Optimized Finite Difference Schemes for Wave Propagation in High Loss Viscoelastic Material

by

Mark A. Hayner

Submitted to the Department of Ocean Engineering in partial fulfillment of the requirements for the degree of

Master of Science in Ocean Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1994

Author.....
Department of Ocean Engineering
April 12, 1994

Certified by.....
J. Robert Fricke
Assistant Professor
Thesis Supervisor

Accepted by.....
A. Douglas Carmichael
Chairman, Departmental Committee on Graduate Students

Optimized Finite Difference Schemes for Wave Propagation in High Loss Viscoelastic Material

by
Mark A. Hayner

Submitted to the Department of Ocean Engineering
on April 12, 1994, in partial fulfillment of the
requirements for the degree of
Master of Science in Ocean Engineering

Abstract

An efficient time domain method is needed to analyze problems containing high loss viscoelastic materials with complex geometry. One presently available method for low loss, constant η, materials is the *finite difference method with memory variables* attributed to Carcione et al., and Blanch et al. [2, 5]. In this method, the constitutive relation is approximated by a sum of decaying exponential functions with matched or *optimized* relaxation spectra. This allows the time domain convolutions, appearing in the constitutive relation, to be eliminated at the cost of additional field variables or "memory variables". The total number of field variables is roughly doubled for both 2D and 3D models with attendant increases in computer speed and storage requirements. Greater efficiencies are needed to make this method more competitive.

In this thesis, I investigate the idea of cancelling the error due to the constitutive relation approximations (optimization error) with the error due to the finite difference time domain model (discretization error) by a process referred to as *reoptimiztion.* Error cancellation is possible because the discretization error is completely predictable and can be accounted for during the optimization procedure. To this end, the work of Blanch et al. is extended to high loss factor materials with $\eta = O(1)$ by matching both real and imaginary parts of the complex modulus with experimental data and implemented in a finite difference model using a Predictor-Corrector scheme with 3-point centered spatial differencing. Using this model, the reoptimization process is compared to the optimization process for three cases: 1.) high loss factor, narrow band, moderate Courant number; 2.) high loss factor, wide band, moderate Courant number; and 3.) high loss factor, wide band, low Courant number. Results show that, for a given accuracy, if the reoptimization process is used, then the temporal and spatial step sizes of the finite difference model are roughly doubled for all three cases. This represents a decrease in model run time by 8 times in 2D and 16 times in 3D. The corresponding reduction in storage requirements is 4 times in 2D and 8 times in 3D.

Thesis Supervisor: J. Robert Fricke
Title: Assistant Professor

Acknowledgments

I would like to first thank my family, without their support and encouragement, I simply would not be here. Special thanks to my sister, Maryhelen, for grammatical editing of this beastly thing. I'm sure there are errors, still, but you don't need a new career. Thanks to my advisor, Rob Fricke, for his patient and steady guidance throughout this work and never refusing to read yet another revision. Finally, I would like to thank my girlfriend, Kathy, for *staying with me* through it all. How many deadlines did I miss? I'm essentially done, you know!

This thesis is dedicated to my Dad, Paul F. Hayner Sr., who forty years ago, gave up his chance to receive a Masters degree from MIT because the immediate needs and responsibility of eleven children were his priority.

Contents

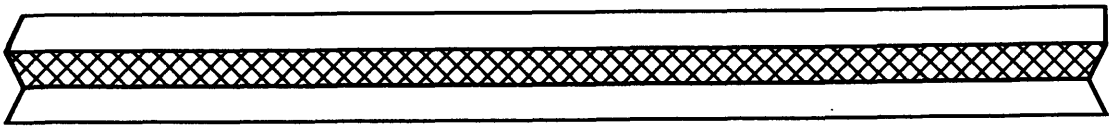
Chapter 1

Overview

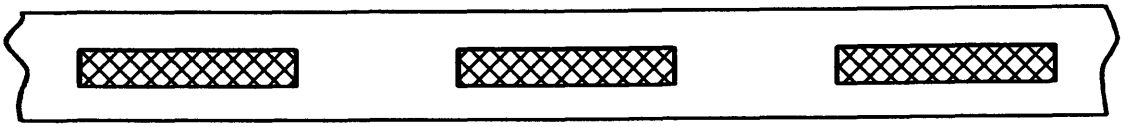
1.1 Introduction

A viscoelastic material is any material that dissipates energy when subjected to stress or strain. In accordance with the second law of thermodynamics, all materials are viscoelastic, which is ample motivation for the present study. Materials related to ocean engineering, known to exhibit overt viscoelastic effects under rather common conditions include arctic sea ice, seawater, and fluidized seabeds. Viscoelasticity is also important in the design of submarine hulls which are often covered with a layer of viscoelastic material to absorb interrogating sound waves. Clearly, viscoelasticity is of general interest and importance to all who study the ocean and ocean related technologies.

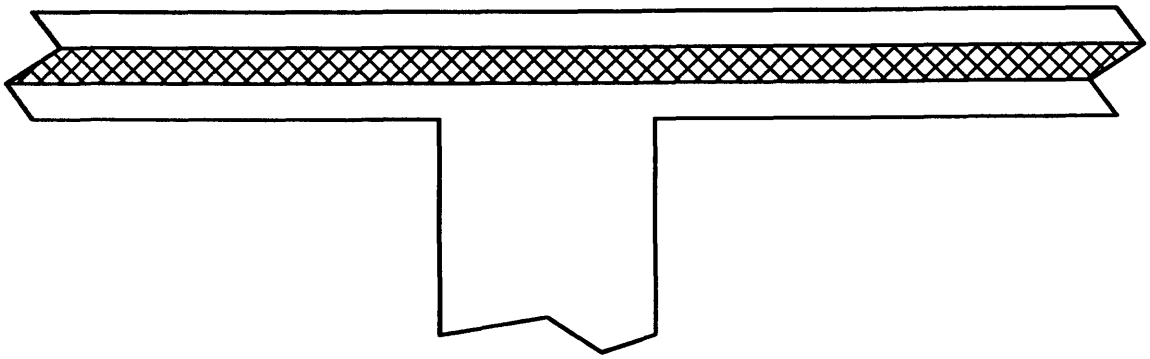
A problem of particular interest at MIT's Department of Ocean Engineering is to analyze how a constrained viscoelastic layer at the center of a submarine hull thickness can be used to reduce scattering noise levels due to shear waves. Four hull designs under consideration are shown in Figure 1.1. Each design includes regions of a high loss elastomer constrained by a pressure bearing steel shell. The basic constrained layer configuration is shown in Figure 1.1a. Figure 1.1b is configured to tradeoff dissipative characteristics with pressure bearing capacity by replacing areas of elastomer with steel. Figure 1.1c shows the conventional constrained layer design with added complexity due to a bulkhead. The final figure is a more radical approach to breaking up propagating shear waves with resonant sawtooth cavities. Notice that the geometries are, in general, complex, requiring a time domain numerical solution.



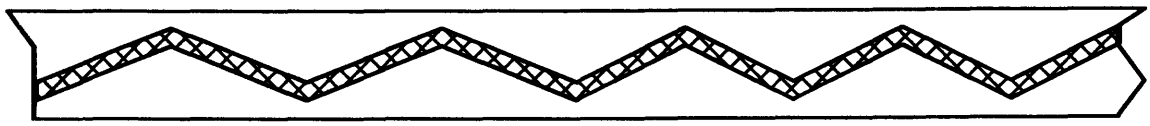
a.) Basic Constrained Layer Configuration



b.) Interspersed Elastomer



c.) Bulkhead Interface



d.) Jagged Layering

Figure 1.1

Cross section of various constrained layer submarine hull designs used to attenuate shear wave energy. Constrained layer is an elastomer with a loss factor $\eta = O(1)$. Constraining layer is made of steel.

More incentive for using a time domain numerical approach is that viscoelastic materials often exhibit material non-linearities due to temperature, ambient pressure, or large dynamic strain rates [17]. Because spectral methods rely on linear superposition, they are not suited for analyzing non-linear effects. In short, situations commonly arise that dictate the use of time domain methods.

The problem with time domain solutions is that the computational burden, in terms of both computer storage and run time, is immense. The problem lies in the fact that the constitutive relation for viscoelastic materials is a convolution in the time domain given in one dimension by

$$\tau(t) = G * \epsilon_{,t} , \tag{1.1}$$

where τ is the shear stress, G is the stress relaxation function, and $\epsilon_{,t}$ is the time derivative of the strain. Not only does the time convolution operator consume computer run time but the strain history at every point in the modeled domain must be stored, which requires preclusive amounts of memory, even for moderate size models. In 1988, J. Carcionne, D. Kosloff, and R. Kosloff developed the *memory variable* concept [5], which eliminated the convolution. In short, the convolution is traded for additional field variables referred to as memory variables. This created a resurgence in the use of time domain methods, the finite difference method among them. While the trade-off between the additional memory variables is a good one, the computational burden of time domain modeling for viscoelastic materials is still high. A comparison of the total number of field variables required for an elastic model compared with narrow band and wide band (about 2 octaves) memory variable models is presented in Table 1.1 using values generated in Section 2.7.3. For narrow band models, the number of field variables in 2D increases by 80% and in 3D by 67%. For wide band, the respective increases are 160% and 155%. Storage requirements increase directly with the number of field variables and computation time also increases (roughly) with the number of variables. These numbers indicate the need to improve the efficiency of these types of methods.

TABLE 1.1

TOTAL NUMBER OF FIELD VARIABLES REQUIRED FOR ELASTIC MODELING VERSUS VISCOELASTIC MODELING.

DIMENSION	ELASTIC	VISCOELASTIC, NARROW BAND	VISCOELASTIC, WIDE BAND
2D	5	9	13
3D	9	15	23

1.2 Objectives

In light of the importance of viscoelastic problems in the field of ocean engineering, the necessity to analyze these problems in the time domain, and the computational intensity of time domain solutions, the objective of this thesis is to increase the generality and computational efficiency of a popular time domain solution, namely, the *finite difference method with memory variables.* Specifically, the finite difference method with memory variables applied to low loss factor materials, $\eta = O(10^{-2})$, by Blanch et al. [2] is extended to high loss factor materials with $\eta = O(1)$. Computational efficiency is improved by a new process referred to as *reoptimization.* How these improvements are made is described in the following section. While the layered submarine hull problem is not analyzed in this thesis, the capability to do so is demonstrated in sufficient detail. The damped hull problem motivates the present study and is used as the context in which to interpret results. The focus of this thesis is on the development of a general and computationally efficient numerical tool.

1.3 A Few Definitions and Concepts

Before discussing the approach to generalizing and improving numerical efficiency of existing time domain methods, I need to introduce a few concepts and define a few terms relative to viscoelastic materials. I am always gratified to learn something new by recasting things I already know and there is opportunity to do that here. Let us begin with the hypothesis that the modulus for a viscoelastic material is complex in terms of frequency. If we consider the complex shear modulus then

$$\mu_s(\omega) = \mu' + j \cdot \mu'' , \tag{1.2}$$

where the real part, μ', is called the storage modulus and the imaginary part, μ'', is the loss modulus. Phase speed, frequency, and wavenumber are related by the same relationships as elastic materials (relationships we already know),

$$c_s = \frac{\omega}{k} = \left(\frac{\mu_s}{\rho} \right)^{\frac{1}{2}} , \tag{1.3}$$

except now all quantities can be complex. If ω is real, then k must be complex; if k is real, then ω is complex. We also know that a right going shear wave is given by

$$\tau = e^{j \cdot (\omega \cdot t - k \cdot x)} . \tag{1.4}$$

But if ω is real, then k is a complex function of ω and Equation 1.4 is

$$\tau(x,t) = e^{+\Im(k) \cdot x} \cdot e^{j \cdot (\omega \cdot t - \Re(k) \cdot x)} . \tag{1.5}$$

A viscoelastic wave must therefore exponentially decay (or grow) depending on the imaginary part of k and propagate at a phase speed depending on the real part of k. Both are functions of frequency and viscoelastic waves are, therefore, dispersive and dissipative.

A useful measure of energy dissipation, which is often referred to in this thesis, is the spatial loss factor η defined by

$$\eta = \frac{\mu''}{\mu'} . \tag{1.6}$$

If η <<1, Equations 1.2, 1.3, and 1.6 can be manipulated to show that $\Im(k) \rightarrow -|k| \cdot \frac{\eta}{2}$ and $\Re(k) \rightarrow |k| \rightarrow k$. In words, the phase speed is independant of the level of dissipation. Equation 1.5 can then be recast as

$$\tau(x,t) \approx e^{-\frac{1}{2} \cdot \eta \cdot k \cdot x} \cdot e^{j \cdot (\omega \cdot t + k \cdot x)} . \tag{1.7}$$

The decay has the same form as the free vibration of a damped single degree oscillator. In η wavelengths, the wave amplitude decays by $e^{-\pi \cdot \eta}$. For small η, there is no distinction between ordinary (temporal) loss factor and spatial loss factor. In general, they are space and time duels of each other. For those readers accustomed to using quality factor measures of dissipation, the spatial quality factor is the inverse of the loss factor. By hypothesizing that the modulus of a viscoelastic material is complex, and applying what we already know, we have uncovered the fundamental dispersive and dissipative character of viscoelasticity and have a quantitative understanding for the spatial loss factor.

Before proceeding to the next section, I need to define three different types of wave dispersion. All three are defined in terms of the basic dispersion relation, Equation 1.3. The first two are easily defined. If the complex modulus is given by experimental data, then Equation 1.3 is the *material* dispersion relation. If the modulus is given by an analytic model, Equation 1.3 is the *analytic* dispersion relation. Finally, there is *numeric* dispersion, which is analytic dispersion with

error due to the finite difference time domain model. The precise definition and nature of numeric dispersion is fully developed in Chapter 3.

1.4 Approach

This thesis is broken into three main tasks. The first task is to review the theoretical basis leading to the governing differential equations using memory variables. The second task is to extend the memory variable method used by Blanch for low loss materials to high loss materials. The third task is to present the *reoptimization procedure* including the theoretical basis for reoptimization and demonstration in a working time domain model. The first task is detailed in Chapter 2, the second and third are covered in Chapter 3.

The new concept of reoptimization is introduced here by first reviewing the "conventional" finite difference method with memory variables used by Blanch and then discussing how the method is modified to make it generally applicable and more efficient. This is merely an overview of the method, greater detail is given in Chapter 3. The "conventional" finite difference method with memory variables is shown in block diagram form in Figure 1.2. There are four components (blocks) of the method: the complex modulus experimental data, a complex modulus model, a least squares equation solver, and a finite difference model. The equation solver is used to *optimize* the parameters of the assumed complex modulus model with the data in terms of loss factor, η, over a preselected frequency band. The complex modulus model, in terms of the optimized parameters, is inverse Fourier transformed and incorporated into the finite difference model. Note that there are two models used here; the complex modulus which models the behavior of a point in the viscoelastic material; and the time domain model which models the behavior of a collection of points. Together they model waves in viscoelastic media[†]. As indicated in the Figure 1.2, there are two sources of error in the conventional method. The first is

[†] For brevity, I will at times refer to either the complex modulus model or the finite difference model as "the model". The context of the statement will make the distinction clear.

referred to as optimization error and is due to the finite number of complex modulus parameters. The second is referred to as discretization error and is due to the finite size of the temporal and spatial grid size of the finite difference model. In Chapter 3, the origin and nature of these error is described in detail.

In Figure 1.3, a new approach, referred to as the *finite difference method with reoptimized memory variable parameters* is illustrated. The approach is different than the conventional method in several significant ways. While the conventional method is used for low loss materials, the new method models high loss materials. To do this, it is necessary to optimize to both the real and imaginary parts of the complex modulus rather than their ratio, the loss factor. The ability to model high loss materials is, therefore, more general and needed to model the constrained layer submarine hull. The other important difference is the reoptimization process, the focus of this thesis. The basic idea, conceived jointly with my advisor, J. R. Fricke, is a simple one and its implementation requires only slightly more recurring work than standard optimization. In short, reoptimization is the process of cancelling discretization error with optimization error. Because the finite difference error is completely predictable for a given temporal and spatial step size, it can be accounted for during the optimization process. In practice, the reoptimization process is the same as the optimization process, only the numeric dispersion relation is substituted for the analytic dispersion relation. For reasons relating to the stability, it is often better to optimize first. Then the numeric dispersion relation is substituted for the analytic dispersion relation and, using the optimized parameters as an initial guess, optimization is carried out again, thus the term reoptimization.

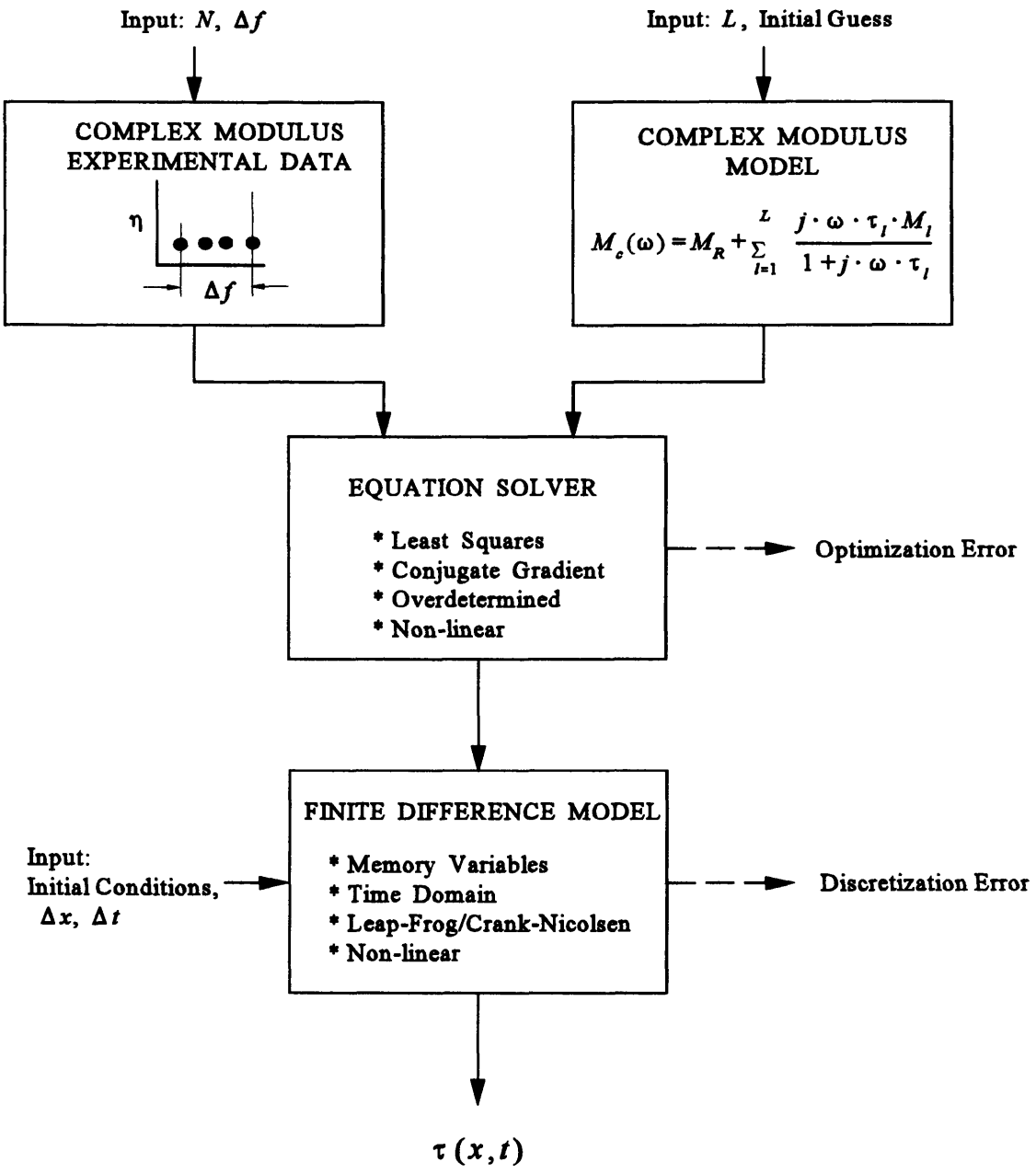


Figure 1.2

Block diagram of *Finite Difference Method with Optimized Memory Variables* used to simulate wave propagation in low constant η materials [Carcione et al. (1988), Blanch et al. (1993)]. Decaying exponential constitutive model parameters are matched to complex η data within a specified frequency band using a non-linear solver. The inverse Fourier transform of the complex modulus model is incorporated into a time domain finite difference model used to predict wave propagation in viscoelastic media.

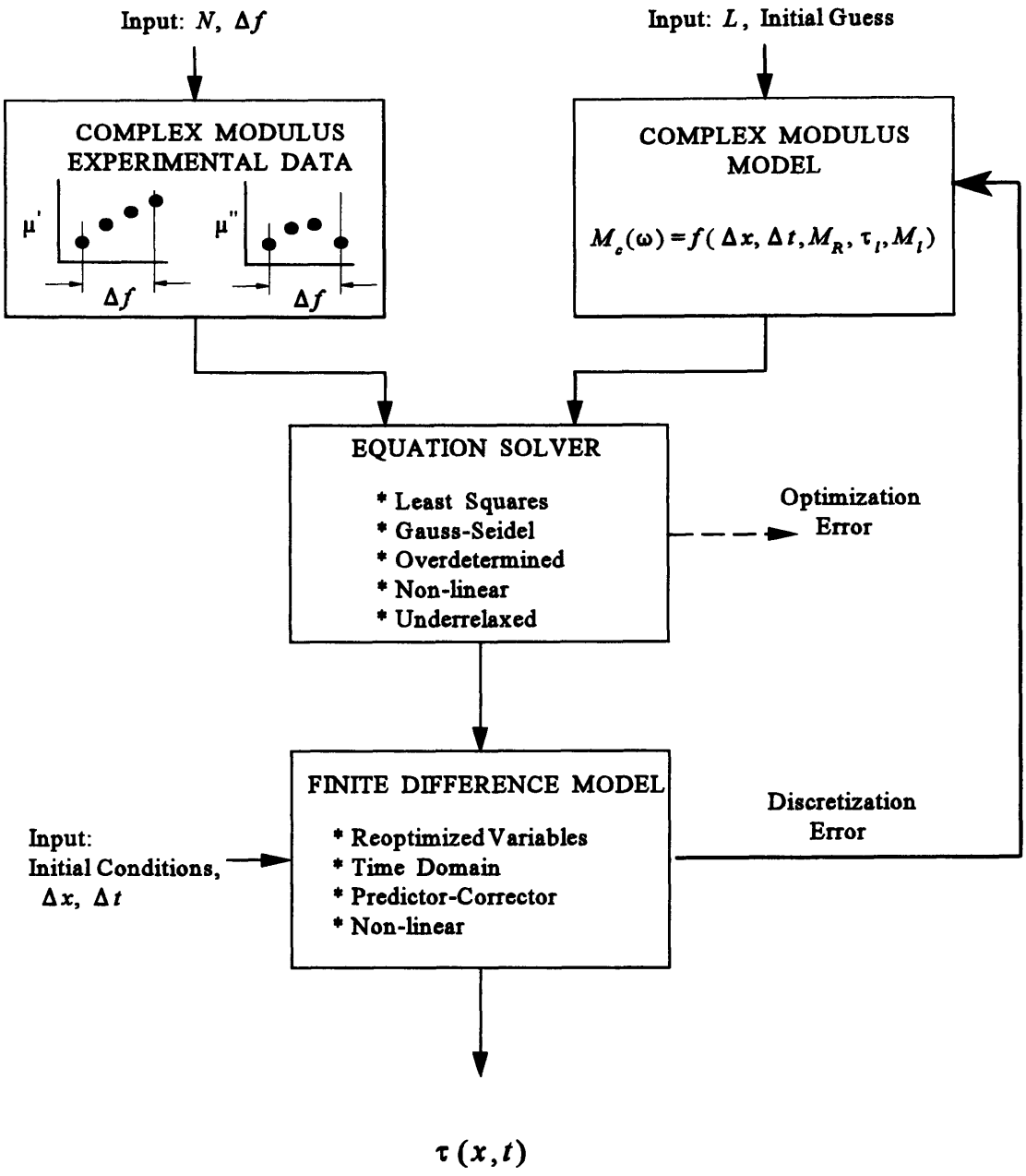


Figure 1.3

Block diagram of *Finite Difference Method with Reoptimized Memory Variables* used to simulate wave propagation in high loss η materials. Complex model parameters are matched to both real and imaginary parts of data within a specified frequency band using a non-linear solver. Because the complex modulus relation includes error due to the finite temporal and spatial step sizes of the finite difference model, discretization error is canceled by optimization.

Chapter 2

Viscoelastic Theory

2.1 Introduction

This chapter presents background information on viscoelastic theory needed to fully understand the reoptimization process presented in Chapter 3. I set out to accomplish three things. First, I discuss the general properties and nature of viscoelasticity in both the time and frequency domains, expanding on the concept and significance of complex modulus begun in Section 1.3. Second, I review four constitutive models which approximate the true constitutive relation including the decaying exponential model used by Carcione et al. The other models are included for completeness and comparison. Third, I review the memory variable formulation of the 1D and 3D viscoelastic wave equations. Introduction of these memory variables eliminates memory hogging convolutions required for time domain modeling. The 3D formulation is important to justify that conclusions based on 1D results are generally applicable. It is also presented to emphasize the computational intensity of time domain solutions to viscoelastic problems and the need to improve their efficiency.

2.2 Overview of Viscoelastic Materials

Viscoelasticity is a general term for materials that dissipate energy when subjected to stress and strain. All materials are dissipative to some degree; a condition linked to the second law of thermodynamics. Viscoelastic materials can be considered to be somewhere between an elastic solid, responding to stress and strain instantaneously but non-dissipative, and a fluid, dissipating mechanical energy as heat but unable to support a sustained load. Viscoelasticity is characterized by material memory; the material "remembers" and is forever responding to earlier loads. In the time domain, viscoelastic materials creep or relax and propagating waves decay and disperse. In the frequency domain, constitutive relations are functions of frequency.

Viscoelasticity is quantified by the loss factor, η, or alternatively its inverse, the spatial quality factor, Q. Both are measures of energy dissipation. As shown in Chapter 1, the amplitude and energy of free waves in viscoelastic media decay exponentially in proportion to η. Waves in low Q or high η materials decay rapidly. Stiffer materials tend to have low loss factors; soft materials tend to have high loss factors. Steel has an η of order .001, fiber reinforced polymers of order .01, and polymers of order one. Materials may have η higher than one; there is no limiting physical mechanism. However, they are typically not much higher than one. Because polymers have such high η, they are considered typical viscoelastic materials. I show in later sections that another way to express viscoelasticity is to allow the modulus to be complex, that is,

$$M \rightarrow M(\omega) = M' + j \cdot M'', \tag{2.1}$$

where M' is referred to as the storage modulus, and M'' as the loss modulus[†]. Elastic moduli are,

[†] I use the symbol M to refer to complex modulus in a general sense, either compressional or shear.

therefore, pure real numbers. The loss factor defined in terms of the complex modulus is

$$\eta = \frac{M''}{M'}. \tag{2.2}$$

The second law of thermodynamics requires that M' monotonically increase with frequency and that η be positive. Typical complex shear modulus data for a viscoelastic elastomer is shown in Figure 2.1. Shear loss factors are generally an order of magnitude greater than the extensional loss factor. The figure shows three distinct regimes: the glassy region at high frequencies and low temperatures; the rubbery region at low frequencies and high temperatures; and the glass-transition region between the other two where the most dramatic changes occur. Note that as the frequency goes to either infinitity or zero, the material approaches elastic behavior, since M' approaches constant values and $M'' \to 0$. The high frequency limit of the storage modulus is referred to as the infinite frequency modulus, M_∞, and the low frequency limit is the relaxed modulus, M_R.

Many environmental variables effect the η of elastomers: preload, age, chemical exposure, sunlight, static stress, and displacement amplitude [17]. Frequency and temperature generally have the strongest influence. For polymers there is a close inverse relationship between temperature on an absolute scale and frequency on a logarithmic scale. Therefore data is usually given as a function of these two parameters. Because of the strong dependance on displacement amplitude and temperature, viscoelasticity is commonly a non-linear process. In this thesis, I restrict the problem to be isothermal and linear but the modeling procedures developed are generally applicable.

2.3 Relaxation, Creep, and Complex Modulus

Early development of viscoelastic theory began with W. Weber's studies of creep in silk threads under a constant load [14]. I begin our more mathematical discussions of viscoelastic theory in the time domain, as well. My goal, in this section, is to show the relationships between relaxation,

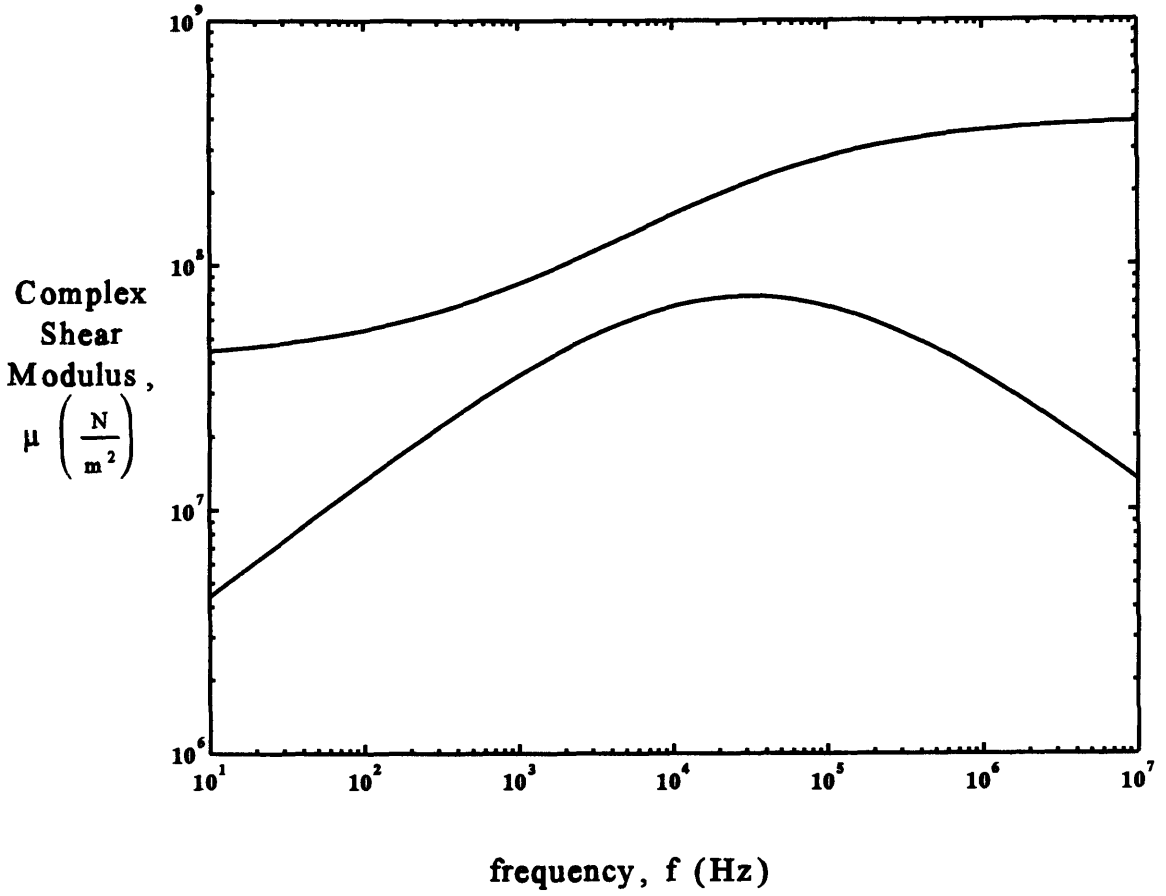


Figure 2.1

Typical complex shear modulus versus frequency for high loss elastomeric material with loss factor, $\eta \approx .5$. The top curve is the storage modulus, μ', and the bottom curve is the loss modulus, μ''. Note that in both the high and low frequency limits, the material approaches elastic behavior. The curves shown are generated using the fractional derivative model fit to data near 2 kHz (refer to Torvik and Bagley [17]).

complex modulus, and creep, which links the time and frequency domains. The derivations are based on those given by R. M. Christensen [6].

