental (%)
Experimental	2.33		0.231	
Voigt-Klevin	2.31	0.9	0.136	41
Maxwell	2.32	0.4	0.109	53
Zener	2.31	0.9	0.212	8
Prony Series for a Constant Poisson's Ratio				
1 Term	2.68	15	0.241	4
2 Terms	2.58	11	0.202	13
3 Terms	2.60	12	0.193	16
Prony Series for a Constant Bulk Modulus				
1 Term	2.63	13	0.241	4
2 Terms	2.57	10	0.187	19
3 Terms	2.57	10	0.179	23
Fractional Voigt-Kelvin	2.49	7	0.121	48

Table 6.15: Natural Frequency and Logarithmic Decrement for the PMMA Beam with a Payload

shear modulus to be underestimated by the Voigt-Kelvin model close to 2 Hz. This explains the underestimation of the damping observed in Figure 6.26. Table 6.15 gives the natural frequency and the logarithmic decrement obtained with the Voigt-Kelvin model. There is a difference of 41% between the experimental data and the simulation results for the logarithmic decrement. The simulation is carried out in 0.1277 h for 769 steps.

The Maxwell Model

Figure 6.27 shows the simulation with a Maxwell model. As for the beam with no payload, the predicted natural frequency is very good, but the damping level is underestimated. Figure 3.8 showed both the magnitude and the phase of the shear modulus to be underestimated by the Maxwell model, leading to the underestimation of the damping level in simulation. The simulation requires 819 steps and 0.1412 h of CPU time.

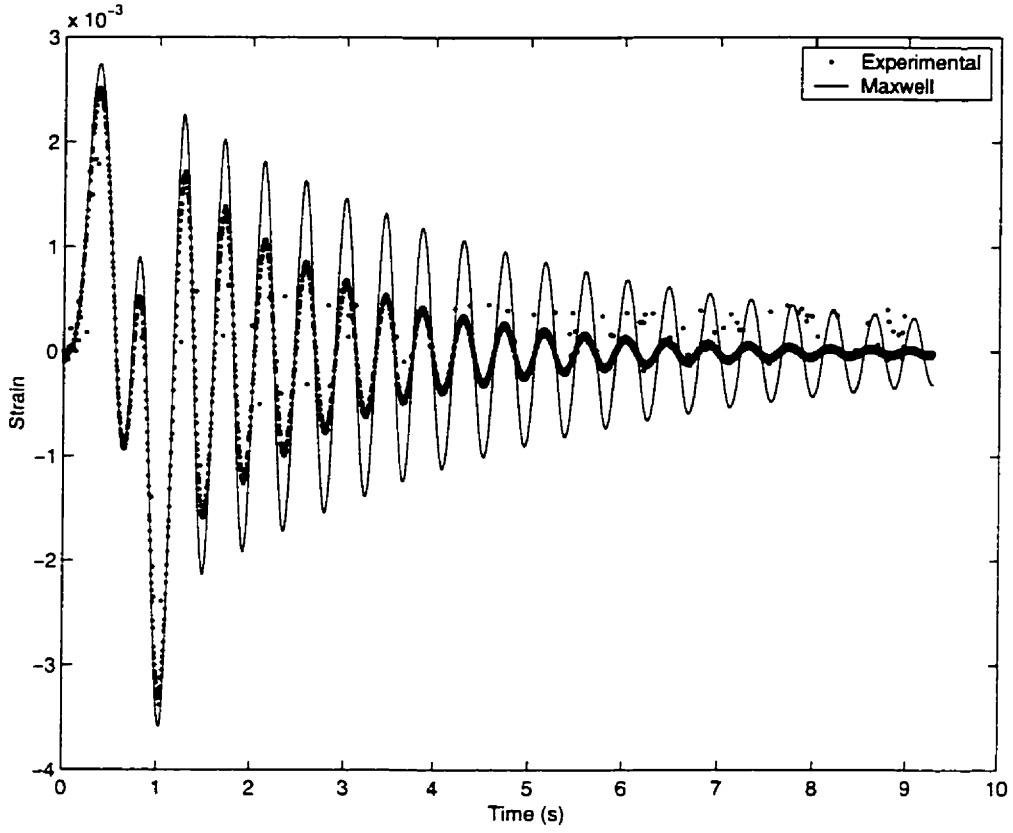

Figure 6.27: Simulated Response Using a Maxwell Model for the PMMA Beam with a Payload

