Effects on Seismic Absorption due to Changed Pore Surface Properties Resulting from Exposure to Propanol:

A Study Utilizing Artificial Glass Cracks

by

Oliver S. Boyd

B.A., University of Colorado at Boulder

A thesis submitted to the Faculty of the Graduate School of the University of Colorado in partial fulfillment of the requirement for the degree of

Master of Science

Department of Geology

1997

This thesis for the Master of Science degree by

Oliver S. Boyd

has been approved for the

Department of Geology

by

________________________________.
Prof. Hartmut Spetzler

________________________________.
Prof. Shemin Ge

________________________________.
Prof. Anne Sheehan

________________________________.
Mr. Ivan Getting

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Date:______________.

ABSTRACT

Boyd, Oliver S. (M.S. Geology)

Effects on Seismic Absorption due to Changed Pore Surface Properties Resulting From Exposure to Propanol: A Study Utilizing Partially Saturated Artificial Glass Cracks

Thesis directed by Professor Hartmut Spetzler

The study of multi-phase flow has been expanding in recent times due to a deepening appreciation of, and a greater ability to predict, the system's complex behavior. This thesis shows how altering the solid phase in the presence of gas and liquid phases produces definite changes in the measured quantities, attenuation and stiffness, for a partially saturated, sinusoidally stressed, artificial glass crack.

The two key means of dissipation for this system are energy lost due to viscous effects and energy lost due to restricted movement of the contact line, the interface between the gas, liquid, and solid phases. The viscous effects are relatively well known and simple, but contact line movement, involving the interaction of the three phases, requires a broader understanding. The chemistry between the solid phase and liquid phase and the intermolecular forces in the liquid phase become exceedingly important.

Model predictions are tested against experimental data obtained on an attenuation spectrometer. The spectrometer produces oscillatory deformation of a partially saturated artificial crack made of glass from .001 to 100 Hz with amplitudes up to .5 um. The phase of the displacement relative to an elastic standard, a measure of dissipated energy, is deduced after digitizing and fitting the measured signal.

Viscous effects are found to explain the measured attenuation above 10 Hz and restricted contact line movement does its job below 1 Hz. There combination allows for nearly perfect agreement over the entire frequency range. However, large discrepancies exist between predicted and measured stiffness.

ACKNOWLEDGMENTS

Many people are responsible for the completion of this thesis. I would like to thank Hartmut Spetzler for all of his ideas and constant encouragement. Ivan Getting has been an invaluable resource in the experimental design and implementation of the attenuation meter. I would also like to thank Pamela Burnley and Bill Waite for attempting to answer most of my questions. A special thanks go to Rainer Moerig for all that I learned during our endless conversations and my father, Jeff Boyd, whose programming experience and keen mind helped reduce my programming bugs to theoretical contradiction. I am also grateful to my mom and sister who also helped keep me together.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

Characterizing upper crustal response to imposing pressure gradients provides valuable information to all who use pressure waves to image subsurface phenomena. Much effort has been expended in making observations and developing theories about how energy is propagated in the subsurface. This thesis is primarily concerned with the role of partial fluid saturation in the dissipation of seismic energy.

The partially saturated system has been studied extensively yet little is known about the role of the contact line, the interface between the three phases, gas, liquid, and solid. Through the application of experimental and theoretical techniques, this thesis will show how the properties of the solid phase at its surface affect the mobility of the contact line and thereby substantially affect the magnitude and frequency dependence of contact line and viscosity related dissipation. The measurements are performed on a cylindrical crack geometry with Propanol as the surface affecting agent.

Background

Many precious resources lie below our feet. There are reservoirs full of water, oil and gas. There are faults and folds and the remains of our ancestors. Approximately 100% of the Earth's mass is below our feet. The scientific, cultural, and economic evolution of our species depends on an understanding of the subsurface. There are several ways to gather information about the material within the Earth depending on the depth and attribute of inquiry and the resources available to pursue such inquiry. Direct methods such as bore holes provide a wealth of direct observations that allow for characterization on a sub-meter scale. However, they are relatively expensive and suffer an extremely low resolution on larger scales. Direct methods also become unfeasible with depths greater than several kilometers. Indirect methods allow a much greater resolution on a much larger scale. Measurements can be made on phenomena from the crust to the core to distant stars and their planets. The indirect methods require a significant amount of theory to describe the observed phenomena and frequently result in non-unique interpretations. At the forefront of acquisition techniques for the earth's interior are the methods associated with seismic wave propagation.

Seismic waves are pressure waves that propagate through the subsurface. Their velocity and phase is dependent on the complex modulus of the medium the waves are passing through. Their frequency content is dependent on the depth of the source and the dispersion of the intervening material. Common velocities are on the order of several km/s and frequencies on the order of 1 Hz. These conditions make measurement difficult in the lab because the samples are small compared to the wavelength of the passing waves. The sample's influence on the dissipation of seismic energy is virtually undetectable and so empirical development has relied on measurements made in the sonic and ultrasonic frequency ranges.

Important steps toward an understanding of wave propagation in porous rocks were made by Born and Owen in 1935. Measuring the width of a resonant peak in the sonic to sub-sonic frequency range, they determined the attenuation and discovered that the behavior of their rock samples possess a strong dependence on pore content. To explain this Biot published several papers outlining the theory behind the propagation of seismic waves in saturated porous solids (1956a; 1956b; 1962). Much of his theory is based on inertial effects of the pore content. But since its inception, the theory has not adequately modeled experimental results for velocity dispersion and attenuation (O'Connell and Budiansky, 1977; Mavko and Jizba, 1991). Bulau et al. (1984), Jones and Nur (1983), and Dvorkin et al. (1993) note that experimental observations show a frequency dependence that is in contradiction to Biot theory. With increasing oil viscosity in a Berea sandstone, measured peaks in attenuation and dispersions in velocity moved to lower frequencies (Winkler, 1983). Biot predicted their movement to higher frequencies. Nur and Simmons (1969) and O'Connell and Budiansky (1977) suggested that another mechanism more strongly dependent on the viscosity of the pore content was involved. This mechanism was referred to and is called local fluid flow.

During and after the development of local fluid flow theory (Murphy et al., 1986; Norris 1993; Gurevich and Lopotnikov, 1995) and because of improvements in measurement techniques, additional patterns began to emerge. Scattering in dry rocks (Burridge and Chang, 1989; Shapiro et al., 1994) and partially saturated rocks, simulated (Gurevich et al., 1997), squirt flow between pores (Palmer and Traviolia, 1980; Murphy et al., 1986; Akbar et al., 1994), and contact line movement (Miksis, 1988; Waite et al., 1997) have been proposed.

Gurevich et al. (1997) combined Biot theory, local fluid flow, and scattering in a simulated stratified rock and found that in the seismic to sonic frequency range, local fluid low and scattering dominate over Biot theory. They also found that the attenuation peak due to interlayer fluid flow occurred at lower frequencies than scattering. Akbar et al. (1994) modeled squirt flow between pores and local fluid flow within them and found good agreement to data measured by Paffenholz and Burkhardt (1989). Akbar et al. note that the resulting squirt flow attenuation occurs at lower frequencies than do losses due to local fluid flow. Ongoing measurements by Moerig et al. (1996) in the seismic frequency range begin to address the role of restricted contact line movement and subsequent energy loss. In their measurements the solid surfaces of an artificial crack made of microscope slides separated by thin wires are brought together at varying velocities causing pressure to be built up within the fluid and energy to be stored in the elastic spring mechanism of the sample. The fluid is displaced producing some combination of shear, contact line movement and change in shape of the meniscus depending on the chemical properties of the solid surface (Waite et al., 1997, Moerig et al., 1997). Moerig et al. (1996) made the observation that on surfaces that have been exposed to carbon based molecules, an increase in low frequency attenuation and an increase in stiffness occurs (figure I.1).

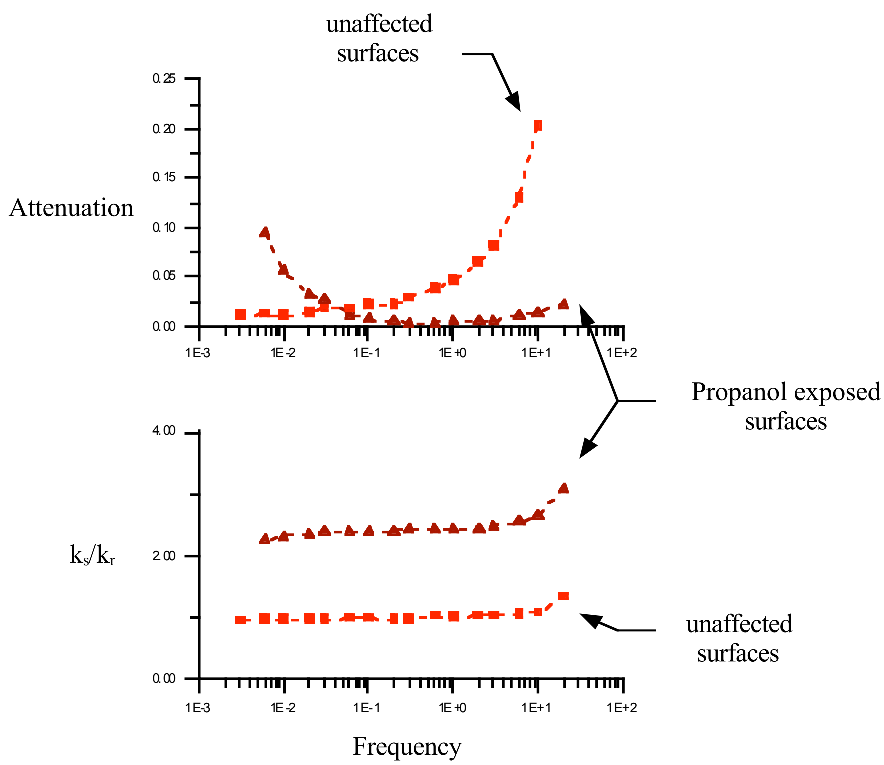


Figure I.1 Attenuation and relative stiffness (k_r = 50 kN/m) versus frequency for artificial cracks partially saturated with water. Squares represent data taken on surfaces that have not been exposed to carbon based molecules. Triangles represent data taken on surfaces exposed to Propanol.

The types of mechanisms addressed in this thesis were originally investigated using real rock samples. The measured attenuation and stiffness of these samples were very difficult to interpret (Chelidze et al., 1996). There was no quantitative means for predicting the measured values. Since that time, efforts have been made to reduce the complexity of the system by studying single cracks. Observations by Moerig et al. (1996) utilized a rectangular crack where one side was fixed, the opposite side allowed to move and deform, and the orthoginal sides allowed to deform. The geometry for this sample was still fairly complex but pointed towards viscous relaxation and a definte dissipation mechanism related to a restricted contact line. With this information, the single crack was simplified further such that the fluid film was nearly cylindrical. This geometry allows for quantitative interpretations and are the focus of this thesis. Future work will begin with measurements on cintered glass bead samples and then real rocks.

Chemistry

A silica surface that is exposed to a humid environment will adsorb water to produce hydroxyls (Iler, 1979). On a flat quartz surface, the surface density of hydroxyls are on average 4.6 OH nm^{-2} (Peri and Hensley, 1968). Amorphous silica has a slightly smaller surface density of hydroxyls. The hydroxyls allow hydrogen bonding to other molecules containing OH groups such as Water, H_2O (Iler, 1979), Methanol, CH_3OH (George, 1994), Acetone, C_3H_5OH, and Propanol, C_3H_7OH (Appendix A). The most likely molecular configurations appear in figure I.2. Others have been noted (Iler 1979).

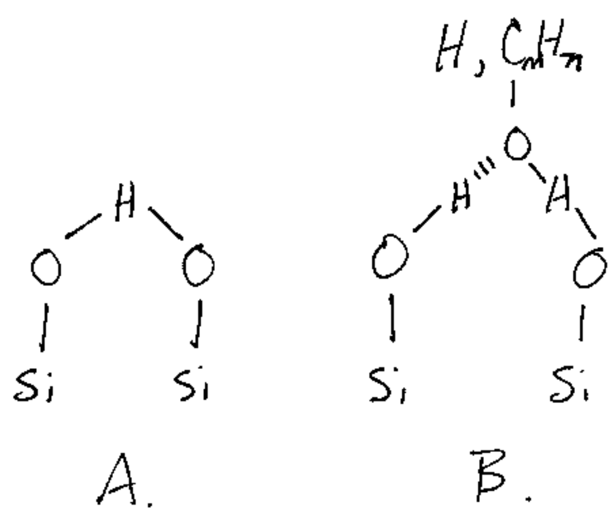


Figure I.2 A. Hydroxylated silica surface. B. Hydrogen bound water and carbon based molecules through the OH group.

Water reacts quite differently to a surface that contains carbon based groups as opposed to a simple hydroxylated surface. A simple hydroxylated surface allows polar interactions between water molecules and surface hydroxyls, a process that is considered hydrophilic. Carbon groups containing hydrogen are nonpolar and have no affinity for water producing a hydrophobic effect (Iler, 1979; Ben-Naim, 1980). In terms of a moving contact line this means that on a hydrophilic surface the water molecules have a relatively high probability of bonding with a surface site and moving some distance in some interval of time. As the surface sites become hydrophobic, the probability for water molecules to bond to the surface decreases resulting in a lower contact line velocity. The water molecules must either find a surface hydroxyl or replace a nonpolar molecule with a smaller probability.

Attenuation Spectrometer

Definition of 1/Q

Attenuation, the inverse of the quality factor, Q, is defined as the fractional energy lost during one cycle of deformation of a material,

$$\frac{1}{Q} = \frac{\Delta W}{2\pi W}. \qquad \text{(I.1)}$$

The work in terms of stress and strain is

$$W = \int_0^{\omega t} \sigma \partial \varepsilon . \qquad \text{(I.2)}$$

The phase with which the strain lags the stress is a measure of attenuation. To show this, let's begin by writing the stress as

$$\sigma = M^* \varepsilon \qquad \text{(I.3)}$$

where M^* is the complex modulus and ε is the strain. The strain can be expressed as

$$\varepsilon = \varepsilon_0 \sin(\omega t + \delta) \qquad \text{(I.4)}$$

and $$\partial \varepsilon = \varepsilon_0 \cos(\omega t + \delta) \partial \omega t \qquad \text{(I.5)}$$

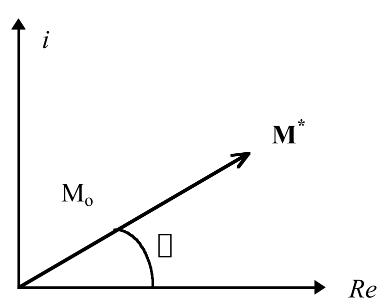


Figure I.3 Complex vector M in the real-imaginary plane. δ is the angle between M* and the real axis and corresponds to the phase lag between stress and strain. The anelastic component of M* is $M_0 \sin\delta$ and the elastic, $M_0 \cos\delta$.*

where ε_0 is the strain amplitude, ω is the circular frequency, and δ is the phase lag between stress and strain and the angle between the real axis and the complex vector M* on the real-imaginary plane (figure I.3). The energy lost is the solution to eq. I.2 where the limits of integration are one full cycle,

$$\Delta W = \oint \sigma \partial \varepsilon = \pi \varepsilon_0^2 M_0 \sin \delta . \qquad \text{(I.6)}$$

The maximum energy stored is the solution to eq. I.2 where the limits of integration are over a quarter cycle,

$$W = \int_0^{\pi/2} \sigma \partial \varepsilon = \frac{\varepsilon_0^2 M_0 \cos \delta}{2} . \qquad \text{(I.7)}$$

The fractional energy lost is then eq. I.6 divided by eq. I.7,

$$\frac{\Delta W}{W} = \frac{2\pi \sin \delta}{\cos \delta} = 2\pi \tan \delta . \qquad \text{(I.8)}$$