Consider the time domain response of a slender massless viscoelastic rod (a silk thread perhaps) to a sudden unit step strain as shown in Figure 2.2. Neglecting inertial effects, since it is massless, the second law of thermodynamics requires that the response, $G(t)$ for $t>0$, monotonically decrease with time to a some value greater than zero. The shape of the decaying stress curve is otherwise general. $G(t)$ is referred to as the step response function or relaxation function. The initial response, M_∞, is elastic with an asymptotic decay to either zero or some offset M_R, where M_∞ and M_R are the infinite and zero frequency moduli defined earlier. If M_R is zero the material is defined to be a fluid; otherwise it is a solid. This distinction is nebulous when the decay is very gradual.

I have shown the response of a viscoelastic material to a step function. How do we model the response to a general strain input? A general input strain function, $\epsilon(t)$, can be thought of as the sum of a series of delayed and scaled step strain functions as shown in Figure 2.3. The scale of each step is given by the slope of the strain at the corresponding delay time, $t=\tau$. The stress response is the sum of the product of each input step strain with the step relaxation function evaluated at $t=\tau$. For infinitesimal sample size, $\Delta\tau \to 0$, the stress response becomes a convolution,

$$\sigma(t) = \int_0^t G(t-\tau) \cdot \left(\frac{d\epsilon}{dt}\right)_{|t=\tau} d\tau \ . \tag{2.3}$$

This equation applies only if the input strain, $\epsilon(t)$, and its derivative are smooth, and $\epsilon(t)$ is zero for $t<0$. The convolution is expressed more compactly as

$$\sigma(t) = G * \epsilon_{,t} \ . \tag{2.4}$$

By Fourier transforming Equation 2.4, the expression for complex modulus is in terms of the

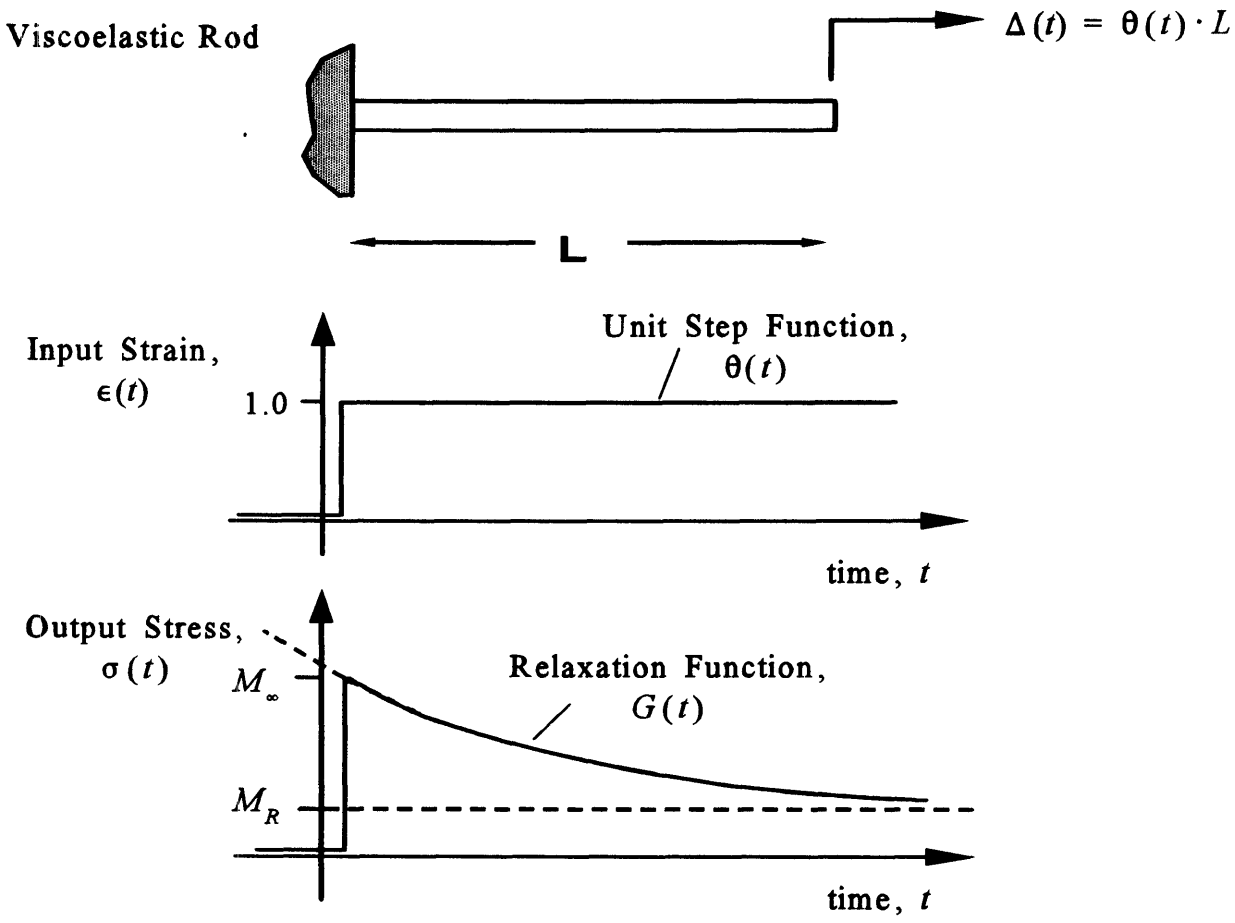


Figure 2.2

Relaxation response of viscoelastic rod to unit step strain input. The upper figure shows the loading of the rod; the middle figure is the unit step input strain, $\theta(t)$, plotted versus time; and the bottom figure is the output stress versus time. Since the input strain is a unit step function, the output stress is the relaxation function. The initial output response is elastic with magnitude, M_∞, which relaxes over time to the asymptotic value, M_R.

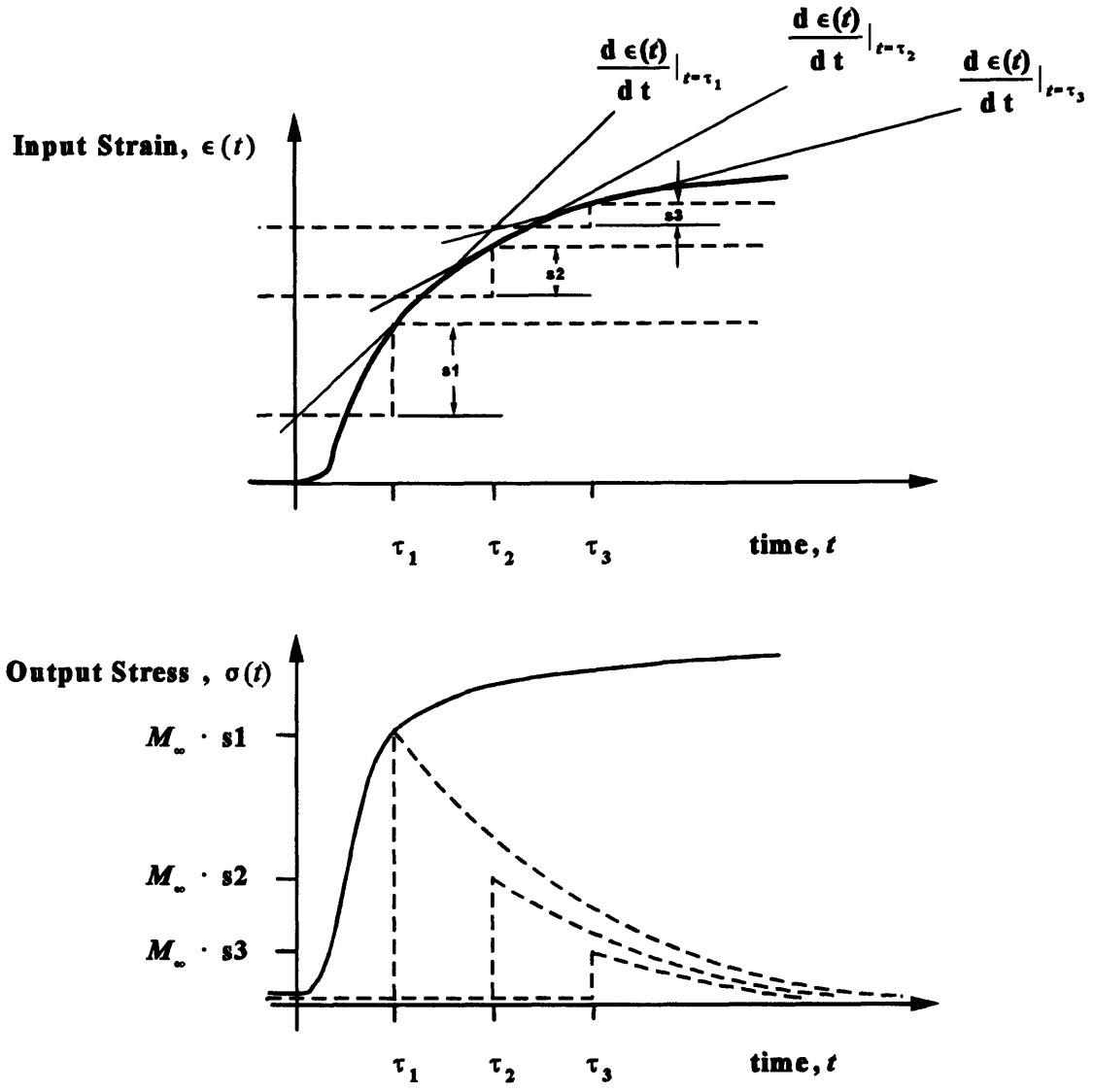


Figure 2.3

Schematic showing output stress response, $\sigma(t)$, to a general input strain function, $\epsilon(t)$. The input is constructed from a series of step functions delayed by time, $t = \tau_i$ and scaled by the increase in strain occurring since the previous step function, s_i. The output stress response is given by the sum of the product of the constructed series of step functions with the relaxation response of the material. For infinitesimally small step sizes, $\Delta\tau$, the summation becomes a convolution integral.

relaxation function is given by

$$\frac{\hat{\sigma}}{\hat{\epsilon}} = M_c(\omega) = j \cdot \omega \cdot \hat{G} \ , \tag{2.5}$$

where the hat overscript denotes the transformed variable. Using Initial and Final-Value Theorems for Fourier transforms, we get a formal verification that the initial and infinite time response to a step function are indeed M_∞ and M_R, respectively, that is,

$$M_\infty = \lim_{j\cdot\omega \to \infty} j \cdot \omega \cdot \hat{G} \tag{2.6}$$

and

$$M_R = \lim_{j\cdot\omega \to 0} j \cdot \omega \cdot \hat{G}. \tag{2.7}$$

It is now easy to show how the step relaxation function, $G(t)$, is related to the step creep function, $Y(t)$. By following the same arguments used previously it can be shown that the strain response to a general stress input is

$$\epsilon(t) = Y * \sigma_{,t} \ . \tag{2.8}$$

If we now Fourier transform Equation 2.8 and rearrange the terms, we get a new expression for complex modulus,

$$\frac{\hat{\sigma}}{\hat{\epsilon}} = M_c(\omega) = \frac{1}{j \cdot \omega \cdot \hat{Y}} \ . \tag{2.9}$$

By equating Equation 2.9 and Equation 2.5, the creep and relaxation functions are related by

$$\hat{G} = \frac{1}{(j \cdot \omega)^2 \cdot \hat{Y}} . \tag{2.10}$$

The creep and relaxation function are not inverses of each other but they are inversely related. In this thesis, the wave equations are formulated using the relaxation functions, but the choice is arbitrary.

2.4 Constitutive Models

In this section, I discuss the four constitutive models. By the end of this section, we see that two of them are special cases of the remaining two. I discuss each in terms of either the time domain, the frequency domain, or both depending on what information is needed in other sections of this thesis.

2.4.1 Decaying Exponential Model

The decaying exponential model is the constitutive model implemented in the time marching finite difference model. I discuss it in detail because it gives us insight into the nature of viscoelastic materials and the optimization process that is central to this thesis. The model is based on an expansion that is akin to the Fourier transform, except that the integration is along the real axis rather than the imaginary axis. More specifically, Golden and Graham show that it is possible to expand the stress relaxation function, $G(t)$, without error in a continuous spectrum of decaying exponentials given by

$$G(t) = \int_{-\infty}^{+\infty} M(\tau) \cdot e^{\frac{-t}{\tau}} \, d\tau, \tag{2.11}$$

where τ is the relaxation time spectra and $M(\tau)$ is the magnitude of the spectral content of

relaxation times [14]. The only restriction on the relaxation function, $G(t)$, is that it be causal and monotonically decrease. The relaxation spectra, τ , can be either positive or negative.

For a sampled response, $G(t)$ can be expressed exactly as an infinite sum of discrete relaxation spectra. By using a finite number of spectra, the sampled relaxation function is

$$G(t) = \left(M_R + \sum_{l=1}^{L} M_l \cdot e^{\frac{-t}{\tau_l}} \right) \cdot \theta(t), \tag{2.12}$$

where $\theta(t)$ is the Heaviside function, τ_l is the relaxation time, and M_l is its corresponding magnitude. The infinite τ_l term has been brought out front and identified as, our now familiar friend, the relaxed modulus, M_R. To insure monotonic decay, we will restrict ourselves to positive τ_l values.

It is possible, at this point, to use some optimization technique to determine the values of the M_l and τ_l parameters that would best fit the true response. Figure 2.4 shows schematically how Equation 2.12, with L=2, approximates the true relation function. Each term or "relaxation mechanism" decays at its own relaxation time τ_l and is scaled by M_l. Notice that the share of the initial infinite frequency response for each mechanism is M_l. This is expressed mathematically as

$$M_\infty = M_R + \sum_{l=1}^{L} M_l. \tag{2.13}$$

However, because the source used in the time domain model is band limited, a more accurate technique is to optimize in the frequency domain. To find an expression for complex modulus, we Fourier transform the approximate relaxation function, Equation 2.12, and substitute it into Equation 2.5, yielding

$$M_c(\omega) = M_R + \sum_{l=1}^{L} \frac{j \cdot \omega \cdot \tau_l \cdot M_l}{1 + j \cdot \omega \cdot \tau_l}. \tag{2.14}$$

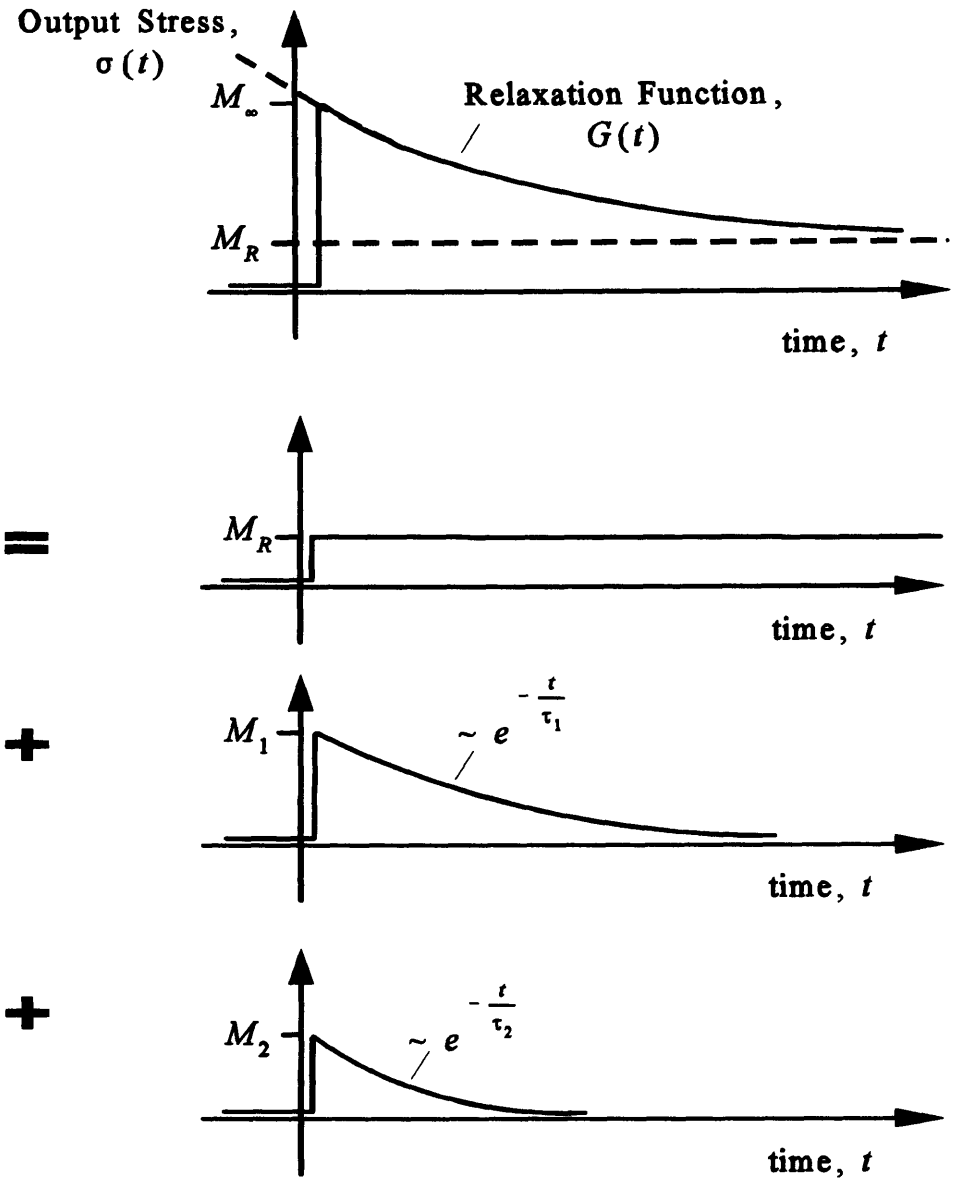


Figure 2.4

Schematic of a time domain synthesis of a general relaxation function for a viscoelastic solid, $G(t)$, from a series of three decaying exponential functions of the form, $M_l \cdot \exp(-t/\tau_l)$. The top curve shows the general relaxation function, $G(t)$, versus time. The first term in the series has an infinite relaxation time, $\tau_l \to \infty$, with a scale factor, M_l equal to the relaxed modulus of the material, M_R. The second and third terms have finite relaxation times and moduli as shown.

Equation 2.14, in terms of the real and imaginary parts, is

$$M'(\omega) = M_R + \sum_{l=1}^{L} \frac{\left(\frac{\omega}{\omega_l}\right)^2 \cdot M_l}{1 + \left(\frac{\omega}{\omega_l}\right)^2} \tag{2.15}$$

and

$$M'' = \sum_{l=1}^{L} \frac{\left(\frac{\omega}{\omega_l}\right) \cdot M_l}{1 + \left(\frac{\omega}{\omega_l}\right)^2}, \tag{2.16}$$

where ω_l is just the inverse of τ_l. The parameters, τ_l and M_l, can now be varied to determine the best fit between Equation 2.15 and Equation 2.16 and the real and imaginary parts of the complex modulus experimental data, within the frequency band of the source.

To get a better idea of how this might work, consider an approximation with only one term in the expansion and examine Equations 2.15 and 2.16 graphically, as shown in Figure 2.5. The mechanism "corners" at a frequency given by $\omega_l = 2 \cdot \pi \cdot f_l = 100\,\text{Hz}$. The loss modulus is greatest at the corner frequency and falls off toward zero at about 1 decade to each side. Over that same interval, the elastic modulus increases by M_l. Note the similarity in form between the model and the real data (Figure-2.1). We should expect this since the relaxation time functions have similar shapes. For both the experimental data and the model, the magnitude of the loss modulus is related to the slope of the storage modulus. Several sections in Chapter 3 are devoted to the optimization of the model parameters to experimental data. I postpone further discussion of the complex modulus matching until then.

One more note before moving ahead. The reader may recognize that the complex modulus expansion, Equation 2.14, has the equivalent circuit shown in Figure 2.6. The circuit components

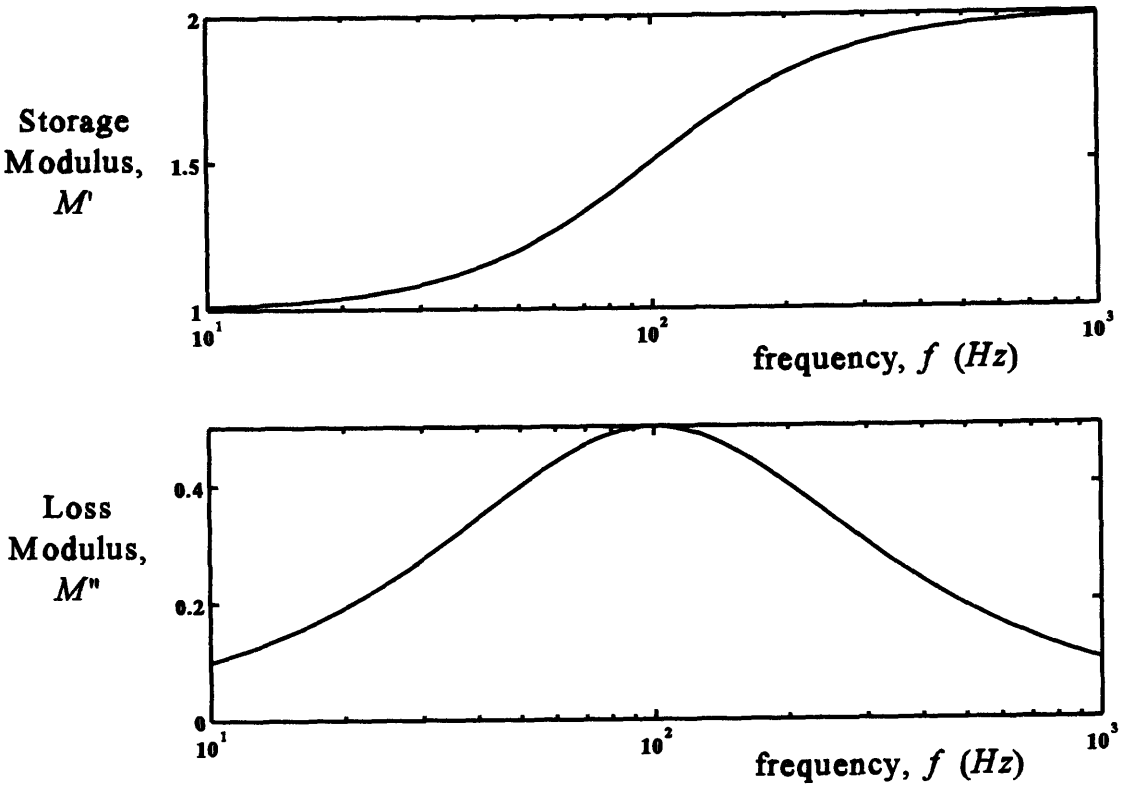


Figure 2.5

Complex modulus, M_c, versus frequency using decaying exponential model with one term in the expansion. The top and bottom curves are storage modulus and loss modulus, respectively. For this example, the relaxed modulus is $M_R = 1$, the infinite frequency modulus is $M_\infty = 2$, and the the corner frequency is $f_{cl} = \frac{1}{2 \cdot \pi \cdot \tau_l} = 100\,Hz$. Note that the shape of the curves are similar to those in Figure 2.1 for the complex modulus experimental data.

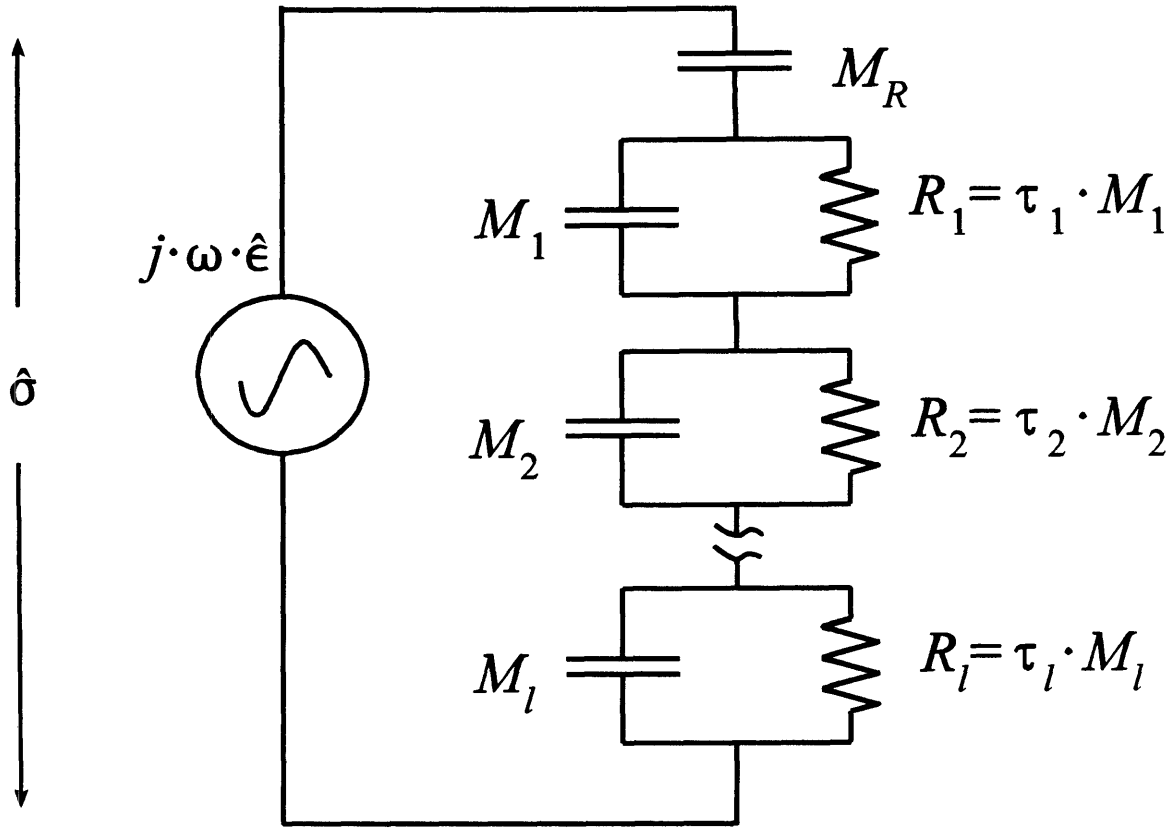


Figure 2.6
Equivalent circuit representation of the relaxation function using the decaying exponential model. The time derivative of the input strain is analogous to the input current; the output stress is analogous to output potential; and the parameters of the decaying exponential functions are analogous to circuit element impedances.

are a series of L resistor-capacitor pairs in parallel, where the capacitance is $C_l = \frac{1}{M_l}$ and the resistance is, $R_l = \tau_l \cdot M_l$. The output voltage (stress) is the response of the circut impedance (complex modulus) to the input current (time derivative of strain). The decaying exponential model is a bank of discharging capacitors. The electrical circuit analogy is used in the next section to motivate an extension to the viscoelastic model.

2.4.2 Mini-oscillator Model

The second of the four constitutive models is the Mini-Oscillator Model (MOT). Once equivalent circuit theory has been invoked and the circuit contains only resistive and capacitive elements, it doesn't take long before wondering, "Why not inductive elements?". Mactavish provided the answer [15]. He extended the resistor-capacitor (RC) model (or the decaying exponential model) to include an inductor element in parallel with the RC parallel combination. He is clever to require that each RLC combination be an overdamped oscillator, which guarantees a causal system. The relaxation function is now

$$G(t) = M_R \cdot \left[1 + \sum_{l=1}^{L} \alpha_l \cdot \left(\frac{b_{2l}}{b_{2l} - b_{1l}} \cdot e^{-b_{1l} \cdot t} - \frac{b_{1l}}{b_{2l} - b_{1l}} \cdot e^{-b_{2l} \cdot t} \right) \right], \quad (2.17)$$

where α_l is a scaling factor for each mini-oscillator. Parameters b_{1l} and b_{2l} (inverse relaxation times) are given by

$$b_{k1},\ b_{k2} = \omega_l \cdot [\ \zeta_l \mp (\zeta_l^2 - 1)^{1/2}\ ], \quad (2.18)$$

where the damping ratio is $\zeta_l > 1$, and ω_l is the natural resonance frequency of each RLC. Both the damping ratio and the natural frequency are defined in the usual sense for a simple oscillator. Restricting $\alpha_l, \zeta_l, \omega_l > 0$ ensures an overdamped, causal system.