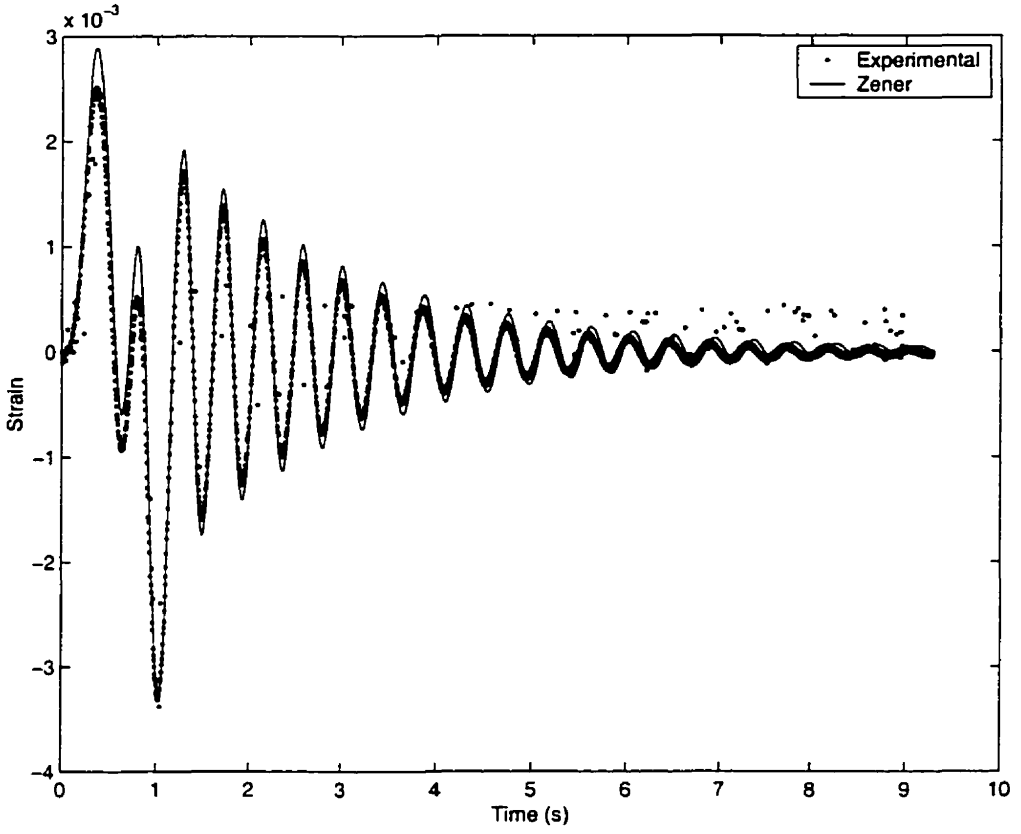

Figure 6.28: Simulated Response Using a Zener Model for the PMMA Beam with a Payload

The Zener Model

Figure 6.28 shows the comparison between the experimental data and the simulation results obtained with a Zener model. The predicted natural frequency and damping level are very good. Figure 3.9 showed both the magnitude to be underestimated by the Zener model, but the phase of the phase modulus is slightly overestimated in the vicinity of 2 Hz. As it is the case for the Voigt-Kelvin model for the beam with no payload, this combination of underestimation of the magnitude of the shear modulus and overestimation of the phase of the shear modulus produced the right parameters to obtain good simulation results. The simulation is done in 674 steps for 0.1273 h of CPU time.

The Prony Series

Table 6.15 shows the natural frequency and the logarithmic decrements predicted for all Prony series models. For the beam with no payload, the two term Prony series gives the best prediction for the logarithmic decrement. However, in the case of the beam with no payload, all models overpredicted the natural frequency and underpredicted the damping level. For the beam with a payload, Table 6.15 shows all models based on Prony series to overestimate the natural frequency, but the one term Prony series overestimate the damping level, whereas the other Prony series underestimate the damping level. Although the one term Prony series predicts an accurate value for the damping level, it is slighlty overestimated. This overestimation added to the overestimation of the natural frequency leads to a much stiffer solution than observed experimentally. Figure 6.29 shows the one term Prony series for a constant Poisson's ratio and the stiffness of the solution is obvious. Figures 3.10 and 3.11 showed the phase of the shear modulus to be well predicted by the two term and three term Prony series in the vicinity of 2 Hz, and overpredicted by the one term Prony series. The one term Prony series also overestimate the magnitude of the shear modulus close to 2 Hz, resulting in larger discrepancies between the predicted and experimental natural frequencies.

In Figure 6.29, against the one term Prony series, the three term Prony series is also shown for a constant bulk modulus. This model is the one which underestimates most severely the damping level. However, since the natural frequency is overestimated, the underestimation of the damping level leads to the best match between the simulation and the experimental results.

Table 6.16 gives the number of steps and the CPU time required by each simulation. The Prony series using a constant Poisson's ratio assumption run faster needing less time steps than the ones with a constant bulk modulus for this example of a PMMA beam with a payload.

For this example, it is difficult to say which solution is the best. The natural frequency is certainly a dominating criteria. The two terms and three

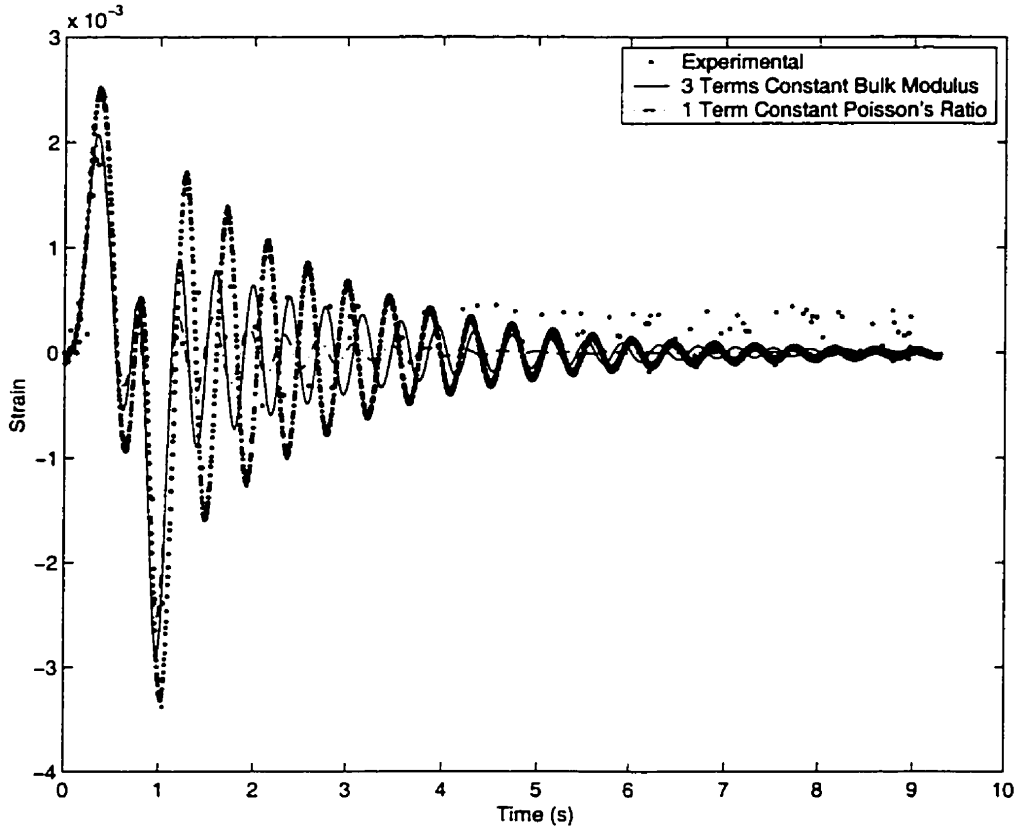

Figure 6.29: Simulated Response Using Prony Series for the PMMA Beam with a Payload