For small δ, tanδ is approximately equal to δ. The result is a direct relationship between the phase lag and the attenuation,

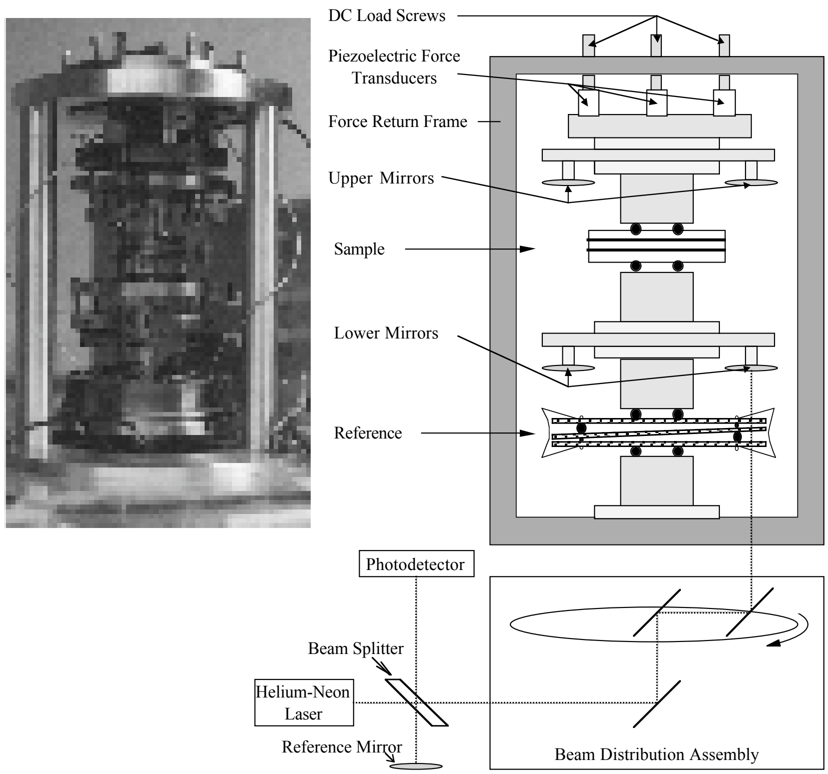
$$\frac{\Delta W}{W} = 2\pi\delta . \qquad \text{(I.9)}$$

Apparatus

Work by Moerig et al. (1996) and continuing measurements made by members of this group utilizing an attenuation spectrometer (Cherry et al., 1996; figure I.4) will be the central focus of the experimental concepts presented in this thesis. An extended discussion of the sample is presented on page 31. The instrument employs optical interferometry techniques to measure the phase and magnitude of the displacement of a sample and elastic standard. With this information, the attenuation and stiffness of the sample are deduced.

Besides the sample/standard column, the apparatus consists of a waveform generator, amplifier, force transducers, He-Ne laser, photodetector, A-D converter, a computer, and other miscellaneous devicesThe waveform generator produces a sinusoidal voltage that varies in frequency between .001 Hz and 100 Hz. The waveform passes through three independently adjustable amplifier channels. From there each amplified signal is sent to a piezoelectric transducer where up to 1 μm of displacement is produced. This displacement compresses the column causing deformation in the sample and standard. A Michelson interferometer detects the movement though changes in the lengths of 6 object beams relative to a reference beam. Three of the object mirrors constitute a plane above the sample and three constitute a plane between the sample and standard. The object and reference beams are interfered at the photodetector which continuously feeds the signal to the A-D converter. The A-D converter stores 16384 data points comprising several cycles of

Instrument



Sample

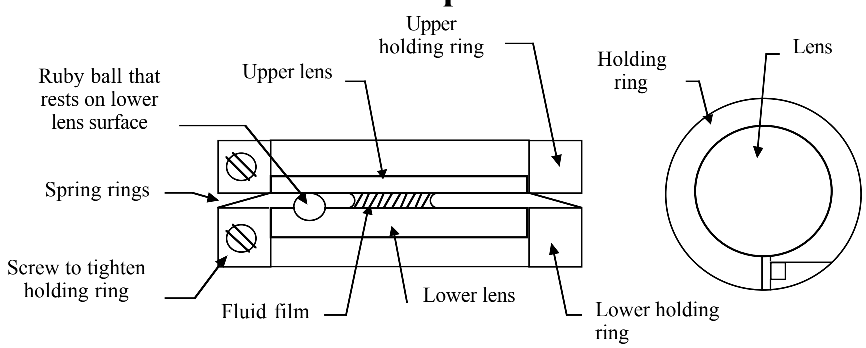


Figure I.4 Photograph and schematic of the attenuation spectrometer and sample. In the photograph, the two mirror plates can be seen along with the cylindrical sample and rectangular standard. The sample/standard column is surrounded by a rigid exoskeleton made of aluminum. Leads from a 500 volt amplifier wind up the outer structure to the top of the column to meet the transducers. The schematic shows the interferometer technique including laser, reference mirror, beam distribution assembly, and photodetector. A basic schematic of the sample includes upper and lower holding rings, upper and lower lenses, screws to tighten holding rings, spring rings, ruby ball, and fluid film.

the seismic waveform and relays it on to the computer at the end of the measurement (Appendix D). After several signals have been acquired, they are reduced to a seven parameter equation (eq. I.10, figure I.5) using a Levenberg-Marquardt nonlinear least-squares algorithm (Appendix E).

$$z(t) = A_3 \sin[A_1 \sin(2\pi ft + A_2) + A_5 + A_6 t + A_7 t^2] + A_4 \quad \text{(I.10)}$$

A_1 is the sinusoidal displacement of the objective mirror. A_2 is the phase of the objective mirror relative to the waveform generator. $A_3 + A_4$ and A_4 represent the

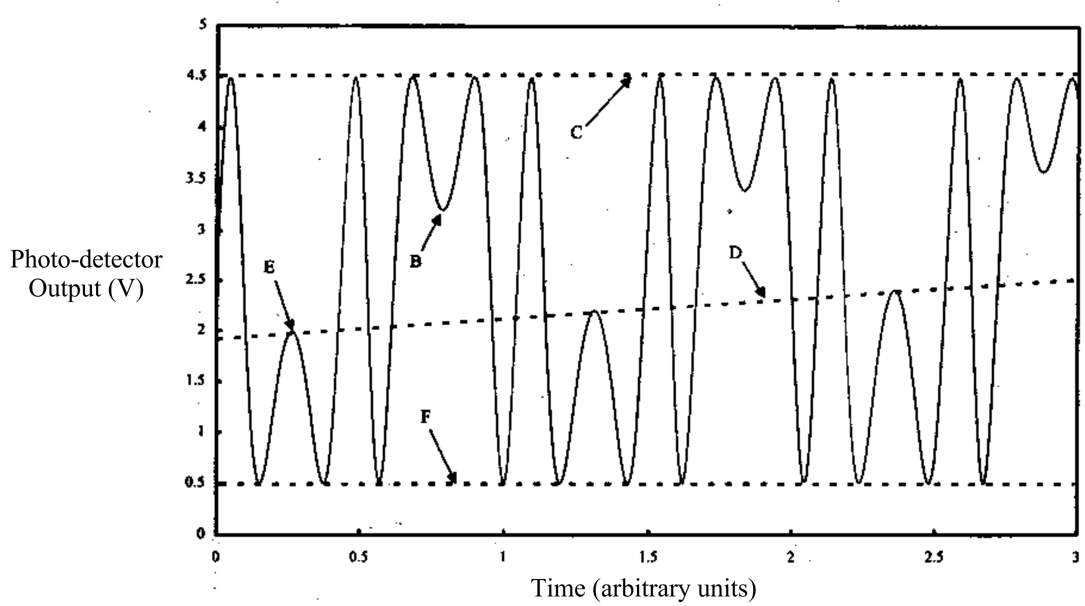


Figure I.5 Figure from Cherry et al., 1996. Interference pattern produced by combining the moving object beam and stationary reference beam. A_1 is the sinusoidal displacement of the objective mirror represented by the number of fringes between locations E and B. A_2 is the phase of the displacement. A_4 is the mean amplitude. $A_3 + A_4$ is the peak amplitude of the photodetector output corresponding to the line labeled C. A_4-A_3 is the minimum output represented by the line labeled F. A_5 is the starting position of the objective mirror relative to the reference mirror measured along the beam path, labeled E. A_6 and A_7 are measurements of the linear and second order drift of the objective mirror, the line labeled D.

maximum and mean amplitude respectively of the photodetector output. A_5 is the starting position of the objective mirror relative to the reference mirror as measured along the beam path, and A_6 and A_7 are the values of the linear and second order drift of the objective mirror.

To deduce the attenuation and stiffness of the sample, the parameters A_2 and A_5 must be known. With these coefficients, the complex displacement of the sample is calculated. The vectorial difference between the upper and lower objective beams gives the displacement and phase of the sample. The phase of the sample minus the phase of the standard is a direct measure of the attenuation in the sample. The displacement of the standard relative to the displacement of the sample is the stiffness ratio between the sample and standard.

Chapter II

THEORETICAL OVERVIEW

The sample which these models describe consists of a cylindrical film of fluid between parallel plates (figure II.1). The plates are separated by an elastic spring of known stiffness. The distance between the plates, referred to as the gap height h, is approximately 100μm and is varied sinusoidally with frequencies ω, from .001 Hz to 100 Hz. The amplitude of oscillation is approximately .02% of the initial gap height and the aspect ratio of the fluid film, h/r, is approximately .01.

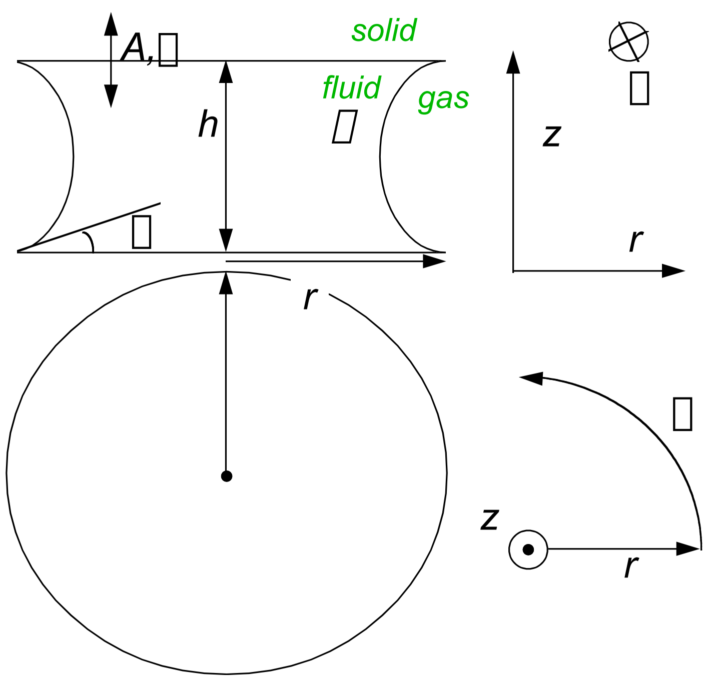


Figure II.1. Schematic illustrating the relation of key parameters in the sample. The fluids has viscosity η and sample has stiffness K. h is the distance between the plates, r is the radius of the fluid film, A is the amplitude of vibration of the upper plate and ω is the frequency of vibration of the upper plate.

Attenuation Due to Viscous Dissipation

Viscosity is a property of materials that describes how quickly a material will strain in response to an applied stress. The energy given to the material through viscous dissipation is unrecoverable and by definition leads to attenuation.

In order to calculate the energy absorbed by viscous dissipation, the strain rate or, equivalently, the velocity field within the material and its viscosity must be known. The velocity field can be described by the Navier-Stokes equations and the viscosity has been empirically measured for many materials.

Energy and Force

One path for the transfer of energy is to impart a force on an object over a distance. Quantitatively this is given by the expression

$$E = \int \hat{F} \cdot d\hat{l} \,. \qquad \text{(II.1)}$$

The total force, F, acting externally on a body is the normal component of stress, $\square \cdot \square$, acting on that body integrated over the body's entire surface area, A. Gausses theorem states that the integral over the surface of a body is related to the integral over its volume through the divergence of the integrals argument. The force field can then be expressed as the total change in stress as a function of space within the volume, V,

$$\hat{F} = \oint \tau \cdot n\, dA = \int \nabla \cdot \tau\, dV . \qquad \text{(II.2)}$$

The stress relates to the strain rate as

$$\tau = \mu \dot{\gamma} \qquad \text{(II.3)}$$

where μ is the materials Newtonian viscosity and $\dot{\gamma}$ is the strain rate . The strain rate is expressed as the gradient of the velocity field, $\nabla \hat{U}$. In terms of the velocity and a constant viscosity, the viscous force becomes

$$\hat{F} = \mu \int \nabla^2 \hat{U}\, dV . \qquad \text{(II.4)}$$

Velocity Field

The Navier-Stokes equation (Gerhart and Gross, 1985), a statement of cause and effect, is the governing principle that allows for the solution of the velocity field within continuous bodies. It simply states that the time rate of change in linear momentum of a body is equal to the sum of forces acting on that body. This relationship is customarily expressed in terms of the density ρ, velocity U, pressure gradients ∇p, and viscous forces $\mu \nabla^2 \hat{U}$,

$$\frac{1}{V}\frac{d\left(m\hat{U}\right)}{dt} = \rho\frac{d\hat{U}}{dt} = -\nabla p + \mu\nabla^2\hat{U}. \qquad \text{(II.5)}$$

The general solution to this equation does not exist due to the equations complexity. In order to extract the velocity field, the system must be known well enough to understand where simplifications can be made.

Some of the terms in the Navier-Stokes equation can be neglected due to their relatively small magnitude. The expression on the left hand side of the Navier-Stokes equation can be broken up into the force per unit volume needed to accelerate the body and a force per unit volume associated with the inertia of the body,

$$\rho\frac{\partial\hat{U}}{\partial t} + \rho\hat{U}\cdot\nabla\hat{U}, \qquad \text{(II.6)}$$

respectively. The velocity of the upper plate is on the order of $2\pi A\omega$ and dt on the order of the period, $1/\omega$. The Spatial derivatives, dr and dz, are on the order of the gap dimensions, r and h. The expressions giving rise to the order of magnitude of the accelerating, inertial, and viscous terms are

$$\rho 2\pi A\omega^2, \rho\frac{4\pi^2 A^2\omega^2}{h}, \mu\frac{2\pi A\omega}{h^2}. \qquad \text{(II.7)}$$

For the values of the parameters used in this model, ρ = 1000 kg/m^3, μ = .001 kg/ms, A = .2 μm, ω = 10 Hz, and h = 100 μm, the size of these equations are .0126, .0016, and .126 N/m^3, respectively. At 10 Hz, an error of less than 1% is incurred in the Navier-Stokes equation by neglecting the inertial effects of the fluid and an error of 10% develops by neglecting the acceleration of the fluid. These errors decrease quickly with decreasing frequency. In terms of the velocity profile, which is the parameter of interest, there may be no error for all frequencies by neglecting equation II.6, the acceleration and inertia of the fluid film. The simplified Navier-Stokes equation is the Stokes equation,

$$\nabla p = \mu \nabla^2 \hat{U}. \tag{II.8}$$

The current form of this force balance is still cumbersome and can be reduced further by evaluating the Laplacian.