The complex modulus for the MOT model is

$$M_c(\omega) = M_R \cdot \left[1 + \sum_{l=1}^{L} \alpha_l \cdot \frac{-\omega^2 + j \cdot 2 \cdot \zeta_l \cdot \omega_l \cdot \omega}{\omega_l^2 - \omega^2 + j \cdot 2 \cdot \zeta_l \cdot \omega_l \cdot \omega} \right]. \qquad (2.19)$$

Equation 2.17 shows that one term in the MOT model contains two decaying exponential terms. The magnitude and relaxation time for each exponential are not independant, however. Also, note that the MOT model allows one of the exponentials to have a negative magnitude. In the decaying exponential model, a negative magnitude (M_l) would violate the second law of thermodynamics (a rod in tension would contract). This extra degree of freedom can potentially result in a better fit with experimental data. The MOT model is, therefore, the more general application of Equation 2.11. It should be noted that the inductor or "mass" element is not physical mass but rather a convenient mathematical construct. Mactavish applied the MOT model to a finite element beam model.

2.4.3 Generalized Model

I introduce the third constitutive model, the generalized model, because it is a precursor to the fractional derivative model (yes, the generalized model is not most general) introduced in the next section. The derivation follows that given by Nashif [17]. We begin with the simple statement that stress is some function of strain,

$$\sigma(t) = f(\epsilon(t)). \qquad (2.20)$$

We have seen previously that stress and strain for viscoelastic materials are functions of time. Both sides of Equation 2.20 can, then, be expanded in a (Taylor-like) series about a point in time to give

$$\sigma + b_1 \cdot \frac{\partial \sigma}{\partial t} + b_2 \cdot \frac{\partial^2 \sigma}{\partial t^2} + \ldots = a_0 \cdot \epsilon + a_1 \cdot \frac{\partial \epsilon}{\partial t} + a_2 \cdot \frac{\partial^2 \epsilon}{\partial t^2} + \ldots , \qquad (2.21)$$

or more compactly,

$$\sigma + \sum_{m=1}^{\infty} b_l \cdot \frac{\partial^m \sigma}{\partial t^m} = a_0 \cdot \epsilon + \sum_{n=1}^{\infty} a_l \cdot \frac{\partial^n \epsilon}{\partial t^n} \; . \tag{2.22}$$

If we now assume harmonic solutions for both σ and ϵ, substitute them into Equation 2.22, and solve for $M_c = \frac{\delta}{\epsilon}$, then the complex modulus for the generalized constitutive model is

$$M_c(\omega) = \frac{a_o + a_n \cdot \sum_{n=1}^{N} (j \cdot \omega)^n}{1 + b_m \cdot \sum_{m=1}^{M} (j \cdot \omega)^m} \qquad \text{for M , N} \rightarrow \infty. \tag{2.23}$$

In practice, of course, a finite number of terms for M† and N are used. It is not required that M and N be equal. A problem with the generalized model is that even for moderate values for M and N, the matrix inversion process required to solve for the coefficients, a_n and b_m, is ill-conditioned and only the simplest of generalized models are used [17].

2.4.4 Fractional Derivative Model

The fourth and final constitutive model is the Fractional Derivative Model. The fractional derivative model is a more general form of the generalized derivative model and allows derivatives

† The integer M should not be confused with the script symbol for modulus, *M*.

to have fractional powers rather than just the integer powers. Equation 2.23 becomes

$$M_c(\omega) = \frac{a_o + a_n \cdot \sum_{n=1}^{N} (j \cdot \omega)^{\alpha_n}}{1 + b_m \cdot \sum_{m=1}^{M} (j \cdot \omega)^{\beta_m}}, \tag{2.24}$$

where α_n and β_m are real numbers. Of all the different models, the fractional derivative model is the most powerful and usually requires the fewest number of terms for a given accuracy. Usually, M and N are one. In some special cases the physics of the material on a molecular level has been shown to behave according to fractional derivatives [19]. Unfortunately, the time domain representation of fractional derivatives cannot be manipulated to eliminate the convolution integral. Therefore, the fractional derivative model is not a candidate for a time domain finite differencing formulation. I still find several uses for the fractional derivative model in this thesis. In Chapter 3, I actually optimize against a fractional derivative model of experimental data rather than the data itself. I am able to do this because the model is extremely accurate. There are a number of reasons I do this. A broad range of materials has already been characterized using fractional derivatives and results can be generalized in terms of the fractional derivative parameters. Also, the closed form fractional derivative relations are easier to handle than raw data and the model smooths the data. Furthermore, when it is more convenient to work with real wavenumber rather than real frequency, I use the model to easily transformation the data from real frequency to real wavenumber ($c_p(\omega) \rightarrow c_p(k)$). In future studies, I plan to use the fractional derivative model as a benchmark to test the generality of the decaying exponential model. The reader is referred to P. J. Torvik and D. L. Bagley for further details on the uses of this method [19].

2.5 Classical Formulation of the 1D Viscoelastic Wave Equations

In this section, I collect the constitutive equation with the momentum equation, defined in terms of stress and velocity, to form the classical 1D viscoelastic wave equations. The simplicity of these equations belie the insight that can be gained from them. By applying the von Neuman method, the dispersion relation is determined and much of the behavior of viscoelastic waves is revealed.

I begin by restating the constitutive relation presented in Section 2.3,

$$\sigma(t) = G * \epsilon_{,t} . \tag{2.25}$$

But for particle velocity u, $\epsilon_{,t} = u_{,x}$, Equation 2.25 can be written as

$$\sigma = G * u_{,x} . \tag{2.26}$$

Taking the time derivative of both sides of Equation 2.26 and collecting the result along with the conservation of momentum equation, results in the classical formulation of the 1D viscoelastic wave equation in terms of velocity and stress,

$$\begin{aligned} \sigma_{,t} &= G_{,t} * u_{,x} \\ u_{,t} &= \frac{1}{\rho} \cdot \sigma_{,x} , \end{aligned} \tag{2.27}$$

where where ρ is the material density. These two equations form a set of coupled equations governing the propagation of viscoelastic waves in one dimension. As discussed in Section 1.1, applying a numerical method, such as the finite difference method, to solve these equations

directly is not appealing because of the convolution. The memory variable method, presented in the following section, circumvents the convolution shown in Equation 2.27.

To gain more insight into Equation 2.27, I apply the von Neuman method, which assumes that both particle velocity and stress have harmonic solutions of the form

$$u,\ \sigma \sim e^{j \cdot (k \cdot x + \omega \cdot t)}, \tag{2.28}$$

where k is the wavenumber and ω wave frequency. Substituting these forms into the 1D wave equation, eliminating both σ and u, and using the previously derived expression for complex modulus,

$$M_c = (j \cdot \omega) \cdot \hat{G}(\omega), \tag{2.29}$$

results in an expression for the phase speed of sound, $c_p = \frac{\omega}{k}$, in a viscoelastic medium,

$$\left(\frac{\omega}{k}\right)^2 = \frac{M_c}{\rho}. \tag{2.30}$$

This is recognized as the ordinary definition for sound speed except either wavenumber and frequency (or both) are complex (since the modulus is complex). In all cases, this relation tells us how fast the wave travels and decays in both space and time. We can infer that higher frequencies travel faster, since the storage modulus is monotonically increasing, but not faster than $c_\infty = \sqrt{\frac{M_\infty}{\rho}}$ and not slower than $c_R = \sqrt{\frac{M_R}{\rho}}$. For low damping levels, the group speed is given by $c_g = \frac{\partial \omega}{\partial k}$. Equation 2.30 is also referred to as the dispersion relation.

2.6 Memory Variables Formulation of 1D Wave Equation

The memory variable method is, in short, a clever mathematical trick using the properties of exponentials and linearity to eliminate the computationally costly convolution appearing in the constitutive equation. In this section, the derivation of the viscoelastic field equation with memory variables for scalar waves is presented because it is instructive and fundamental to the present study. The memory variable method was conceived by Carcionne et al. but the following derivation offered by Blanch et al. is used because it is clearer and more complete.

If the relaxation function, G, is modeled using the decaying exponential model, then taking the time derivative of G and substituting it into the constitutive equation yields

$$\sigma_{,t} = \left[\left(M_R + \sum_{l=1}^{L} M_l \cdot e^{\frac{-t}{\tau_l}} \right) \cdot \delta(t) + \sum_{l=1}^{L} \left(\frac{1}{\tau_l} \cdot M_l \cdot e^{\frac{-t}{\tau_l}} \right) \cdot \theta(t) \right] * u_{,x} . \qquad (2.31)$$

By defining the L memory variables as

$$r_l = \left(\frac{1}{\tau_l} \cdot M_l \cdot e^{\frac{-t}{\tau_l}} \cdot \theta(t) \right) * u_{,x} \qquad \text{for } l=1, 2,...L , \qquad (2.32)$$

Equation 2.31 becomes

$$\sigma_{,t} = M_\infty \cdot u_{,x} + \sum_{l=1}^{L} r_l . \qquad (2.33)$$

Notice that the convolution has been removed from Equation 2.33. To remove the convolution from Equation 2.32, I take its time derivative,

$$r_{l,t} = \left[-\frac{1}{\tau_l} \cdot \left(\frac{1}{\tau_l} \cdot M_l \cdot e^{\frac{-t}{\tau_l}} \right) \cdot \theta(t) + \left(\frac{1}{\tau_l} \cdot M_l \cdot e^{\frac{-t}{\tau_l}} \right) \cdot \delta(t) \right] * u_{,x} \qquad (2.34)$$

$$\text{for } l = 1, 2, \ldots L \ .$$

Then by substituting Equation 2.32 into Equation 2.34 results in

$$r_{l,t} = -\frac{r_l}{\tau_l} + \frac{M_l}{\tau_l} \cdot u_{,x} \qquad \text{for } l=1, 2, \ldots L \ . \qquad (2.35)$$

Equation 2.35 and Equation 2.33, with the addition of the momentum equation, comprise a set of $2+L$ equations, referred to as the 1D viscoelastic wave equations with memory variables,

$$\sigma_{,t} = M_\infty \cdot u_{,x} + \sum_{l=1}^{L} r_l$$

$$u_{,t} = \frac{1}{\rho} \cdot \sigma_{,x} \qquad (2.36 \text{ a,b, and c})$$

$$r_{l,t} = -\frac{r_l}{\tau_l} + \frac{M_l}{\tau_l} \cdot u_{,x} \qquad \text{for } l=1, 2, \ldots L \ .$$

The memory variables can be thought of as a continuous correction made to the propagation of waves in an elastic material with modulus equal M_∞. Initially, if all memory variables are set to zero, the top two equations appear like their elastic counterparts. The third equation tracks the rate at which the infinite frequency elastic response should be corrected due to the collective effect of past loads and of any new loads (occurring in the past *dt* seconds). Any new strain loads represented by $u_{,x} = \epsilon_{,t}$ are scaled by M_l (which is the share of the new load associated with that particular memory variable) and registered by the second term on the right side of the memory variable equation. The first term on the right side tracks the rate of relaxation of both new and past loads associated with a particular memory variable.

2.7 Full 3D Viscoelastic Wave Equations

Previous discussion of viscoelastic wave behavior in 1D has allowed us to become familiar with four new concepts presented here in bullet form for clarity.

* First, the constitutive relation for viscoelastic waves in the time domain is a convolution of the time derivative of the strain with the relaxation function.

* Second, the modulus is a complex quantity.

* Third, we can hope to model the relaxation function with a series of decaying exponentials.

* Fourth, provided we use this model, the convolution can be eliminated with the introduction of memory variables.

In this section, I use our understanding of 1D viscoelasticity to help derive the full 3D viscoelastic wave equations which includes a full tensor description of the state of stress in a linear isotropic viscoelastic solid. I then proceed to derive the 3D wave equations using memory variables. The increase in the number of field variables emphasizes the need for improved numerical efficiency. I also show that the full 3D description can be decomposed into a set of four waves, one compressional and three shear, that behave like the 1D wave. Therefore, the discussions and

conclusions based on the 1D results are not only appropriate but applicable. The derivation of the classical viscoelastic wave equations follows that given by Christensen [6]; the memory variable formulation is again based on the work of Carcione et al. [5]; and the Helmholtz decomposition is given by Christensen [6] and J. D. Achenbach [1]. All derivations, as presented here, are intended to be more intuitive than rigorous.

2.7.1 Constitutive Equations

Consider the hypothesized stress-strain constitutive relation given by

$$\sigma_{ij} = G_{ijkl} * \epsilon_{kl} , \tag{2.36}$$

where Einstein's subscript notation is used to indicate individual tensor elements. We expect the convolution based on the 1D case. But now the relaxation function, G_{ijkl}, is a fourth order tensor which is the most general relation between two linear second order tensors. The fourth order tensor has, in general, 81 independent variables. By confining our considerations to isotropic materials and using arguments involving symmetry, coordinate system independence, compatibility of the strain field, and stain energy density, the relaxation tensor, G_{ijkl}, written in terms of just two independent relaxation functions, G_1 and G_2, is

$$G_{ijkl} = \frac{1}{3} \cdot (G_2 - G_1)\ \delta_{ij} \cdot \delta_{kl} + \frac{1}{2} \cdot G_1 \cdot (\delta_{ik} \cdot \delta_{jl} + \delta_{il} \cdot \delta_{jk}) , \tag{2.37}$$

where δ_{ij} is the Kronecker delta function. The constitutive relation is simplified if we first define the deviatoric stress and strain by subtracting the dilational stress and strain from the total stress and strain, that is,

$$s_{ij} = \sigma_{ij} - \frac{1}{3} \cdot \sigma_{kk} \qquad (i \neq j) \tag{2.38}$$

and

$$e_{ij} = \epsilon_{ij} - \frac{1}{3} \cdot \epsilon_{kk} \qquad (i \neq j) \ . \tag{2.39}$$

Then, by substituting these Equations 2.37, 2.38, and 2.39 into 2.36, we obtain the constitutive relations for a 3D isotropic viscoelastic solid,

$$\sigma_{kk} = G_2 * \epsilon_{kk,t} \ , \tag{2.40}$$

and

$$s_{ij} = G_1 * e_{ij,t} \ . \tag{2.41}$$

For the full 3D viscoelastic equations, there are now *two* convolutions required; one associated with dilational or bulk stress; and the other associated with deviatoric shear stress. In terms of the more conventional definitions of stress and strain,

$$\sigma_{ij} = \frac{1}{3} \cdot \delta_{ij} \cdot G_2 * \epsilon_{kk,t} + G_1 * \epsilon_{ij,t} \ . \tag{2.42}$$

By Fourier transforming Equation 2.40 and Equation 2.41, we obtain expressions for the complex bulk and shear moduli,

$$\hat{B} = j \cdot \omega \cdot \frac{1}{3} \cdot \hat{G}_2 \ , \tag{2.43}$$

and

$$\hat{\mu} = j \cdot \omega \cdot \hat{G}_1 \ . \tag{2.44}$$

The two relaxation functions, G_1 and G_2, are purposefully defined in Equation 2.37 to arrive at

these simple expressions for bulk and shear modulus. Equation 2.37 could be defined in terms of other relaxation functions to arrive at simple expressions for complex modulus, in terms of other commonly used moduli, such as, the *Lame'* constants, λ and μ. The bulk and shear moduli are chosen because the physical mechanisms that dictate material behavior are separate and distinct when expressed in terms of their dilational and deviatoric components.

2.7.2 Helmholtz Decomposition

We have seen consistently throughout this chapter that, in the frequency domain, the expressions for the moduli are the same as those for elastic materials, except that the moduli are frequency dependant. If we restrict our attentions to infinite domains (no boundaries, no boundary conditions), then the equations of motion in the frequency domain are, therefore, also the same. The equations of motion in terms of displacements, $\hat{q}_i$, are given by

$$(j \cdot \omega) \cdot \hat{\mu} \cdot \hat{q}_{k,jj} + [(j \cdot \omega) \cdot \hat{\lambda} + (j \cdot \omega) \cdot \hat{\mu}] \cdot \hat{q}_{j,jk} = -\rho \cdot \omega^2 \cdot \hat{q}_k \ . \qquad (2.45)$$

Since vector calculus operators holds true for complex as well as for real vectors, the Helmholtz decomposition can be applied to the above equation, thus

$$\hat{q}_i = \hat{\Phi}_{,i} + (\text{curl } \hat{\Psi})_i \ , \qquad (2.46)$$

where $\phi(x_i,t)$ is a scalar potential and $\psi(x_i,t)$ is a vector potential. Substituting this equation into the displacement equations of motion, Equation 2.45, and inverse Fourier transforming yields

four equations governing the propagation of one compressional wave and three shear waves,

$$\nabla^2 \phi = \frac{1}{c_p} \cdot \phi_{,tt}$$
$$\nabla^2 \psi_x = \frac{1}{c_s} \cdot \psi_{x,tt}$$
$$\nabla^2 \psi_y = \frac{1}{c_s} \cdot \psi_{y,tt} \qquad \text{(2.47a, b, c, d)}$$
$$\nabla^2 \psi_z = \frac{1}{c_s} \cdot \psi_{z,tt} ,$$

where the complex compressional and shear phase speeds are $c_p = \sqrt{\frac{\lambda + 2 \cdot \mu}{\rho}}$ and $c_s = \sqrt{\frac{\mu}{\rho}}$, respectively. Equation 2.47 shows that both the dilational and deviatoric waves are governed by four scalar 3D wave equations. For infinite media, the dilational wave is completely uncoupled, while the deviatoric waves remains coupled through the vector potential, $\psi(x_i, t)$. Therefore, results presented throughout this thesis, based on the analysis of the 1D scaler wave equation, are justified provided they are not changed by the dimension of the wave equation or the coupling between equations. The coupling restricts the number of solutions that satisfy Equation 2.47, but does not change the form and nature of those solutions. In other words, the set of solutions for ψ that satisfy Equation 2.47b, 2.47c, and 2.47d, is a subset of the solutions for ψ that satisfy any one of the individual equations. A physical system with boundaries only introduces more coupling between the waves, including coupling between the dilational and deviatoric waves. The dimension of the scalar wave equation is not significant because any (plane) wave traveling in some arbitrary direction in 3D space can be modeled by a 1D model with an appropriate rotation of the coordinate system. In addition to increasing our understanding of waves in 3D viscoelastic solids, the Helmholtz decomposition is important to the present work because it substantiates the earlier claim that my discussion and conclusions in terms of the simpler 1D wave are applicable to the general full 3D wave behavior.

2.7.3 Full 3D Wave Equation with Memory Variables

Before presenting the 3D viscoelastic wave equations with memory variables, it is useful to summarize the key steps in deriving the wave equatioin in 1D with the memory variables. It was shown that the convolution of the time derivative of the relaxation function with some other function , $G_{,t} * \{\ \}$, could be replaced by $M_\infty \cdot \{\ \} + \sum_{l=1}^{L} r_l$ where r_l is given by $r_{l,t} = -\frac{r_l}{\tau_l} + \frac{M_l}{\tau_l} \cdot \{\ \}$, provided G is approximated by the decaying exponential model.

Following these same steps, the constitutive relations for a viscoelastic solid, Equation 2.42, combined with the momentum equations resulting in the following abbreviated set of equations,

$$
\begin{aligned}
\sigma_{xx,t} &= B_\infty \cdot \Xi + \mu_\infty \cdot \left[2 \cdot u_{,x} - \frac{2}{3} \cdot \Xi\right] + \sum_{m=1}^{M} r_{1m} + \sum_{l=1}^{L} r_{11l} \\
r_{1m,t} &= -\frac{r_{1m}}{\tau_m} - \frac{B_m}{\tau_m} \cdot \Xi \\
r_{11l,t} &= -\frac{r_{11l}}{\tau_l} - \frac{\mu_l}{\tau_l} \cdot \left[2 \cdot u_{,x} - \frac{2}{3}\,\Xi\right] \\
\sigma_{xy,t} &= \mu_\infty \cdot (v_{,x} + u_{,y}) + \sum_{l=1}^{L} r_{12l} \\
r_{12l,t} &= -\frac{r_{12l}}{\tau_l} - \frac{\mu_l}{\tau_l} \cdot (v_{,x} + u_{,y}) \\
u_{,t} &= \frac{1}{\rho} \cdot (\sigma_{xx,x} + \sigma_{xy,y} + \sigma_{xz,z}) \ ,
\end{aligned}
\tag{2.48}
$$

where u, v, and w are the velocities in the x, y, and z directions respectively, and $\Xi = u_{,x} + v_{,y} + w_{,z}$ is the divergence of the velocity vector. For clarity only representative components are shown. For the full 3D case, there are a total of $N = 9 + 6 \cdot L + M$ field variables; 3 velocities, 6 independant components of stress, $6 \cdot L$ memory variables corresponding to the shear modulus, and M corresponding to the bulk modulus. In 2D, there are $N = 5 + 3 \cdot L + M$ field variables. These relations for the total number of field variables, N, were used to generate Table 1.1 presented in Chapter 1.

Chapter 3

Numerical Analysis

3.1 Introduction

The benefits of the *reoptimization process*, discussed in Chapter 1, are now described in detail and demonstrated in a working time domain model. To this end, Chapter 3 covers three areas: finite difference time domain modeling, the optimization and reoptimization process, and the benefits of the reoptimization process demonstrated using the time domain model. Section 3.2 covers topics relating to the finite difference model; origins of error, scheme selection, and practical implementation. Because optimization and reoptimization are very similar, both processes are covered in Section 3.3. The results of three case studies comparing optimized and reoptimized models under various bandwidths and Courant numbers are presented in Section 3.4. Results of these studies, the central results to this thesis, are discussed in Section 3.5.

3.2 Finite Difference Time Domain Modeling

This section covers what we need to know about the finite difference method to understand the process of reoptimization. One objective is to trace the sources of error and show how they can be predicted. The relationship we are looking for is the numeric dispersion relation, which dictates how viscoelastic waves travel on the finite difference grid. Another objective is to show that the finite difference scheme selected is properly implemented in a practical model. To avoid confusion between schemes, I make note now that it is the Predictor-Corrector scheme with 3-point centered spatial differencing that is ultimately implemented in the finite difference model.

3.2.1 Overview of the Finite Difference Method

The concept of the finite difference method, that the elastodynamics of a physical continuum can be approximated by considering a large but finite number of points, is intuitively gratifying. A distilled description of the finite difference method is that the operators of the differential equation governing the physical system at a given point can be approximated by a weighted sum of the field variables at that point and at judiciously selected surrounding points. This can be expressed generally for one dimension problems as

$$L[\phi_m] \sim \sum_{n=1}^{N} w_n \cdot \phi_n , \tag{3.1}$$

where L is the differential operator, ϕ_n is the field variable at location $x_n = n \cdot \Delta x$, and w_n is a weighting factor. This contrasts with the finite element method, where the governing differential equation is altered usually by integration over the physical domain and a solution, ϕ, to this new system is found that includes influences from all points within the physical domain. The salient features of the two methods are forever bound to these fundamental differences.

For more insight into the finite difference method, consider the three point centered approximation ϕ' (where the prime denotes spatial derivative) at the grid point m. Using Equation 3.1,

$$\phi_{mx} = w_{m-1} \cdot \phi_{m-1} + w_m \cdot \phi_m + w_{m+1} \cdot \phi_{m+1} + Error. \tag{3.2}$$

By expanding the function at the left and right most points in a Taylor series about the center point yields

$$\phi_{m-1} = \phi_m - \phi_m' \cdot \Delta x + \phi_m'' \cdot \frac{\Delta x^2}{2} - \phi_m''' \cdot \frac{\Delta x^3}{6} + \ldots, \tag{3.3}$$

and

$$\phi_{m+1} = \phi_m + \phi_m' \cdot \Delta x + \phi_m'' \cdot \frac{\Delta x^2}{2} + \phi_m''' \cdot \frac{\Delta x^3}{6} + \ldots . \tag{3.4}$$

If Equations 3.3 and 3.4 are substituted in Equation 3.3, then it is clear that $w_{m+1} = -w_{m-1} = \frac{1}{2 \cdot \Delta x}$ for the right and left side of Equation 3.3 to be generally equivalent. Furthermore, for Equation 3.2 to be true when $\Delta x \to 0$, the weight of the center point must be $w_m = 0$. This is the referred to as the *consistency* requirement. The weights are now all determined and the error is given by the remaining truncated terms in the expansions,

$$Error = 0 + \phi_m''' \cdot \frac{\Delta x^2}{6} + 0 + \phi_m''''' \cdot \frac{\Delta x^4}{120} \ldots . \tag{3.5}$$

Since the leading error term is proportional to Δx^2, the scheme is referred to as order two accurate in space or, in short, O(2). The leading error term governs the convergence of the finite difference approximation to the exact solution for increasingly smaller step sizes.

Another spatial differencing scheme that is considered is the five point centered scheme. Using the same procedure as above, the weights are

$$w = [\frac{1}{12}, -\frac{8}{12}, 0, \frac{8}{12}, -\frac{1}{12}] , \tag{3.6}$$

and the leading error terms are

$$Error = \frac{-\phi_m^{/////} \cdot \Delta x^4}{30} + \frac{-\phi_m^{/////} \cdot \Delta x^5}{180} + \ldots . \tag{3.7}$$

Note that the points closest to the center point have the greatest weight as expected and that five point centered stencil is O(4) accurate. I will discuss these error terms more extensively and look at them in a number of different ways in subsequent sections. I will also consider the additional complexity due to error from the approximation to the time derivative operator and error due to sets of differential equations with multiple field variables. Since I can approximate operators by selecting any points within the spatial and temporal domains, the number of possible schemes is infinite. Usually known schemes are used that have been shown to be computationally efficient and accurate. It is interesting to note that as the number of points used in the finite difference approximation approach all grid points within the domain, then the finite difference method approaches the pseudo-spectral method [18]. The spirit of the finite difference method is, however, to use only local points and schemes with moderate order accuracies, e.g., O(2) to O(6) accuracy.

3.2.2 The Predictor-Corrector Scheme

There are a great many finite difference schemes from which to choose and many factors to be considered when choosing one. The major concerns are stability, error, and computational speed. It is shown in the following section that, even for hyperbolic systems, the finite difference scheme can introduce both phase and magnitude error. Other considerations include two dimensional scheme isotropy, boundary conditions, material contrast, problem

size, and source type. No scheme treats all requirements satisfactorily. The analyst seeks to strike a balance among the most important factors.

A quick survey of a number of popular schemes was conducted. Since typical problem grid sizes are 500 by 500 with at least five field variables, implicit methods are not considered because they would require inversion of a 125,000 by 125,000 matrix at each time step. The explicit Lax-Wendroff method depends on cancellation of error due to temporal and spatial finite differencing of elastic waves and is thought to be inaccurate for modeling high loss materials. It also requires that 3 point centered differencing be used for spatial derivatives which was thought to be limiting (Blanch suggests that higher order spatial accuracies are needed). This method was also not used because the complexity of its dispersion relation would make results too difficult to generalize.

Thankfully, not all methods were discarded or this would be the end of this thesis. Two schemes were investigated in detail including a hybrid Leap-Frog/Crank-Nicolson (LFCN) scheme and Huen's Predictor-Corrector (PC). Blanch suggested LFCN, reasoning that the non-dissipative characteristics of Leap-Frog are favorable. The LFCN scheme was initially chosen but after running into difficulties related to parasitic modes, the PC scheme was ultimately used. Although the PC scheme is weakly unstable for elastic waves, it will be shown to be stable for high loss materials. Unless specified, Predictor-Corrector is the scheme discussed throughout this thesis and it is the scheme implemented in the two dimensional finite difference model.

The time evolution stencils for the two schemes are shown in Figure 3.1. All field variables, p[†], u, and each r_l are calculated at every grid point. For the LFCN scheme p and u are evolved in time using Leap-Frog, while Crank-Nicolson is used for the memory variable equations. Both LF and CN are O(2) accurate in time and therefore the overall scheme will not exceed that accuracy. The LFCN combination is used by Blanch but apparently in a different manner since the resulting dispersion relation is different than reported [2]. Blanch's

---

[†] The pressure and stress field variables are related by $p = -\sigma$.

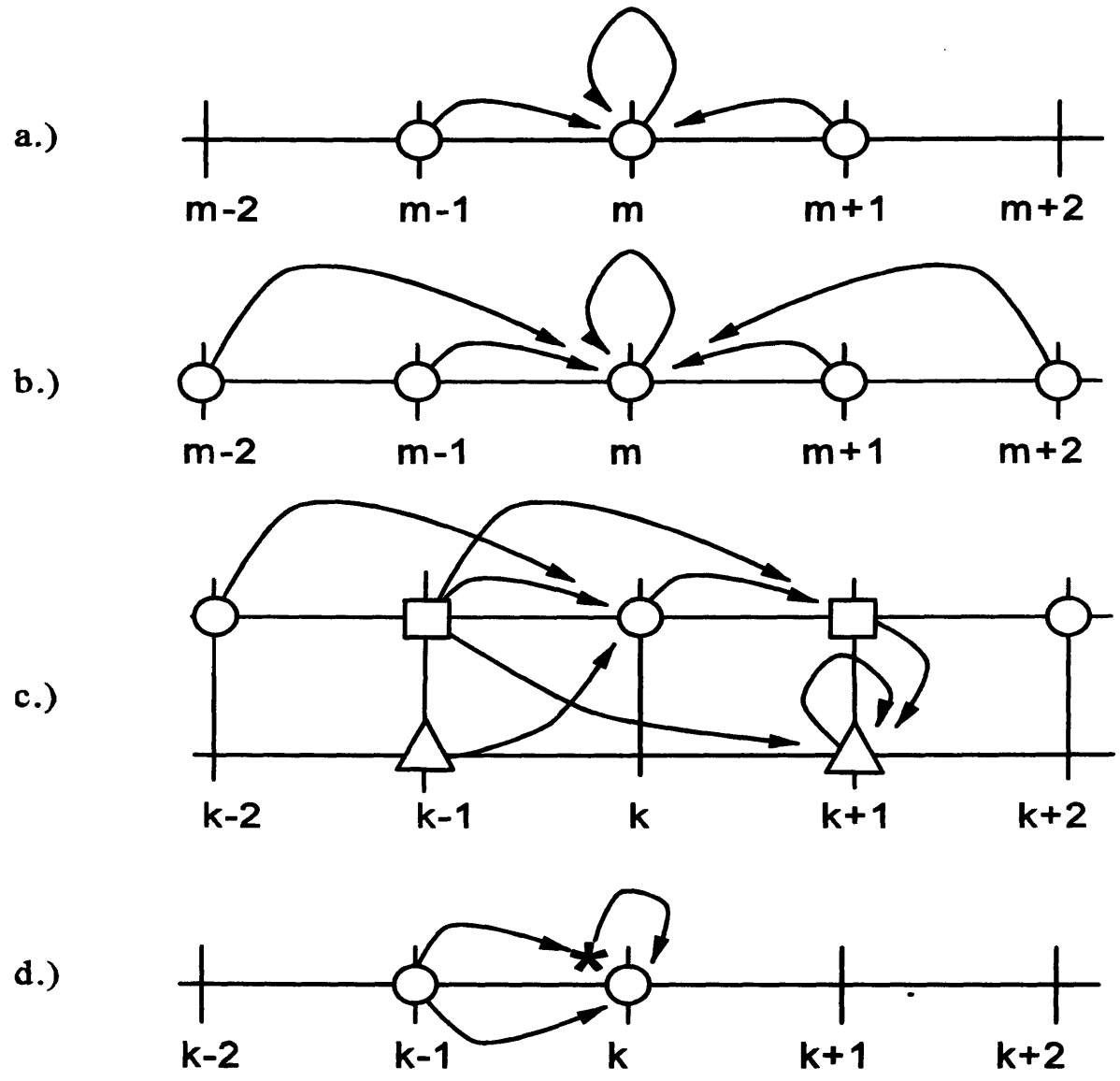


Figure 3.1

Various finite difference stensils considered in this thesis. Figure 3.1a is O(2) accurate three point centered stensil used for spatial integration. Figure 3.1b is O(4) five point stensil also used for spatial integration. Figure3.1c is hybrid Leap-Frog/Crank-Nicolsen used for O(2) accurate time integration. The pressure, velocity, and, memory field variables are symbolized by the circle, square, and triangle respectively. Figure3.1d is Huen's Predictor-Corrector O(2) accurate time integration scheme. The predictor (*) is Euler-Foward and the corrector is also Euler-Foward using the average of the old and predicted values.

reasons for using LF are that it is unconditionally stable and non-dissipative for the elastic wave equation. The scheme is also weakly unstable; for any attempt to add dissipation to LF renders it so [10]. A slight inconvenience of LF is that it requires a jump start. Some method must be used to determine the initial conditions of either u or p. Although CN is an implicit scheme, it is possible to show that the overall scheme is effectively explicit and does not require matrix inversion.