Prony Series	Constant Poisson's Ratio		Constant Bulk Modulus	
	Number of Time Steps	CPU Time (h)	Number of Time Steps	CPU Time (h)
1 Term	389	0.0675	429	0.0764
2 Terms	556	0.0946	602	0.1062
3 Terms	556	0.0961	610	0.1020

Table 6.16: Number of Time Steps and CPU Time for the Simulations Using Prony Series for the PMMA Beam with a Payload

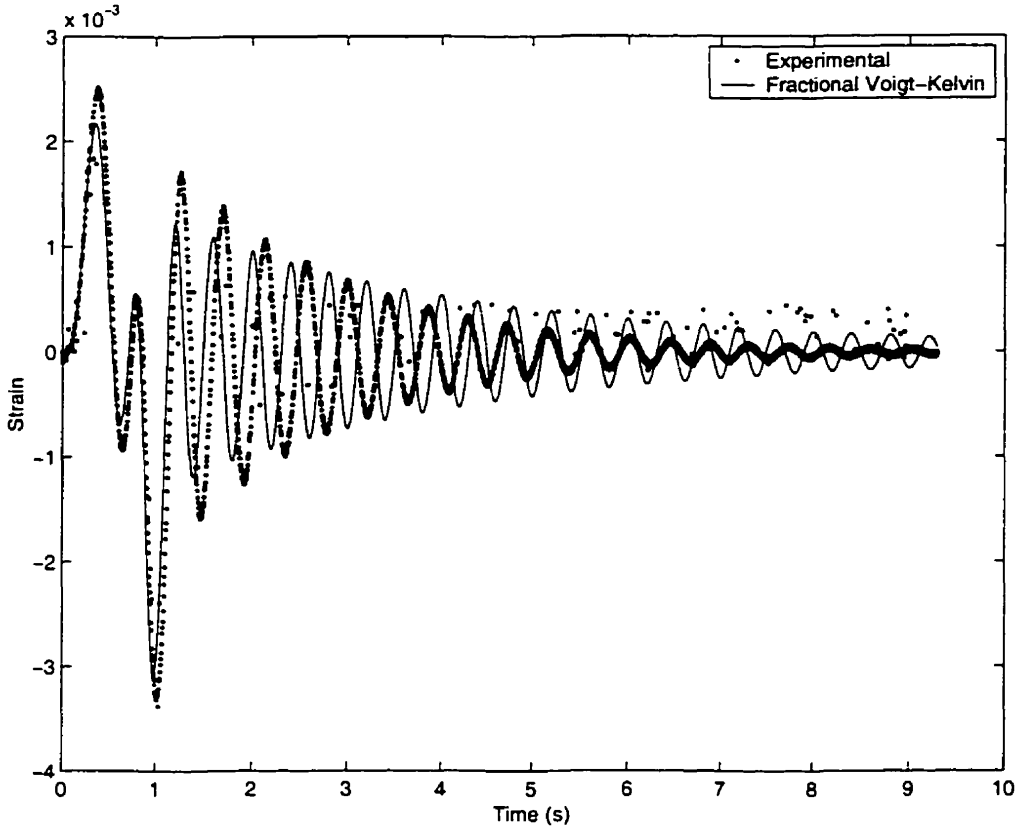

Figure 6.30: Simulated Response Using a Fractional Voigt-Kelvin Model for the PMMA Beam with a Payload

terms Prony series for a constant bulk modulus generate results with a more precise natural frequency.

The Fractional Voigt-Kelvin Model

Figure 6.30 shows the simulation results obtained with a fractional Voigt-Kelvin model against the experimental data. The damping of the solution is underestimated, but since the natural frequency is overestimated, the solution and the simulated results remain close to the experimental results.

The results shown in Figure 6.24 are obtained with a MEMMAX value of 1500, when the total number of time steps is 767 and the CPU time is 6.3671 h, which means no compression of the data has been done in the computation of the integral term of the fractional derivative. A simulation resulting in almost the same natural frequency, 2.51 Hz rather than 2.49 Hz, and almost

the same logarithmic decrement, 0.116 rather than 0.121, are obtained with a value of MEMMAX of 400. When MEMMAX is set at 400, the simulation is accomplished in 755 steps and 3.3476 h, a net improvement in terms of CPU time over the case for which MEMMAX is set at 1500. Reducing further the value of MEMMAX to 200 prevents the solution from converging. If compression is possible in this case, when it was not possible for the beam with no payload, it is because the time for which the forced oscillation occurs is small. Data compression is possible when the system is in free oscillation, but accurate simulation of the forced oscillation is essential. The finite element package reacts similarly by taking small steps during a forced oscillation and larger steps during a free oscillation. A scheme which would compress only the data during the free oscillations could be advantageous.

Conclusion

As in the case of a beam with no payload, the right combination of underestimation of the magnitude of the shear modulus and overestimation of the phase of the shear modulus gives a very good fit to the experimental data. In this case, it happens for the Zener model, whereas it was the Voigt-Kelvin model which gave a very good fit for the beam with no payload. For this beam, the Voigt-Kelvin and Maxwell models both severely underestimate the damping level.

The Prony series all overestimate the natural frequency, leading to a stiffer model. Models which underestimate the damping level correct in part the overestimated natural frequency and produce a better fit. Again, this stiffness of the model compared to the experimental data could be a result of the colder ambient temperature when the parameter identification was done or of other experimental factors in the acquisition of the material parameters.

The fractional Voigt-Kelvin model does not overestimate the natural frequency as much as the Prony series, but it underestimates more severely the damping level. In Figure 3.12, the model underestimates the magnitude and

the phase of the shear modulus. The overestimation of the natural frequency is unexpected and could show the material data to represent a stiffer behaviour than what is observed with the slewing beams.

The Prony series require on average 524 steps and the fractional Voigt-Kelvin model on average for the different values of MEMMAX, 761 steps, a significant 45% increase over the Prony series. However, this represents a sampling frequency of approximately 30 data points per cycle, which is not excessive. The Maxwell model also require 819 time steps, a number close to what is obtained with the fractional Voigt-Kelvin model. The Prony series require on average 0.0905 h compared to 3.3476 h for the fractional Voigt-Kelvin model with a MEMMAX of 400, a large difference.