The Laplacian of a vector, $\hat{U}$, in cylindrical coordinates is

$$\begin{aligned}
\nabla^2 \hat{U} = & \left[\frac{\partial}{\partial r}\left(\nabla \cdot \hat{U}\right) + \frac{1}{r}\left[\frac{\partial}{\partial z}\left(r\left[\frac{\partial U_r}{\partial z} - \frac{\partial U_z}{\partial r}\right]\right) - \frac{\partial}{\partial \theta}\left[\frac{1}{r}\left(\frac{\partial}{\partial r}(rU_\theta) - \frac{\partial U_r}{\partial \theta}\right)\right]\right]\right]e_r \\
& + \left[\frac{1}{r}\frac{\partial}{\partial \theta}\left(\nabla \cdot \hat{U}\right) + \left[\frac{\partial}{\partial r}\left[\frac{1}{r}\left(\frac{\partial}{\partial r}(rU_\theta) - \frac{\partial U_r}{\partial \theta}\right)\right] - \frac{\partial}{\partial z}\left[\frac{1}{r}\left(\frac{\partial U_z}{\partial \theta} - \frac{\partial}{\partial z}(rU_\theta)\right)\right]\right]\right]e_\theta \\
& + \left[\frac{\partial}{\partial z}\left(\nabla \cdot \hat{U}\right) + \frac{1}{r}\left[\frac{\partial}{\partial \theta}\left[\frac{1}{r}\left(\frac{\partial U_z}{\partial \theta} - \frac{\partial}{\partial z}(rU_\theta)\right)\right] - \frac{\partial}{\partial r}\left(r\left[\frac{\partial U_r}{\partial z} - \frac{\partial U_z}{\partial r}\right]\right)\right]\right]e_z
\end{aligned}$$

Due to the high modulus of water and the lack of confining pressure, the water volume can be considered incompressible, thus $\nabla \cdot \hat{U} = 0$. Because of the symmetry of the fluid volume and of the forces acting on it, angular dependence is absent. This condition requires all terms containing derivatives with respect to the angle θ to be zero. Further simplification arises when the assumption is made that the velocity in the z direction does not depend on the radius, i.e. $\partial U_z / \partial r = 0$. This assumption is made without proof but is believed to be true for parallel plates. The resulting Stokes equations are

$$\frac{\partial p}{\partial r} = \mu \frac{\partial^2 U_r}{\partial z^2} \qquad \text{(II.9)}$$

and

$$\frac{\partial p}{\partial z} = \frac{\mu}{r} \frac{\partial}{\partial r}\left(r \frac{\partial U_r}{\partial z} \right). \qquad \text{(II.10)}$$

After taking the derivative of the first with respect to z and the second with respect to r, the two equations can be set equal to each other.

$$\mu \frac{\partial^3 U_r}{\partial z^3} = \mu \frac{\partial}{\partial r} \frac{1}{r} \frac{\partial}{\partial r}\left(r \frac{\partial U_r}{\partial z} \right) \qquad \text{(II.11)}$$

The assumption of the water volume being incompressible allows U_r to be written in terms of U_z. Starting with the divergence of the velocity vector,

$$\nabla \cdot \hat{U} = \frac{1}{r} \frac{\partial}{\partial r}\left(r U_r \right) + \frac{\partial U_z}{\partial z} = 0, \qquad \text{(II.12)}$$

U_r is found to be

$$U_r = -\frac{r}{2}\frac{\partial U_z}{\partial z}. \qquad \text{(II.13)}$$

Substituting U_r in terms of U_z into equation II.11 remembering that the z component of velocity does not vary in the radial direction, namely $\partial U_z / \partial r = 0$, equation II.11 becomes

$$\frac{\partial^4 U_z}{\partial z^4} = 0 \qquad \text{(II.14)}$$

where the general solution is

$$U_z = a\frac{z^3}{6} + b\frac{z^2}{2} + cz + d. \qquad \text{(II.15)}$$

It is now time to invoke the boundary conditions. At the lower solid boundary where $z = 0$, $U_z = 0$ and $U_r = 0$, a no slip boundary condition. At the upper solid boundary where $z = h$, $U_z = U_h = A\square$ and $U_r = 0$, also a no slip boundary condition. Using equation II.13 and applying these boundary conditions reveals U_r to be

$$U_r = 6\square A\square\frac{r}{h}\left(\frac{z^2}{h^2} - \frac{z}{h}\right). \qquad \text{(II.16)}$$

The velocity profile for various r is shown in figure II.2. At $z = 0$ and $z = h$, the velocity is zero. At $h/2$, the velocity is at a maximum and as r increases, so does the velocity.

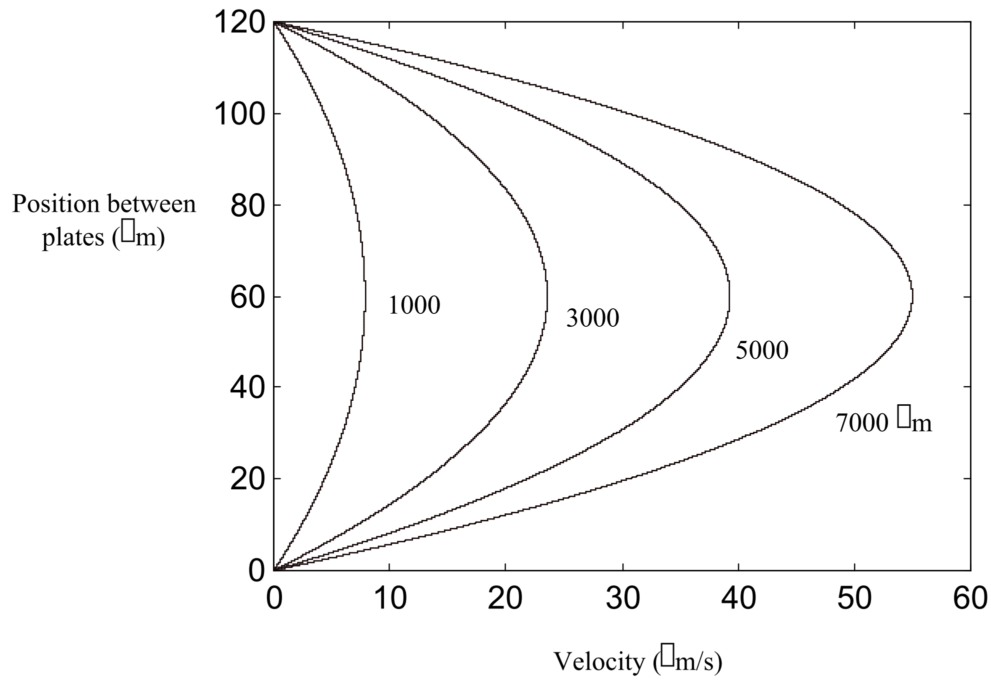


Figure II.2 Velocity field dependence on position between plates for various distances from center of fluid film. Curves are calculated using A = .2 μm, h = 120 μm, □= 1 Hz, and r = 1000, 3000, 5000, 7000 μm.

Dissipation

All of the ingredients for determining the energy lost through viscous dissipation are present. Applying the Laplacian to the radial component of the velocity (eq. II.16), integrating over the volume and multiplying by the viscosity yields the restrictive force acting in the radial direction (eq. II.4). Integrating this force over the radius of the fluid volume gives the energy lost due to viscous dissipation (eq. II.1,II.17)

$$E = 2\square^2 \square A \square \frac{r^4}{h^2} \tag{II.17}$$

The key to this effect lies within two fundamental concepts, viscosity and the Laplacian of the velocity field. As discussed earlier, viscosity is a property of materials that governs the rate at which the material will strain in response to an applied stress. The Laplacian of a vector is a measure of how the individual components of that vector vary with respect to each of the coordinate directions. For the geometry discussed in this paper, the magnitude of the resistive force depends on

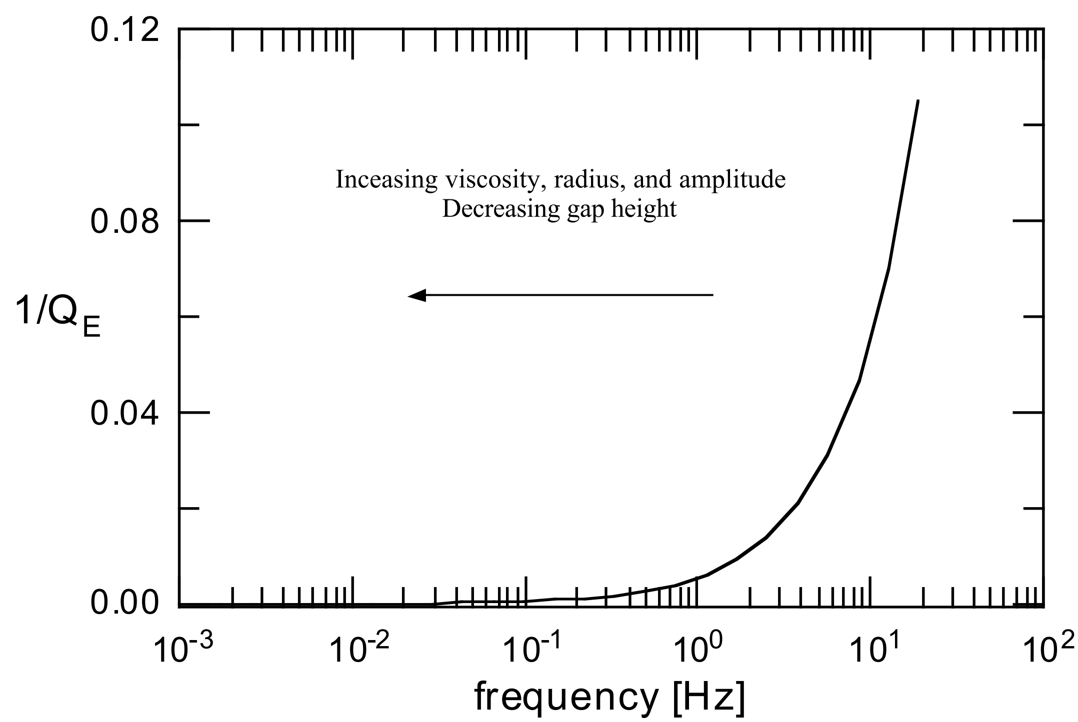


Figure II.3 - Attenuation versus Frequency. Increasing viscosity, radius, and amplitude, or decreased gap height result in increased shear and would move the curve to the left. This causes greater energy lost and higher attenuation for a given frequency. Input parameters are h = 120 μm, r = 6mm, A = .2 μm and μ = .001 kg/ms. The curve does not account for inertial effects or the force needed to accelerate the fluid.

how quickly the velocity in the radial direction changes with respect to z. The dissipated energy is a result of the shearing of the fluid volume. A faster change in velocity produces greater shear and increased attenuation (figure II.3).

Attenuation due to Contact Line Movement

As was inferred in the introduction, the mobility of the contact line can be represented as the probability of water molecules at the contact line to bond to the solid surface. The probability for a new bond to be created depends on the departure from equilibrium of the forces on the contact line. As pressure is built up within the fluid due to a decrease in volume of the pore space, the meniscus changes shape bringing molecules on its surface closer to the solid surface. This is represented by an increase in contact angle (figure II.1). Given enough time, new bonds will form and the contact line will move effortlessly. But if the pressure within the fluid builds too quickly, the molecules at the gas fluid interface will not have enough opportunity to bond to the solid surface and the sample will stiffen. Let this regime of contact line motion be considered a sticking regime and described with a probability of the form

$$p_i = \left(\frac{\theta_i - \theta_e}{\theta_h}\right)^b \qquad \text{(II.18)}$$

where θ_i is the contact angle at the *i*th interval of time, θ_e is the equilibrium contact angle, θ_h is the contact angle hysteresis and b is a constant that controls how quickly the probability changes during sticking. Slipping occurs above θ_h where the contact line motion is governed by other processes.

Energy is lost when some change in fluid pressure balances a related change in fluid radius. To determine the energy lost due to restricted contact line motion, the resistive force on the contact line and the distance over which the contact line moves must be known. Because of the expression for the volume of the fluid film, the force on the contact line is not known analytically as a function of distance. The integral in equation II.1 must be made a sum and the force and distance calculated for each interval of time.

$$E_{lost} \approx \sum_i F_i dr_i \qquad \text{(II.19)}$$

The force on the contact line is due to a change in capillary pressure (Corey, 1994)

$$\Delta P_c = \frac{4\gamma}{h_i}\left(cos\theta_i - cos\theta_e\right) \qquad \text{(II.20)}$$

where γ is the surface tension of the fluid. The force is given by

$$F_i = \pi r_i h_i \Delta P_c = 4\pi r_i \gamma\left(cos\theta_i - cos\theta_e\right) \qquad \text{(II.21)}$$

The calculation of the change in radius is broken up into several stages. At each time step, dt, the velocity of the upper plate, ∂h/∂t, is calculated from the frequency, ν, and amplitude, A, of the passing waveform to determine the new height. For infitesimal dt, h is

$$h_i = h_0 - \frac{\partial h}{\partial t}_i \, dt$$
$$\frac{\partial h}{\partial t}_i = 2\pi\nu A\cos\left(2\pi\nu t_i\right) \qquad \text{(II.22)}$$

The velocity is used to determine the change in separation between the solid surfaces. For an incompressible fluid and stationary contact line, the contact angle must change. The volume for a cylindrical film of fluid with a semicircular meniscus between parallel surfaces is approximately the volume of a cylinder, $\pi r^2 h$, minus the fractional volume of the surrounding annulus (Jeff Boyd, personal communication),

$$V = \pi r_i^2 h_i - \frac{\pi h_i^2 r_i}{2}\left[\frac{\frac{\pi}{2} - \theta_i}{\cos^2\theta_i} - \tan\theta_i\right]. \qquad \text{(II.23)}$$

Because the contact angle can not be solved for analytically, an iterative procedure is used to converge on the correct value. The resulting change in contact angle produces some probability for a bond to be created at the contact line. If the probability is less than 1 and greater than a randomly generated number between 0 and 1, the contact line advances an appropriate distance, dr_m, the distance considered to be the maximum change in fluid radius when sticking. Averaged over the entire contact line, the change in fluid radius is

$$dr_i = p_i dr_m \qquad (\theta < \theta_h) \qquad \text{(II.24)}$$

If the probability is greater than 1, the contact line motion is determined by alternate processes. The behavior has been studied by Moerig et. al. (1996). They found that the velocity of the contact line has an approximately linear dependence on contact angle. The fluid radius may then be given by

$$dr_i = m\frac{dr_m}{\theta_h}\left[\theta_i - \left(\theta_e + \theta_h\right)\right] + dr_m \qquad (\theta > \theta_h) \quad \text{(II.25)}$$

where m scales the slope of the line, dr_m/θ_h. The resulting velocity versus contact angle relationship is seen in figure II.4. The lost energy (eq II.19) is solved using either equation II.25 or II.24 and II.21.