Both PC and LFCN are O(2) accurate in time. Both O(2) and O(4) spatial order of accuracies were considered corresponding to the three and five point centered spatial differencing discussed earlier. For the high loss factor modeling investigated in this thesis, the error turns out to be magnitude limited [†] and the three point scheme provides more than sufficient accuracy.

[†] Error between the exact and modeled solution that is correctable by scaling the modeled solution by some constant value is referred to as magnitude error, error that is correctable by phase shifting is phase error. When magnitude error is much worse than phase error, this situation is referred to as magnitude error limited. What constitutes "much worse" is a subjective judgement. Later, in Section 3.3.2, I subjectively define $\pi/4$ radians as the maximum acceptable phase error and a 1dB as the maximum acceptable magnitude error.

The finite differencing of the viscoelastic difference equations using the Predictor-Corrector scheme with 3 point centered differencing is straightforward and is demonstrated here by differencing the 1D viscoelastic wave equations given in Chapter 2. Euler-Foward is used for the predictor which gives

$$\frac{p_m^* - p_m^k}{\Delta t} = -M_\infty \cdot \frac{u_{m+1}^k - u_{m-1}^k}{2\cdot\Delta x} - \sum_{l=1}^{L} r_{l,m}^k$$

$$\frac{u_m^* - u_m^k}{\Delta t} = -\frac{1}{\rho} \cdot \frac{p_{m+1}^k - p_{m-1}^k}{2\cdot\Delta x} \tag{3.8}$$

$$\frac{r_{l,m}^* - r_{l,m}^k}{\Delta t} = -\frac{1}{\tau_l} \cdot r_{m,l}^k - \frac{M_l}{\tau_l} \cdot \frac{u_{m+1}^k - u_{m-1}^k}{2\cdot\Delta x} \qquad \text{for } l = 1 \ldots L \ ,$$

where the m and k subscript and superscript denote the field variable at spatial step m and time step k; all other variables have been previously defined. The corrector is also Euler-Foward except the average of the k and *-step values are used rather than just the k-step. The corrector finite difference equations are then

$$\frac{p_m^{k+1} - p_m^k}{\Delta t} = -\frac{M_\infty}{2} \cdot \left(\frac{u_{m+1}^k - u_{m-1}^k}{2\cdot\Delta x} + \frac{u_{m+1}^* - u_{m-1}^*}{2\cdot\Delta x} \right) - \sum_{l=1}^{L} \frac{1}{2} \cdot (r_{l,m}^k + r_{l,m}^*)$$

$$\frac{u_m^{k+1} - u_m^k}{\Delta t} = -\frac{1}{\rho} \cdot \left(\frac{p_{m+1}^k - p_{m-1}^k}{2\cdot\Delta x} + \frac{p_{m+1}^* - p_{m-1}^*}{2\cdot\Delta x} \right) \tag{3.9}$$

$$\frac{r_{l,m}^{k+1} - r_{l,m}^k}{\Delta t} = -\frac{1}{2\cdot\tau_l} \cdot (r_{l,m}^k + r_{l,m}^*) - \frac{M_l}{2\cdot\tau_l} \cdot \left(\frac{u_{m+1}^k - u_{m-1}^k}{2\cdot\Delta x} + \frac{u_{m+1}^* - u_{m-1}^*}{2\cdot\Delta x} \right) \qquad \text{for } l = 1 \ldots L \ .$$

These are the equations that are implemented in the time domain finite difference model.

3.2.3 Two Dimensional Time Marching Model

A two dimensional finite difference model was written to simulate the propagation of scalar waves in a homogeneous isotropic viscoelastic medium. The model is written using *Matlab* and uses the Predictor-Corrector scheme described earlier. The PC scheme, as implemented, uses the centered three point stencil for spatial derivatives and is, therefore, second order accurate in space and time. Periodic boundary conditions are imposed to eliminate errors introduced by boundaries [3], since periodic boundary conditions are effectively a boundary-less system. Energy is injected into the model using initial conditions for either a sinusoidal wave or a modulated Gaussian pulse. The model is described in Appendix A.2 with actual computer codes used given in Appendix A.4. For this thesis, all physical domains modeled are one dimensional.

Convergence tests are run on the model to verify that the finite difference scheme is correctly implemented. Initial conditions are set to establish a sine wave traveling in the x-direction. The model is run using progressively finer mesh sizes that are two times finer in both space and time while keeping the physical size and final time constant. The expected error between successive runs should approach four because the scheme is O(2) accurate in both space and time. Model convergence tests are conducted for both elastic and viscoelastic wave propagation. Results, presented in Figure 3.2 and Figure 3.3, show that the error between successive runs approaches a factor of 4 times which verifies that the model is correctly implemented. Note that in both figures the model results and the results predicted based on the numerical dispersion relation are indistinguishable (hashed line is on top of solid line) because the boundary-less model has only numerical roundoff error. The results for a wave traveling in the y-direction are identical.

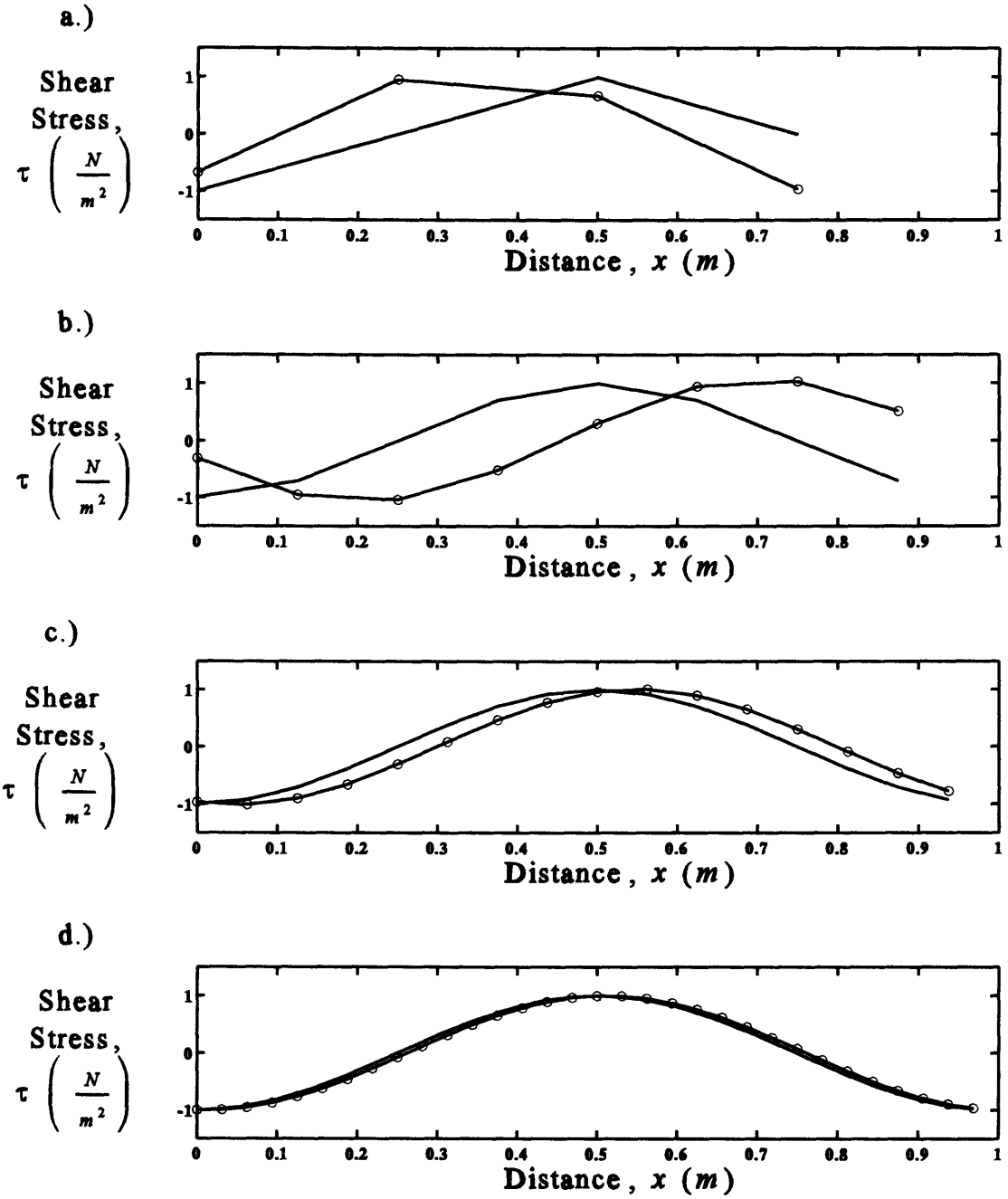


Figure 3.2

Model convergence tests for elastic wave propagation. Circles mark finite difference model result, hashed line is predicted result based on numerical dispersion relation, and solid line is result based on analytic dispersion relation. Figures a.) thru d.) are results using progressively finer mesh sizes; each figure is two times finer in both space and time than the preceding figure while maintaining a constant physical size and final time. Average rms error between model and analytic prediction for each run is 71, 88, 21, and 5.3%.

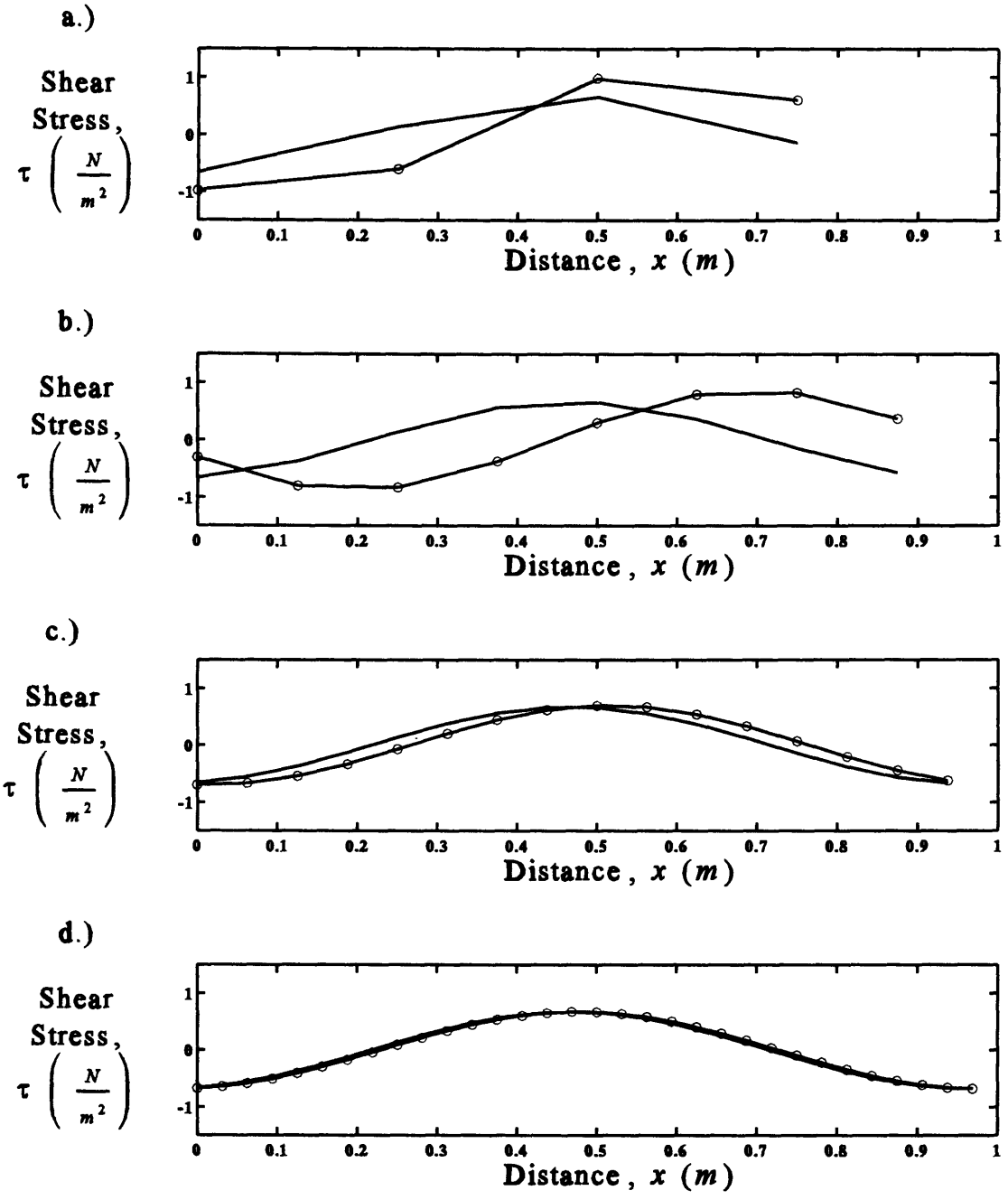


Figure 3.3

Model convergence tests for viscoelastic wave propagation. Circles mark finite difference model result, hashed line is predicted result based on numerical dispersion relation, and solid line is result based on analytic dispersion relation. Figures a.) thru d.) are results using progressively finer mesh sizes; each figure is two times finer in both space and time than the preceding figure while maintaining a constant physical size and final time. Average rms error between model and analytic prediction for each run is 87, 110, 22, and 5.1%.

3.2.4 Numeric Dispersion and Dissipation

It has been said that finite difference equations have a life of their own and in the following paragraphs we will see what is meant by this statement. In this section, I describe and discuss the derivation of the dispersion relations for the both the differential and finite differenced viscoelastic wave equations in 1D. The dispersion relations hold the secrets of how waves behave. As has been the theme throughout, the 1D study sufficiently describes the important phenomena of wave behavior for the more general 2D and 3D cases. A closed form solution of a dispersion relation is presented to bring out what non-dimensional quantities control dispersion. I will then examine the dispersion of elastic waves graphically in non-dimensional frequency-wavenumber space to demonstrate most dramatically how finite differencing alters wave behavior. The ω-k space gives a complete picture of wave behavior including error and stability.

I often refer to three different types of dispersion relations, which I define here. The first is the *material* dispersion relation which is the true (measured) physical dispersion relation of the material being modeled. The *analytic* dispersion relations is defined as the dispersion relation for a scheme with infinitesimal step sizes. In general, the material and analytic dispersion differ since the number of memory variables used is finite and the matching or *optimization* process between the complex modulus of the model and the material is not perfect. The *numeric* dispersion is the dispersion relation for the finite difference scheme and includes two types of errors. One type due to the matching process (the same errors as the analytic dispersion) and the other due to finite temporal and spatial step sizes of the finite difference method. In the section on optimization, I investigate the notion of allowing these two types of error to cancel! This is referred to as *reoptimization.*

One of the important advantages of finite difference methods over other methods, such as finite elements, is that dispersion errors can be determined *a priori* using the von Neumann method. To determine the analytic dispersion relation, all field variables are assumed to be proportional to $e^{j\cdot(k\cdot x-\omega\cdot t)}$, where ω is the frequency and k is the wavenumber. By

substituting the assumed form of the field variables into the viscoelastic differential equations, Equations 2.35, and taking temporal and spatial derivatives as required, results in the following system of equations,

$$\omega \cdot \begin{bmatrix} p \\ u \\ r_1 \end{bmatrix} = j \cdot \begin{bmatrix} 1 & -k \cdot M_\infty & -1 \\ -\frac{k}{\rho} & 1 & 0 \\ 0 & \frac{-k M_1}{\tau_1} & 1-\frac{1}{\tau_1} \end{bmatrix} \cdot \begin{bmatrix} p \\ u \\ r_1 \end{bmatrix}. \tag{3.10}$$

A similar set of equations results if more memory variables are used. This equation has the form of a standard eigenvalue problem with complex eigenvalues equal to radial frequency ω. For one dimensional problems with one memory variable, there are two eigenvalues corresponding to a left and right going decaying wave and one corresponding to a stationary decaying wave. The eigenvectors, which are in general complex, give the relative phase and amplitudes between the field variables. These eigenvalues and their corresponding eigenvectors are used in the finite difference model to determine initial conditions for a cosine modulated Gaussian source.

To determine the numerical dispersion relation, the discretized field variables are assumed proportional to $e^{j\cdot(k\cdot m\cdot \Delta x-\omega\cdot n\cdot \Delta t)}$. These are then substituted into Equations 3.9 resulting in the system of equations,

$$\Phi^* = [\, I + A \,] \cdot \Phi^k , \tag{3.11}$$

where $\phi = [p, u, r_1]^T$ the superscript (*) denotes the predictor value, the k superscript denotes the *kth* time step, the matrix I is the identity matrix, and the matrix A is given by

$$A = j \cdot \begin{bmatrix} 0 & \frac{-M_\infty \cdot \Delta t \cdot j \cdot \sin(k \cdot \Delta x)}{2 \cdot \Delta x} & -\Delta t \\ \frac{-\Delta t \cdot j \cdot \sin(k \cdot \Delta x)}{2 \cdot \rho \cdot \Delta x} & 0 & 0 \\ 0 & \frac{-M_1 \cdot \Delta t \cdot j \cdot \sin(k \cdot \Delta x)}{2 \cdot \tau_1 \cdot \Delta x} & -\frac{\Delta t}{\tau_1} \end{bmatrix} . \qquad (3.12)$$

The corresponding set of equations for the both the predictor and the corrector steps is

$$e^{-j \cdot \omega \cdot \Delta t} \cdot \Phi^k = [\, I + A + .5 \cdot A^2 \,] \cdot \Phi^k \, . \qquad (3.13)$$

Like the derivation for the analytic dispersion relation, we arrive at a system of equations in the form of an eigenvalue problem and the earlier discussion of eigenvalues and eigenvectors apply. They are, of course, different than their analytic counterparts but approach each other in the limit of infinitesimally small step sizes. Equation 3.13 is solved using standard matrix eigenvalue solution routines. In this form, the dispersion relation is easily solved for complex frequency, ω, given real wavenumber, k (not to be confused with time step, k).

It is possible to solve for the numerical dispersion relation in closed form for certain schemes, such as LFCN, but not PC. This is done by eliminating all L+2 field variables from the L+2 equations, which leaves an equation relating ω and k in terms of the material properties modeled in terms of the memory variable parameters and the temporal and spatial step sizes, in short, the dispersion relation. When LFCN is applied to model the 1D wave equation, the dispersion relation is

$$\sin^2(\bar{\omega}) = \sin^2(\bar{k}) \cdot {N_C}^2 \cdot \left[1 - \sum_{l=1}^{L} \frac{\bar{M}_l}{1 - j \cdot \bar{\tau}_l \cdot \tan(\bar{\omega})} \right], \qquad (3.14)$$

where dimensionless radial frequency is $\bar{\omega}=\omega \cdot \Delta t$, dimensionless wavenumber is $\bar{k}=k \cdot \Delta x$, and the Courant number is $N_C = \frac{c_\infty \cdot \Delta t}{\Delta x}$. Also, the dimensionless memory variable moduli are

$\bar{M}_l = \frac{M_l}{M_\infty}$ and the dimensionless relaxation times are $\bar{\tau}_l = \frac{\tau_l}{\Delta t}$. A similar relation was given by Blanch when only one memory variable is used.

In the limit as $\omega \to \infty$, the summation terms inside the square brackets go to zero and Equation 3.14 becomes the dispersion relation for the elastic wave equation using Leap-Frog with a sound speed equal to the unrelaxed sound speed. The summation terms, therefore, represent the viscoelastic effects. As $\Delta x \to 0$ and $\Delta t \to 0$, the numerical dispersion approaches the theoretical dispersion relation,

$$\rho \cdot \frac{\omega^2}{k^2} = \left(M_R + \sum_{l=1}^{L} \frac{j \cdot \tau_l \cdot M_l}{1 + j \cdot \omega \cdot \tau_l} \right) . \qquad (3.15)$$

For real ω, the right hand side of Equation 3.15 is recognized as the original expression for complex modulus, and we have come full circle. Note that the closed form dispersion relations clearly show that waves propagate with radial frequency ω, which is in general complex, and that they are non-linear in frequency and wavenumber and must be solved iteratively to find either one in terms of the other.

The closed form dispersion relation for LFCN has been useful to see what dimensionless variables govern dispersion. In practice, when the full wave equations are used a closed form dispersion relation is too difficult to derive and no further insight could be gleaned from it even if it could be derived. I now return to discussing the PC scheme, which is our principal interest, where dispersion is determined numerically.

To understand the dispersion relations better, let's graphically compare the analytic dispersion to the numeric dispersion for the PC scheme for the one dimensional case. To keeps things clear and simple, I restrict our discussions to real wavenumber. Also, I consider only elastic waves to emphasize that the dispersion and dissipation error is due to the finite differencing In a continuum, the elastic material dispersion relation is two infinitely extended straight lines which intersect the ω-k origin and have slopes of +/- c. For a material with sound speed c=.5, the elastic material dispersion in ω-k space is shown in Figure 3.4. The real part of ω represents the phase speed of the wave related by the slope of the curve. The positive

sloping curve is a right traveling wave and the negative slope curve is a left traveling wave. The imaginary part of ω is, of course zero, so there is no amplitude decay. The group speed is given by $c_g = \frac{\partial \omega}{\partial k}$ which is equal to the phase speed, c, in the elastic case. This relation applies exactly only when dissipation is small; when there is heavy damping the group speed relation is only crudely valid [18]. Since there is two fold symmetry between the four quadrants, it is enough to consider only the first quadrant

Waves on finite difference grids are sampled waves. Both spatial and temporal aliasing occurs and it is sufficient to consider only the non-dimensional domain $-\pi < \bar{w} < \pi$ and $-\pi < \bar{k} < \pi$, where $\bar{w}$ and $\bar{k}$ are dimensionless wavenumber and frequency [18]. In Figure 3.5, the numerical dispersion relation for PC is applied to the elastic wave equation and plotted versus non-dimensional wavenumber and frequency. In Figure 3.5a, near the origin, the curve starts out at the correct slope, which fulfills the consistency requirement discussed earlier. But away from the origin, the curve bends over and waves become more and more dispersive, a completely non-physical numerical artifact. Beyond $\bar{k} = \pi/2$, the group speed is negative and energy propagates in the opposite direction. In Figure 3.5b, notice that there is dissipation even though the material modeled is elastic. These dispersive and the dissipative errors occur in all finite difference models and are completely predictable in the manner discussed.

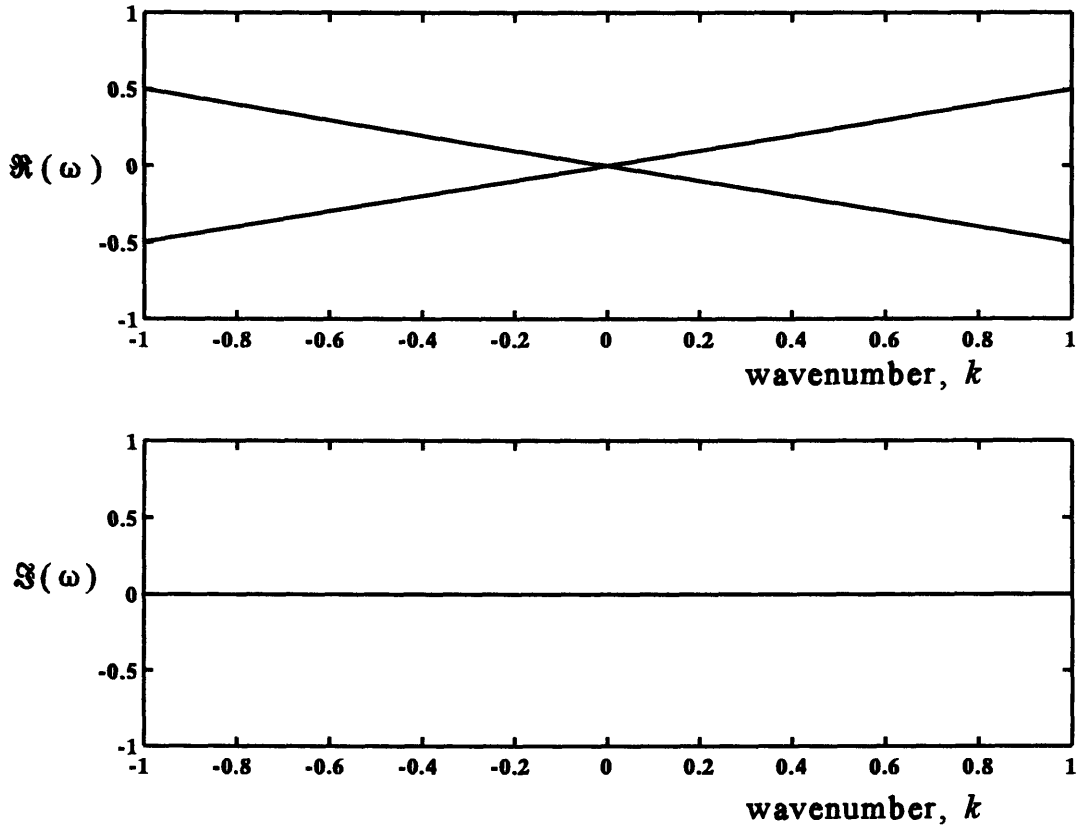


Figure 3.4

Top and bottom figures show real and imaginary frequency versus wavenumber for waves in an elastic medium, respectively. In the top figure, the upward sloping line represents a right traveling wave with sound speed, c=.5, and the downward sloping line represents a left traveling wave. The phase speed is constant versus frequency (or wavenumber) and there is, therefore, no dispersion. As shown in the bottom figure, elastic waves are non-dissipative, as well.

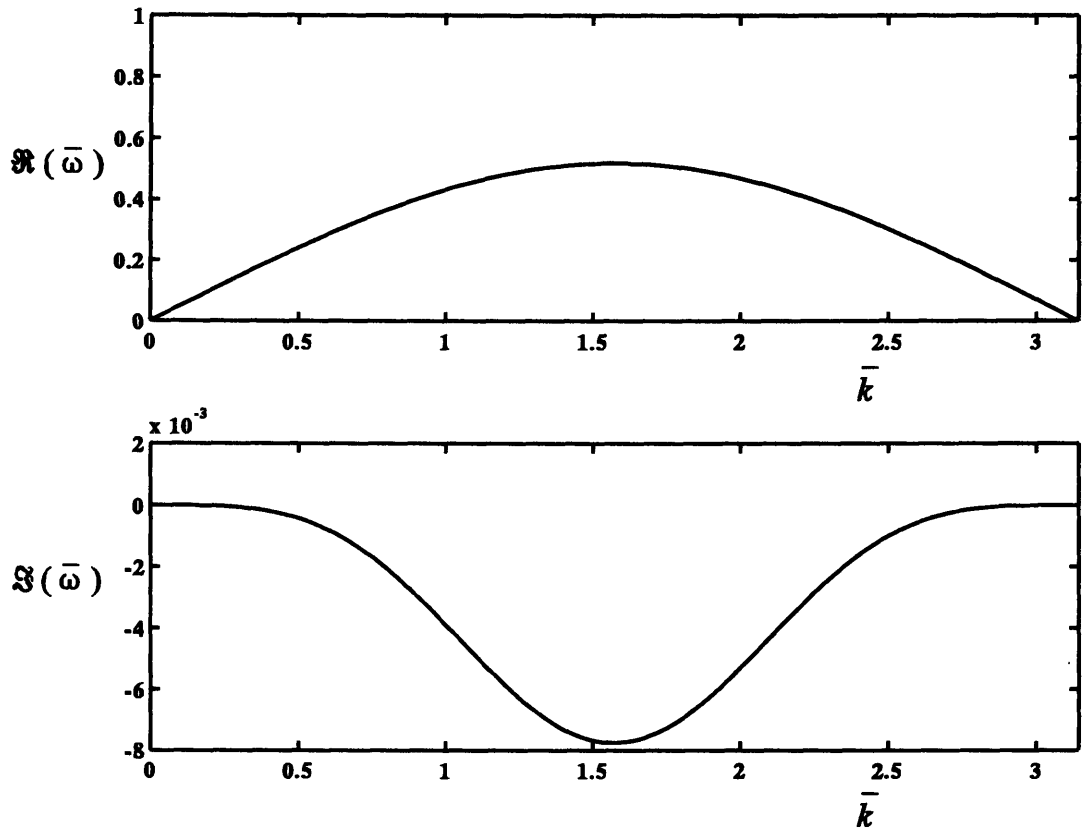


Figure 3.5

Top and bottom figures show real and imaginary dimensionless frequency versus dimensionless wavenumber for elastic waves modeled using the Predictor-Corrector scheme, respectively. The dimensionless material sound speed, the Courant number, is $N_c = .5$.