6.4 Conclusion

Although the numerical algorithm used for the fractional derivative is efficient in terms of number of time steps needed to complete the simulations, the required CPU time is still very high compared to classical models. In the case of the steel beam covered by a constrained viscoelastic layer, the Prony series are as accurate as the fractional Voigt-Kelvin model. In the case of the PMMA beams, simulations are not as close to the experimental results. The natural frequency tends to be significantly overestimated and this is probably due to colder ambient temperatures when the parameter identification was done or to other experimental factors. Despite this, the Prony series and the fractional Voigt-Kelvin model are reliable. They give a certain level of error which remains similar from case to case. The Voigt-Kelvin, Maxwell, and Zener models can give arbitrarily good or poor fits to experimental data depending if the models have the right combination of underestimation of the natural frequency and overestimation of the damping level. The three examples are all oscillating at low frequencies and there is no significant difference between using a constant Poisson's ratio assumption or a constant bulk modulus assumption.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

Viscoelastic models based on fractional derivatives are recongnized in the literature for accurately modelling the experimental behaviour with few parameters. In this thesis, classical viscoelastic models and models based on fractional derivatives are fitted to the experimental behaviour of polymethylmethacrylate and an acrylic based viscoelastic layer, 3M ISD 112. The shear modulus with respect to frequency is the experimental behaviour studied for both materials. In both cases, the best fit is obtained with the Prony series, but this classical model requires several parameters. The fractional Voigt-Kelvin model does not approximate perfectly the behaviour of PMMA, but it does give a good fit for 3M ISD 112 for the first half of the frequency range. Although the fractional Voigt-Kelvin model does not fit the experimental behaviour as well as the Prony series, it has the advantage of requiring few parameters. In certain situations in which little experimental data is available, a model with a low number of parameters to identify is essential.

Fractional derivatives are hereditary integrals, and as such, their computation is tedious. Authors have developed algorithms to solve equations with fractional derivatives in the time domain. Most of these algorithms are based on finite differences or the Grünwald series and they require small time steps

for a good accuracy. With the finite element method, three-dimensional equations are used and this adds to the computation burden. As an alternative, in this thesis, an algorithm based on an approximation of the fractional derivative is developed. The integral resulting from this approximation is solved with a trapezoidal rule. The algorithm is tested on a one-dimensional constitutive equation with one fractional derivative. The results are compared to the results obtained with a numerical inverse Laplace transform for a sinusoidal strain input and a decaying sinusoidal strain input. The algorithm based on the trapezoidal rule does not require small time steps and gives accurate results. Rather than storing all the data points to be used for the hereditary integral, only one data out of two is kept for data points far from the actual computing point. This approach reduces significantly the computing time.

The algorithm based on the trapezoidal rule is implemented in a commercial finite element package, *Samcef*. Again, the results for a cube submitted to a sinusoidal strain and a decaying sinusoidal strain are compared to the results from the numerical inverse Laplace transform. In this case, however, the finite element package computes three-dimensional equations even though the prescribed displacement is one-dimensional. For this case too, the algorithm does not require small time steps to produce accurate results. Again, storing only one data point out of two for data points far away from the actual data point reduces the computation time. The effect of the various convergence parameters inherent to the finite element package is studied and no unexpected effect is observed.

Finally, simulations are done for experimental cases. Two PMMA slewing beams are studied, one having a mass at the extremity. A slewing steel beam covered by a constrained viscoelastic layer is also studied. The viscoelastic material is 3M ISD 112. The simulations are compared to the experimental results for the fractional Voigt-Kelvin model and for classical models: Prony series, Voigt-Kelvin model, Maxwell model, and Zener model. The goal is not to fit the model to the experimental results, but rather to use the parame-

ters determined from the fit to the experimental shear modulus, and see the consistency with the simulation results for slewing beams. The Voigt-Kelvin, Maxwell, and Zener models give accurate or imprecise results depending on the natural frequency being excited, but they are not reliable to represent the experimental behaviour over a large range of situations. The Prony series are accurate and efficient in terms of CPU time. The fractional Voigt-Kelvin model is also accurate, but it requires a significant amount of CPU time. Storing one data out of two for data points far away from the actual data point helps, but it still is not competitive with the CPU time required by the Prony series. The long CPU time for the fractional Voigt-Kelvin model is due to the hereditary integral, since with the algorithm based on the trapezoidal rule, the time step is not smaller than the time step required by the classical models. In the case of the PMMA beam with no mass at the extremity, the forced oscillation time is quite long compared to the overall simulation time, and in this case, it is not possible to reduce significantly the storage vector. The main advantage of the fractional Voigt-Kelvin model remains the low number of parameters to identify and this model should be used whenever it is impossible to identify correctly the parameters of the Prony series. The experimental cases are all low frequency examples, and for these cases, no significant difference is observed between assuming a constant Poisson's ratio or a constant bulk modulus.

7.2 Recommendations

Different aspects of the algorithm based on the trapezoidal rule could be improved:

- The examples of the PMMA beams have shown the difficulty of storing less data during the forced oscillations. A more flexible algorithm could be devised to store all data during forced oscillations and less data during natural oscillations.

- The accuracy of the model would be improved by adding a fractional derivative of the stress. This would, however, add significant CPU time.
- The temperature dependency could be modelled.
- Further approximations of the hereditary integral could be sought with the aim of reducing the CPU time without sacrificing accuracy.

Bibliography

[1] V. Jha and P. Tremblay, "Application of passive damping material to reduce acoustically induced vibration in radarsat," in *44th Congress of the International Astronautical Federation*, (Paris, France), pp. 1–6, Canadian Space Agency, SPAR Aerospace Ltd., International Astronautical Federation, October 16-22 1993.

[2] V. Jha and P. Tremblay, "Correlation of theoretical & experimental results for reduction of acoustically induced vibration due to passive damping for a typical spacecraft panel," in *45th Congress of the International Astronautical Federation*, (Jerusalem, Israel), pp. 1–6. International Astronautical Federation, October 9-14 1994.