The time steps progress until a minimum in separation exists between the solid surfaces, 1/4 cycle. After the last time step, the change in stiffness of the sample and energy stored in the fluid film are calculated. The added stiffness is simply the change in capillary pressure (eq. II.20) multiplied by the area of the fluid-solid interface, πr^2, divided by the amplitude of motion of the upper surface,

$$\partial k = \frac{\pi r_i^2 \Delta P_c}{A} = \frac{4\pi r_i^2 \gamma}{A h_i}\left(\cos\theta_i - \cos\theta_e\right). \qquad \text{(II.25)}$$

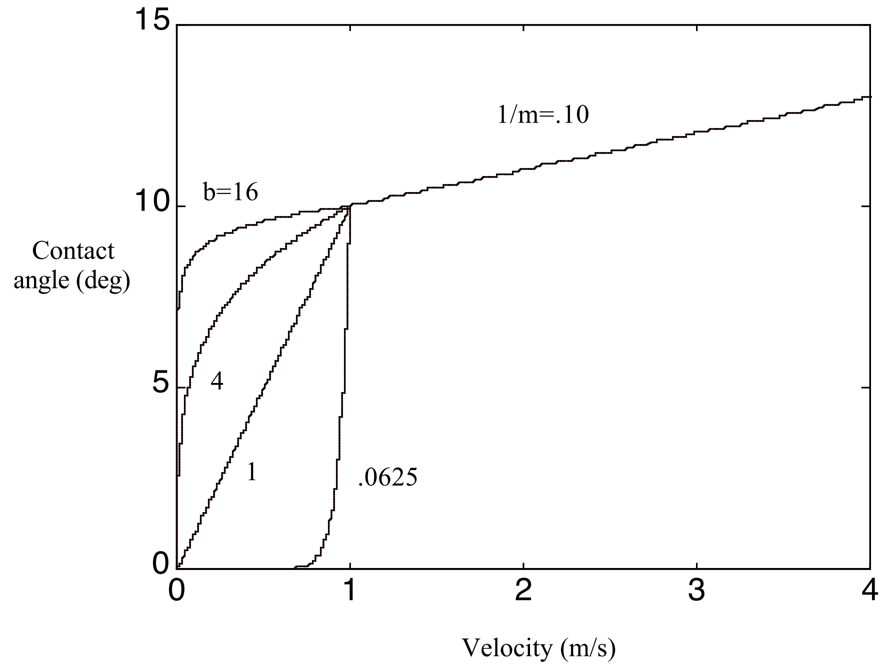


Figure II.4 Contact angle versus velocity; $\partial r_m/dt = 1$ and $\theta_h = 10°$. Values used to calculate fits are given in table V.1.

The energy stored is the energy that would be given back by the system if all external stresses were relaxed. The energy is approximately equal to the change in capillary pressure times the area of the fluid solid interface times the distance over which the capillary force acts *A/2*, plus the energy stored in the spring rings of the sample,

$$E_{stor} \approx A \pi r_i^2 \Delta P_c + \frac{1}{2} k A^2 . \qquad \text{(II.26)}$$

By having multiple regimes of contact line movement, multiple shapes in attenuation and stiffness result. In figure II.5, all parameters are held fixed while the constant b is varied. Recall that b is the power to which the ratio θ/θ_h is raised. As b increases, the peak in attenuation and dispersion in stiffness below 1 Hz. move to lower frequencies. If the slope of the contact line velocity versus contact angle relationship decreases then the peak in attenuation and dispersion in stiffness shift to lower frequencies. With increasing equilibrium contact angle the magnitude of both

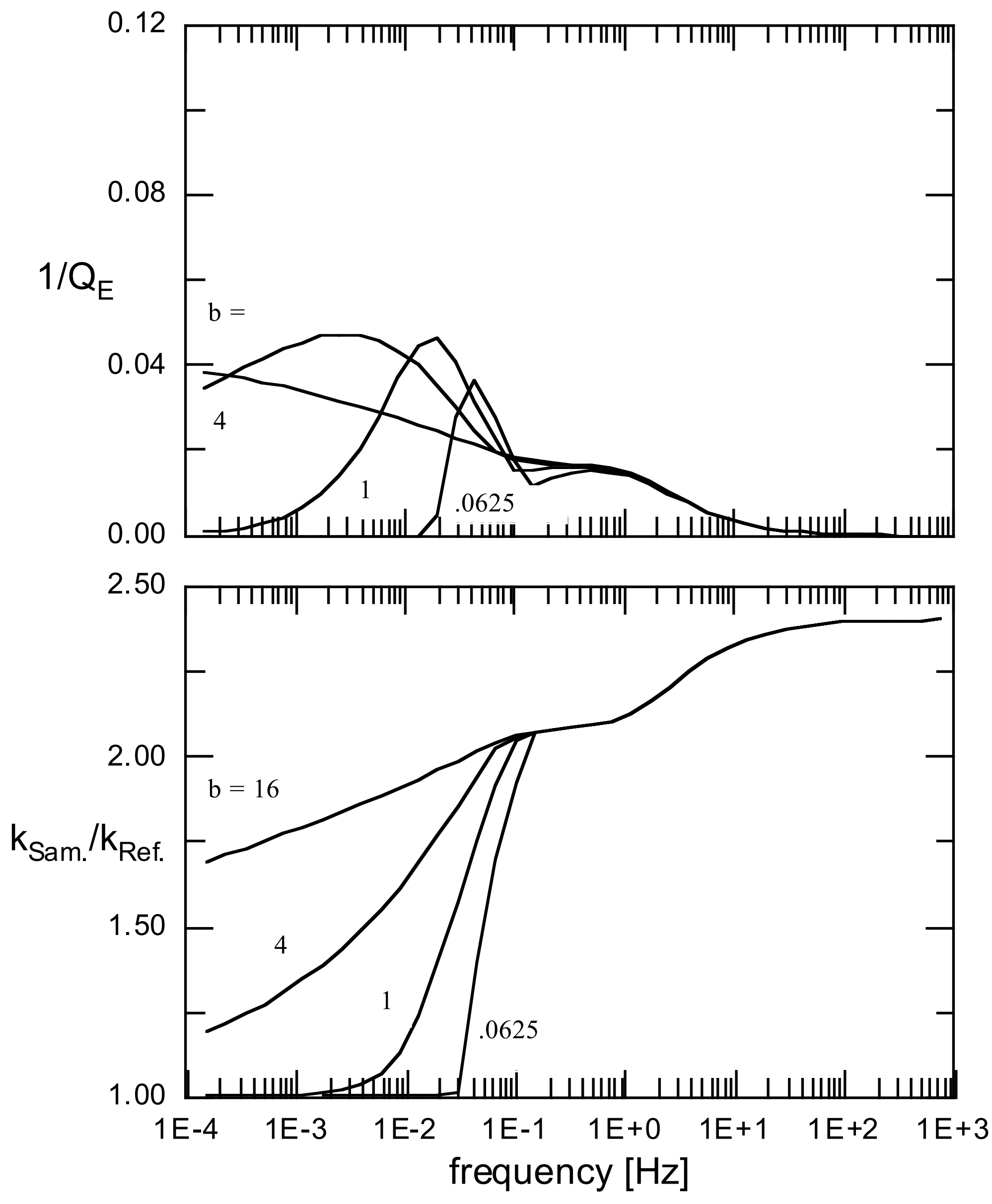


Figure II.5 Attenuation and stiffness versus frequency in dependence on the constant b. The constant b controls the behavior of sticking during low velocities or low frequencies. As b increases from .0625 to 16, the attenuation peak and stiffness dispersion move to lower frequency. This is due to the fact that the slope of the contact line velocity versus contact angle relationship is decreasing. Values for b used in the modeling of attenuation and stiffness versus frequency span this range.

the attenuation and stiffness increase. This is due to the fact that to displace the same volume of fluid during deformation, a higher equilibrium contact angle needs a greater contact angle change. With increasing hysteresis, the attenuation due to slipping decreases. This is simply because more fluid can be taken up in deformation of the meniscus rather than moving the contact line. This also means that the stiffness increases due to the increased contact angle. Above .1 Hz the attenuation and stiffness are independent of b. This means that at any given frequency or contact line velocity, the measured attenuation and stiffness do not depend on the shape of the contact line velocity versus contact angle relationship below that point.

Chapter III

EXPERIMENTAL PROCEDURE

Sample

The sample consists of nearly parallel glass surfaces separated by a thin cylindrical film of fluid (figure III.1) and elastic spring.

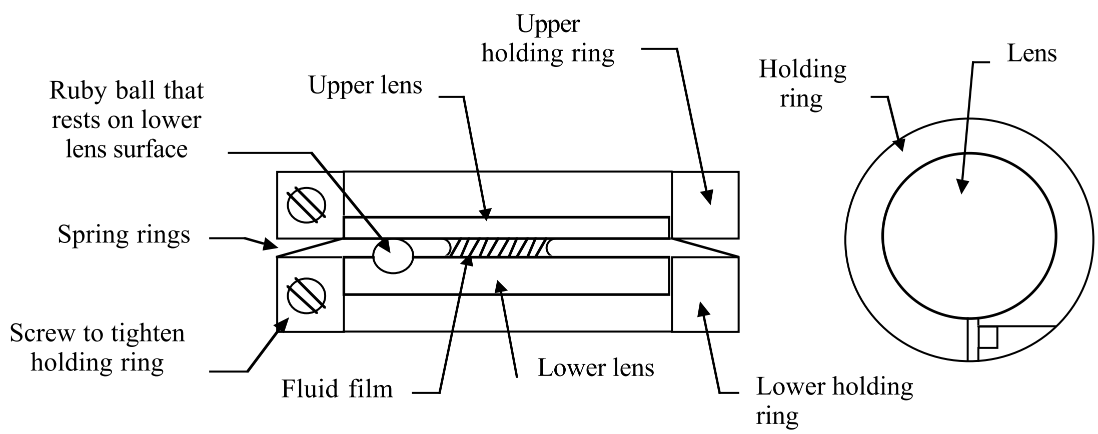


Figure III.1 Basic schematic of the sample design. Included are upper and lower holding rings, upper and lower lenses, screws to tighten holding rings, spring rings, ruby ball, and fluid film.

A small ruby ball rests on the lower surface acting both as a spacer between the solid surfaces and as an indicator for the gap height, a crucial parameter in the models considered for this sample. The glass surfaces are held rigidly by holding rings. These rings are separated by an elastic spring mechanism consisting of donut like metal disks. Although elastic, the mode of deformation of the metal disks is such that a stress dependent stiffness emerges (Appendix B). This produces problems in reproducing the same dry stiffness between measurements.

Preparation

The basic procedure for preparing the sample is the following: 1. The glass lenses are cleaned. 2. The lenses are oriented in the holding rings. 3. The ruby ball is placed on the bottom lens. 4. The sample is partially fluid saturated and placed in a plastic bag to reduce evaporation.

To clean the lenses, they are placed in an oven at 420 degrees C for two hours. This cleaning procedure produces a condition of the surfaces which is relatively reproducible and referred to as unaltered. The condition is reflected in a relatively mobile wetting front. The sample surfaces may then be contaminated. Before any data are taken, all of the instruments are turned on and allowed to warm up. This reduces the chance of fluctuations in the movement of the upper mirror plate and wavelength of the laser light. The block upon which the sample and column sit is checked to see that the only differential forces acting on it are due to the piezoelectric transducers.

During sample assembly, the critical task is orienting the upper lens such that during all measurements, the sample is in the same prestressed state for a given surface separation. This will insure the same dry sample stiffness between measurements. The upper surface of the lower lens is made flush with the top of the bottom holding ring. On this the spring rings are placed. The top holding ring with its lens loosely inside is then added. The upper lens slides down and comes in contact with the lower lens. A Plexiglas container holding up to 2 liters of water is placed on the upper holding ring. In the center of the container is a dry cylinder 8 cm in diameter with a 3 cm diameter hole at its center through which the upper surface of the upper lens is exposed. With the desired mass of water in the container, ~2 kg, the exposed surface of the upper lens is tapped by a small Allen wrench. Newton rings are observed at the contact between the two inner glass surfaces. When the Newton rings are centered, the upper holding ring is tightened, thereby orienting the upper lens. The actual point of contact is up to 1 Newton of reduced force. The sample may then be saturated and placed in a plastic bag.

When the instruments are ready, the sample is incorporated into the column. The reference, lower mirror plate and upper mirror plate are aligned with a mechanism fixed to the side of the column. The sample is aligned manually. Each component of the column is positioned such that its mass is distributed symmetrically about a single axis, $\hat{a}$, parallel to gravity. The upper mirror plate is allowed to rest freely on the sample and reference producing an initial prestress. Further stress is added by the transducer screws. The column is compressed until the upper surface within the sample comes into contact with the ruby ball on the lower surface. For partially saturated measurements, the fluid film must be centered on the $\check{a}$ axis in order to determine the separation between upper and lower surfaces. The transducer screws are then backed off slightly so that there is no contact between the ruby ball and upper lens.

All six mirrors are aligned and the gains of the amplifier channels adjusted until the amplitudes of the upper mirrors are between 2.8 and 3.0 fringes of laser light, approximately 1 μm. One fringe corresponds to one wavelength of laser light, which for a He-Ne laser is ~ 600 nm. Three fringes is 1800 nm. The displacement of the upper mirror plate is 1/2 the change in distance of the sample beam, ~1 μm. Two Styrofoam boxes are then placed around the column to reduce air flow.

Dry Sample

Before reliable measurements can be made on a partially saturated sample, the dry system must be understood. The two elements of the dry sample are the spring rings and the plastic bag. First, the sample is constructed such that it lacks only fluid. This is done to insure that the basic sample design will not effect the results obtained later. Frequency independent and low magnitude attenuation along with a constant stiffness should be observed. According to the process described above, measurements are made on this sample multiple times with slight variations of the sample orientation. Initially nothing changes, only time. Then the column is disassembled and reassembled and the measurements are repeated.

The next step involves the removal of the upper holding ring, the process involved when saturating the sample. As before, the sample is placed in a plastic bag and measurements are taken. The sample is removed from the plastic bag and the upper holding ring set aside. The sample is reassembled, placed back in a plastic bag, and measurements are again performed. Like the basic sample design, the removal and reassembly of the sample should not give rise to attenuation or changing stiffness that would interfere with the principle investigation.

Partially Saturated Unaltered Sample

The sample surfaces are cleaned (see preparation p. 32) before the sample is partially saturated with water. At this point, the saturant is seen to easily flow along the sample surfaces. Because of the unaltered condition of the sample surfaces, great care must be taken when placing the sample within the column so that the fluid does not flow into the spring rings. The sample and column are assembled as described above. A measurement is taken and repeated until all of the fluid has evaporated. A dry measurement follows to determine the dry sample stiffness. Before and after the measurements, the saturation state of the sample is observed to give an estimate of the fluid radius. This process is repeated for several clean samples.

Partially Saturated Surface Affected Sample

After a good understanding of the unaltered surface measurements has been obtained, the sample surfaces are contaminated with Propanol. The contamination is incurred by wetting the sample surfaces and allowing the contaminant to fully evaporate. This is done twice to insure the conversion of the surface from hydrophilic to hydrophobic. The sample and column are assembled and measurements are taken and repeated until the fluid has evaporated. This process is repeated once more for comparison.

Chapter IV

RESULTS

The following results were obtained using the sample discussed in the previous chapter. All samples were measured between .002 and 20 Hz. For those results that do not encompass this frequency range, the data has been discarded because the phase and amplitude values could not be extrapolated or the A/D converter stopped functioning during the measurement. The amplitude for all measurements can be found by solving the equation

$$A = \frac{x_A}{\frac{2k_2}{k_1} + 1} \qquad \text{(IV.1)}$$

where x_A is the peak to peak displacement at the upper mirror plate (~ .9 um), k_1 is the sample stiffness, and k_2 is the reference stiffness (~ 60 kN/m).

Dry

The dry system was measured on three different samples. For each sample, a new bag was used and measurements were made multiple times. Initially the sample was left untouched and the measurements were repeated. Without taking the sample out of the plastic bag, the sample column was disassembled and reassembled. All measurements show roughly the same behavior (figure IV.1). The attenuation averages around zero, has a standard deviation of ~.004 rad, and increases slightly with increasing frequency. The stiffness has a virtually constant value up to 10 Hz. Above this frequency, the data exhibit a repeatable rise.