3.3 Optimization and Reoptimization

The optimization procedures are discussed in general terms in Section 1.4 and throughout this thesis. The non-linear solver is discussed in more detail in this section, followed by the definition of the figure-of-merit error analysis which is used to determine how well the complex modulus parameters have been optimized.

3.3.1 Non-linear Equation Set Solver

There are three instances when we need a method to solve a non-linear set of coupled complex equations: first, to find ω in terms of k in the dispersion relations; second, to find the optimum values of the constitutive model parameters that best match experimental complex modulus data; and third, to convert the complex modulus data given versus real frequency to real wavenumber using the fractional derivative model. Since there are so many uses for the non-linear solver, it is presented here in general terms.

Consider how a real non-linear set of equations would be solved using the Gauss-Seidel method. In general terms, a set of I non-linear equations, each a function of J variables can be expressed using indicial notation as

$$F_i(x_j) = 0 \ , \qquad \text{where } i = 1, 2, \ldots \text{I and } j = 1, 2, \ldots \text{J} \ . \tag{3.16}$$

The set of I equations is linearized by expanding each function in a Taylor series with respect to all the x_js and keeping first order terms yielding

$$\left[\frac{\partial F_i(x_j^n)}{\partial x_j} \right] \cdot [x_j^{n+1} - x_j^n]^{\mathrm{T}} = - [F_i(x_j^n)]^{\mathrm{T}} \ , \tag{3.17}$$

where x_j^n is an initial guess at the solution and x_j^{n+1} is the improved prediction found by solving Equation 3.17. The form of Equation 3.17 is $A \cdot x = b$, an eigenvalue problem, which

is easily solved by standard methods. I use *Matlab*'s matrix inversion routine. By iterating, x_j^{n+1} may or may not converge on the solution x_j depending on the initial guess, the condition of matrix A, and the strength of the non-linearities. Equation 3.17 can be solved in a deterministic manner using the same number of equations as unknowns or as an overdetermined system where $I > J$. The derivative is evaluated in a discrete sense by perturbing the variables a given percentage of their total value, usually .001 percent. Matrix A then becomes the sensitivity matrix. Convergence is determined by testing if the improvement of all variables between iterations is below some acceptable threshold, usually 10 times higher than the perturbation. For systems with high sensitivity, the solution can be underrelaxed by letting $\bar{x}_j^{n+1}$ be the solution of Equation 3.17 and then defining

$$x_j^{n+1} = \alpha \cdot x_j^n + (1 - \alpha) \cdot \bar{x}_j^{n+1} , \tag{3.18}$$

where $\alpha = 0$ is normal iteration and $\alpha = 1$ is complete underrelaxation (the solution doesn't move).

A complex set of N equations is handled as a set of $2N$ real equations. The Gauss-Seidel Method can now be used to solve the dispersion relations for either complex radial velocity in terms of real wavenumber or for complex k in terms of real ω. From these values the complex modulus is computed using Equation 1.3. Both the analytic and numeric dispersion relations are solved in a deterministic manner without any underrelaxation. An initial guess of $[\,.1 + j \cdot .1\,]$ is used in all cases.

Use of the non-linear solver to fit the constitutive model to the data is described in Section 3.4 for each individual case. In general, the solutions are found in an overdetermined manner with each data point given equal weight Underrelaxation is not necessary for optimization (unless the initial guess is poor) but is needed in some cases for reoptimization.

3.3.2 Figure-Of-Merit Errors

As we have seen, finite difference methods introduce errors that are both dissipative and dispersive. As a basis for judging these errors and comparing different finite difference schemes, the figure-of-merit approach as described by Fricke [12] is presented here. The phase error figure-of-merit, N_ϕ, is defined as the number of time steps to $\frac{\pi}{4}$ phase error and is given by

$$N_\phi = \frac{\frac{\pi}{4}}{| \Re(\bar{\omega}_n - \bar{\omega}_m) |} , \tag{3.19}$$

where $\bar{\omega}_n$ is either the numerical or analytic phase speeds, $\bar{\omega}_m$ is the material phase speed, and $\Re( )$ takes the real part of the expression in parenthesis. The complex wavenumbers are determined by solving the dispersion relations for a range of wavenumber using the non-linear solver. Since only real wavenumber waves exist in the finite difference model described in the Section 3.2, we take the real part of ω to find the phase speed. The magnitude error figure-of-merit, N_{mag} is the number of steps to 1 dB magnitude error given by

$$N_{mag} = \frac{1}{20 \cdot \log(e) \cdot | \Im(\bar{\omega}_n - \bar{\omega}_m) |} . \tag{3.20}$$

These figures-of-merit offer a more quantitative judgement of how well the model fits the data and are used to balance, if possible, the two types of errors for more efficient computation. Their use will be demonstrated in Section 3.4.

3.4 Optimization and Reoptimization Results

Three cases are presented here, which cover a number of important aspects of optimization for high loss materials. The first case is narrow band optimization at moderate Courant number where the bandwidth is less than an octave and the Courant number is $N_C = .5$. The second case is wide band (two octaves) at the same Courant number. The third case is wide band at a low Courant number, $N_C = .1$. Only one term in the complex modulus expansion is required (L=1 in Equation 2.14) for the narrow band case while two terms are used for the wide band. The center wavenumber is the same for all cases.

The same material is used for all three cases. The material is a high loss elastomeric material given by Torvik and Bagley [19]. It has a peak loss factor of about $\eta = .5$ and is representative of many elastomers with properties close to the more common Nitrile rubber. The shear modulus data, originally collected versus real frequency, is converted to real wavenumber using the fractional derivative model as described in Chapter 2. The material is optimized in the glass transition region near the peak loss factor where the wavenumber dependence is the strongest.

Since the benefit of reoptimization increases as the step size increases, an effort was made to find the largest step size at which the reoptimization process would converge. While the exact upper limit is not precisely determined, an argument will be made that it is near the step size investigated, $\Delta x = .004\ \mathrm{m}$, which is fixed for all three cases. The effect of changing the temporal step size is found by comparing the second and third cases where the temporal step size is decreased by a factor of five to change the Courant number by the same amount.

To gain more insight into the optimization process [†], I present the results in tabulated form and several graphical forms. The first graphical form shows the original complex modulus matched with the experimental data versus wavenumber. The second shows the match in

---

[†] I use the term optimization in a general sense, which can include reoptimization as well. If I am distinguishing between the two, then the context of the statement will make this clear.

terms of the phase and magnitude time step limits, N_ϕ and N_{mag} defined earlier in Section 3.3.2. The third is time domain simulations showing the propagation of a cosine modulated Gaussian pulse using a finite difference model with either optimized or reoptimized parameters. Each form brings out certain aspects of the results that increase our understanding of the optimization process and its potential benefits. Let us now consider the results for each of the three cases.

3.4.1 Case 1: Narrow Band, Moderate Courant Number

For the first case, a one term expansion in the decaying exponential constitutive model is used, and therefore, there are three model parameters to be optimized, M_R, M_1, and f_1. Forty equally spaced data points † in the 100 to 150 m^{-1} wavenumber band are used. Optimization requires the solution of the following overdetermined set of non-linear equations,

$$M_{c,data}(k_i, x_j) - M_{c,model}(k_i, x_j) = 0 \qquad \text{for } i = 1, 2, \ldots 20, \text{ and } j = 1, 2, \text{and } 3. \tag{3.21}$$

This has the same form as Equation 3.16 where x_j are the constitutive model parameters and $M_{c,model}$ is given by the analytic dispersion relation for optimization and the numeric dispersion relation for reoptimization. The non-linear equation solver described earlier is used to solve this system in iterative fashion. It should be noted that the solution of the complex modulus model also requires the use of a non-linear solver and therefore optimization requires nested iterations.

The optimization process begins with initial guesses of the constitutive model parameters. For this example, the initial guesses for the model parameter values are given in Table 3-1. The initial guess is required input to the non-linear solver described earlier and implemented in the

---

† Forty data points is equivalent to twenty wavenumber points because each wavenumber gives two data points corresponding to the storage and loss moduli.

program, *opt1.m,* described in the Appendix A. The fixed inputs, Δt and Δx, are not needed for optimization when the analytic dispersion relation is used. To improve the conditioning of the optimization process, all moduli are normalized by the relaxed modulus. Normalization is not needed for the corner frequencies. The program is run for 50 iterations yielding the optimized parameter values shown in line two of Table 3-1. The modeled complex modulus using the initial and optimized parameters is graphed with the data in Figure 3.6.

Reoptimization is now performed using the optimized values as the initial guess and the numerical dispersion relation for Predictor-Corrector determined in Section 3.4. *Opt1.m* is allowed to run for 50 more iterations resulting in the optimized parameters shown in Table 3-1. The finite difference grid step sizes assumed are also shown in the table. The spatial step size is the maximum step size (within a factor of two) at which the solution would converge. I refer to this as the *reference* step size, which is fixed at $\Delta x = .004\,\mathrm{m}$ for all three of the cases considered. The difficulty of finding a convergent solution can be explained by Equation 3.14, the dispersion relation for the LFCN scheme. As the spatial step size becomes large, the sinusoidal terms in the dispersion relation begin to oscillate and the non-linear solver cannot lock onto a consistent trend in parameter space.

Table 3.1

Initial, Optimized, and Reoptimized Constitutive Model Parameters for Narrow Band and Moderate Courant Number.

Item#	Name	Δx (m)	Δt (sec)	M_R (N/m^2)	M_1 (N/m^2)	f_1 (Hz)
1	Initial Guess	NA	NA	8.00e+07	7.00e+07	5.00e+03
2	Optimized	NA	NA	1.05e+08	1.25e+08	1.03e+04
3	Reoptimized	4.00e-03	4.00e-06	1.16e+08	1.35e+08	1.06e+04

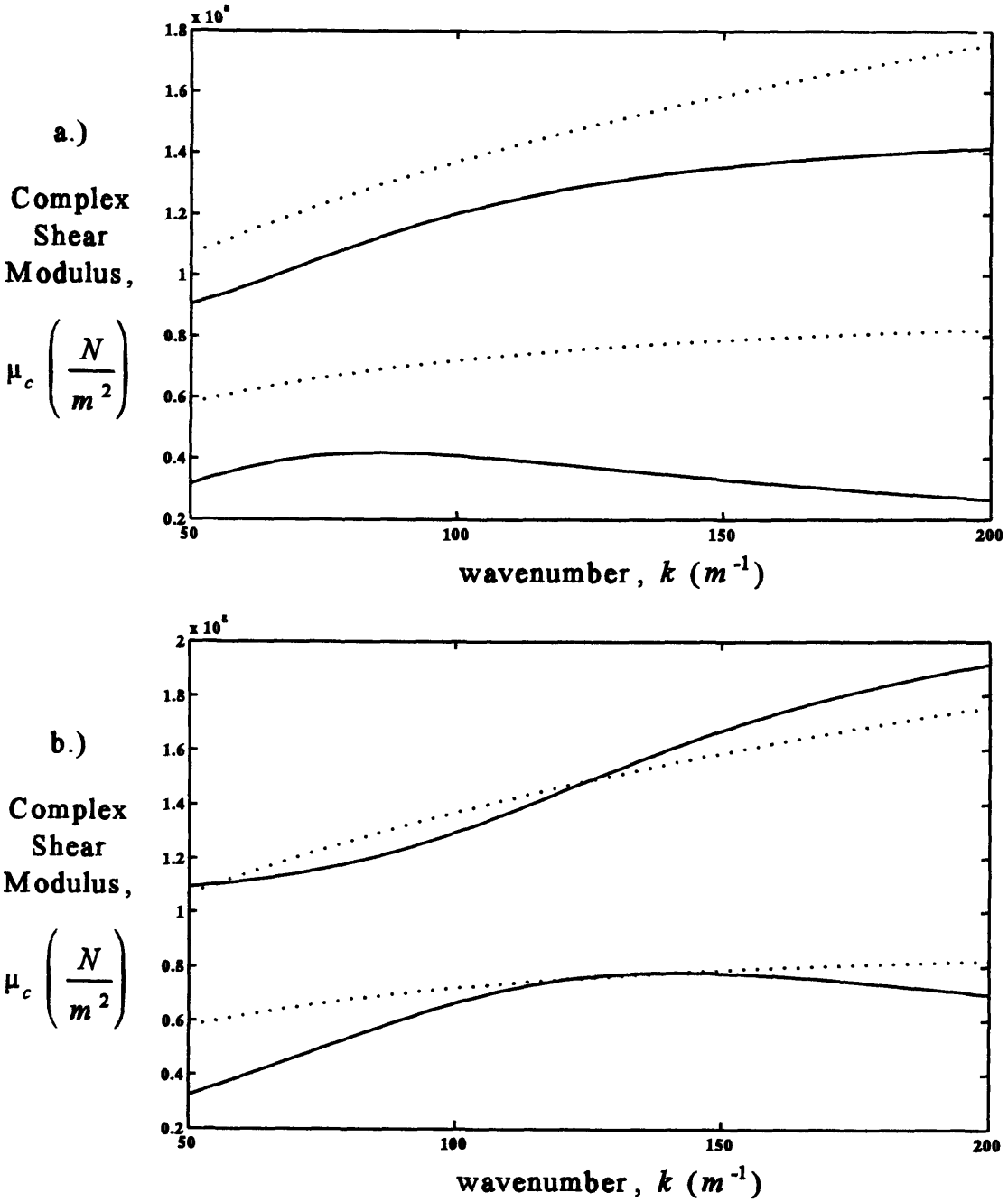


Figure 3.6

Complex shear modulus optimization results for narrow band, moderate Courant number with $L=1$ and $N_c = .5$. Both figures show experimental data (dotted) compared with complex modulus determined by the analytic dispersion relation (solid). For this material, the storage modulus, μ', is always greater than the loss modulus, μ''. Top figure shows complex modulus using initial guess as model parameters, while the bottom figure is modulus using optimized parameters after 50 iterations. The parameters are optimized over a narrow band between $k = 100$ and $150\ m^{-1}$.

To judge how well the parameters have been matched to the data and to compare optimization with reoptimization, several additional plots are now presented. The plots are formatted to determine, for a given accuracy, how fine the step sizes must be made for a finite difference model using the optimized parameters compared with the step size at which the reoptimized parameters are determined (the reference step size). Since the Courant number is fixed for each case, if the spatial step size is halved then the temporal step size is also halved. For example, Figure 3.7 shows experimental data versus wavenumber compared with the modeled complex modulus determined by the numerical dispersion. The optimized parameters are used to calculate the modulus for the Figure 3.7a, 3.7b, and 3.7c with spatial step sizes of 0, 1/2, and 1 times the reference grid size. The reoptimized parameters are used in Figure 3.7d (using the reference grid size, of course). By using the numeric rather than the analytic dispersion relation (even when the optimized parameters are used) the behavior of the viscoelastic wave on the finite difference grid is revealed. Judging from this figure, it is necessary to half the grid size of a model using optimized parameters in order to obtain the same accuracy as the reoptimized model.

While optimization and reoptimization performed in terms of complex modulus improves conditioning [†], it is not the best way to judge how well the data is matched. A better way is shown in Figure 3.8 where the magnitude and phase error figures-of-merit, N_{mag} and N_{ϕ}, defined in Section 3.3.2 are plotted. Figures 3.8a through 3.8d correspond to the same four axes in Figure 3.7. The advantage of using N_{mag} and N_{ϕ} is that phase and magnitude errors are now clearly distinguished. It is evident that the phase error is drastically improved by reoptimization and is better than the optimized parameters at zero grid size (I explain this in Section 3.5). Furthermore, the magnitude error is on average the same as the optimized parameters at half the step sizes.

[†]Blanch recognized that conditioning is improved by optimizing to loss factor rather than its inverse, the quality factor. Optimizing to complex modulus is an equivalent notion. The key idea is that loss factor and modulus are impedance type quantities that add independently to some total impedance while their inverses do not. The independence of the elements that make up the whole leads to a more stable and better conditioned process.

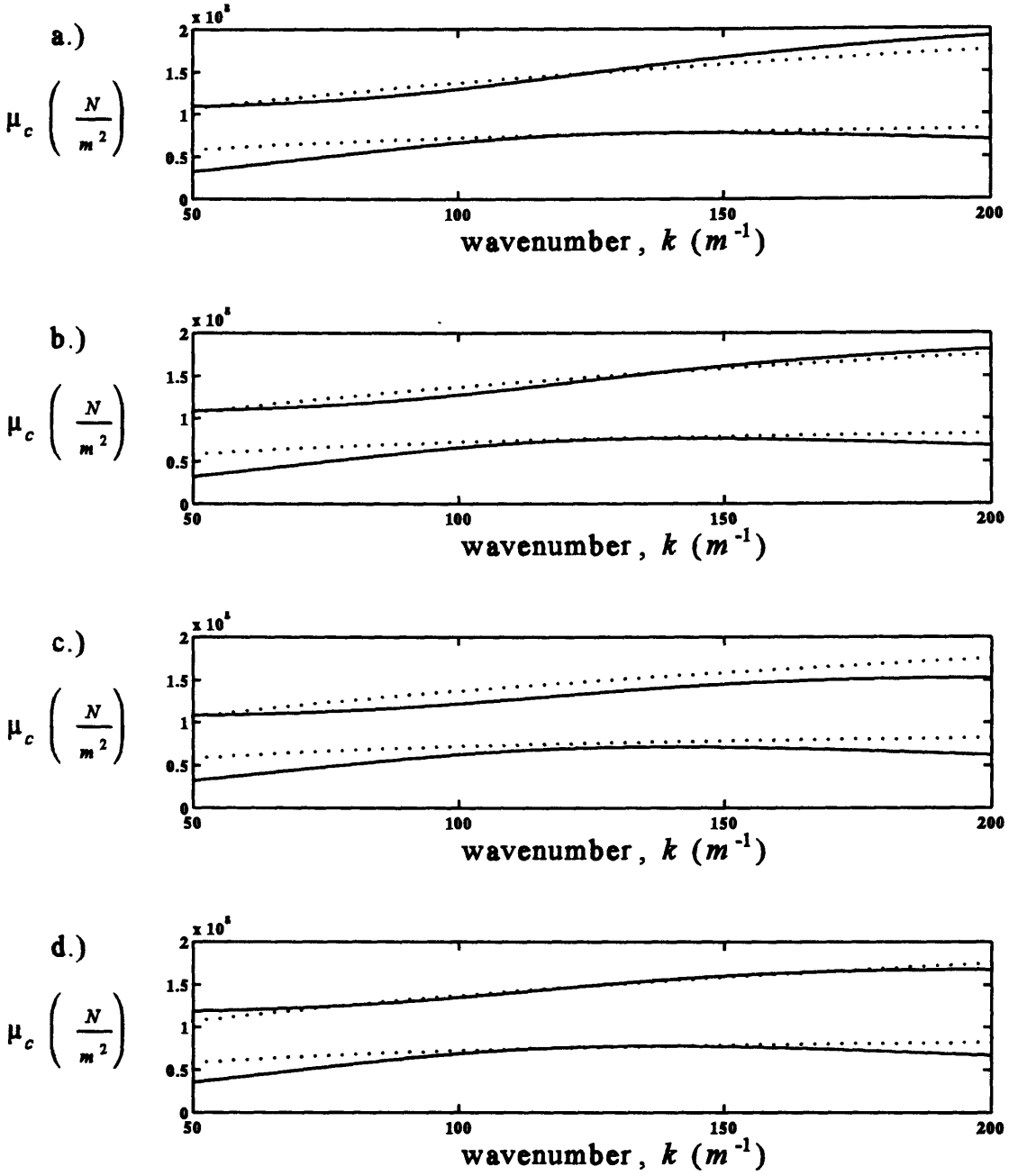


Figure 3.7

Complex shear modulus optimization results for narrow band, moderate Courant number with $L=1$ and $N_c = .5$. All figures show experimental data (dotted) compared with the modeled complex modulus determined by the numerical dispersion relation (solid). The storage modulus, μ', is greater than the loss modulus, μ'', in all cases. Optimized parameters are used to calculate the modulus for the top three figures while reoptimized parameters are used in the bottom figure. The temporal step size is $\Delta t = 4e{-}6$ sec and spatial step sizes used for the top through the bottom figure are 0, .5, 1, and 1 times the reference step sizes, $\Delta x = .004$ m.

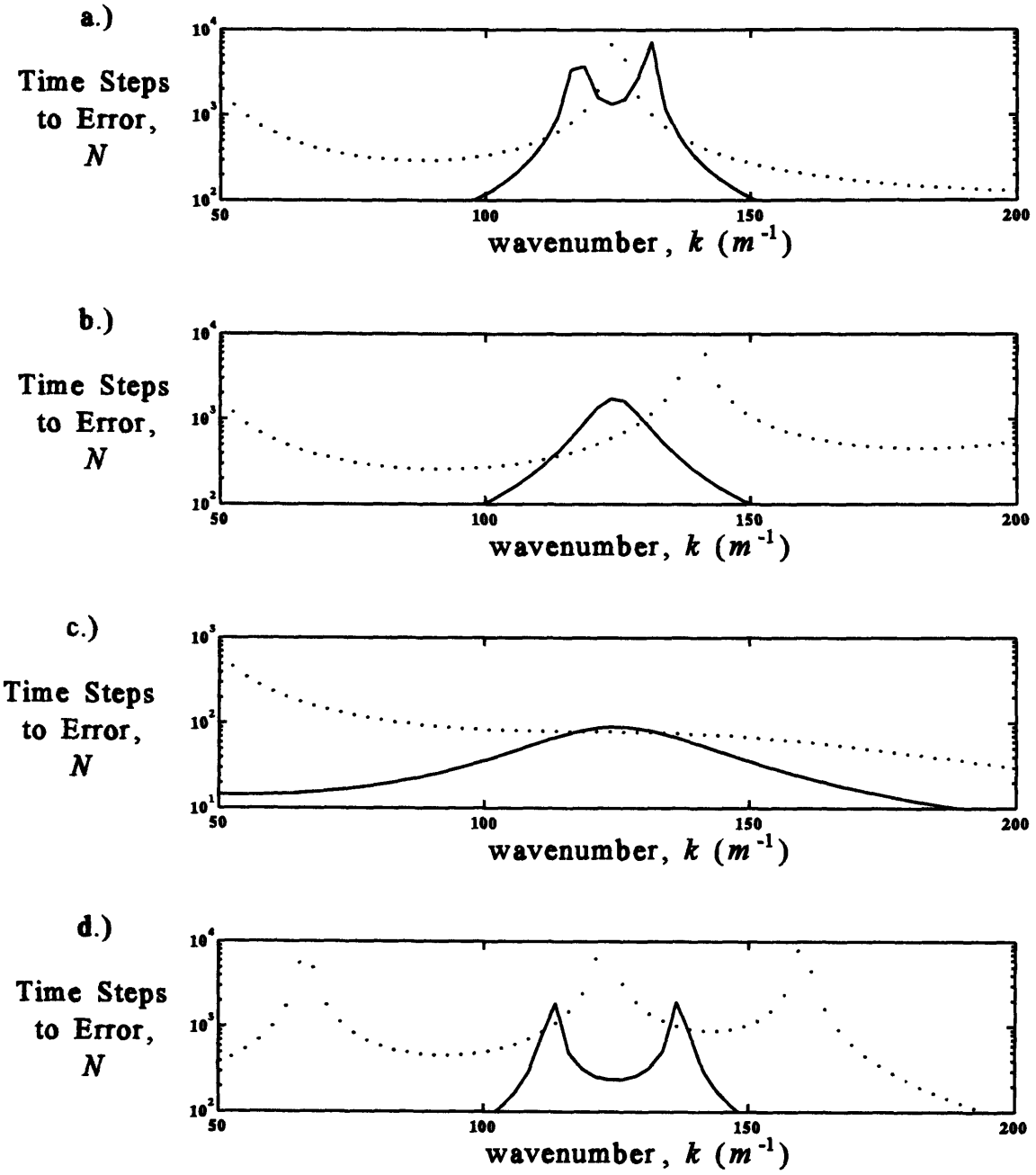


Figure 3.8

Error analysis of results for narrow band, moderate Courant number optimization with $L=1$ and $N_c = .5$. Dotted curves represent number of time steps to $\frac{\pi}{4}$ phase error, N_ϕ, solid curves represent time step limit to 1 dB magnitude error, N_{mag}. The four figures correspond to the four figures shown in Figure 3.7. Note that N_{mag} for the reoptimized parameters (Figure 3.8d) is roughly the same as the optimized parameters at half the step size and N_ϕ is greater than the optimized parameters at half the step size (Figure 3.8b).

Yet another way to judge error is to examine the time domain behavior of a propagating shear pulse. Since the ultimate objective is time domain modeling, the finite difference model described in Section 3.2 is used to compare optimized and reoptimized parameters. The initial pulse shapes that are used to test both the narrow and wide band optimization are shown in Figure 3.9 with their respective power spectral densities. The narrow band results are shown in Figure 3.10 in the same customary format. The figure shows the pulse versus position after 100 time steps have elapsed. The pulse has travelled less than .5 m and already the peak amplitude has decayed by 20 dB. Also, there is very little dispersion because the bandwidth is narrow. Comparing Figure 3.10a through 3.10d, it is again clear that a reoptimized model is at least as accurate as the optimized model. From the time domain viewpoint, the reoptimized model is more accurate than even the zero step size optimized model (see Section 3.5)!

3.4.2 Case 2: Wide Band, Moderate Courant Number

A two term decaying exponential model is needed to accurately match the data over the wider two octave band investigated here. This is consistent with the finding by Blanch for low constant loss factor modeling. There are now five constitutive model parameters to be optimized. The same techniques used in the narrow band optimization are used with two exceptions. In order to reoptimize at the same reference spatial step size, I first optimize at $\Delta x = .003$ m, then reoptimize in sequence at $\Delta x = .003$ m, .0035 m, and .004 m. The result of the earlier reoptimization is used as the initial guess for the subsequent reoptimization. A limit of 20 iterations for each step is imposed for a total of 80 iterations. In addition, the solution is underrelaxed as described in Section 3.6 using an underrelaxation parameter, $\alpha = .1$. The extra effort required for the wide band reoptimization indicates that the numeric dispersion relation is nearing the unstable oscillatory region discussed earlier and, therefore, the upper limit for the reference spatial step size has been obtained. It should be noted that the technique is systematic and could easily be programmed to automatically search for the maximum spatial step. Furthermore, the "extra effort" is non-recurring and does not detract

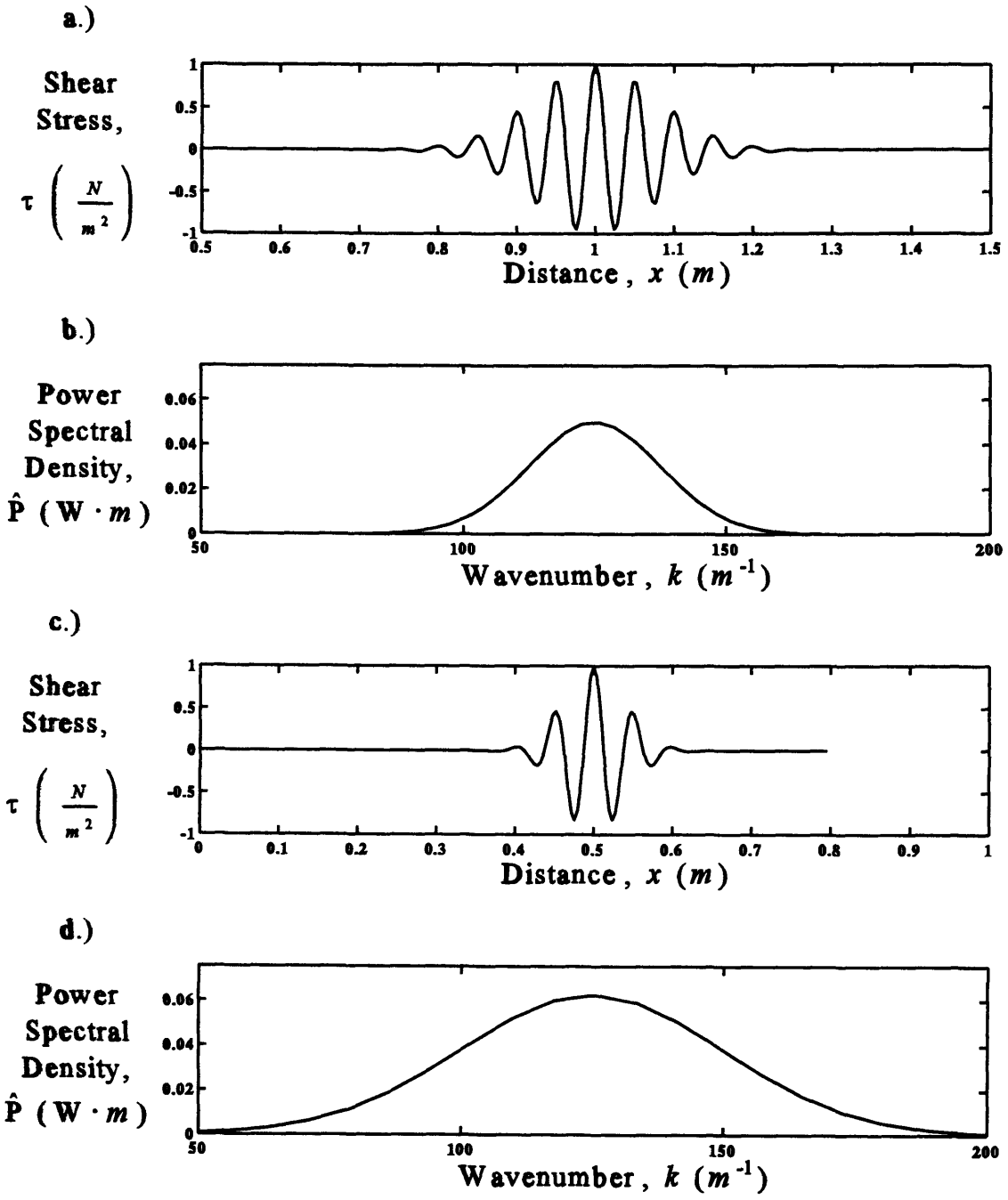


Figure 3.9

Initial shear stress pulse shapes versus distance for narrow and wide band sources and their respective power spectral densities versus wavenumber used to test the benefits of optimization and reoptimization in the time domain. Since the constitutive model is only accurate over a limited bandwidth, it is neccessary to limit the bandwidth of the energy injected into the finite difference model by using initial pulse shapes such as these.