[3] D. I. G. Jones, J. S. Kirby, and D. Edberg, "Effect of long space exposure upon properties of viscoelastic materials," in *Proceedings of Damping '93*, (San Francisco, California), pp. ICD–1 to ICD–8, February 1993.

[4] M. Poizat, P. Vialatoux, P. Cochery, and M. Vedrenne, "Viscoelastic damping system use as a remedy for pogo effect on the diamant satellite launch vehicle," *The Shock and Vibration Bulletin*, no. 46, pp. 245–466, 1976. Part 2.

[5] T. E. Albert, H. Xia, and Y. Chen, "Dynamic analysis to evaluate passive damping augmentation for the space shuttle remote manipulator system," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 468–475, September 1992.

[6] C. D. Johnson and D. A. Kienholz, "Finite element prediction of damping in structures with constrained viscoelastic layers," *AIAA Journal*, vol. 20, no. 9, pp. 1284–1290, 1982.

[7] C. D. Johnson, D. A. Kienholz, E. M. Austin, and M. E. Schneider, "Design and analysis of damped structures using finite element techniques," in *Proceedings of the ASME Design Engineering Division Conference and Exhibit on Mechanical Vibration and Noise*, (Cincinnati, Ohio), pp. 1–8, September 10-13 1985.

[8] C. T. Sun, B. V. Sankar, and V. S. Rao, "Damping and vibration control of unidirectional composite laminates using add-on viscoelastic materials," *Journal of Sound and Vibration*, vol. 139, no. 2, pp. 277–287, 1990.

[9] M. L. Slanik, J. A. Nemes, M.-J. Potvin, and J.-C. Piedbœuf, "Damping of cantilever beams with constrained viscoelastic layers using a prony series representation," in *1998 ASME International Mechanical Engineering Congress and Exposition*, (Anaheim, California), 16-20 November 1998.

[10] D. R. Axelrad, "Mechanical models of relaxation phenomena," *Advances in Molicular Relaxation Processes*, vol. 2, pp. 41–68, 1970.

[11] M. L. Slanik, J. A. Nemes, M.-J. Potvin, and J.-C. Piedboeuf, "Time domain finite element simulations of damped multilayered beams using a prony series representation," *Mechanics of Time-Dependent Materials*, vol. 4, pp. 211–230, 2000.

[12] Y. A. Rossikhin and M. V. Shitikova, "Applications of fractional calculus to dynamic problems of linear and nonlinear hereditary mechanics of solids," *Applied Mechanics Review*, vol. 50, pp. 15–67, January 1997.

[13] R. L. Bagley and P. J. Torvik, "Fractional calculus in the transient analysis of viscoelastically damped structures," *AIAA Journal*, vol. 23, pp. 918–925, June 1985.

[14] J. Cooke and R. Keltie, "Determination of the impulse response of a viscoelastic beam using a fractional derivative constitutive model," in *Proceedings of the 1987 ASME Design Technology Conferences - 11th Biennial Conference on Mechanical Vibration and Noise*, pp. 137–141, September 1987.

[15] J. Padovan and Y. Guo, "General response of viscoelastic systems modelled by fractional operators," *Journal of the Franklin Institute*, vol. 325, no. 2, pp. 247–275, 1988.

[16] A. Makroglou, R. K. Miller, and S. Skaar, "Computational results for a feedback control for a rotating viscoelastic beam," *Journal of Guidance, Control, and Dynamics*, vol. 17, pp. 84–90, January-February 1994.

[17] L. Suarez and A. Shokooh, "Response of systems with damping materials modeled using fractional calculus," *Applied Mechanics Review*, vol. 48, pp. S118–S126, November 1995.

[18] L. E. Suarez and A. Shokooh, "An eigenvector expansion method for the solution of motion containing fractional derivatives," *ASME Journal of Applied Mechanics*, vol. 64, pp. 629–635, 1997.

[19] W. P. Baker, L. B. Eldred, and A. N. Palazotto, "Viscoelastic material response with a fractional-derivative constitutive model," *AIAA Journal*, vol. 34, pp. 596–600, March 1996.

[20] R. A. DiTaranto, "Theory of vibratory bending for elastic and viscoelatic layered finite-length beams," *Journal of Applied Mechanics*, vol. 32, pp. 881–886, 1965.

[21] R. L. Bagley and R. A. Calico, "Fractional order state equations for the control pf viscoelastically damped structures," *Journal of Guidance, Control, and Dynamics*, vol. 14, pp. 304–311, March-April 1991.

[22] C. G. Koh and J. M. Kelly, "Application of fractional derivatives to seismic analysis of base-isolated models," *Earthquake Engineering and Structural Dynamics*, vol. 19, pp. 229–241, 1990.

[23] K. B. Oldham and J. Spanier, *The Fractional Calculus*. New York: Academic Press, 1974.

[24] N. Makris and M. Constantinou, "Fractional-derivative Maxwell model for viscous dampers," *Journal of Structural Engineering*, vol. 117, pp. 2708–2724, September 1991.

[25] L. B. Eldred, W. P. Baker, and A. N. Palazotto, "Kelvin-Voigt vs fractional derivative model as constitutive relations for viscoelastic materials," *AIAA Journal*, vol. 33, pp. 547–550, March 1995.

[26] L. B. Eldred, W. P. Baker, and A. N. Palazotto, "Application of fractional derviative model constitutive relations for viscoelastic materials," in *Proceedings of the 36th AIAA/ASME/ASCE/ANS/ASC Structures, Structural Dynamics and Materials Conference*, (New Orleans), pp. 2360–2370, American Institute of Aeronautics and Astronautics, Inc., 1995.

[27] L. B. Eldred, W. P. Baker, and A. N. Palazotto, "Numerical application of fractional derivative model constitutive relations for viscoelastic materials," *Computers and Structures*, vol. 60, no. 6, pp. 875–882, 1996.

[28] J. Padovan, "Computational algorithms for FE formulations involving fractional operators," *Computational Mechanics*, vol. 2, pp. 271–287, 1987.