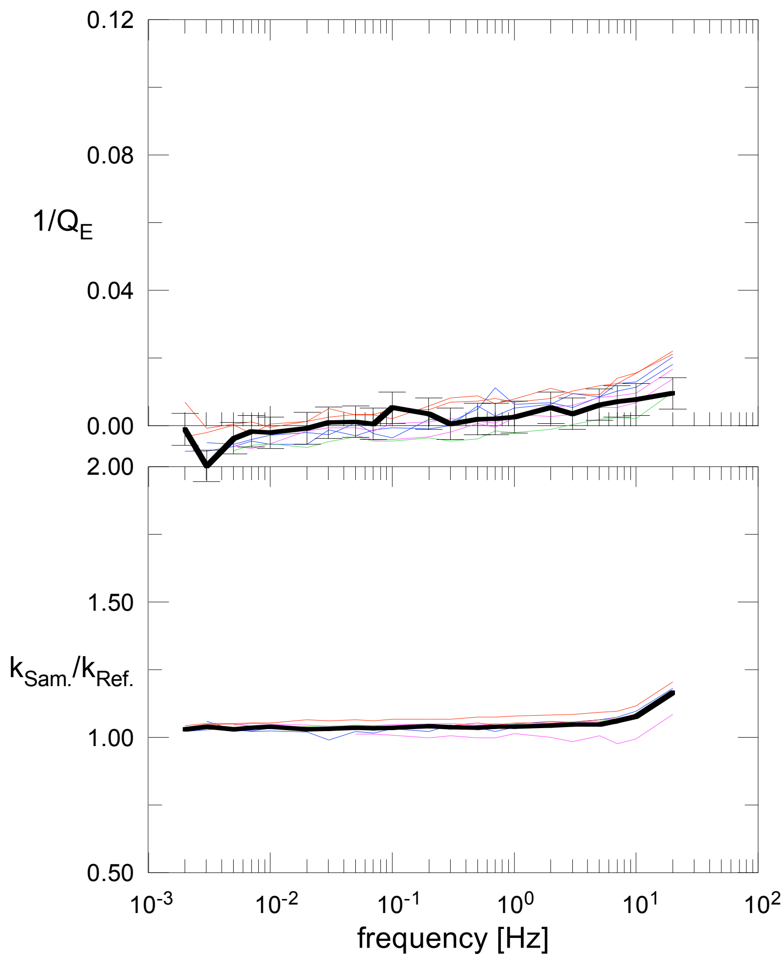


Figure IV.1 Measured data on dry cylindrical samples. The upper graph shows attenuation vs. frequency. The error bars on the thick curve are an averaged standard deviation over three of the measurements. The lower graph contains measurements of the sample/reference stiffness ratio.

Saturated

For all saturated cases presented, the saturant was water. Water has a viscosity of .001 kg/m s and a surface tension of .073 N/m at 20 °C. The fluid volume injected was $\sim 2\times10^{-8}$ m^3. The fluid radius ranged from ~4 to 6 mm depending on the amount of evaporation between measurement and injection. The amplitude of deformation is given by equation IV.1. The following values are unknown: equilibrium contact angle, contact angle hysteresis, velocity of the contact line at the stick-slip transition, dr_m/dt, slope of the contact line velocity versus contact angle above θ_h, and dry sample stiffness The range of the contact line-contact angle parameters is currently being measured. Unfortunately the samples dry stiffness was not obtained due to a precipitate that had formed after evaporation of the fluid (figure IV.2).

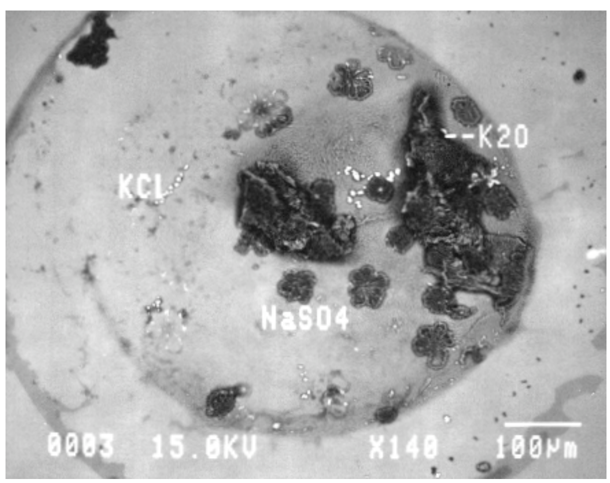


Figure IV.2 Precipitates that had formed at the end of all fluid saturated measurements. The small white speckles are potassium chloride. The flower like disk structures are sodium sulfate, and the larger structures are potassium oxide. This last precipitate was responsible for preventing the acquisition of a dry sample stiffness. There was enough of the compound to span the gap separation.. Photo and analysis courtesy of John Drexler, University of Colorado at Boulder.

Clean

During injection of the fluid, the contact line moves smoothly. There is little resistance. A slight tilt of the sample is enough to cause the fluid to flow quickly. As stated before, great care must be taken to keep the fluid from flowing into the spring rings. Looking down the axis of axial symmetry of the sample, the fluid film is nearly a perfect circle. This was the basic theme behind all three sets of partially saturated unaltered samples.

The first sample shows measurable attenuation in the entire frequency range, ~.04 rad (figure IV.3). Generally, the attenuation increases slightly with increasing frequency. Above 1 Hz, the attenuation seems to increase more rapidly. At the mid to low frequencies the attenuation shows little frequency dependence. The corresponding stiffness exhibits a gradual increase from 1.25 k_{ref} at .002 Hz to 1.5 k_{ref} at 10 Hz.

The second sample looks much like the first (figure IV.4). Because more of the high frequency data was measured, a rapid increase in attenuation with increasing frequency is very apparent. The sample displays similar frequency independent attenuation between .01 and 1 Hz. What has not been seen before is the sudden decrease in attenuation below .01 Hz.

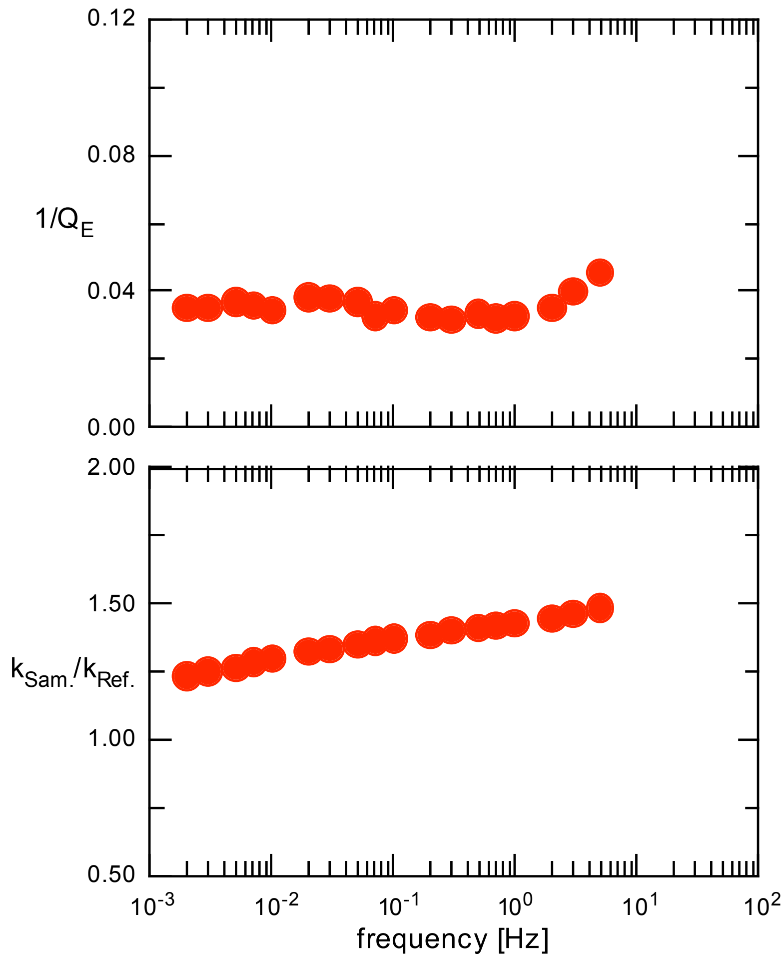


Figure IV.3 First clean sample; attenuation and stiffness versus frequency. The attenuation and stiffness show a slight increase with increasing frequency. A more rapid increase occurs in the attenuation above 1 Hz. Each solid circle represents a data point. For the attenuation curve, the vertical size of the circle is on the order of the standard deviation as derived from the dry experiments.

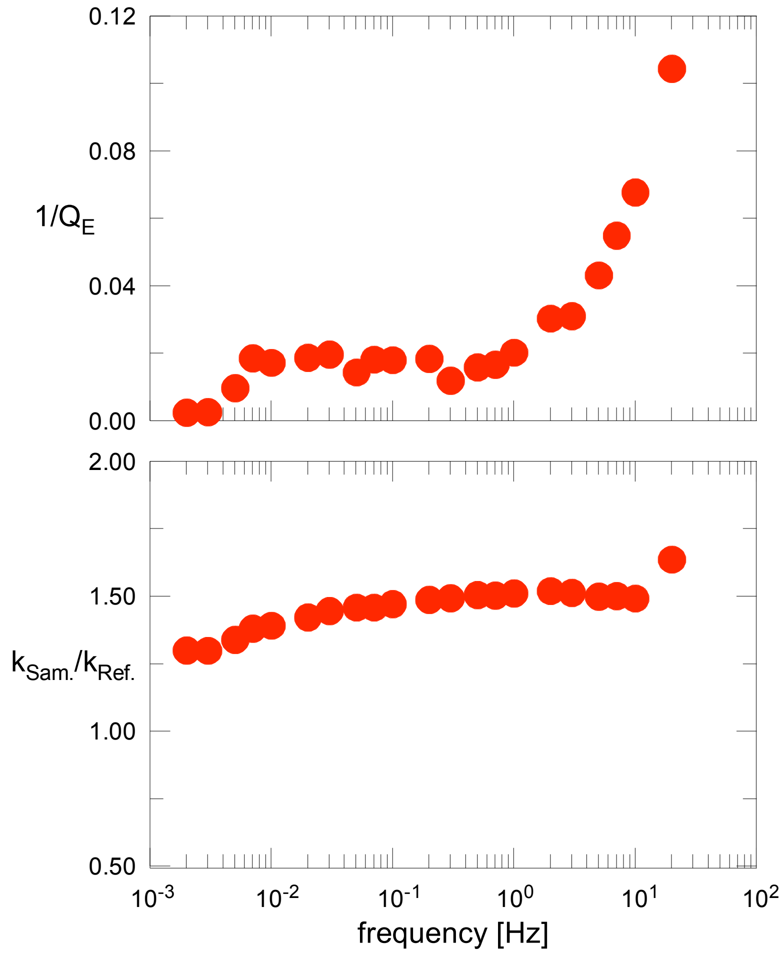


Figure IV.4 Second clean sample; attenuation and stiffness versus frequency. Attenuation and stiffness continue to increase with increasing frequency. As compared to the previous sample, a more dramatic rise in attenuation occurs above 1 Hz. Below .01 Hz, a sudden decease is noted.

The third sample was not as fortunate as the first two due to a failure of the A/D converter to finish much of the first measurement. A second measurement was completed but at a lower saturation (figure IV.5). The fluid radius was approximately 4 mm with all other parameters as before. The resulting measurement closely resembles the data taken on the second sample. The major difference is lower attenuation and stiffness at all frequencies.

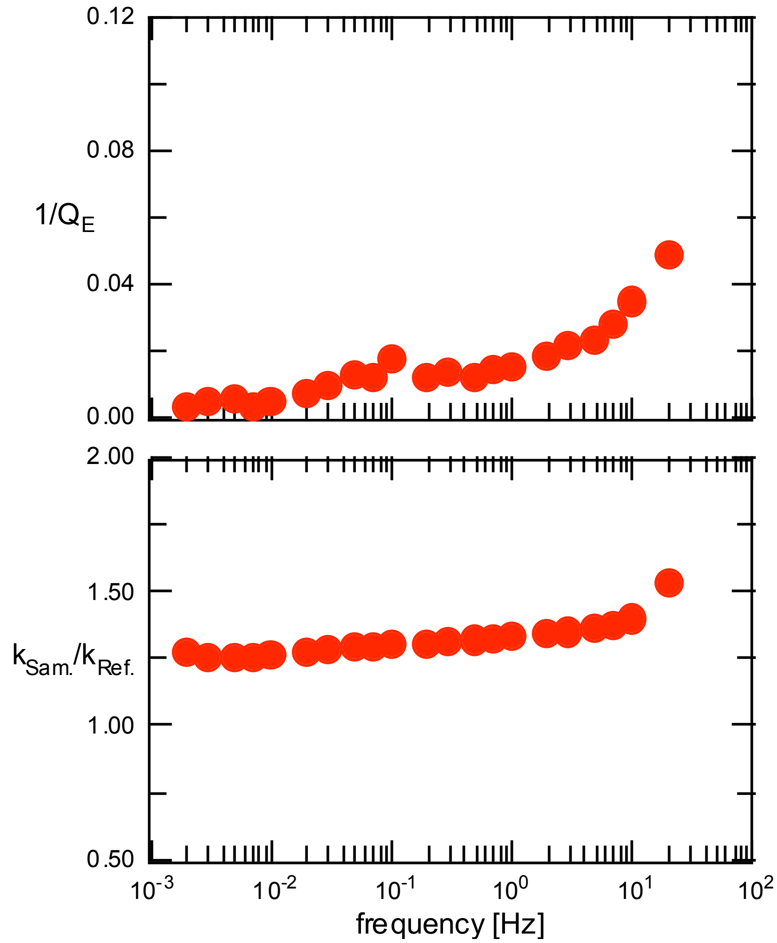


Figure IV.5 Third clean sample; attenuation and stiffness versus frequency for a fluid radius of 4 mm. The frequency dependence resembles that of the second sample with the exception of a nearly constant decrease at all frequencies.

Contaminated

After exposing the sample surfaces to either Propanol or acetone, new sets of attenuation and stiffness measurements were made. The surface alteration produces an observable effect on fluid flow. When saturating the sample, the contact line moves irregularly. A slight tilt of the sample now has visually no observable effect.

Looking down the axis of axial symmetry of the sample, the fluid film takes on a very irregular, oblong shape. The sample is stressed multiple times to force the fluid into a more circular form and more closely match the model constraints, i.e. cylindrical symmetry.

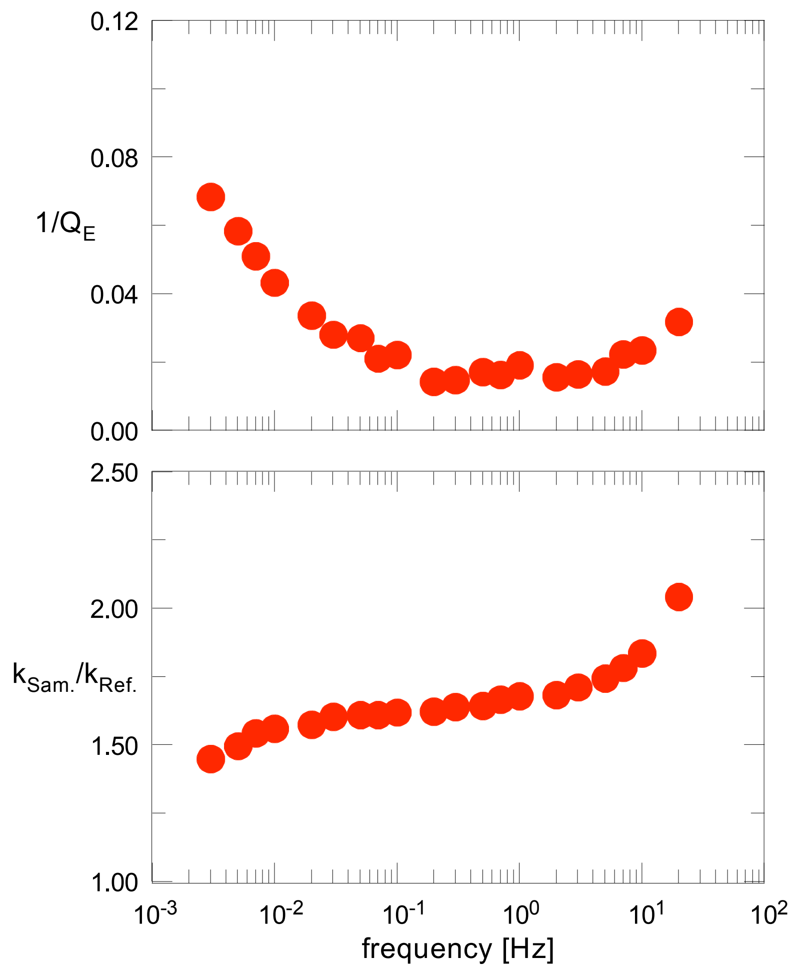


Figure IV.6 First Propanol contaminated sample; attenuation and stiffness versus frequency. The sample shows a very different frequency dependence in comparison to the clean measurements. Note the change in stiffness axis from .5 to 2.0 in the clean figures to 1 to 2.5 in the contaminated figures.