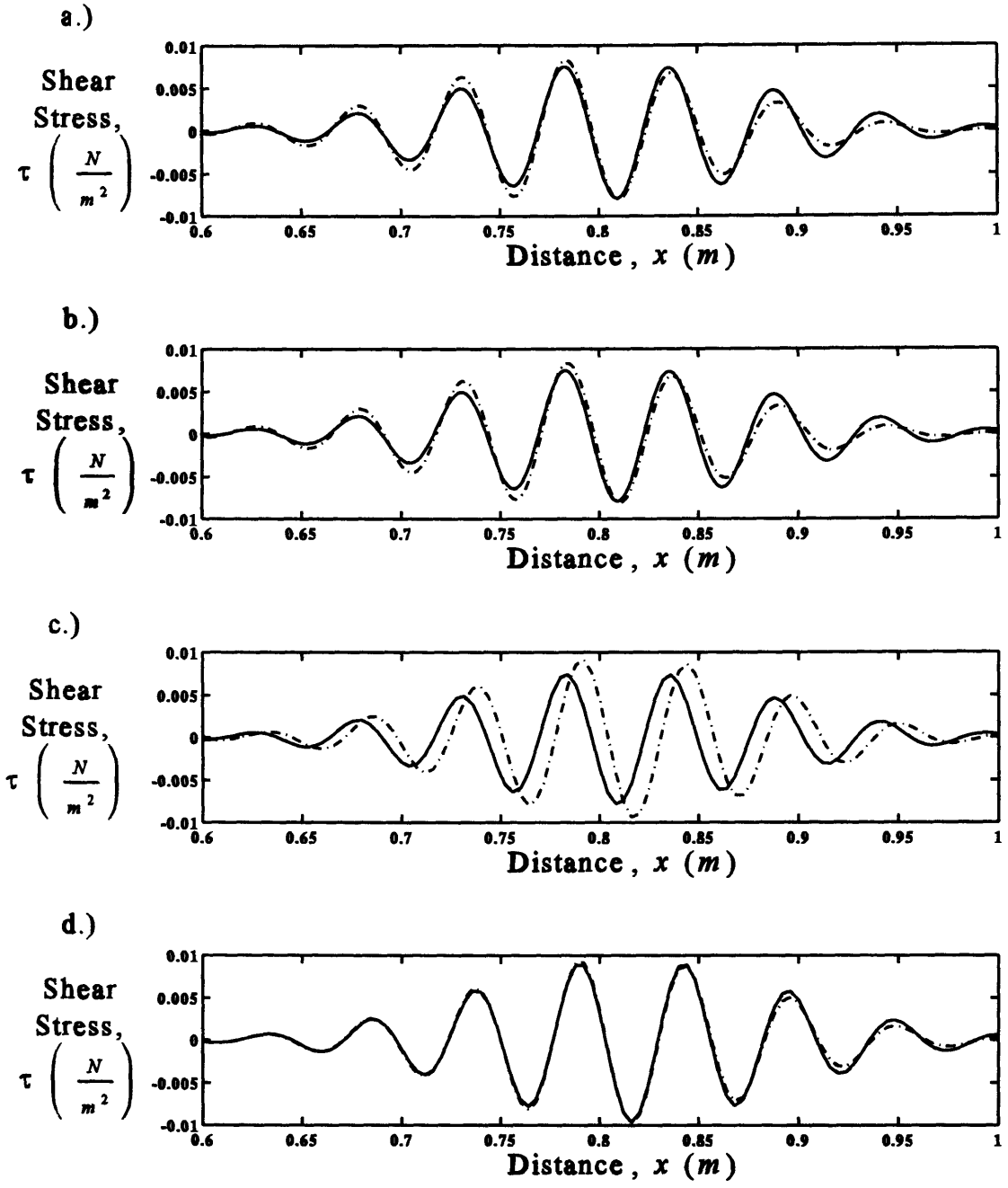


Figure 3.10

Time domain results for narrow band, moderate Courant number optimization with $L=1$ and $N_C = .5$. All figures show a cosine modulated Gaussian pulse at fixed time, $t = 100 \cdot \Delta t_{ref}$, as predicted by a spectral method (solid) and a finite difference model. (dash-dot). The original pulse shapes are shown in Figure 3.9. The temporal step size is $\Delta t = 4e{-}6$ sec and spatial step sizes used for the top through the bottom figure are 0, .5, 1, and 1 times the reference step sizes, $\Delta x = .004$ m. These four figures correspond to those shown in Figure 3.7. Note that the reoptimized pulse shape follows the spectral method results more closely than the optimized parameters at half the step sizes.

from the potential payoff of reoptimization. The initial, optimized, and reoptimized parameters are given in Table 3-2 with intermediate steps omitted for clarity. There is one more point of interest. Since the extra effort was not required for the narrow band reoptimization, it indicates that the maximum reference step size was not reached and that more improvement due to reoptimization could be achieved in that case.

Table 3.2
Initial, Optimized, and Reoptimized Constitutive Model Parameters for Wide Band and Moderate Courant Number.

Item	Name	M_R (N/m^2)	M_1 (N/m^2)	M_2 (N/m^2)	$f_{c,1}$ (Hz)	$f_{c,2}$ (Hz)
1	Initial	5.00e+07	1.00e+07	2.00e+08	1.00e+03	6.00e+03
2	Optimized	8.02e+07	6.55e+07	1.18e+08	3.37e+03	2.05e+04
3	Reoptimized	9.39e+07	4.16e+07	1.27e+08	3.04e+03	1.32e+04

The complex modulus model is compared with the experimental data in Figure 3.11. The agreement over the band for the optimized parameters at zero step size is very close. As was observed in the narrow band case, the additional error introduced due to discretization is apparent at the half reference step size (Figure 3.11b) and gross at the reference grid size (Figure 3.11c). Reoptimization again improves the fit and is closer in appearance to the optimized parameters at half step size than to the full reference step size.

Figure 3.12 gives the more quantitative N_ϕ and N_{mag} measures of error. The same trends observed for the narrow band are here again. The phase error after optimization is roughly the same on average as the optimized model at half the step sizes. The magnitude error is improved to some point between the optimized models at the half and full step sizes. An

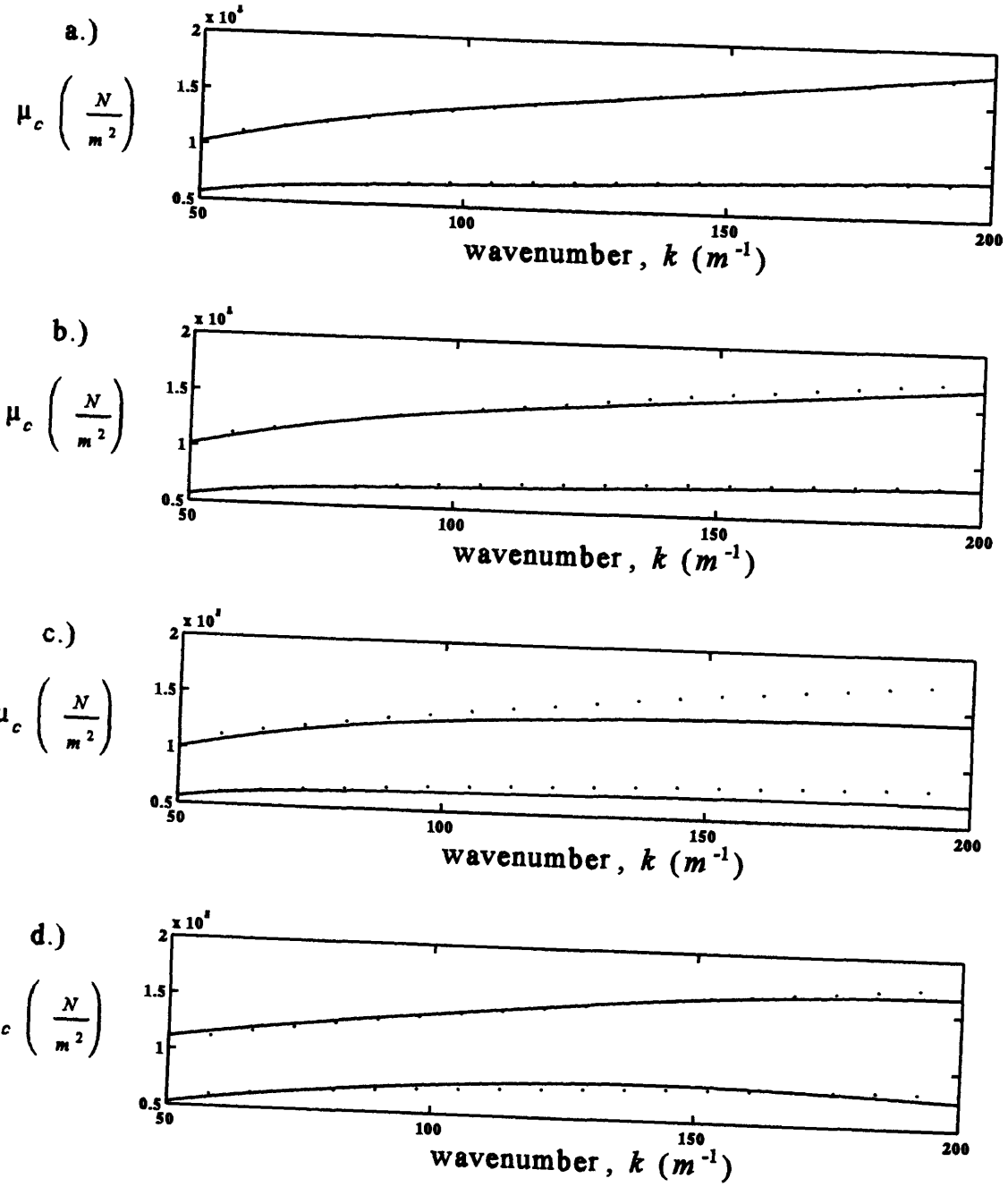


Figure 3.11

Complex shear modulus optimization results for wide band, moderate Courant number with $L=2$ and $N_C = .5$. All figures show experimental data (dotted) compared with the modeled complex modulus determined by the numerical dispersion relation (solid). The storage modulus, μ', is greater than the loss modulus, μ'', in all cases. Optimized parameters are used to calculate the modulus for the top three figures, while reoptimized parameters are used in the bottom figure. The temporal step size is $\Delta t = 4e\text{-}6$ sec and spatial step sizes used for the top through the bottom figure are 0, .5, 1, and 1 times the reference step sizes, $\Delta x = .004$ m.

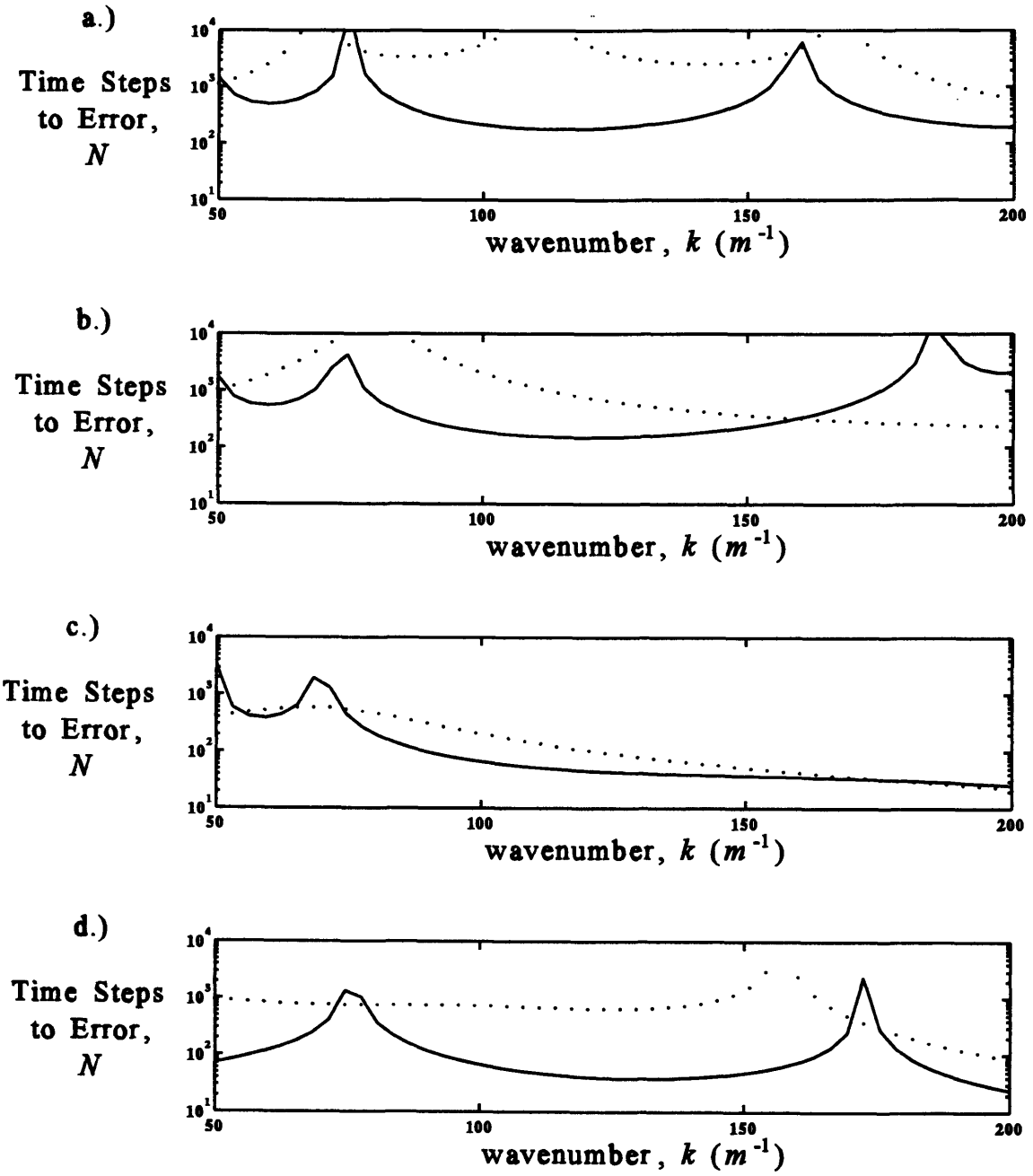


Figure 3.12

Error analysis of results for wide band, moderate Courant number optimization with $L=2$ and $N_c=.5$. Dotted curves represent number of time steps to $\frac{\pi}{4}$ phase error, N_ϕ, solid curves represent time step limit to 1 dB magnitude error, N_{mag}. The four figures correspond to the four figures shown in Figure 3.11. Note that N_{mag} for the reoptimized parameters (Figure 3.12d) is higher than the optimized parameters at the same step sizes (Figure 3.12c) but less than the N_{mag} at half the step size (Figure 3.12b). The N_ϕ for the reoptimized parameters is on average about the same as the optimized parameters at half the step sizes.

unsuccessful attempt was made at this point to balance the phase and magnitude error by weighting the loss modulus more heavily than the storage modulus with the idea that this would yield a more efficient model. For low loss materials, the loss modulus controls the magnitude error while the storage modulus controls phase error (this is only roughly true for the high loss materials). For light amounts of weighting, the improvement is negligible. As the weighting is increased, before any significant improvement is realized, the solution will no longer converge. I attribute this to the limited degrees of freedom in the decaying exponential model where there is a fixed relationship between the loss and storage modulus. By changing the weighting, I am is trying to force a change in this relationship which is not possible. In other words, the solution I am trying to achieve is not contained in the decaying exponential model and therefore cannot possibly converge.

The results from the time domain experiments are shown in Figure 3.13. The wider band pulse shown in Figure 3.9 is now used. Figure 3.13 gives the pulse shapes versus distance after 100 time steps. Due to the wider bandwidth, dispersion of the pulse shape is evident. The improvement in phase error due to reoptimization is evident by comparing the reoptimized model with the optimized model at the same step size. The improvement in magnitude error is somewhat better than anticipated from the wavenumber domain error analysis figures. One reason for this may be the filtering effect that the source spectrum has on the wavenumber domain error. Another reason is that there may be some phase and magnitude error cancellation. In any event, once again, based on these results the reoptimized model at the reference step size is as good as the optimized model at half the reference step size.

To compare error at different times, the reoptimized model and the optimized model at the reference step size is plotted (along with the "exact" spectral model solution) after 25, 50, 100, and 200 time steps in Figure 3.14. All three solutions are close together after 25 time steps. The error in the optimized model is seen to increase more rapidly than the reoptimized solution. The stress levels are extremely small after 200 time steps.

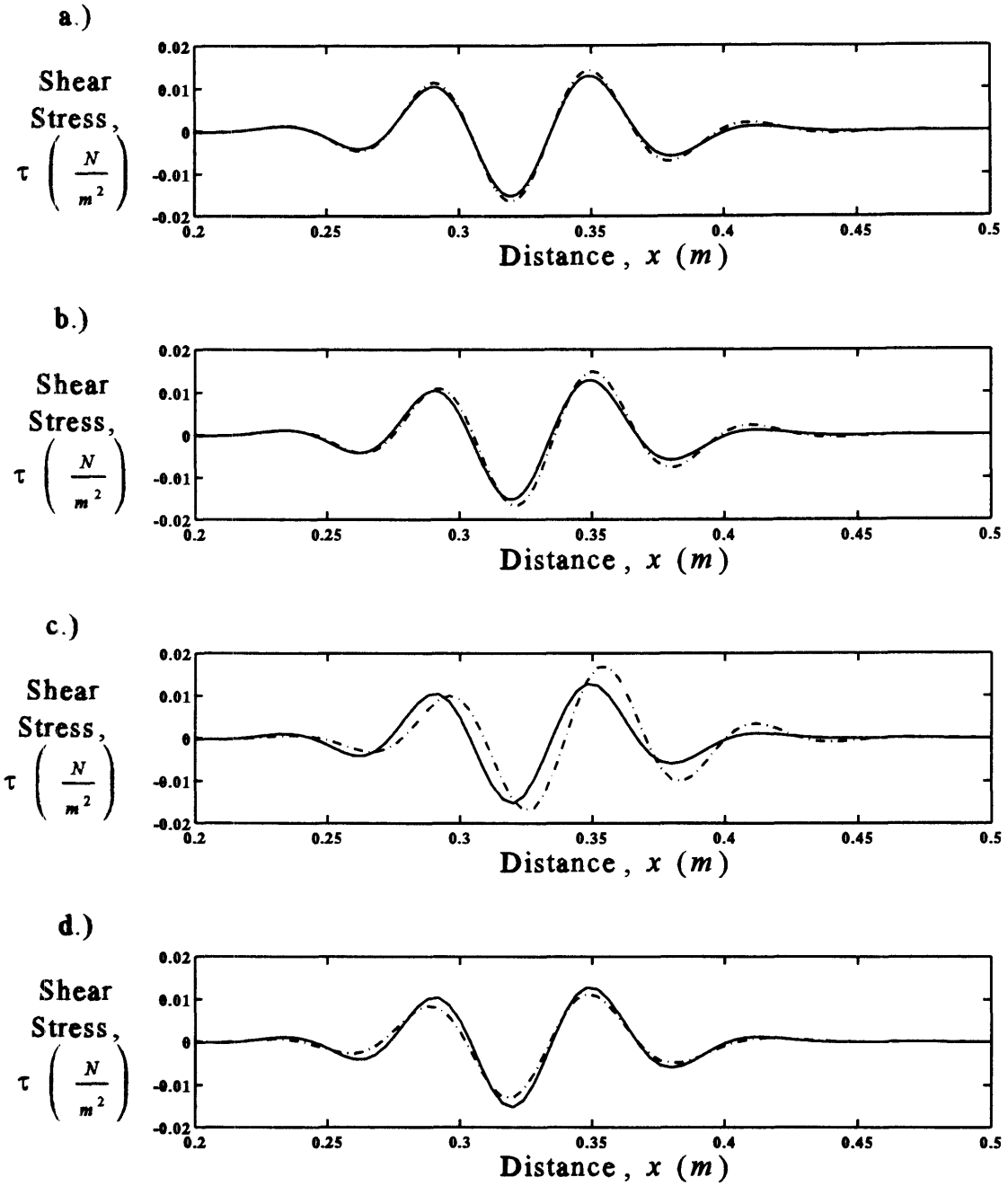


Figure 3.13

Time domain results for wide band, moderate Courant number optimization with $L=2$ and $N_C = .5$. All figures show a cosine modulated Gaussian pulse at fixed time, $t = 100 \cdot \Delta t_{ref}$, as predicted by a spectral method (solid) and a finite difference model. (dash-dot). The original pulse shapes are shown in Figure 3.9. The temporal step size is $\Delta t = 4e{-}6$ sec and spatial step sizes used for the top through the bottom figure are 0, .5, 1, and 1 times the reference step sizes, $\Delta x = .004$ m. Optimized constitutive model parameters are used for the top three figures while the reoptimized parameters are used for the bottom figure.

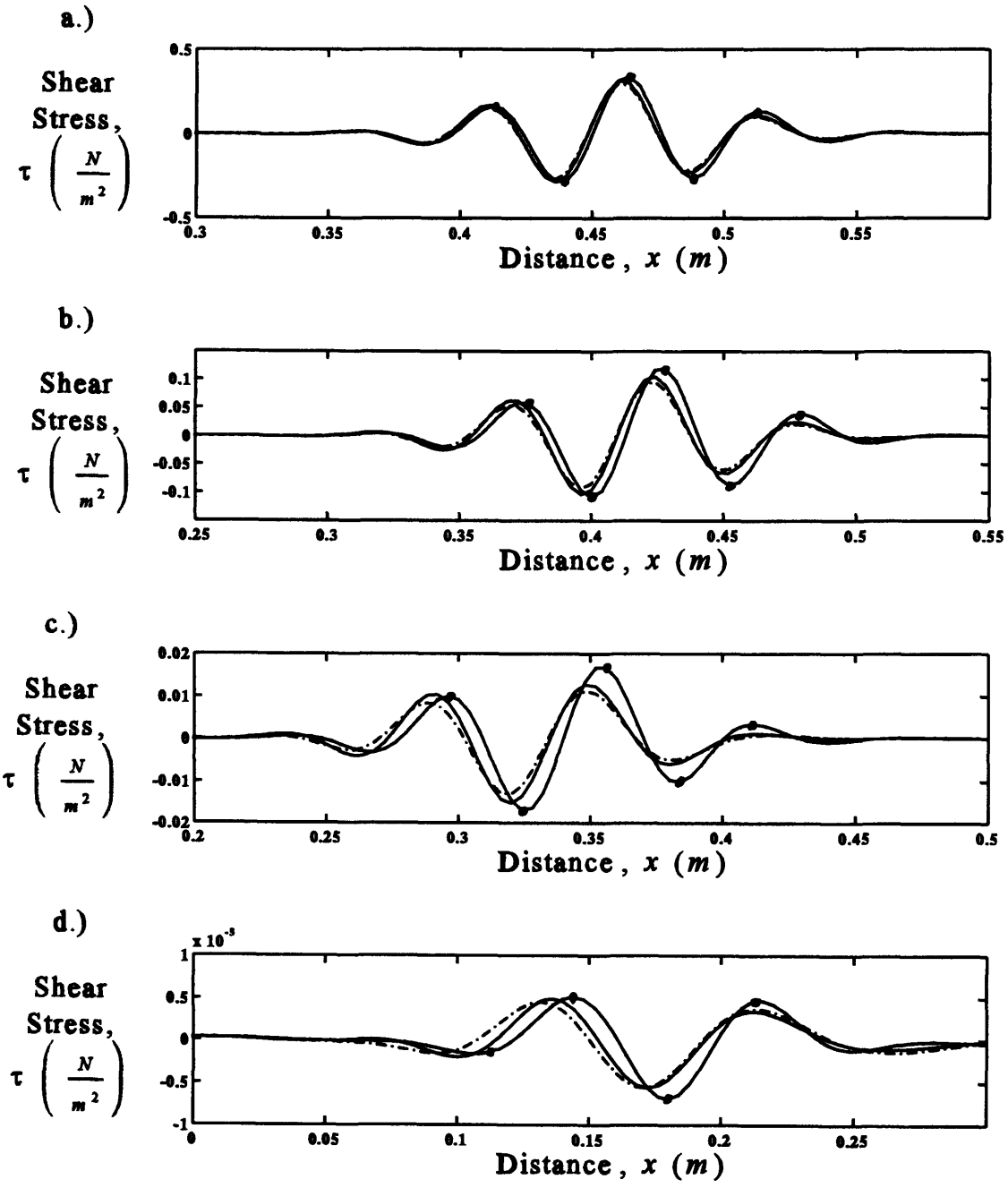


Figure 3.14

Time domain results for wide band, moderate Courant number optimization with $L=2$ and $N_C = .5$. Figures from top through the bottom show a cosine modulated Gaussian pulse at 25, 50, 100, and 200 time steps where $t = 100 \cdot \Delta t_{ref} = .4e\text{-}6$ sec. Each figure has three curves representing shear stress as predicted by a spectral method (solid), the finite difference model with optimized parameters (solid-dot), and the finite difference method with reoptimized parameters (dash-dot). The original pulse shapes are shown in Figure 3.9 and the spatial step size is $\Delta x = .004$ m.

3.4.3 Case 3: Wide Band, Low Courant Number

Due to the low sound speed of high loss materials, it will often be required to model wave propagation at low Courant numbers near $N_C = .1$, which is the reason this case is considered. The reference spatial step size is the same as the earlier cases; the temporal step size is decreased by a factor of five to achieve the low Courant number. Exactly the same optimization procedure used for the wide band, moderate Courant number is used for this case. The results are given in Table 3-3. It is interesting to note that the reoptimized parameters are within several percent of those for the moderate Courant number.

Table 3.3
Initial, Optimized, and Reoptimized Constitutive Model Parameters for Wide Band and Low Courant Number.

Item	Name	M_R (N/m^2)	M_1 (N/m^2)	M_2 (N/m^2)	$f_{c,1}$ (Hz)	$f_{c,2}$ (Hz)
1	Initial	5.00e+07	1.00e+07	2.00e+08	1.00e+03	6.00e+03
2	Optimized	8.02e+07	6.55e+07	1.18e+08	3.37e+03	2.05e+04
3	Reoptimized	9.56e+07	3.97e+07	1.32e+08	3.05e+03	1.32e+04

The complex modulus match versus wavenumber is graphed in Figure 3.15. The appearance is similar to the corresponding graph for the moderate Courant number indicating that the discretization error is driven by the spatial step size. The phase and magnitude errors versus wavenumber are shown in Figure 3.16. Both N_ϕ and N_{mag} are five times greater than the moderate Courant number case merely due to the smaller temporal step size. The differences beyond that are not pronounced. The time domain experiments for the wide band wave packets after 500 time steps (same time as the moderate Courant number cases after 100 time

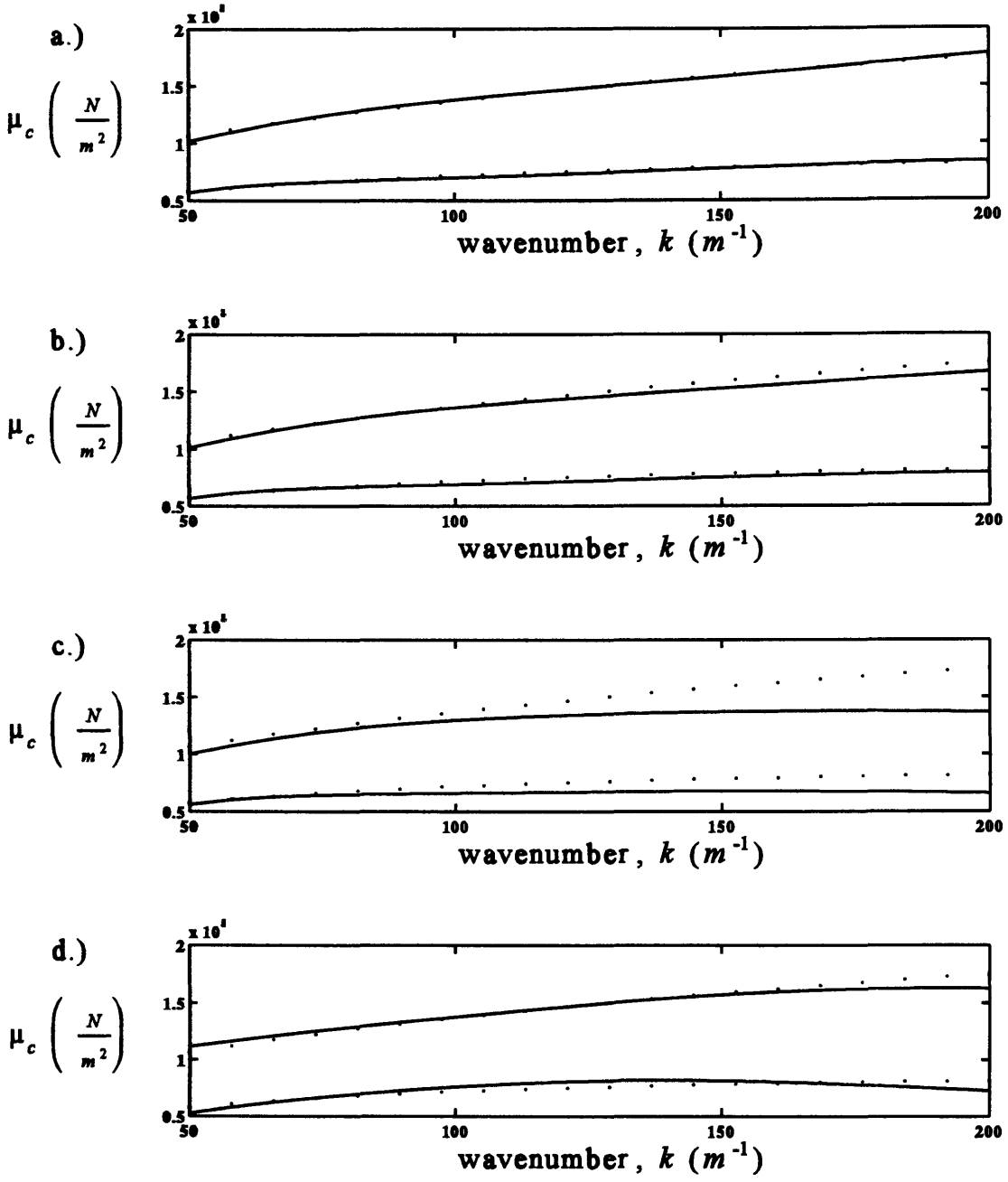


Figure 3.15

Complex shear modulus optimization results for wide band, low Courant number with $L=2$ and $N_C = .1$. All figures show experimental data (dotted) compared with the modeled complex modulus determined by the numerical dispersion relation (solid). The storage modulus, μ', is greater than the loss modulus, μ'', in all cases. Optimized parameters are used to calculate the modulus for the top three figures while reoptimized parameters are used in the bottom figure. The temporal step size is $\Delta t = .8e\text{-}6$ sec and spatial step sizes used for the top through the bottom figure are 0, .5, 1, and 1 times the reference step sizes, $\Delta x = .004$ m.