[29] J.-T. Chern, *Finite Element Modeling of Viscoelastic Materials on the Theory of Fractional Calculus*. PhD thesis, Pennsylvania State University, 1993.

[30] M. Enelund and P. Olsson, "Damping described by fading memory - analysis and application to fractional derivative models," *International Journal of Solids and Structures*, vol. 36, pp. 939–970, 1999.

[31] M. Enelund, A. Fenander, and P. Olsson, "Fractional integral formulation of constitutive equations of viscoelasticity," *AIAA Journal*, vol. 35, pp. 1356–1362, August 1997.

[32] M. Enelund and B. L. Josefson, "Time-domain finite element analysis of viscoelastic structures with fractional derivatives constitutive relations." *AIAA Journal*, vol. 35, pp. 1630–1637, October 1997.

[33] M. Enelund and G. A. Lesieutre, "Time domain modeling of damping using anelastic displacement fields and fractional calculus," *International Journal of Solids and Structures*, vol. 36, pp. 4447–4472, 1999.

[34] M. Enelund, L. Mähler, K. Runesson, and B. L. Josefson, "Formulation and integration of the standard linear viscoelastic solid with fractional order rate laws," *International Journal of Solids and Structures*, vol. 36, pp. 2417–2442, 1999.

[35] M.-J. Potvin, J.-C. Piedbœuf, and J. Jeswiet, "A fractional Voigt-Kelvin model for the dynamic beaviour of a PMMA beam," in *Proceedings of the 12th Symposium on Engineering Applications of Mechanics*, (Montréal), pp. 315–324, June 27-29 1994.

[36] M.-J. Potvin, J. Jeswiet, and J.-C. Piedbœuf, "Réponse temporelle d'une poutre en polymère modélisée par un amortissement fractionnaire de Voigt-Kelvin," *Transactions of the CSME*, vol. 19, no. 1, pp. 13–24, 1995.

[37] M.-J. Potvin, J.-C. Piedbœuf, and J. A. Nemes, "Solution d'équations dynamiques non-linéaires avec loi de comportement viscoélastique d'ordre fractionnaire," in *Proceedings of the 16th Canadian Congress of Applied*

Mechanics (CANCAM 97), pp. 75–76, Université Laval, Québec, Canada, June 1-5 1997.

[38] M.-J. Potvin, J.-C. Piedbœuf, and J. A. Nemes, "Comparison of viscoelastic models in simulating the transient response of a slewing polymer arm," *Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 120, pp. 340–345, September 1998.

[39] H. M. Hilber, T. J. R. Hughes, and R. L. Taylor, "Improved numerical dissipation for time integration algorithms in structural dynamics," *Earthquake Engineering and Structural Dynamics*, vol. 5, pp. 284–292, 1977.

[40] L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium.* New Jersey: Prentice-Hall, Inc., 1969.

[41] N. W. Tschoegl, *The Phenomenological Theory of Linear Viscoelastic Behavior.* Berlin: Springer-Verlag, 1989.

[42] D. L. Rogers, "Damping, on modeling viscoelastic behavior," *The Shock and Vibration Bulletin*, pp. 55–69, May 1981. Part 1.

[43] R. L. Bagley and P. J. Torvik, "Fractional calculus - a different approach to the analysis of viscoelastically damped structures," *AIAA Journal*, vol. 21, pp. 741–748, May 1983.

[44] M. R. Spiegel, *Formules et tables de mathématiques.* Schaum, New York: McGraw-Hill, 1978.

[45] R. L. Bagley, "The thermorheologically complex material," *International Journal of Engineering Science*, vol. 29, no. 7, pp. 797–806, 1991.

[46] *Samcef User Manual - Version 8.1*, 2000.

[47] D. W. van Krevelen, *Properties of polymers: their correlation with chemical structure, their numerical estimation and prediction from additive group contributions.* Amsterdam: Elsevier, 3rd ed., 1990.

[48] M. Slanik, "A numerical and experimental investigation of steel beams damped with constrained viscoelastic layers," Master's thesis, McGill University, Montréal, Québec, Canada, June 1998.

[49] R. L. Burden and J. D. Faires, *Numerical Analysis.* Brooks/Cole, California, United States. 6th ed., 1997.

[50] D. Wilcox, "Numerical laplace transformation and inversion," *International Journal of Electric Enging Education*, vol. 15, pp. 247–265, 1978.

[51] M.-J. Potvin, "A time-domain algorithm for the dynamic equations of robots modelled with fractional voigt-kelvin damping," Master's thesis, Queen's University, Kingston, Ontario, Canada, January 1994.

[52] D. J. Wilcox and I. S. Gibson, "Numerical laplace transformation and inversion in the analysis of physical systems," *International Journal for Numerical Method in Engineering*, vol. 20, pp. 1507–1519, 1984.

[53] J.-M. Dorlot, J.-P. Baïlon, and J. Masounave, *Des matériaux.* Montréal: École Polytechnique de Montréal, 2 ed., 1986.

[54] I. Tremblay, "Poutres couvertes de couches viscoélastiques," Master's thesis, École Polytechnique de Montréal, To be submitted in 2001.

[55] J.-C. Piedbœuf, L.L.Pagé, I. Tremblay, and M.-J. Potvin, "Efficient simulation of a multilayer viscoelastic beam using an equivalent homogeneous beam," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, (Detroit, Michignan), pp. 1188–1193, 10-15 May 1999.

[56] N. Makris, "Three-dimensional constitutive viscoelastic laws with fractional order time derivatives," *Journal of Rheology*, vol. 41, pp. 1007–1020. September-October 1997.

[57] R. H. Boyd, M. Robertsson, and J. F. Jansson, "The nature of the nonlinear anelasticity of glassy polymers," *Journal of Polymer Science : Polymer Physics Edition*, vol. 20, pp. 73–81, 1982.

[58] F. Povolo and O. A. Lambri, "Nonlinear mechanical spectrometry of poly (methyl methacrylate)," *Journal or Applied Polymer Science*, vol. 56, pp. 161–168, 1995.

[59] J.-P. Trotignon, J. Verdu, M. Piperaud, and A. Dobraczinski, *Précis de matières plastiques*. Paris: Nathan, 4th ed., 1988.