Two contaminated measurements were made, both on surfaces exposed to Propanol (figure IV.6 and figure IV.7). The Propanol measurements look very similar to each other but remarkably different from the clean samples. In addition to the high frequency attenuation, they both show a large increase in attenuation at low frequencies. Accompanying this new attenuation behavior is an increased stiffness.

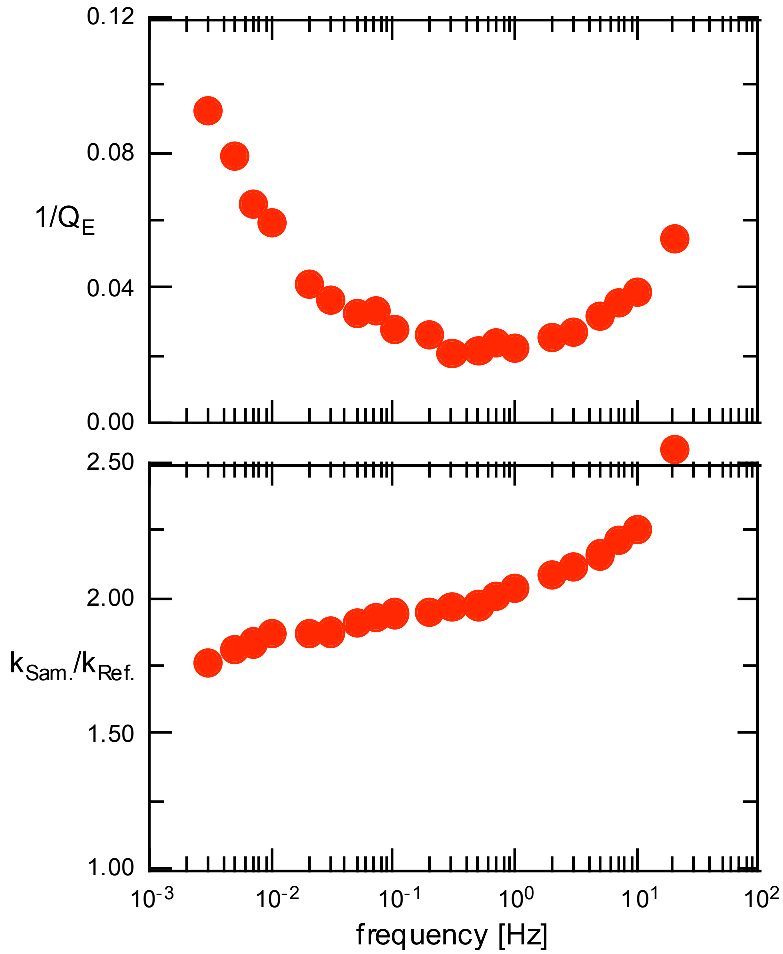


Figure IV.7 Second Propanol contaminated sample; attenuation and stiffness versus frequency. The sample looks almost identical to the other Propanol contaminated sample. The only difference is a slight increase in stiffness and attenuation over the entire frequency range.

Chapter V

INTERPRETATION

In the modeling of the attenuation and stiffness for the three clean samples and first Propanol contaminated sample, there are several characteristics that need to be kept in mind. All samples show some amount of the following trends except where noted: 1. Attenuation increases with increasing frequency above 1 Hz. 2. Attenuation is nearly frequency independent between .01 and 1 Hz. 2. For the clean samples, the attenuation decreases at the low frequencies. 3. For the contaminated sample, the attenuation increases below .1 Hz. 4. Stiffness increases with increasing frequency throughout the entire frequency range. 5. The contaminated sample is markedly stiffer than clean samples.

All calculations assume a gap height of 123 μm, displacement amplitude of .19 μm, and dry sample stiffness 1.05 times that of the reference. A water volume of .02 cm^3 was injected into the samples. For a cylindrical fluid film, this volume and gap height, a radius of 6 mm results but due to the curvature of the upper lens, a radius of 5 mm is used. The true value is unknown. A radius of 4 mm is applied to the third clean measurement because approximately 30% of its volume had evaporated. The values for the equilibrium contact angle, contact angle hysteresis, velocity of the contact line at the contact angle hysteresis, and constants b and m that control the shape of the contact angle-contact line velocity relationship are all chosen to provide the best fit to the data. The ranges for the equilibrium contact angle and contact angle hysteresis are currently being measured. The ranges for the velocity of the contact line at the contact angle hysteresis and constants b and m are unknown. Table V.1 lists the values used for these five free parameters.

Table V.1 Free parameters used in the modeling of contact line movement, eqs. II.18, II.24, & II.25.

	θ_e(deg)	**θ_h(deg)**	**v_m(m/s)**	**b**	**m**
first clean	28.0	5.0	.03	8	194
second clean	19.5	4.0	.00007	.5	12467
third clean	24.0	3.5	.0007	.0625	582
first prop.	35.0	6.0	.0005	4	209

Attenuation is described using both viscous and contact line effects. Stiffness is described only by contact line effects. There is a contribution to stiffness from viscosity, but it is not considered. Moerig et al. (1996) showed that the contribution to stiffness from viscosity is negligible.

Viscous Dissipation

All measurements exhibit increasing attenuation with increasing frequency. This phenomena can be easily explained by viscous dissipation. However, the fits in figure V.1 clearly demonstrate that this mechanism for attenuation is inadequate to explain the data below 10 Hz. The measured attenuation is always higher than that predicted by viscous effects. For the first contaminated measurement, a serious discrepancy is observed. Not only is the measured attenuation much higher below .01 Hz, it is lower above 10 Hz, indicating an additional source of stored energy.

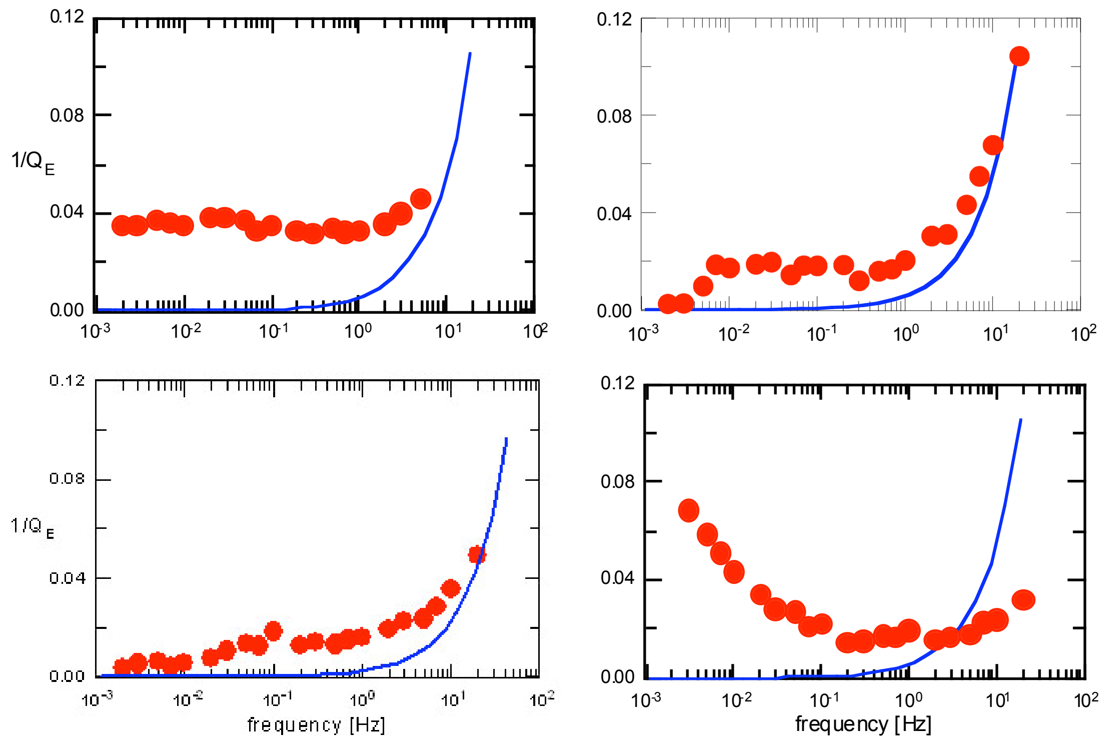


Figure V.1 Attenuation versus frequency; measured data and calculated viscous dissipation. From left to right, top to bottom, first second and third clean measurements, and first contaminated measurement. The clean measurements are modeled well by viscous dissipation at the higher frequencies. The contaminated measurement falls below theoretical predictions indicating either a lack of lost energy or a source of stored energy.

Restricted Contact Line Motion

Below 10 Hz, a definite trend in attenuation is observed on all samples. The clean measurements have a plateau down to ~.01 Hz and then tend to decrease. The contaminated measurements have a similar plateau but begin to rise below .1 Hz. The stiffness increases with increasing frequency and is much higher for the contaminated samples.

The attenuation measurements for all samples can be modeled fairly well by restricted contact line motion between ~.01 and 1 Hz (figures V.2 V.3 V.4 V.5). The modeled attenuation diverges more strongly for the contaminated sample at very low frequencies. The stiffness is much more sensitive. Above ~.1 Hz, the second and third clean measurements have stiffness' that coincide with the model. Below this frequency, the modeled stiffness decreases quickly to the samples dry stiffness. The measured data continues with little change. The first clean sample has a measured stiffness with a slope closely resembling the model. The magnitude, however, is less than calculated. The first contaminated sample is even more anomalous. Measured values are far too low above .01 Hz.

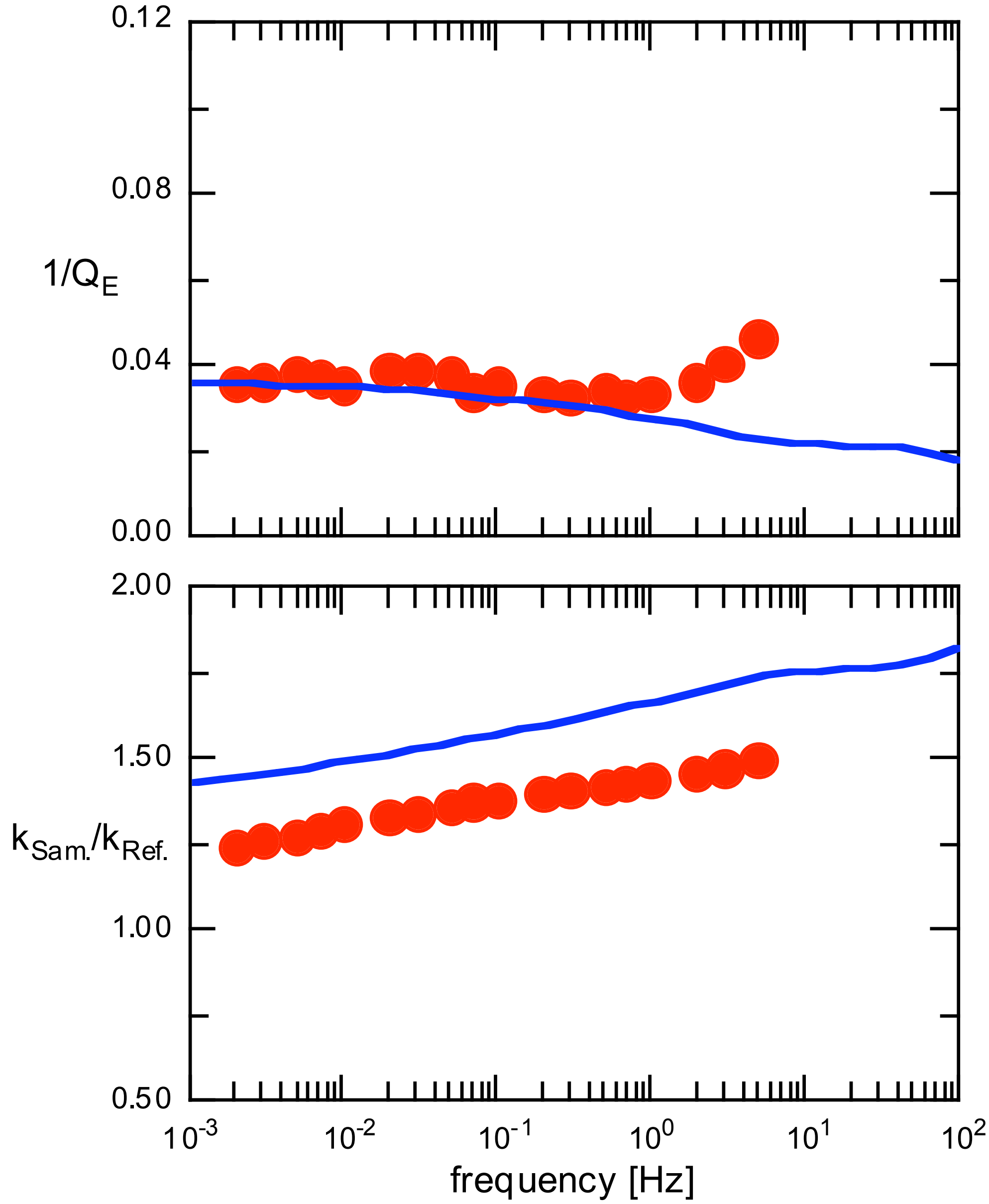


Figure V.2 First clean sample; attenuation and stiffness versus frequency. Solid line indicates modeling from restricted contact line motion only. Note the discrepancy in the stiffness between model and experiment. Model and experimental attenuation diverge above 1 Hz.