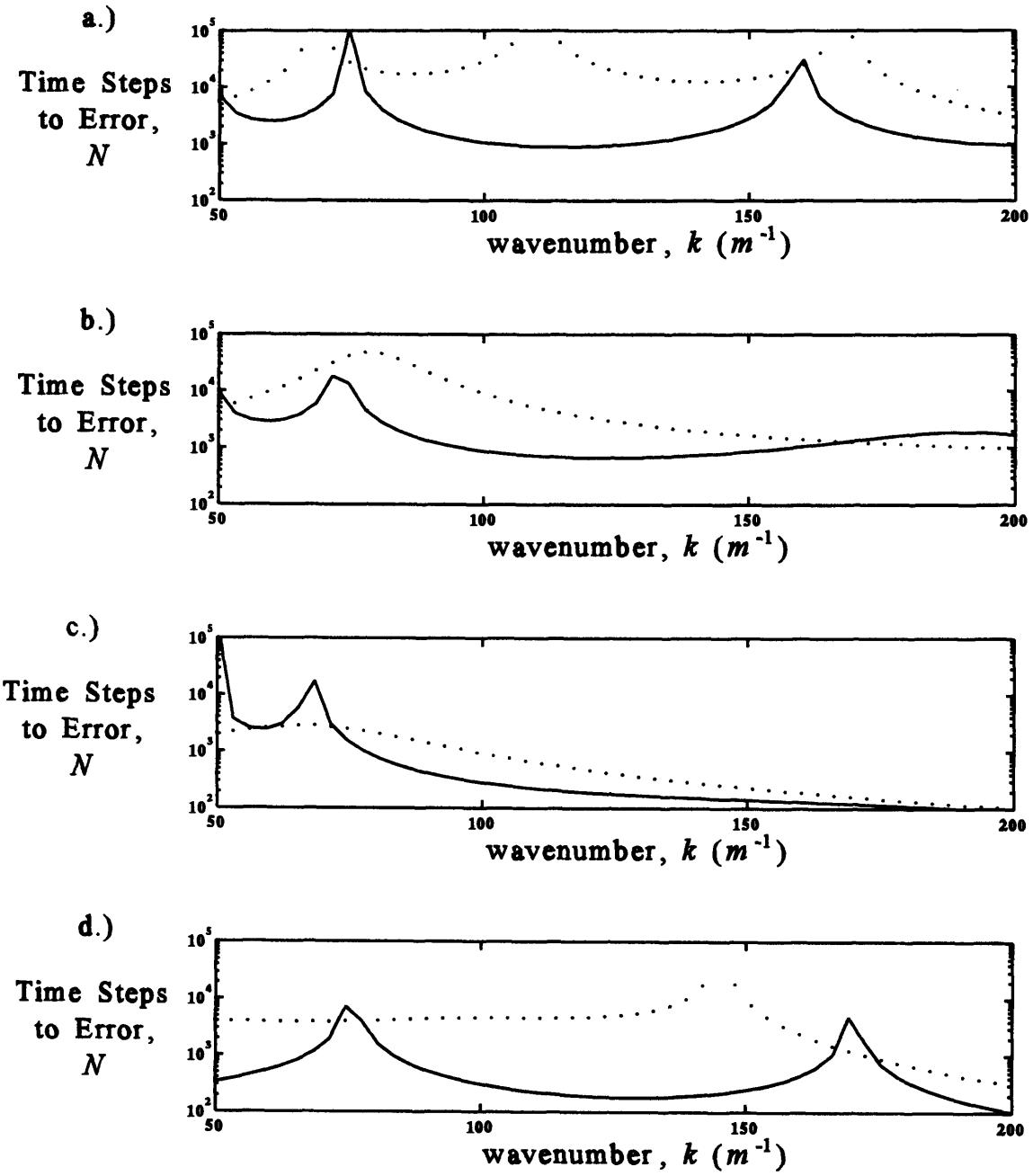


Figure 3.16

Error analysis of results for wide band, moderate Courant number optimization with $L=2$ and $N_c = .1$. Dotted curves represent number of time steps to $\frac{\pi}{4}$ phase error, N_ϕ, solid curves represent time step limit to 1 dB magnitude error, N_{mag}. The four figures correspond to the four figures shown in Figure 3.15. Note that N_{mag} for the reoptimized parameters (Figure 3.16d) is higher than the optimized parameters at the same step sizes (Figure 3.16c) but less than the N_{mag} at half the step size (Figure 3.16b). The N_ϕ for the reoptimized parameters is on average about the same as the optimized parameters at half the step sizes.

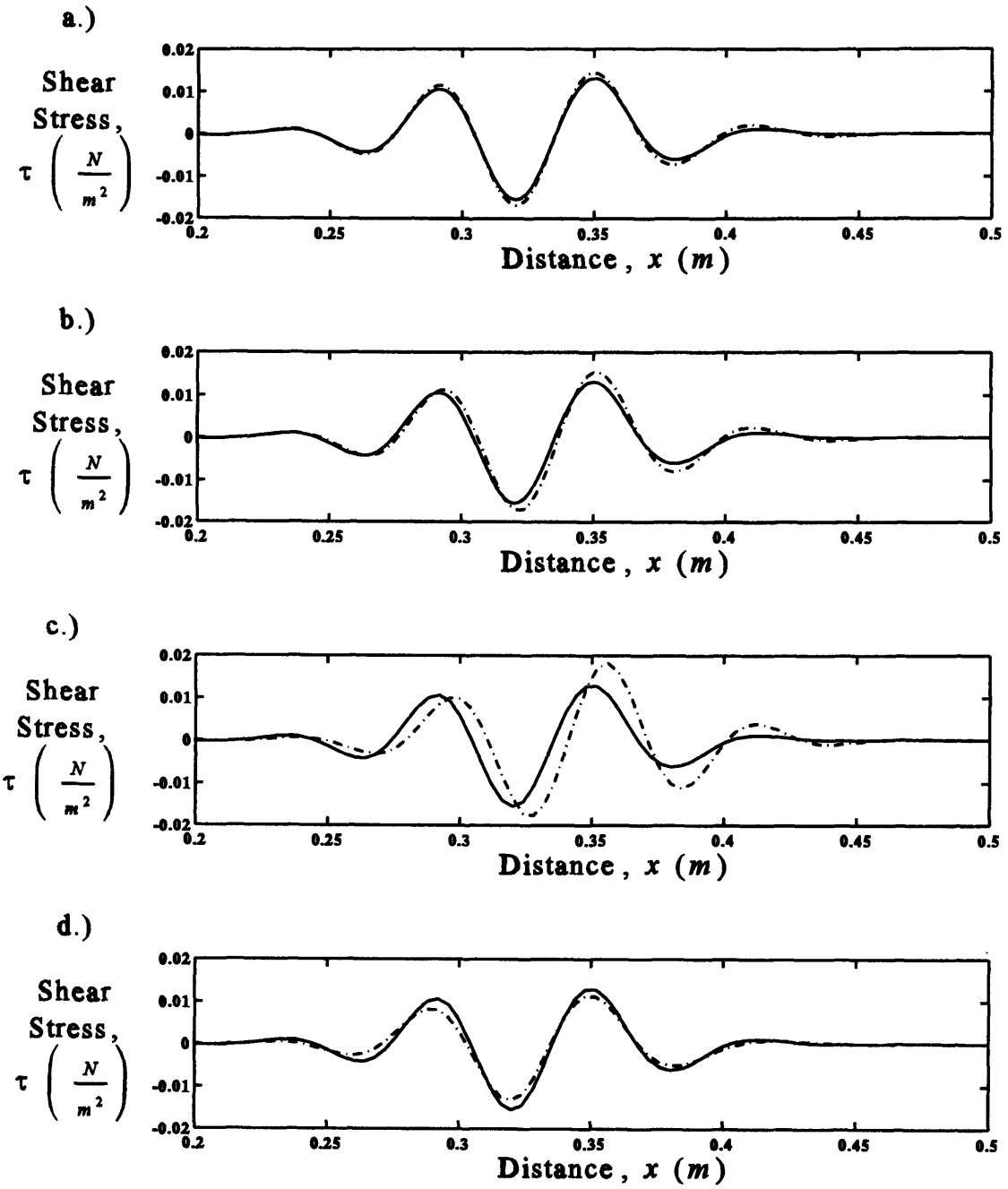


Figure 3.17

Time domain results for wide band, low Courant number optimization with $L=2$ and $N_C = .1$. All figures show a cosine modulated Gaussian pulse at fixed time, $t = 500 \cdot \Delta t_{ref}$, as predicted by a spectral method (solid) and a finite difference model. (dash-dot). The original pulse shapes are shown in Figure 3.9. The temporal step size is $\Delta t = .8e{-}6$ sec and spatial step sizes used for the top through the bottom figure are 0, .5, 1, and 1 times the reference step sizes, $\Delta x = .004$ m. Optimized constitutive model parameters are used for the top three figures while the reoptimized parameters are used for the bottom figure.

steps) are shown in Figure 3.17. The error for the optimized model at the reference step size is worse than the same conditions at the moderate Courant number. For this reason, the improvement due to optimization is greater than the moderate Courant number case.

3.5 Discussion

In most cases of interest, the domain of the finite difference model is heterogeneous and will include several materials of various degrees of viscoelasticity including non-dissipative elastic materials. Because high loss materials are soft and typically have the lowest sound speeds, they dictate the spatial step size of the entire grid. In extreme cases, such as the constrained layer submarine hull problem discussed in Chapter 1, the sound speed contrast between the steel hull and the elastomer is approximately 10 times. Therefore by increasing the step size required by the viscoelastic material, the step size for the entire domain is also increased. This is how reoptimization is used to increase computational efficiency.

The results of the previous section show that for narrow band modeling, the time steps to error for the reoptimized parameters have a larger average N_ϕ and nearly the same average N_{mag} in a narrow band about the center wavenumber than the optimized values at half the spatial step size. Therefore, reoptimization allows a two-fold increase in spatial and temporal step sizes which increases computational speeds by eight times for 2D models and by sixteen times for 3D. The attendant saving in computer memory is also significant, especially considering the roughly two fold increase in field variables due to the memory variables (refer to Section 2.8.). The contention that, for a given accuracy, reoptimization allows the grid size to be doubled has been verified in both the frequency domain and in the time domain by the experiments presented. The time domain results are most important since that is the where the final solution is desired. The use of narrow band modeling is particularly useful for modeling complex geometry, precluding the use of spectral methods, and whenever the computational cost of wide band is prohibitive due to additional terms in the constitutive

model expansion. This is likely to arise when modeling 3D viscoelastic waves where the computational burden is already immense and each additional term in the constitutive model expansion adds six additional field variables.

How is it possible that the reoptimized results (see Figure 3.7) are better than the optimized results at zero grid size? I discussed in Chapter 2, how the fractional derivative model required the fewest number of model parameters for a given accuracy. Remember that the this model allowed for the frequency dependance to have fractional powers (more degrees of freedom). This means that the physical nature of most viscoelastic materials follow fractional power relationships. In contrast, the decay exponential model has integer power relationships (see Equation 2.15 and Equation 2.16). Fitting the decay exponential model to experimental data is somewhat like fitting a square peg in a round hole. As shown in Figure 3.7, the numerical error distorts the frequency (and wavenumber) relationship by making it flatter. It is fortuitous that this happens to be closer to the frequency behavior of the material. Other materials must be investigated to determine whether this is a general result.

While the benefit of wide band reoptimization is slightly less dramatic, clearly the reoptimized model performance at the reference step size is closer to the accuracy of the optimized model at half the reference step size than at the full reference step size. In hindsight, a wider band source spectrum should be used to take full advantage of reoptimization. Reoptimization works best where discretization error is greatest, which for wide band is towards the upper band edge (see Figures 3.7, 3.11, and 3.15). The bandwidth of the source used is too narrow compared with the optimized bandwidth. One idea is to weight the optimization process by the power spectral density of the source to ensure a proper relationship between the two. The wideband results at different Courant numbers indicate that the benefits of reoptimization is effective at all Courant numbers. The wide band results are important because the wider bandwidth models are more efficient and because wide bandwidth models are needed to reveal the dispersive nature of viscoelastic waves.

How do the time step limits for a viscoelastic model compare with those for an elastic model at the same spatial sample rate? For the three cases studied, the time step limit for the

reoptimized parameters is magnitude limited at roughly 100 time steps corresponding to a spatial sample rate of 12 grid points per wavelength (12 gpw) at the center wavenumber. These numbers are competitive with time step limits for elastic schemes at the same sample rate and Courant numbers. For example, the Lax-Wendroff scheme used by Fricke [10] is phase error limited to about 200 time steps at 12 gpw and $N_C = .5$, and 800 time steps at 12 gpw and $N_C = .1$. But if the viscoelastic material is modeled at 12 gpw, then the elastic material must be near 120 gpw due to the high sound speed contrasts. Time step error limits for elastic materials at these sample rates are on the order of 1e5. This means that the viscoelastic material is severely limiting the accuracy of the entire model and indicates inefficient computation. All is not lost, however, when one considers that the magnitude error time step limit, as defined, is unduly hard on rapidly decaying waves. The limit is defined in terms of the error relative to the amplitude of the wave at the present time step (not at its original amplitude). If we consider two packets of energy, one traveling in the elastic region of a heterogeneous medium and the other in the viscoelastic, a 1 dB magnitude error in the first wave packet is much more significant than a 1 dB error in the viscoelastic packet. As an example, for the moderate Courant number cases studied, the amplitude has decayed more than 20 dB after 100 time steps. This suggests that a redefined error limit is needed for the high loss material modeling that adjusts for the rapid amplitude decay.

To model heterogeneous materials with high contrast in sound speed, the Courant number in the high sound speed material will be close to unity which means that the viscoelastic material will be low, perhaps as low as Nc=.1 (at least for our submarine hull problem). Optimization at low Courant number is therefore important and is the reason why in the previous section it is run at Nc=.1 and .5 by changing the temporal step size. The reoptimization at the moderate Courant number shows that the error is severely magnitude limited. It was hoped that decreasing the temporal step size would reduce the magnitude error relative to the phase error and therefore increase the modeling efficiency. The results show (compare the reoptimized error limits for case 2 and 3 in Figure 3.12 and 3.16) that there is little change in the relative error.

It makes sense that the high loss materials would be magnitude error limited. The rapid decay rate associated with high loss materials requires higher temporal than spatial sampling. This contention is in agreement with the Blanch's results for the low loss materials [2]. Blanch uses a five point centered spatial differencing with O(4) accuracy, and their model is still phase error limited. I use three point differencing with O(2) accuracy and error is magnitude limited (refer to Section 3-2 for discussion of spatial differencing). As observed in the error analysis for all three cases (see Figures 3.8, 3.12, and 3.16), reoptimization improves phase error more than magnitude error. This suggests that the benefits of reoptimization will be even more significant for lower loss materials.

3.6 Future Work

Since reoptimization is truly a new method, there remain untested ways in which it can be implemented and numerous conditions under which it may be useful; too many to cover in the present study. I mention ones that I have considered here without full and adequate disclosure.

* Low or Intermediate Loss Material Modeling- I have already discussed the possible use of reoptimization to improve low loss material modeling.

* Application to the Pseudo-Spectral Method- Carcione [5] used pseudo-spectral methods rather than finite differencing for spatial integration of the viscoelastic wave equation with memory variables. Finite differencing was still used for time domain integration and therefore reoptimization may still be applied to the constitutive model parameters.

* Wider Bandwidth Optimization and Split Finite Difference Schemes- Wider bandwidths beyond two octaves using more than two memory variables need to be investigated. It was mentioned earlier that finite difference schemes are more meaningful and efficient for wider bandwidth source spectrum. The wider the bandwidth, the more information the model carries and evolves in time. If optimization and reoptimization is done for wider bandwidth

using more memory variables (also with a wider spectrum of relaxation spectra), the more one would be inclined to use a split finite difference scheme [8]. A split scheme updates or samples the slow relaxation spectra less often; the relaxation spectra are, in effect, each treated with a different (or split) scheme. I believe that reoptimization applied to a split scheme would therefore realize the benefit closer to the narrow band results rather than the wider band results.

* Use of Mini-Oscillator Technique (MOT) to Improve Reoptimization- While the MOT model [15] by itself does not significantly improve optimization, it may help to improve reoptimization. As seen from Figures 3.11 and 3.15, the loss modulus is matched well on average and it seems that, with the ability to "tweak" the match using the MOT model, error could be drastically reduced.

* Use of Memory Variables in *ELF* Model- The Eulerian Finite Difference model (ELF), which models elastic waves in heterogeneous media with high contrast boundaries uses added dissipation of the form ϕ_{xx} to control instabilities originating at these boundaries [11]. The spectral selectivity of the memory variables may be used to improve the accuracy of such models by adding dissipation outside the source spectrum where it is needed.

3.7 Conclusion

While the focus of this chapter has been on reoptimization, it should not be overlooked that the application of optimized memory variables for high loss materials using the finite difference method has been successfully demonstrated. Specifically, that the use of optimized memory variable parameters, matched to both the real and imaginary parts of the complex modulus of a high loss viscoelastic material, and formulated in a finite difference model is collectively new. Furthermore, that the method has been analyzed in terms of stability, both phase and magnitude error, and implemented in a practical two dimensional time domain model using Huen's Predictor-Corrector scheme with three point centered spatial differencing.

I claim that reoptimization of the memory variables used for narrow band, high loss materials reduces the model size requirements by a factor of 8 times in 2D and 16 times in 3D. Given that the number of field variables is roughly doubled from that required for the full viscoelastic equation with memory variables in both 2D and 3D, the reduction in storage requirements is reduced by a factor of 4 times in 2D and 8 times in 3D. Furthermore these reductions require negligible additional work over ordinary optimization.

Similar results have been demonstrated for reoptimization over a two octave wide band. While the reduction in error is slightly less pronounced, with proper weighting between the source bandwidth and optimization bandwidth, the same improvements are possible for wideband modeling.

Bibliography

[1] J. D. Achenbach, *Wave Popagation in Elastic Solids*. North-Holland Pub. Co., Amsterdam, 1973.

[2] J. O. Blanch, J. O. A. Robertson, and W. W. Symes, "Viscoelastic finite difference modeling (tr93-94)". Technical report, Rice University, 1993.

[3] M. Burger, "On Conservation at Grid Interfaces". *SIAM Review, Vol, 24, No. 5, October 1987.*

[4] J. M. Carcione, "Seismic modeling in viscoelastic media". 59th Annual International Meeting, Society of Exploration Geophysicists, 1993.

[5] J. M. Carcione, D. Kosloff, and R. Kosloff, "Viscoacoustic Wave Propogation Simulation in the earth". *Geophysics, Vol 53, No. 6*, June, 1988.

[6] R. M. Christensen, *Theory of Viscoelasticity*. Academic Press, 1971.

[7] C. Corrado, Mid-Frequency acoustic backscattering from finite cylindrical shells and the influence of helical membrane waves. PhD thesis, MIT, 1993.

[8] J. H. Ferziger, *Numerical Methods for Engineering Application.* John Wiley & Sons, 1981.

[9] B. Fornberg, "The pseudospectral method: Comparisons with finite differences for the elastic wave equation". *Geophysics, Vol. 52. No. 4, April 1987.*

[10] J. R. Fricke, Acoustic scattering from elastic ice: A finite difference solution. PhD thesis, MIT/WHOI, 1991.

[11] J. R. Fricke and M. A. Hayner, Users Manual: *ELF* Modeling Package, Version 5.0. August 1992.

[12] J. R. Fricke, "Quasi-linear elastodynamic equations for finite difference solutions in discontinuous media", Journal of Computational Acoustics, Vol. 1, No. 3, 1993.

[13] J. R. Fricke, "Acoustic scattering from elemental Arctic ice features: Numerical modeling results." JASA, **93**, April 1993.

[14] J. M. Golden and G. A. C. Graham, *Boundary Value Problems*. Springer-Verlag, 1988.

[15] D. J. McTavish and P. C. Hughes. "Finite element modeling of linear viscoelastic structures". *The Role of Damping in Vibration and Noise Control,* ASME, 1987.

[16] C. Moler, J. Little, and S. Bangert. *MATLAB for Unix Computers*. The MathWorks, Inc., 1987.

[17] A. D. Nashif, D. Jones, and J. Henderson, *Vibration Damping*. John Wiley & Sons, 1985.

[18] Lloyd N. Trefethan, "Group velocity in finite difference schemes". SIAM Review, 1982.

[19] P. J. Torvik and D. L. Bagley, "Fractional Derivatives in the Description of Damping Materials and Phenomena". *The Role of Damping in Vibration and Noise Control,* ASME, 1987.

Appendix A

Computer Programs

As described in Chapter 1, the use of the *finite difference method with memory variables* to predict the behavior of waves in viscoelastic media requires two computer models, the constitutive model at a point in a viscoelastic material, and the finite difference model of the physical domain (a collection of points). I describe the subroutines which compose each program here, followed by a listing of the *Matlab* codes.

A.1 Description of Constitutive Model Parameter Optimization

The computer program, *opt1.m*, finds the optimum complex modulus parameters (M_R, M_l, $f_{c,l}$) for given complex modulus data in a specified wavenumber band. Opt1.m is used for either optimization or reoptimiztion. If the complex modulus model is based on the analytic dispersion relation, then the process is the ordinary optimization process. If the model is based on the numeric dispersion relation, the process is the new reoptimization process. A block diagram of opt1.m is shown in Figure A.1. A brief description of the function of each subroutine[†],

[†] All *Matlab* subroutines have a ".m" extension and are referred to, in *Matlab* vernacular, as "m" files.

represented by individual blocks, is given here. The *Matlab* computer codes are given in Section A.3.

* *opt1.m*- Top level subroutine which prompts user for input file name and calls other subroutines as shown in Figure A.1.

* inXXX.m- Input file to opt1.m provides: initial guess at model parameters; scale factors used to normalize parameters and improve conditioning of sensitivity matrix (refer to Section 3.3.1); spatial and temporal step sizes needed for reoptimization process; and wavenumber optimization band.

* apost2.m- Calculates complex modulus "data" points at specified wavenumbers from hardwired fractional derivative model parameters.

* solve1.m- Advances iteration of modulus parameters.

* fit4.m- Calculates sensitivity matrix (refer to Section 3.3.1).

* trans2.m- Buffer program which allows fit4.m to talk to ica3.m or icn4.m.

* ica3.m- Calculates analytic complex frequency at specified wavenumber.

* icn4.m- Calculates numeric complex frequency (eigenvalue) at specified wavenumber and spatial and temporal step sizes for Predictor-Corrector scheme with 3-point spatial differencing. Also, provides scale factors and phase relationships (eigenvectors) between field variables needed to set initial conditions for time domain model.

A.2 Description of Time Domain Model

The Matlab program used to verify the effectiveness of the reoptimization process is described in this section. The subroutine, referred to as m2d7.m, is shown in block diagram form in Figure A.2. The model predicts the behavior of viscoelastic waves in two dimensional homogeneous viscoelastic medium. The following is a brief description of the function of each subroutine (that has not already been described in the previous section). A listing of each subroutine is given in Section A.4.

* m2d7.m- Main subroutine: reads input file and calls other subroutines as shown in figure.

* runXXX.m- Input file specifies: complex modulus parameters; grid size and spacing; type of initial conditions; and normalization factors.

* source1.m- Calculates source function for specific model run.

* syn4.m- Calculates or synthesizes initial conditions for model run for given source function. Also, uses spectral method to predict *material* model behavior and *analytic* model behavior (refer to Section 3.2.4).

* apost2.m/icn4.m/icn3.m/fit1.m/dat5.m- Described in previous section.

* pc3.m- Advances all field variables by one time step using the Predictor-Corrector method.

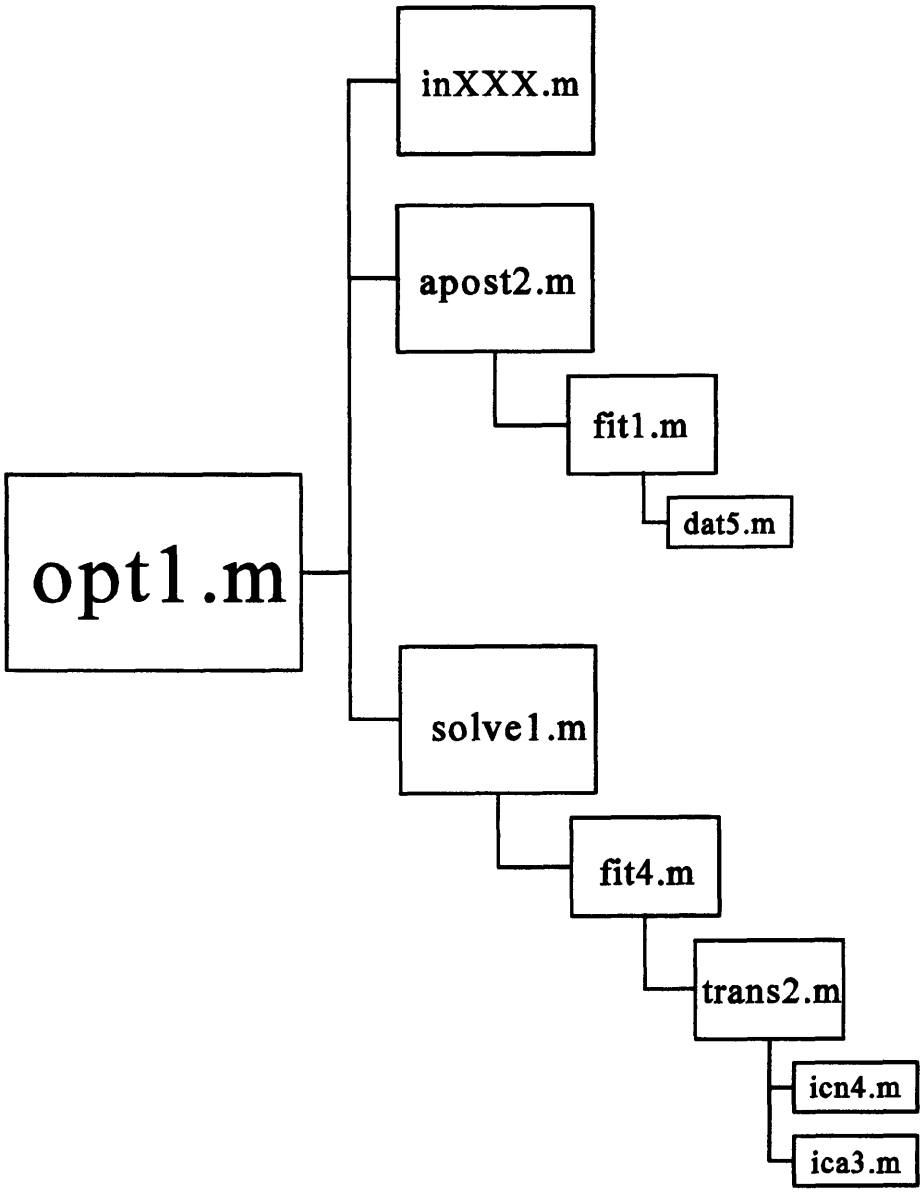


Figure A.1
Block diagram of *opt1.m* program used to find the optimized and reoptimized constitutive model parameters. Program is written in *Matlab* and uses the Gauss-Seidel method with underrelaxation to solve a set of non-linear equations.

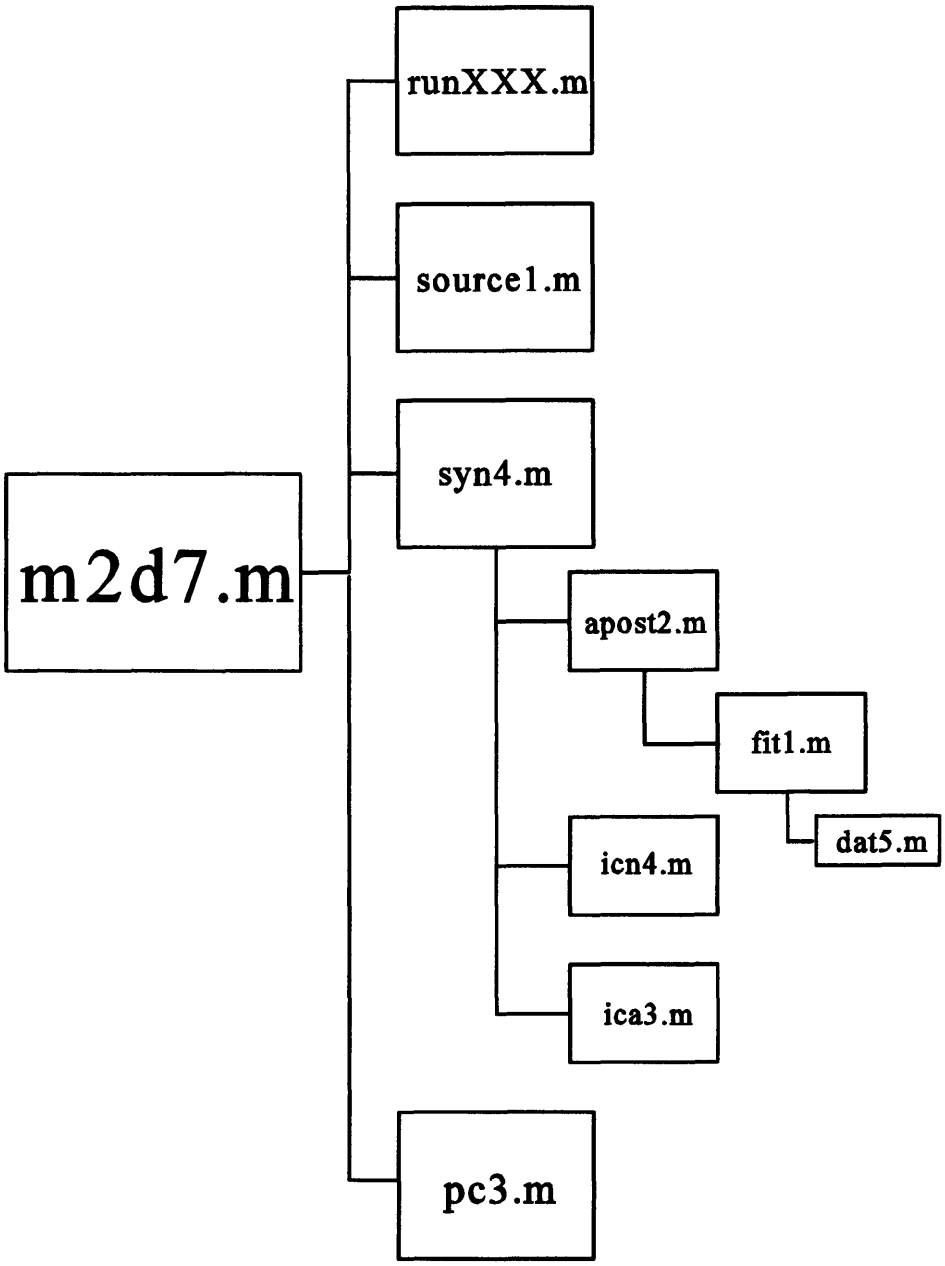


Figure A.2
Block diagram of the finite difference time domain model, *m2d7.m*. Program is written in *Matlab* and uses Huen's Predictor-Corrector scheme with 3-point centered spatial differencing to model waves in a two dimensional, high loss, homogeneous, viscoelastic medium.