[60] G. Arfken, *Mathematical Methods for Physicists*. Orlando: Academic Press, 3rd ed., 1985.

[61] The MathWorks, South Natick, *Matlab*, October 1990.

Appendix A

The Numerical Laplace Transform

The chosen method to invert the Laplace transform is based on the use of the Fast Fourier Transform (FFT). The FFT is a method to calculate the Fourier transform which drastically reduces the number of operations required and speeds up the inversion considerably [60]. Thus, the first step to inverse the Laplace transform is to express it in terms of a transform pair of a discrete Fourier transform.

Wilcox [50] gives the following discrete transform pair for the Laplace transform:

$$\begin{aligned} F(s_K) &= \frac{t_f}{2n} \sum_{M=1}^{2n} f(t_M) e^{-s_K t_M} \\ f(t_M) &= \frac{1}{t_f} \sum_{K=-n, K\neq 0}^{n} F(s_K) e^{s_K t_M} \end{aligned} \tag{A.1}$$

where F is the Laplace transform and f the corresponding time domain solution. The constant t_f is the maximum time of integration, the final time, and it is normally taken to be 1.1 times larger than the time duration of interest because of inaccuracies as t tends toward t_f. The factor $2n$ is the number of points in the time domain. The sampling frequency in hertz is $2n/t_f$ and must respect Shannon's theorem: it must be at least twice as large as the bandwidth. The sampling points in the frequency domain are:

$$s_K = \begin{cases} a + i(2K-1)\pi/t_f & \text{for} \quad K = 1, \ldots, n \\ a + i(2K+1)\pi/t_f & \text{for} \quad K = -n, \ldots, -1 \end{cases} \tag{A.2}$$

where $i = \sqrt{-1}$ and $a = 2\pi/t_f$. The sampling points in the time domain are:

$$t_M = (2M-1)t_f/4n \quad \text{with} \quad M = 1, \ldots, 2n \tag{A.3}$$

The sums are performed using the FFT algorithm of *Matlab* which is based on the following transform pair [61]:

$$\begin{aligned} \mathcal{F}_{k+1} &= \sum_{m=0}^{N-1} \mathrm{f}_{m+1} e^{-i2\pi km/N} \\ \mathrm{f}_{m+1} &= \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{F}_{k+1} e^{i2\pi km/N} \end{aligned} \tag{A.4}$$

where N is the number of samples in the time domain.

The Wilcox pair must be tansformed to suit the *Matlab* pair. The first step is to recognize that $2n = N$. Thus the Wilcox pair becomes:

$$\begin{aligned} F(s_K) &= \frac{t_f}{N} \sum_{M=1}^{N} f(t_M) e^{-s_K t_M} \\ f(t_M) &= \frac{1}{t_f} \sum_{K=-N/2, K\neq 0}^{N/2} F(s_K) e^{s_K t_M} \end{aligned} \tag{A.5}$$

with:

$$t_M = (2M-1)t_f/2N \quad \text{with} \quad M = 1, \ldots, N \tag{A.6}$$

The sums in the *Matlab* pair are not preceded by a factor t_f. The time domain function f in the Wilcox pair is redefined as $t_f f(t_M)$:

$$\begin{aligned} F(s_K) &= \frac{1}{N} \sum_{M=1}^{N} (t_f f(t_M)) e^{-s_K t_M} \\ (t_f f(t_M)) &= \sum_{K=-N/2, K\neq 0}^{N/2} F(s_K) e^{s_K t_M} \end{aligned} \tag{A.7}$$

The index m of the *Matlab* pair goes from 0 to $(N-1)$, whereas in the Wilcox pair M goes from 1 to N. Hence, $M = m + 1$ and t_M becomes:

$$t_m = (2m+1)t_f/2N \quad \text{with} \quad m = 0, \ldots, N-1 \tag{A.8}$$

In the *Matlab* pair, the index k runs from 0 to $(N-1)$. For the values of s_K to remain the same, s_k must be written:

$$s_k = a + i(2k - (N-1))\pi/T \quad \text{with} \quad k = 0, ..., (N-1) \tag{A.9}$$

This is found by inspection. The values in the vectors F and s are the same as originally, even though k runs from 0 to $(N-1)$. The Wilcox pair becomes:

$$\begin{aligned} F(s_k) &= \frac{1}{N}\sum_{m=0}^{N-1}(t_f f(t_m))e^{-s_k t_m} \\ (t_f f(t_m)) &= \sum_{k=0}^{N-1} F(s_k)e^{s_k t_m} \end{aligned} \tag{A.10}$$

with s defined by Equation A.9 and t by Equation A.8. Multiplying s and t in the exponential results in:

$$e^{s_k t_m} = e^{(2m+1)(aT-i\pi(N-1))/(2N)}e^{ik\pi/N}e^{i2mk\pi/N} \tag{A.11}$$

The exponential term that depends only on m will be attached to $f(t_m)$ since f varies with m through the variation of t, and the exponential term depending uniquely on k will be attached to $F(s_k)$ for similar reasons. The Wilcox pair is now:

$$\begin{aligned} (e^{i\pi k/N}F(s_k)) &= \frac{1}{N}\sum_{m=0}^{N-1}(t_f e^{-(2m+1)(at_f-i\pi(N-1))/(2N)}f(t_m))e^{-i2\pi km/N} \\ (t_f e^{-(2m+1)(at_f-i\pi(N-1))/(2N)}f(t_m)) &= \sum_{k=0}^{N-1}(e^{i\pi k/N}F(s_k))e^{i2\pi km/N} \end{aligned} \tag{A.12}$$

The *Matlab* pair has a factor N attached to the second equation rather than the first as in the Wilcox pair. To correct this, N is attached to $F(s_k)$ and the pair becomes:

$$\begin{aligned} (Ne^{i\pi k/N}F(s_k)) &= \sum_{m=0}^{N-1}(t_f e^{-(2m+1)(at_f-i\pi(N-1))/(2N)}f(t_m))e^{-i2\pi km/N} \\ (t_f e^{-(2m+1)(at_f-i\pi(N-1))/(2N)}f(t_m)) &= \frac{1}{N}\sum_{k=0}^{N-1}(Ne^{i\pi k/N}F(s_k))e^{i2\pi km/N} \end{aligned} \tag{A.13}$$