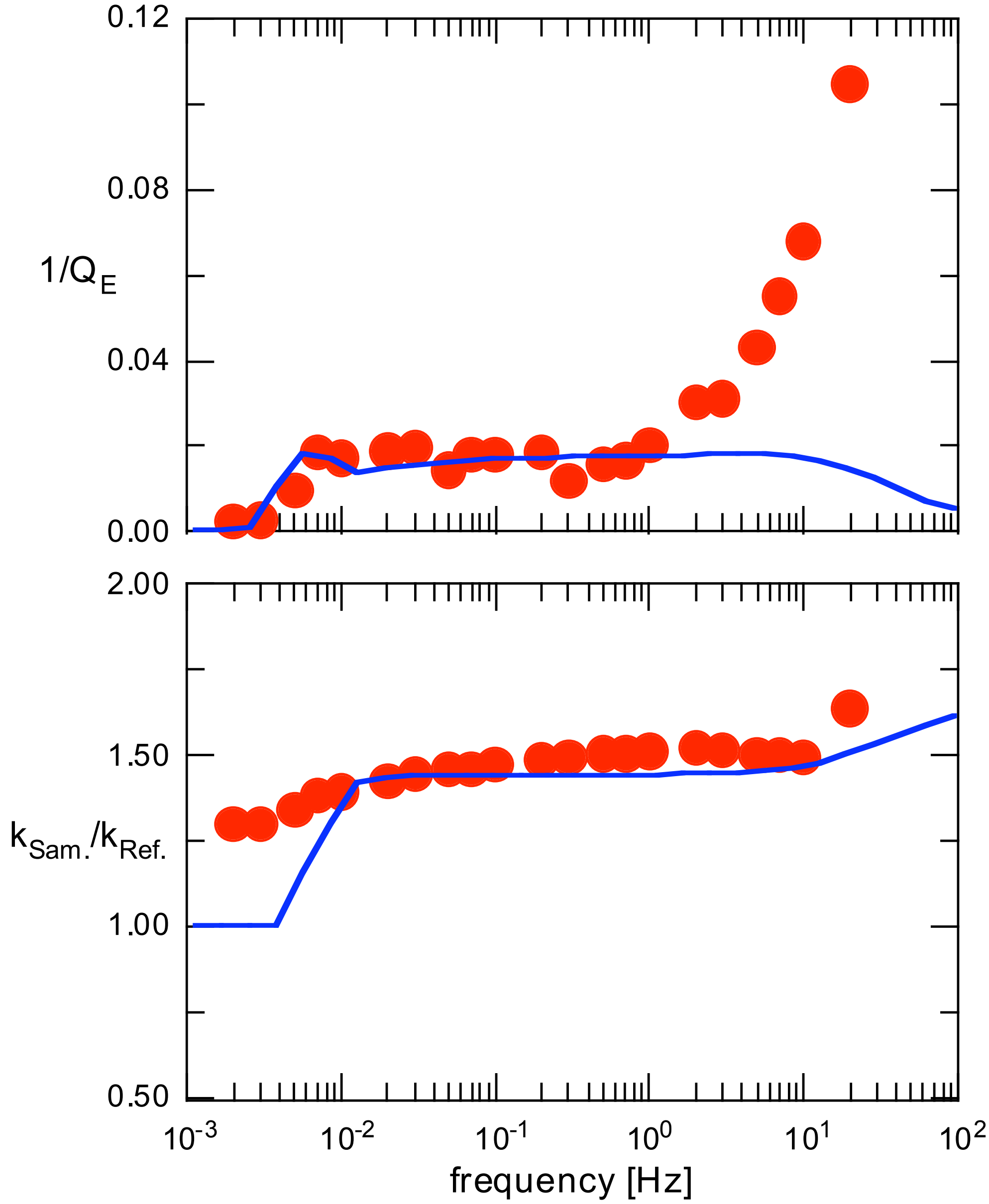


Figure V.3 Second clean sample; attenuation and stiffness versus frequency. Solid line indicates modeling from restricted contact line motion only. Model and experimental attenuation diverge above 1 Hz. Model and experimental stiffness diverge below .01 Hz.

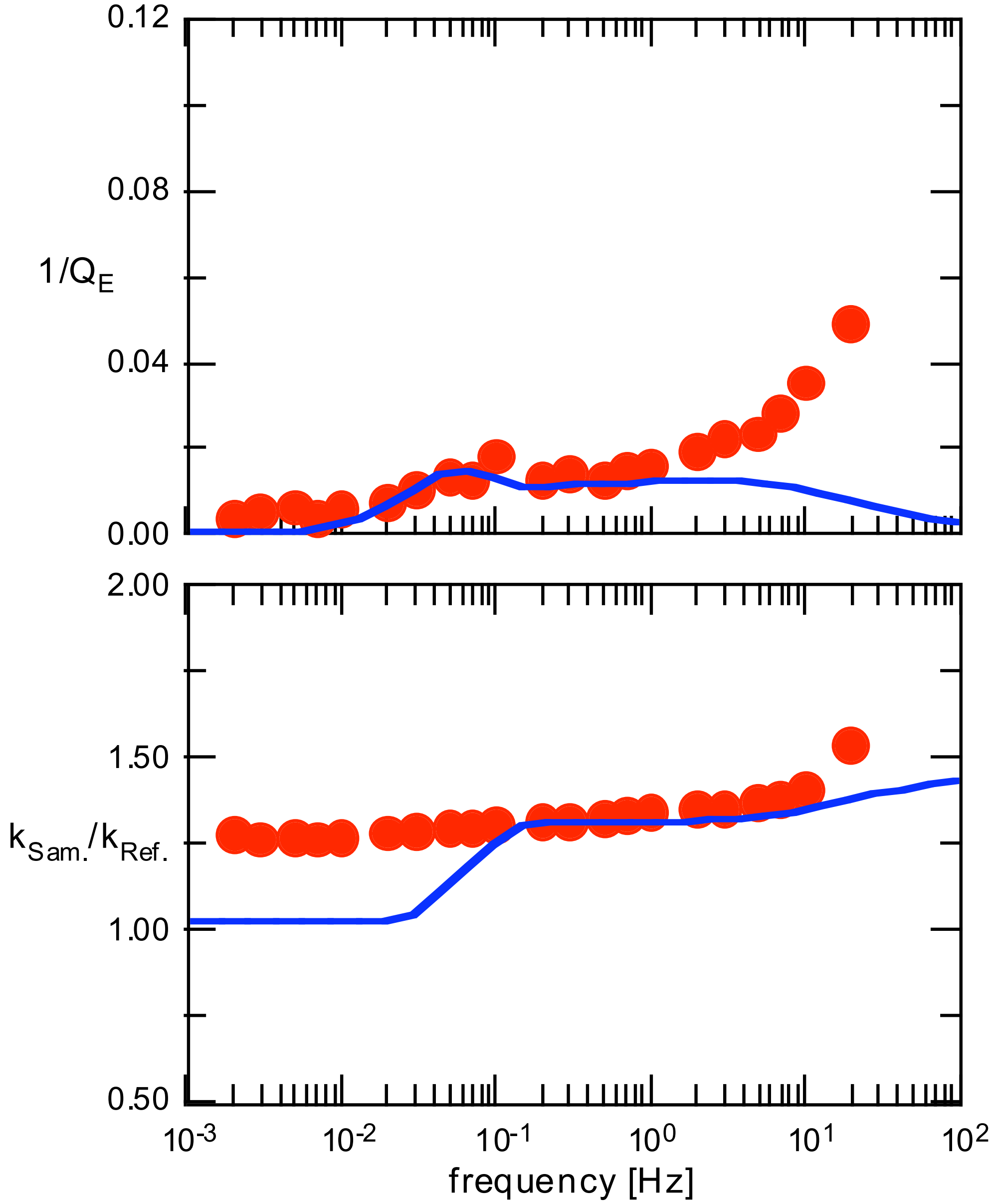


Figure V.4 Third clean sample; attenuation and stiffness versus frequency. Solid line indicates modeling from restricted contact line motion only. Model and experimental attenuation diverge above 1 Hz. Model and experimental stiffness diverge below .1 Hz.

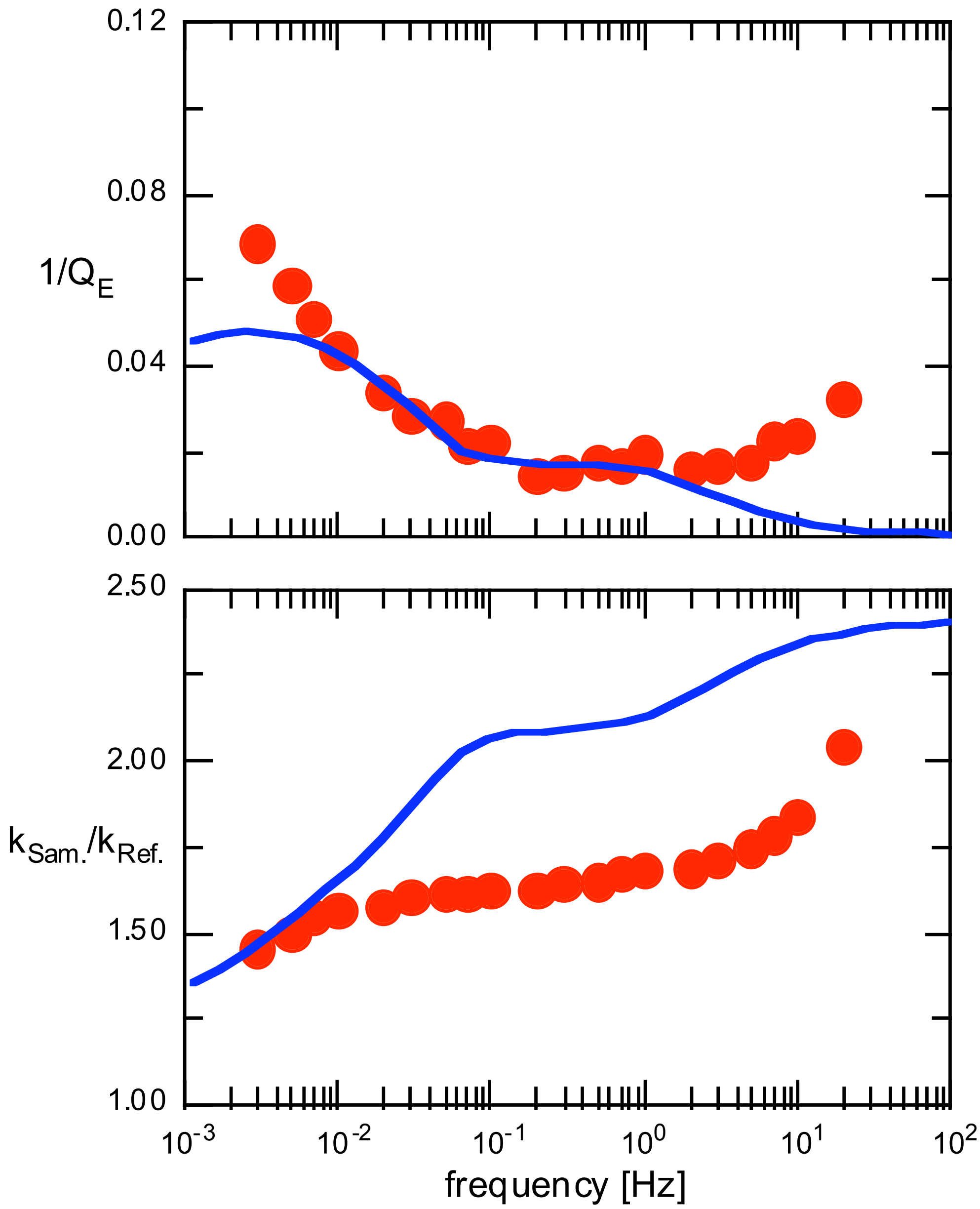


Figure V.5 First Propanol contaminated sample; attenuation and stiffness versus frequency. Solid line indicates modeling from restricted contact line motion only. Note the large discrepancy between model and experimental stiffness. Model and experimental attenuation diverge above 1 Hz and below .01 Hz.

Viscous Dissipation and Restricted Contact Line Motion

The combination of the two mechanisms, viscous dissipation and restricted contact line motion, accounts for nearly all of the attenuation (figure V.6). The restricted contact line model breaks down for contaminated samples below .01 Hz. The contact line resistance in the contaminated sample has provided a source for stored energy and reduced the attenuation at high frequencies.

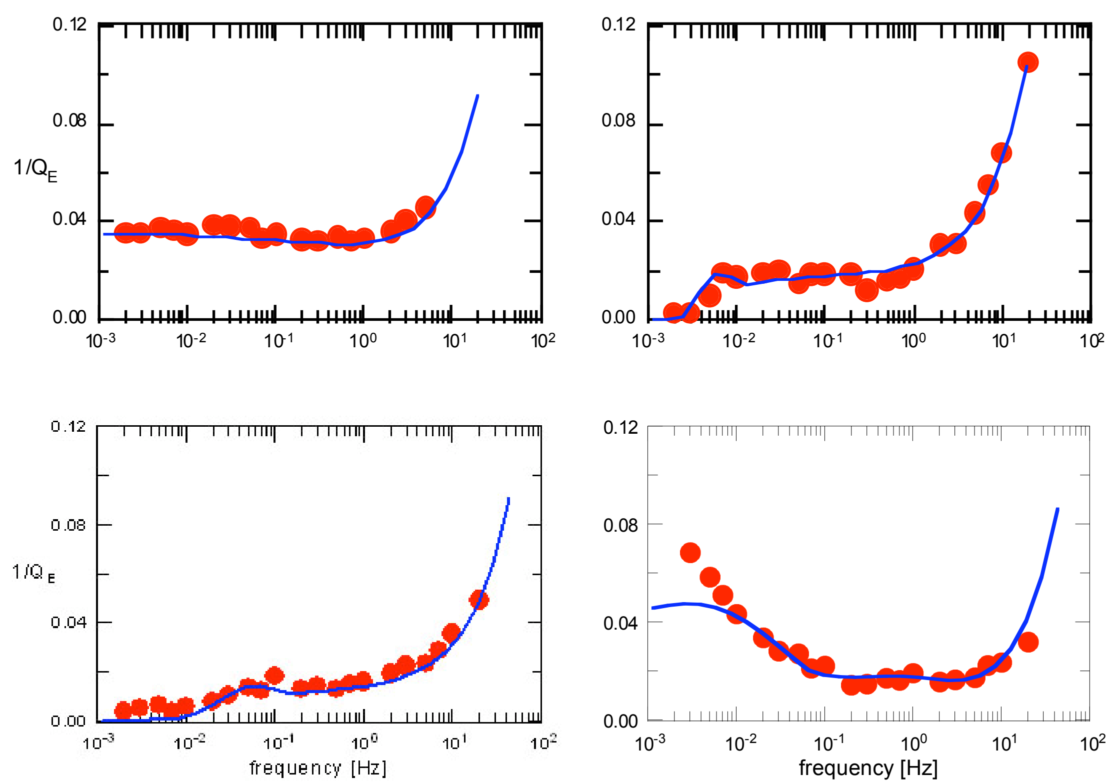


Figure V.6 Attenuation versus frequency; measured data and calculated viscous and contact line dissipation. The combination of the two mechanisms account for nearly all of the experimentally measured attenuation. The source of stored energy in the contaminated measurement needed to reduce the measured attenuation noted earlier can be attributed to contact line resistance.

Chapter VI

DISCUSSION

The description of contact line movement coupled with viscous dissipation as being a source for attenuation shows quantitative agreement with the measured data. The differences between model and experiment in both attenuation and stiffness can probably be attributed to a slightly different shape of the contact line velocity versus contact angle relationship both while sticking and slipping.

The first step would be to make the transition between sticking and slipping continuous. For the sticking regime, a probability that considers the activation energy and other realistic parameters may give the needed relation between contact line velocity and contact angle. It may be possible to derive the contact line velocity versus contact angle relationship empirically from the measured data. Another point to keep in mind is that the dry sample attenuation increased slightly with increasing frequency. This attenuation was not accounted for in the calculations of viscous and contact line related dissipation.

In terms of the values used to model restricted contact line motion, preliminary work suggest that the contact angles are in the proper range. Measurements were made on the slope of the contact line velocity versus contact angle but to date the results are unreliable. Further work should be done to determine the contact line velocity versus contact angle relationship, especially in the sticking regime.

Further work will include measurements on cintered glass bead samples and eventually real rocks. With these new measurements, a higher sample stiffness will increase stored energy and may obscure contact line dissipation.

BIBLIOGRAPHY

Akbar, N., G. Mavko, A. Nur, and J. Dvorkin, Seismic signatures of transport properties and pore fluid distribution, Geophysics, 59, 1222-1236, 1994

Biot, M. A., Theory of propagation of seismic waves in a fluid saturated porous solid, I. Low-frequency range, J. Acoust. Soc. Am., 28, 168-178, 1956a.

Biot, M. A., Theory of propagation of seismic waves in a fluid saturated porous solid, II. Higher frequency range, J. Acoust. Soc. Am., 28, 179-191, 1956b.

Biot, M. A., Mechanics of deformation and acoustic propagation in porous media, J. Appl. Phys., 33, 1482-1498, 1962.