A.3 Listing of Optimization Program

```
% in101a.m
% input file for opt1.m

%------------------ input variables ----------------------%
 tol=1.e-4;                  % tol on optimized x parameters
 delt=1e-4;                  % percent perturb of x
 urp=1;                      % underrelax. parameter, 1=normal, 0=stop
 keys=[0,2,2];               % key(1): not used
                             % key(2): 1=anal dd, 2=num. dd(re trans2.m)
                             % key(3): optimization weighting
 itmax=1;                    % maximum iterations allowed
 N=20;                                % number k points
 k=linspace(10,40,N);                 % wavenumber band

 rho=1000;                            % density
 Mr=8.0232e7;                         % relaxed modulus
 Mm=[.6550,1.1866]*1e8;               % memory variable moduli
 fc=[3.375,20.485]*1e3;               % mv corner freq


 dx=1e-2;                             % spatial step
 NC=.5;                               % Courant number
 theta=0;                             % grid angle (radians)

 MOD=5e7;                             % ref modulus
 RHO=1000;                            % ref density
 LEN=1;                               % ref length
 TIME=LEN*(RHO/MOD)^.5;               % ref time
%-------------------------------------------------------%
```

```
% opt1.m

 clear all
 global  keys itmax tol delt urp kh theta
 global  Nmag Nph Nlim gpw iter_reqd
%------------------- get input file ----------------------%
 infile=input('input file name? ','s');
 eval(infile);

 Mu=Mr+sum(Mm);                          % unrelaxed modulus
 cu=(Mu/rho)^.5;                         % unrelaxed sound speed
 dt=NC*dx/cu;                            % grid time step
 kh=k*dx;                                % non-dimensional k
%------------------- experimental data --------------------%
 [Gp,Gpp]=apost2(k);
%------- normalized quantities ---------%
  nrho=rho/RHO;
  nMu=Mu/MOD;
  nMm=Mm/MOD;
  nfc=fc*TIME;
  ndx=dx/LEN;
  ndt=dt/TIME;
  nk=k*LEN;
  nGp=Gp/MOD;
  nGpp=Gpp/MOD;
%------------------ optimize and renormalize ----------------%
  temp1='solve1(nMu,nMm,nfc,ndx,ndt,nk,nGp,nGpp,nrho)';
  [nMu,nMm,nfc,nEp,nEpp]=eval(temp1);

  Mu=nMu*MOD;
  Mm=nMm*MOD;
  fc=nfc/TIME;
  Ep=nEp*MOD;
  Epp=nEpp*MOD;

  Mr=Mu-sum(Mm);
  cu=(Mu/rho)^.5;
  NC=cu*dt/dx;
  NT=dt./(2*pi*fc);
  NM=Mm./Mu;
  wx=((1/rho)*(Ep+j*Epp).^.5).*k;
  wxd=((1/rho)*(Gp+j*Gpp).^.5).*k;       % wk data
```

```
% solve1.m

function[Mu,Mm,fc,Ep,Epp]=solve1(Mu,Mm,fc,dx,dt,k,Gp,Gpp,rho)';

  global  keys itmax tol delt urp kh theta
  global  k rho Nmv dt dx yexp keys
  global  Nmag Nph Nlim gpw iter_reqd

  yexp=[Gp,Gpp]; % experimental data for matching
  khx=kh*cos(theta); % kh in x-dir

%------------------------- fit ----------------------------%
  iter_reqd=0;
  x=[Mu,Mm,fc].';                 % (Nvx1) not (1xNv)
  N=max(size(yexp))/2;            % number of data points (N must be even)
  Nmv=max(size(Mm));              % number of memory var. param. pairs
  Nv=size(x);                     % total number of parameters
  f1='trans2(x,k,rho,Nmv,dt,dx,yexp,keys)';
  if itmax>0  % iterate to find best fit for x parameters
   [xiter,fiter,Am]=fit4(f1,itmax,tol,delt,urp,x);
   [dummy,iter_reqd]=size(xiter);     % iter required for converg.
   if iter_reqd==[]
   end;
   x=xiter(:,iter_reqd); % new x's(Nv x 1)
  end

%------------------------ post ----------------------------%
  Mu=x(1);
  Mm=x(2:1+Nmv).';
  fc=x(2+Nmv:Nv).';
  Mr=Mu-sum(Mm);
  tau=(1)./(2*pi.*fc);
  cu=(Mu/rho)^.5;
  NC=cu*dt/dx;

% find final value of Ep and Epp vs k (will not work for vs f)
  mode=1;
  ky=0;
  sgn=-1;
  nopt=[1,2,sgn,1];
  wx=[];
  for n=1:max(size(k))
     kx=k(n);
```

```
    if keys(2)==1
     [temp1]=ica3(mode,kx,ky,Mu,rho,Mm,tau,sgn);          %temp1=wc
    elseif keys(2)==2
     [temp1]=icn4(mode,kx,ky,Mu,Mm,rho,tau,dt,dx,nopt);
    end
    wx=[wx,temp1];
  end
  cx=wx./k;
  Ex=rho*cx.^2;
  Ep=real(Ex);
  Epp=imag(Ex);
%---------------------- error analysis --------------------%
  wkd=((1/rho)*(Gp+j*Gpp).^.5).*kh*(NC/cu);  % wk data
  wk=((1/rho)*(Ep+j*Epp).^.5).*kh*(NC/cu);
  Nph=(pi/4)./abs(real(wk-wkd));
  Nmag=(1)./(20*log10(exp(1))*abs(imag(wk-wkd)));
  gpw=2*pi./khx; % grids per wavelength along x-direction
  Nlim=min([Nph;Nmag]);
```

```
function[xiter,fiter,Am]=fit4(f1,itmax,tol,delt,urp,x)
global  k rho Nmv dt dx yexp keys
%--------------------------------%
 M=max(size(x));
 xiter=[];
 fiter=[];
 iter=1;
 if itmax>0;
   key1='gogo';
 else
   key1='nogo';
 end;
 while key1=='gogo'
      iter
   Am=[];
       for n=1:M                  % M, total no. param (2XL)
         n;
         Fa=eval(f1);             % (Nx1)
         x(n)=(1+delt)*x(n);      % perturb x delt perc. (Nx1)
         Fb=eval(f1);             % (Nx1)
         x(n)=x(n)/(1+delt);      % reset x
         dF=(Fb-Fa)/(delt*x(n));  % (Nx1)
         Am=[Am,dF];              % (NxM)
       end % for n
   delx=Am\(-Fa);                 % del x params, not dx
   xnew=x+delx;
   condAm=cond(Am);
   xerr=abs((x-xnew)./x);         % percent abs error (1XN)
   merr=max(xerr);                % max percent abs error (1X1)
   x=(1-urp)*x+urp*xnew;          % underrelax x
   xiter=[xiter,x];               % track x thru iter (1XN)
   fiter=[fiter,Fa];              % track x thru iter (1XN)
   if iter >= itmax;              % enough trys?
     key1='nogo';
     message='fit4 did not converge';
   end
   iter=iter+1;
   if merr<tol;                   % solution converge?
       key1='nogo';
       message='fit4 converged';
   end
end % for iter
xiter=xiter; message
```

```
function[wc,amp]=ica3(mode,kx,ky,Mu,rho,Mm,tau,sgn);
Nmv=max(size(Mm));
temp0=-sgn*i*[0,-i*kx*Mu,-i*ky*Mu,-1*ones(1,Nmv);
        -i*kx/rho,0,0,zeros(1,Nmv);
        -i*ky/rho,0,0,zeros(1,Nmv)];
temp2=[];
for n=1:Nmv
  tt1=[0,-i*kx*Mm(n)/tau(n)];
  tt2=[-i*ky*Mm(n)/tau(n),zeros(1,n-1)];
  tt3=[-1/tau(n),zeros(1,Nmv-n)];
  temp1=i*[tt1,tt2,tt3];
  temp2=[temp2;temp1];
end
A=[temp0;temp2];
[vect,val]=eig(A);                    % val is a diagonal matrix
valv=diag(val).';                     % diagonal of eigenval matrix (1xNv)
%-------------------------%
[dum,index1]=sort(real(valv));
temp1=valv(index1);
temp2=[temp1(1),temp1(3+Nmv),temp1(2:2+Nmv)];
wc=temp2(mode);
vect1=vect(:,index1);
vect2=vect1(:,[1,3+Nmv,2:2+Nmv]);
amp=vect2(:,mode)./vect2(1,mode);
```

```
function[wc,vect]=icn4(mode,kx,ky,Mu,Mm,rho,tau,dt,dx,opt)
%--------------------------------------%
Nmv=max(size(tau));
khx=kx*dx;
khy=ky*dx;
cg=dx/dt;
%--------------------------------------%
if opt(1)==1                          % 3pt centered spatial differentiation
  sx=j*sin(khx);
  sy=j*sin(khy);
 elseif opt(1)==2                     % 5pt centered spatial differentiation
  sx=j*(-sin(2*khx)+8*sin(khx))/6;
  sy=j*(-sin(2*khy)+8*sin(khy))/6;
end
 tt1=[0, -Mu*sx/cg, -Mu*sy/cg];
 tt2=[-sx/(rho*cg),0,0];
 tt3=[-sy/(rho*cg),0,0];
A1=[tt1;tt2;tt3];
 if Nmv>0
   A1(1,4:Nmv+3)=[-dt*ones(1,Nmv)];
   for n=1:Nmv
       A1(3+n,2)=-Mm(n)*sx/(tau(n)*cg);
       A1(3+n,3)=-Mm(n)*sy/(tau(n)*cg);
       A1(3+n,3+n)=-dt/tau(n);
   end % n
end % if
if opt(2)==1                          % Euler-Foward
  A2=eye(3+Nmv)+A1;
 elseif opt(2)==2                     % Predictor-Corrector
  A2=eye(3+Nmv)+A1+.5*A1^2;
end
%--------------------------------------------%
[uvects,gval]=eig(A2);
g=diag(gval).'; % take diagonal of gval matrix (1xNv)
valv=log(g)/(j*opt(3)*dt);            % solve for w
[dum,index1]=sort(real(valv));        % sort eigenvalues
temp1=valv(index1);
wcs=[temp1(1),temp1(3+Nmv),temp1(2:2+Nmv)];
wc=wcs(mode);
temp1=uvects(:,index1);               % sort eigenvectors
vects=[temp1(:,1),temp1(:,3+Nmv),temp1(:,2:2+Nmv)];
vect=vects(:,mode)./vects(1,mode);
```

```
function[dy]=trans2(x,k,rho,Nmv,dt,dx,yexp,keys);
N=max(size(k)); % prelim calc
Nv=max(size(x));
Mu=x(1);
Mm=x(2:1+Nmv);
fc=x(2+Nmv:Nv);
tau=(1)./(2*pi.*fc);
wx=[];  %
mode=1;
ky=0;
sgn=-1;
nopt=[1,2,sgn,1];
for n=1:max(size(k))
   kx=k(n);
   if keys(2)==1
    [temp1]=ica3(mode,kx,ky,Mu,rho,Mm,tau,sgn); %temp1=wc
   elseif keys(2)==2
    [temp1]=icn4(mode,kx,ky,Mu,Mm,rho,tau,dt,dx,nopt);
   end
   wx=[wx,temp1];
end
cx=wx./k; Ex=rho*cx.^2;
keys(2)=1;
if keys(3)==1
  dy1=((yexp(1:N)-real(Ex))./yexp(1:N));
  dy2=(yexp(N+1:2*N)-imag(Ex))./yexp(N+1:2*N);
  dy=[dy1,dy2]';
elseif keys(3)==2
  dy1=yexp(1:N)-real(Ex);
  dy2=yexp(N+1:2*N)-imag(Ex);
  dy=[dy1,dy2].';
elseif keys(3)==3
  dy1=(yexp(1:N)-real(wx))/(pi/4);
  dy2=(yexp(N+1:2*N)-imag(wx))*20*log10(exp(1));
  dy1=(yexp(1:N)-real(wx))/4;
  dy2=(yexp(N+1:2*N)-imag(wx))*8;
  dy=[dy1,dy2]';
elseif keys(3)==4 % Q opt
  dq=yexp(N+1:2*N)./yexp(1:N)-imag(Ex)./real(Ex);
  dy1=((yexp(1:N)-real(Ex))./yexp(1:N));
  dy=[dy1(N/2),dq(1:N-1),dq].';
end
```

```
function[fcn]=dat5(w,k)

w=w(1)+j*w(2);

ao=4e7;
a1=4e5*(5^.5);
b1=1e-3*(5^.5);
a=.5;
b=.5;
bouy=1e-3;

fcnx=w/k-(bouy*(ao+a1*(j*w).^a)./(1+b1*(j*w).^b) ).^.5;
fcn=[real(fcnx);imag(fcnx)];
```

A.4 Listing of Time Domain Model

```
%-- c:\thesis\model\run701.m -------%

 clear all;
%----------input -----------%
% model parameters
 not=1e-30;                    % convienient small number
 K=21;                         % number of time increments
 M=4;                          % number of y increments
 N=6;                          % number of x increments
 dx=1/(N-2);                   % x and y step
 dt=dx/2;                      % time step
 fact=2^3;                     % increase eff. model size by "fact"or
 sgn=-1;                       % must be neg.
TF=dt*(K-1);

%--- material properties ---%
 rho=1;                        % density
 Mr=1;                         % relaxed modulus
 Mm=[0,0,0,0];                 % memory variable moduli
 fc=[1,2,3,4];                 % mv corner freq
 Nmv=4;                        % number of memory variables
% ---source parameters ---%
 sxkey=2;                      % (1=Gaussian, 2=k-tone, 3=const. 4=Ricker)
 sykey=3;                      % source key for y or m-dir.
 kcx=2*pi;                     % Gaussian source center x-wavenumber
 Ax=1;                         % x-amplitude
 xcx=.1;                       % x-spatial decay factor
 xsx=.5;                       % x-spatial delay
 kcy=.125;                     % center y-wavenumber
 Ay=1;                         % y-amplitude
 xcy=.35;                      % y-spatial decay factor
 xsy=.1;                       % y-spatial delay

% --output control parameters --%
 snap=[1,K*fact]; % k-steps for x-y pressure snapshots
 xtrace=2;        % y or m pos. of x length trace
 ytrace=2;        % x or n pos. of y length trace
```

```
%---------- normalize ------------------%
 MOD=1;
 RHO=1;
 LEN=1;
 TIME=LEN*(RHO/MOD).^.5;

 rho=rho./RHO;
 Mr=Mr/MOD;
 Mm=Mm./MOD;
 fc=fc.*TIME;
 dx=dx/LEN;
 dt=dt/TIME;
 TF=TF/TIME;


%----call main subroutine-------%
  m2d7
%--- plot----%
  figure(1)
  plot(x,pfa,x,pfn,x,pf,'o');
  axis([0,1,-1.5,1.5])
```

```
% m2d7.m
%----------------- precalc--------------------%
 t1=clock;
 K=fact*(K-1)+1;
 N=fact*(N-2)+2;
 dt=dt/fact;
 dx=dx/fact;
 tau=(1)./(2*pi*fc);          % relax. times
 Mu=Mr+sum(Mm);               % unrelax. modulus
 cu=(Mu/rho)^.5;              % unrelax. sound speed
 cg=dx/dt;                    % grid speed
 Nc=cu*dt/dx                  % Courant no.
 Nm=Mm./Mu
 Nt=dt./tau
 x=dx*(0:N-3);                % n-grid coord. (2 to N-1)
 y=dx*(0:M-3);                % m-grid coord. (2 to M-1)
 Lx=max(x)+dx;                % total x length
 Ly=max(y)+dx;                % total y length
% TF=(K-1)*dt;                % final time (for pressure)
 tp=0:dt:TF;                  % time (for pressure)
 c1=-Mu*dt/(2*dx);            % constants
 c2=-dt/(2*dx*rho);
 c3=-dt;
 c4=-dt./tau;
 c5=-Mm.*dt./(tau*2*dx);

%------------ initial conditions----------------%
 param=[kcx,xsx,xcx];
 [fx,Kx,Fx,FFx]=source1(x,param,sxkey);
 param=[kcy,xsy,xcy];
 [fy,Ky,Fy,FFy]=source1(y,param,sykey);
 global M N Mu rho Mm tau dx dt TF sgn
 [p,u,v,r1,r2,r3,r4,pfm,pfa,pfn]=syn4(x,fx,y,fy);

 tpx=[p(xtrace,2:N-1)];       % x pressure trace (1x(N-2))
 tux=[u(xtrace,2:N-1)];       % x velocity trace
 tr1x=[r1(xtrace,2:N-1)];     % x velocity trace
 tpy=[p(2:M-1,ytrace)];       % y pressure trace (1x(M-2))
 tuy=[u(2:M-1,ytrace)];       % y velocity trace

%---------------- iterate:--------%
 t2=clock;
 msg='begin iterations...'
```

```
global  M N Nmv c1 c2 c3 c4 c5
if snap(1)==1; psnap1=p(2:M-1,2:N-1); end;

for k=2:K
    k
   [p,u,v,r1,r2,r3,r4]=pc3(p,u,v,r1,r2,r3,r4);
    for kk=1:max(size(snap))
        if k==snap(kk)

eval(['psnap',int2str(snap(kk)),'=p(2:M-1,2:N-1);']);
        end
    end
    tpx=[tpx;p(xtrace,2:N-1)];
    tux=[tux;u(xtrace,2:N-1)];
    tr1x=[tr1x;r1(xtrace,2:N-1)];
    tpy=[tpy,p(2:M-1,ytrace)];
    tuy=[tuy,u(2:M-1,ytrace)];
end % for k
msg='...completed iteration'
t3=clock;

%------------ post processing----------------%
pfm=pfm(1,:);
pfa=pfa(1,:);
pfn=pfn(1,:);
pf=tpx(K,:);                                % final pressure
uo=tux(1,:);                                % initial vel.
uf=tux(K,:);                                % final vel.
r1o=tr1x(1,:);                              % initial vel.
r1f=tr1x(K,:);                              % final vel.

% max and rms error versus amplitude of pfn and pfa
merrn=max(( ( (pf-pfn)/max(pfn) ).^2).^.5);
merra=max(( ( (pf-pfa)/max(pfa) ).^2).^.5);
errn=( sum(( (pf-pfn)/max(pfn) ).^2)/max(size(pfn)) ).^.5
erra=( sum(( (pf-pfa)/max(pfa) ).^2)/max(size(pfa)) ).^.5
t4=clock;
elapsed_time=etime(t4,t1)
iter_time=etime(t3,t1)
```

```
% source1.m
  function[y,wtot,Y,YY]=source1(t,param,key)

%-------------------------------------------%
  N=max(size(t)); % even
  fo=param(1)/(2*pi);
  if key==1 % cosine modulated Gaussian
     to=param(2);
     tau=param(3);
     y=exp(-pi*((t-to)/tau).^2).*cos(2*pi*fo*(t-to));
   elseif key==2
     y=cos(2*pi*fo*t);
   elseif key==3
     y=ones(1,N);
   elseif key==4 % shifted zero phase Ricker wavelet
     to=param(2);
     y=exp(-.5*fo^2*(t-to).^2).*cos(pi*fo*(t-to));
  end
%-------------------------------------------%
  Y=fftshift(fft(y,N))/N;
  YY=abs(Y)

  dt=t(2)-t(1);
  fsamp=1/(2*dt);
  ftot=linspace(-fsamp,fsamp*(N-2)/N,N);
  wtot=2*pi*ftot;
```

```
% syn4.m
function[p,u,v,r1,r2,r3,r4,pfm,pfa,pfn]=syn4(x,fx,y,fy)
global M N Mu rho Mm tau dx dt TF sgn

I=max(size(fy));
J=max(size(fx));
dx=abs(x(2)-x(1));
dy=dx;

fxy=fy.'*fx;              % MxN
M=I+2;                                  % must be even
N=J+2;                                  % must be even
kxs=2*pi/dx;                    % sample spatial freq in x dir.
kys=2*pi/dy;
Ky=kys/I*[0:I/2-1,-I/2:-1];    % no shifting, I even
Kx=kxs/J*[0:J/2-1,-J/2:-1];    % no shifting, J even
Y=fft2(fxy,I,J);
YY=abs(Y);                      % pwr. spect. density

%-------------------------------------%
P=zeros(I,J);
U=zeros(I,J);
V=zeros(I,J);
R1=zeros(I,J);
R2=zeros(I,J);
R3=zeros(I,J);
R4=zeros(I,J);
PFM=zeros(I,J);
PFA=zeros(I,J);
PFN=zeros(I,J);

%-------- synthesis ----------%
nopt=[1,2,sgn,1];  % options for icn4.m
for ii=1:I
  ky=Ky(ii);
  for jj=1:J
    kx=Kx(jj);
    ko=(kx^2+ky^2)^.5;
    if kx>0;
%    if ky>0;
       mode=1;
      else
       mode=2;
```

```
    end
    if YY(ii,jj)>1e-1
      [wca,ampa]=ica3(mode,kx,ky,Mu,rho,Mm,tau,sgn);
      [wcn,ampn]=icn4(mode,kx,ky,Mu,Mm,rho,tau,dt,dx,nopt);
      amp=ampa;
      P(ii,jj)=amp(1)*Y(ii,jj);
      U(ii,jj)=amp(2)*Y(ii,jj);
      V(ii,jj)=amp(3)*Y(ii,jj);
      R1(ii,jj)=amp(4)*Y(ii,jj);
      R2(ii,jj)=amp(5)*Y(ii,jj);
      R3(ii,jj)=amp(6)*Y(ii,jj);
      R4(ii,jj)=amp(7)*Y(ii,jj);
 if kx~=0
     [sEdata,lEdata,rhodata]=apost2(kx);
      mscale=1/5e7;
      rhoscale=1/1000;
   cdata=(mscale*(sEdata+j*lEdata)/(rhoscale*rhodata))^.5;
      wcm=-cdata*kx;
 else
     wcm=0;
 end
%keyboard
      PFM(ii,jj)=amp(1)*Y(ii,jj)*exp(j*sgn*wcm*TF);
      PFA(ii,jj)=amp(1)*Y(ii,jj)*exp(j*sgn*wca*TF);

      PFN(ii,jj)=amp(1)*Y(ii,jj)*exp(j*sgn*wcn*TF);
    end % if YY
  end % jj
end % ii
p=real(ifft2(P));  % take real(-) to rid roundoff error
u=real(ifft2(U));
v=real(ifft2(V));
r1=real(ifft2(R1));
r2=real(ifft2(R2));
r3=real(ifft2(R3));
r4=real(ifft2(R4));
pfm=real(ifft2(PFM));
pfa=real(ifft2(PFA));
pfn=real(ifft2(PFN));
%-----------------%
p=real([0,p(M-2,:),0; p(:,N-2),p,p(:,1); 0,p(1,:),0]);
u=real([0,u(M-2,:),0; u(:,N-2),u,u(:,1); 0,u(1,:),0]);
v=real([0,v(M-2,:),0; v(:,N-2),v,v(:,1); 0,v(1,:),0]);
```

```
r1=real([0,r1(M-2,:),0; r1(:,N-2),r1,r1(:,1); 0,r1(1,:),0]);
r2=real([0,r2(M-2,:),0; r2(:,N-2),r2,r2(:,1); 0,r2(1,:),0]);
r3=real([0,r3(M-2,:),0; r3(:,N-2),r3,r3(:,1); 0,r3(1,:),0]);
r4=real([0,r4(M-2,:),0; r4(:,N-2),r4,r4(:,1); 0,r4(1,:),0]);
```

```
function[pF,uF,vF,r1F,r2F,r3F,r4F]=pc3(p,u,v,r1,r2,r3,r4);
global  M N Nmv c1 c2 c3 c4 c5

%----- p & u predictor-------%
for m=2:M-1
   for n=2:N-1
      SUMr(m,n)=(r1(m,n)+r2(m,n)+r3(m,n)+r4(m,n)); %(M-1xN-1)
      dpv(m,n)=p(m+1,n)-p(m-1,n);  %(M-1xN-1)
      dpu(m,n)=p(m,n+1)-p(m,n-1);  %(M-1xN-1)
      duv(m,n)=v(m+1,n)-v(m-1,n)+u(m,n+1)-u(m,n-1);%(M-1xN-1)
      pS(m,n)=p(m,n)+c1*duv(m,n)+c3*SUMr(m,n);
      uS(m,n)=u(m,n)+c2*dpu(m,n);
      vS(m,n)=v(m,n)+c2*dpv(m,n);
   end
end
pS(1,:)=pS(M-1,:); pS(M,:)=pS(2,:);
pS(:,1)=pS(:,N-1); pS(:,N)=pS(:,2);
uS(1,:)=uS(M-1,:); uS(M,:)=uS(2,:);
uS(:,1)=uS(:,N-1); uS(:,N)=uS(:,2);
vS(1,:)=vS(M-1,:); vS(M,:)=vS(2,:);
vS(:,1)=vS(:,N-1); vS(:,N)=vS(:,2);
%note: exterior nodes of SUMr, dpv, dpu, and duv not used

%---------- r predictor---------------%
if Nmv>0; %note: Nmv either 0 or 4
   for m=2:M-1
     for n=2:N-1
       r1S(m,n)=r1(m,n)+c4(1)*r1(m,n)+c5(1)*duv(m,n);
       r2S(m,n)=r2(m,n)+c4(2)*r2(m,n)+c5(2)*duv(m,n);
       r3S(m,n)=r3(m,n)+c4(3)*r3(m,n)+c5(3)*duv(m,n);
       r4S(m,n)=r4(m,n)+c4(4)*r4(m,n)+c5(4)*duv(m,n);
       SUMrS(m,n)=(r1S(m,n)+r2S(m,n)+r3S(m,n)+r4S(m,n));
     end
   end
   r1S(1,:)=r1S(M-1,:); r1S(M,:)=r1S(2,:);
   r2S(1,:)=r2S(M-1,:); r2S(M,:)=r2S(2,:);
   r3S(1,:)=r3S(M-1,:); r3S(M,:)=r3S(2,:);
   r4S(1,:)=r4S(M-1,:); r4S(M,:)=r4S(2,:);
   r1S(:,1)=r1S(:,N-1); r1S(:,N)=r1S(:,2);
   r2S(:,1)=r2S(:,N-1); r2S(:,N)=r2S(:,2);
   r3S(:,1)=r3S(:,N-1); r3S(:,N)=r3S(:,2);
   r4S(:,1)=r4S(:,N-1); r4S(:,N)=r4S(:,2);
 else
```

```
 SUMrS(m,n)=0;
end % if Nmv
%---------- p & u corrector----------%
 for m=2:M-1
   for n=2:N-1
     dpuS(m,n)=pS(m,n+1)-pS(m,n-1);
     dpvS(m,n)=pS(m+1,n)-pS(m-1,n);
     duvS(m,n)=vS(m+1,n)-vS(m-1,n)+uS(m,n+1)-uS(m,n-1);
 v1=c1*.5*(duvS(m,n)+duv(m,n))+.5*c3*(SUMrS(m,n)+SUMr(m,n));
     pF(m,n)=p(m,n)+v1;
     uF(m,n)=u(m,n)+c2*.5*(dpuS(m,n)+dpu(m,n));
     vF(m,n)=v(m,n)+c2*.5*(dpvS(m,n)+dpv(m,n));
   end
  end % for m-p
  pF(1,:)=pF(M-1,:); pF(M,:)=pF(2,:);
  pF(:,1)=pF(:,N-1); pF(:,N)=pF(:,2);
  uF(1,:)=uF(M-1,:); uF(M,:)=uF(2,:);
  uF(:,1)=uF(:,N-1); uF(:,N)=uF(:,2);
  vF(1,:)=vF(M-1,:); vF(M,:)=vF(2,:);
  vF(:,1)=vF(:,N-1); vF(:,N)=vF(:,2);
%----------- r corrector----------%
if Nmv>0;
 for m=2:M-1
   for n=2:N-1
v1=c4(1)*.5*(r1(m,n)+r1S(m,n))+.5*c5(1)*(duv(m,n)+duvS(m,n));
v2=c4(2)*.5*(r2(m,n)+r2S(m,n))+.5*c5(2)*(duv(m,n)+duvS(m,n));
v3=c4(3)*.5*(r3(m,n)+r3S(m,n))+.5*c5(3)*(duv(m,n)+duvS(m,n));
v4=c4(4)*.5*(r4(m,n)+r4S(m,n))+.5*c5(4)*(duv(m,n)+duvS(m,n));
     r1F(m,n)=r1(m,n)+v1;
     r2F(m,n)=r2(m,n)+v2;
     r3F(m,n)=r3(m,n)+v3;
     r4F(m,n)=r4(m,n)+v4;
   end
 end
  r1F(1,:)=r1F(M-1,:); r1F(M,:)=r1F(2,:);
  r2F(1,:)=r2F(M-1,:); r2F(M,:)=r2F(2,:);
  r3F(1,:)=r3F(M-1,:); r3F(M,:)=r3F(2,:);
  r4F(1,:)=r4F(M-1,:); r4F(M,:)=r4F(2,:);
  r1F(:,1)=r1F(:,N-1); r1F(:,N)=r1F(:,2);
  r2F(:,1)=r2F(:,N-1); r2F(:,N)=r2F(:,2);
  r3F(:,1)=r3F(:,N-1); r3F(:,N)=r3F(:,2);
  r4F(:,1)=r4F(:,N-1); r4F(:,N)=r4F(:,2);
end % if Nmv
```