The only difference with the *Matlab* pair described by Equation A.4 lies in the indices of f and F. *Matlab* indexes its vectors from 1. It can not take 0 as an index for a vector. The values of the indices are simply augmented of 1 without changing the values of the variables. Thus:

$$t_{m+1} = (2m+1)t_f/2N \quad \text{with} \quad m = 0, 1, ..., (N-1)$$

$$s_{k+1} = a + i(2k - (N-1))\pi/t_f \quad \text{with} \quad k = 0.1.....(N-1)$$

$$(Ne^{i\pi k/N} F_{k+1}) = \sum_{m=0}^{N-1} (t_f e^{-(2m+1)(at_f - i\pi(N-1))/(2N)} f_{m+1}) e^{-i2\pi km/N}$$

$$(t_f e^{-(2m+1)(at_f - i\pi(N-1))/(2N)} f_{m+1}) = \frac{1}{N} \sum_{k=0}^{N-1} (Ne^{i\pi k/N} F_{k+1}) e^{i2\pi km/N} \quad \text{(A.14)}$$

with $\mathcal{F}$ and f of the *Matlab* pair defined as:

$$\begin{aligned} \mathcal{F}_{k+1} &= (Ne^{i\pi k/N} F_{k+1}) \\ \mathrm{f}_{m+1} &= (t_f e^{-(2m+1)(at_f - i\pi(N-1))/(2N)} f_{m+1}) \end{aligned} \quad \text{(A.15)}$$

The function F is the Laplace transfer function written in the s space, but $\mathcal{F}$ is the function that must be invoked with *Matlab* inverse FFT. This function called *ifft* returns f whereas the required time response is f.

Appendix B

Simulation Data

B.1 Simulation Results with the Prony Series

Tables B.1, B.2, and B.3 show results from the Prony series simulations for a slewing beam covered by constrained viscoelastic layers.

B.2 Simulation Results with the Fractional Model

Tables B.4, B.5, and B.6 show results obtained with the fractional Voigt-Kelvin model for a slewing beam covered by constrained viscoelastic layers.

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.87	6.3	3.85	5.8
Log. Decrement	1.1×10^{-1}	9.8×10^{-2}	11	9.7×10^{-2}	12
Second Slew					
Natural Freq. (Hz)	3.67	3.78	3.0	3.80	3.5
Log. Decrement	6.9×10^{-2}	7.4×10^{-2}	7.2	6.4×10^{-2}	7.2
Third Slew					
Natural Freq. (Hz)	3.56	3.85	8.1	3.85	8.1
Log. Decrement	9.9×10^{-2}	1.0×10^{-1}	1.0	1.0×10^{-1}	1.0

Table B.1: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Five Term Prony Series

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.87	6.3	3.86	6.0
Log. Decrement	1.1×10^{-1}	9.5×10^{-2}	14	9.5×10^{-2}	14
Second Slew					
Natural Freq. (Hz)	3.67	3.78	3.0	3.81	3.8
Log. Decrement	6.9×10^{-2}	4.3×10^{-2}	38	7.1×10^{-2}	2.9
Third Slew					
Natural Freq. (Hz)	3.56	3.84	7.9	3.85	8.1
Log. Decrement	9.9×10^{-2}	9.5×10^{-2}	4.0	9.5×10^{-2}	4.0

Table B.2: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Six Term Prony Series

	Experimental	Constant Poisson's Ratio		Constant Bulk Modulus	
		Simulated	Diff. (%)	Simulated	Diff. (%)
First Slew					
Natural Freq.(Hz)	3.64	3.86	6.0	3.86	6.0
Log. Decrement	1.1×10^{-1}	9.1×10^{-2}	17	9.2×10^{-2}	16
Second Slew					
Natural Freq. (Hz)	3.67	3.83	4.4	3.83	4.4
Log. Decrement	6.9×10^{-2}	6.5×10^{-2}	5.8	7.5×10^{-2}	8.7
Third Slew					
Natural Freq. (Hz)	3.56	3.84	7.9	3.85	8.1
Log. Decrement	9.9×10^{-2}	9.3×10^{-2}	6.1	9.2×10^{-2}	7.1

Table B.3: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Seven Term Prony Series

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.90	3.64	7.1
Logarithmic Decrement	9.5×10^{-2}	1.1×10^{-1}	14
Second Slew			
Natural Frequency (Hz)	3.87	3.67	5.4
Logarithmic Decrement	1.4×10^{-1}	6.9×10^{-2}	103
Third Slew			
Natural Frequency (Hz)	3.87	3.56	8.7
Logarithmic Decrement	9.0×10^{-2}	9.9×10^{-2}	9.1

Table B.4: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Fractional Voigt-Kelvin Model and a *memmax* of 1700

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.90	3.64	7.1
Logarithmic Decrement	9.5×10^{-2}	1.1×10^{-1}	14
Second Slew			
Natural Frequency (Hz)	3.88	3.67	5.7
Logarithmic Decrement	1.3×10^{-1}	6.9×10^{-2}	88
Third Slew			
Natural Frequency (Hz)	3.87	3.56	8.7
Logarithmic Decrement	9.2×10^{-2}	9.9×10^{-2}	7.1

Table B.5: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Fractional Voigt-Kelvin Model and a *memmax* of 1100

	Simulated	Experimental	Difference (%)
First Slew			
Natural Frequency (Hz)	3.91	3.64	7.4
Logarithmic Decrement	9.6×10^{-2}	1.1×10^{-1}	13
Second Slew			
Natural Frequency (Hz)	3.91	3.67	6.5
Logarithmic Decrement	1.5×10^{-1}	6.9×10^{-2}	117
Third Slew			
Natural Frequency (Hz)	3.89	3.56	9.3
Logarithmic Decrement	9.0×10^{-2}	9.9×10^{-2}	9.1

Table B.6: Natural Frequencies and Logarithmic Decrements of the Simulation Using a Fractional Voigt-Kelvin Model and a *memmax* of 660