Born, W. T., and J. E. Owen, Effect of moisture upon velocity of elastic waves in Amherst sandstone, Am. Ass. of Pet. Geol., 19, 9-18, 1935.

Bulau, J. R., B. R. Tittman, and M. Abdel-Gawad, Modulus and attenuation in sandstone with hydrocarbon and aqueous pore fluids, 54th Ann. Internat. Mtg. Soc. Expl. Geophys., Expanded Abstracts, 353-356, 1984.

Burridge, R., and H-W. Chang, Multimode, one dimensional wave propegation in a highly discontinuous medium, Wave Motion, 11, 231-249, 1989.

Chelidze, T. L., H. A. Spetzler, and G. A. Sobolev, Absorption of strain waves in porous media at seismic frequencies, Pageoph, 147, 25-55, 1996.

Cherry, R. H., H. A. Spetzler, and J. Paffenholz, A New Wideband [1 mHz to 100 Hz] Seismic Spectrometer, Rev. Sci. Instrum., 67[1], 215-221, 1996.

Corey, A. T., Mechanics of Immiscible Fluids in Porous Media, Water Resources, p. 17, 1994.

Dvorkin, J., G. Mavko, and A. Nur, Squirt flow in fully saturated rocks, 63th Ann. Internat. Mtg. Soc. Expl. Geophys., Expanded Abstracts, 805-808, 1993.

George, S., Thermal Stability of Hydroxyl Groups on a Well-Defined Silica Surface, J of Phys. Chem., 99, 13, 4639-4647, 1994.

Gerhart, P. M., and R. J. Gross, Fundamentals of Fluid Mechanics, Addison-Wesley, p. 215, 1985.

Gurevich, B., V. B. Zyrianov, and S. L. Lopatnikov, Seismic attenuation in finely layered porous rocks: Effects of fluid flow and scattering, Geophysics, 62, 319-324, 1997.

Jones, T, and A. Nur, Velocity and attenuation in sandstone at elevated temperatures and pressures, Geophys. Res. Let., 10, 140-143, 1983.

Iler, R. K., The Chemistry of Silica: Solubility, Polymerization, Colloid and Surface Properties, and Biochemistry, John Wiley and Sons, 1979.

Mavko, G., and D. Jizba, Estimating grain-scale fluid effects on velocity dispersion in rocks, Geophysics, 56, 1940-1949, 1991.

Miksis, M. J., Effects of contact line movement on the dissipation of waves in partially saturated rocks, J. Geophys. Res., 93, 6624-6634, 1988.

Moerig, R., W. F. Waite, O. S. Boyd, I. C. Getting, and H. A. Spetzler, Seismic Attenuation in Artificial Glass Cracks: Physical and Physicochemical Effects of Fluids, Geophys. Res. Lett., 23, 2057-2060, 1996.

Moerig, R., W. F. Waite, and H. A. Spetzler, Effects of Surface Contamination on Fluid Flow, Geophys. Res. Lett., 24, 755-758, 1997.

Murphy, W. F., K. W. Winkler, and R. L. Kleinberg, Acoustic relaxation in sedimentary rocks: Dependence on grain contacts and fluid saturation, Geophysics, 51, 757-766, 1986.

Ben-Naim, A., Hydrophobic Interactions, Plenum Press, 1980.

Norris, A. N., Low-frequency dispersion and attenuation in partially saturated rocks, J. Acoust. Soc. Am., 94, 359-370, 1993.

O'Connell, R., and B. Budiansky, Viscoelastic properties of fluid saturated, cracked solids, J. Geophys. Res., 82, 5719-5735, 1977.

Paffenholz, J., and H. Burkhardt, Absorption and modulus measurements in the seismic frequency and strain range on partially saturated sedimentary rocks, J. Geophys. Res., 94, 9493-9507, 1989.

Palmer, I. D., M. L. Traviola, Attenuation by squirt flow in undersaturated gas sands, Geophysics, 45, 1780-1792, 1980.

Shapiro, S. A., H. Zien, and P. Hubral, A generalized O'Doherty-Anstey formula for waves in finely-layered media, Geophysics, 59, 1750-1762, 1994.

Peri, J. B., and A. L. Hensley, The surface structure of silica gel, J. Phys. Chem., 72, 2926-2933, 1968.

Waite, W. F., R. Moerig, and H. A. Spetzler, Seismic Attenuation in a Partially Saturated, Artificial Crack Due to Restricted Contact Line Motion, In press Geophys. Res. Lett..

Winkler, K. W., Frequency dependent ultrasonic properties of high porosity sandstones, J. Goephys., Res., 88, 9493-9499, 1983.

Appendix A

MEASUREMENT OF THE ACTIVATION ENERGY FOR THE BONDING OF PROPANOL TO A SODA-LIME GLASS SURFACE

Introduction

The wetting of artificial crack surfaces with water is severely inhibited when the surfaces have been contaminated by 1-propanol, methanol, or acetone. This is observed as a highly irregular solid-liquid-gas contact line and a greatly reduced capillary pressure.

Restoration of precontaminated wetting behavior has been achieved through the application of sustained heat. It is proposed that the restoration of Propanol contaminated soda-lime slides is a thermally activated process.

Method

Soda-lime glass slide surfaces are cleaned by heating the slides in an oven at 420 C for two hours. They are then cooled, fully wetted with 1-propanol, allowed to dry, and placed in the oven at a specific temperature. After a known period of time, two slides are removed and allowed to come to room temperature. The slides are separated by wires (figure 1) and the capillary rise of water is measured.

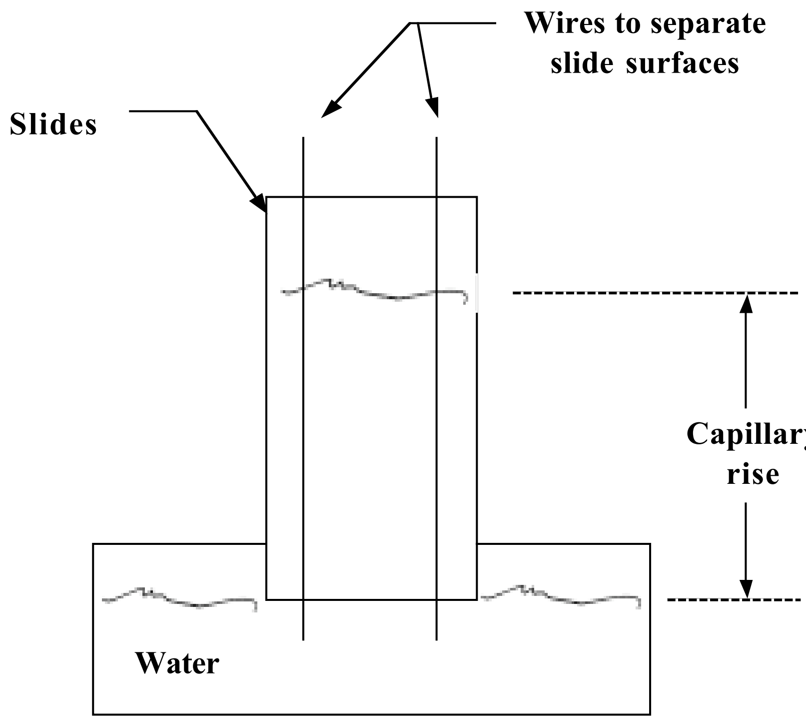


Figure A.1 Measurement of the capillary rise between parallel plates. Water is dawn into the sample. The capillary rise is dependent on the condition of the solid surfaces.

The capillary rise is an indication of the surface contamination. For this experiment, the assumption is made that the greater the capillary rise of water, the fewer the number of affected sites that remain. A site on the surface is one where a bond may be created between the site and an externally imposed molecule (i.e. water).

Assume that the number of affected sites , N, that remain on the surface after exposure to a constant temperature for some period of time is directly proportional to the initial number of affected sites,

$$N = N_0 e^{-Kt}, \qquad \text{(A.1)}$$

where N_0 is the original number of affected sites and K is a rate constant. If this process of reversion is thermally activated, then

$$K = K_0 e^{-\frac{E_A}{k_B T}}, \tag{A.2}$$

where K_0 is the rate constant at infinite temperature, E_A is the activation energy for this process, T is the temperature and k_B is Boltzman's constant.

K_0 can not be measured directly, but remains constant for the same thermally activated process. By taking the ratio between a reaction rate at one time and temperature and a reaction rate at a different time and temperature, K_0 will cancel. If we choose a constant ratio between remaining affected sites and initial affected sites, N/N_0, this ratio will also cancel. After taking the logarithm of both sides and some simple algebra, we are left with

$$\ln\left(\frac{t_1}{t_2}\right) = \frac{E_A}{k_B}\left(\frac{1}{T_1} - \frac{1}{T_2}\right). \tag{A.3}$$

This is a straight line for ln(t) vs. 1/T with slope E_A/k_B, characteristic of a thermally activated process.

In order to make use of equation A.3, we must measure the temperatures and find the times for a constant ratio between remaining affected sites and initial affected sites. A constant ratio, N/N_0, is assumed equivalent to a constant capillary rise. It is important to chose a capillary rise which is substantially less than that for clean surfaces and substantially more than for fully contaminated surfaces.

It is highly unlikely that several sets of slides removed from the oven at different temperatures would provide the same capillary rise. The time for a constant capillary rise must be extrapolated from multiple sets of slides exposed to the same temperature for differing amounts of time. An Arhennious equation (eq A.4) is fitted to the measured data to produce a time for the constant capillary rise.

$$\ln\left[\frac{1}{b}\left(1-\frac{h}{h_0}\right)\right] = -\tau t, \qquad \text{(A.4)}$$

t is the time, τ is a characteristic time provided by the fit, h is the capillary rise, h_0 is the capillary rise at infinite time, and b is a constant to insure that the capillary rise has a particular value at zero time. This technique is repeated for multiple temperatures to provide adequate resolution in the ln(t) vs. 1/T space for the purpose of an estimation of the activation energy.

Results

The collected data are fitted using the values 6 cm for h_0, and .75 for b. The data and associated fits are seen in figure A.2. In figure A.3, the plot of t versus 1/T has a slope of 12520 K where the times are found using a value of 4 cm for the constant capillary rise.

Conclusions

Restoration of Propanol contaminated slides is probably a thermally activated process as shown by the linear relationship between ln(t) and 1/T in figure A.3.

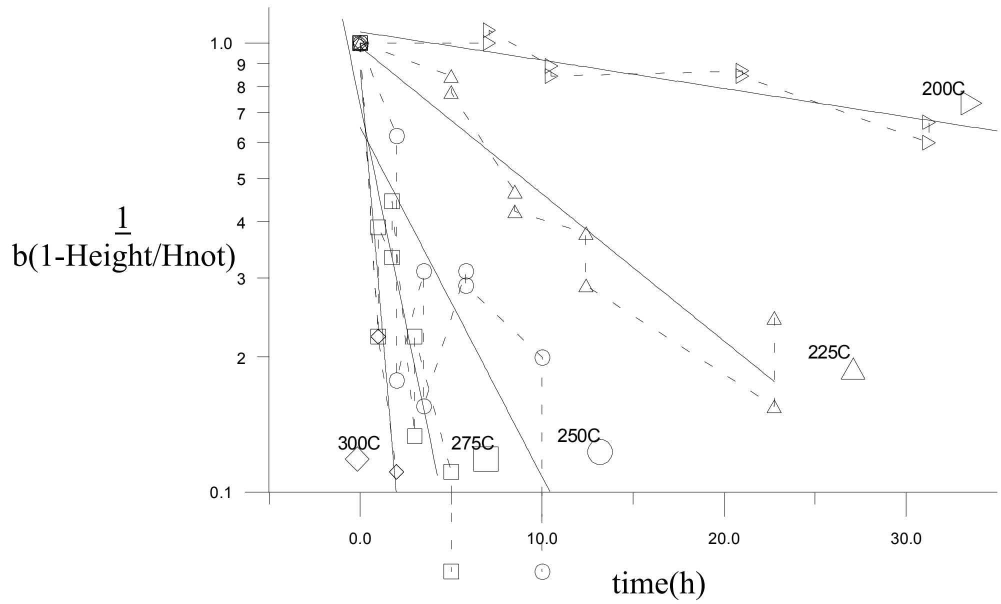


Figure A.2 The logarithmic argument in eq A.9 versus time at temperature. The slope decreases with decreasing temperature as expected.

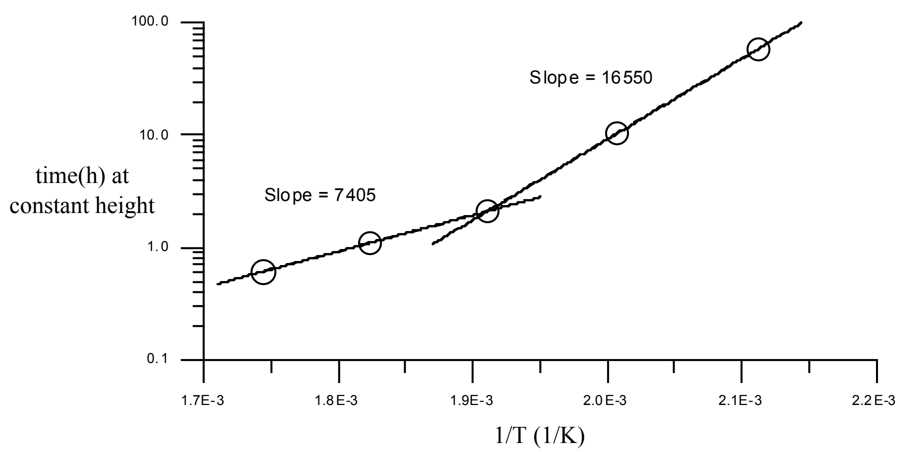


Figure A.3 t versus 1/Temperature. Times are determined from a 4 cm capillary rise. Boltzman's constant is 1.380662 $\times$ 10^{-23} J/K and Avagadro's number is 6.022045 $\times$ 10^{23} 1/mol. A slope of 16550 K corresponds to an activation energy of 138 kJ/mol. A slope of 7405 K corresponds to an activation energy of 62 kJ/mol.

Using Boltzman's constant, $k_B = 1.380662 \times 10^{-23}$ J/K, and Avagadro's number, $N_A = 6.022045 \times 10^{23}$ 1/mol, we find that the activation energy is 138 kJ/mol for a slope of 16550 K and 62 kJ/mol for a slope of 7405 K. These experiments also suggest that the effects of contamination are reversible. It would be highly unlikely that two irreversible processes would occur under less than extreme conditions and with these relatively tame molecules; Propanol, water, and silica.

A reversible, thermally activated process implies that bonds between molecules are involved. The activation energy of the high temperature data is on the order of a hydrogen bond, ~80 kJ/mol. A common assumption is that water hydrogen bonds to a silica surface. Under certain conditions[1], Propanol is likely to hydrogen bond to surface adsorbed water and other surface sites.

In terms of the restricted flow behavior, a possible explanation may be the following. When a solid is heated and then allowed to cool in an environment with some humidity, water will adsorb to the surface of the solid. The Propanol exposed to the surface bonds to the water molecules by means of a hydrogen bond. The energy in a hydrogen bond between water and Propanol molecules is approximately 84 kJ/mol (Barney Ellison, personal communication), roughly 30 % greater than our smallest measured value for the activation energy. The explanation for this discrepancy is unknown. It is possible that the distance over which the bond is acting is smaller or greater and that there is the presence of an additional attraction between the solid surface and the Propanol molecules.

---

[1] A clean sample must be contaminated and allowed to fully dry.

Appendix B

DIRECT MEASUREMENTS

of

CYLINDRICAL SAMPLE STIFFNESS

The samples stiffness is derived by measuring the displacement of the upper holding ring relative to the lower, and the force that caused the displacement (figure B.1).

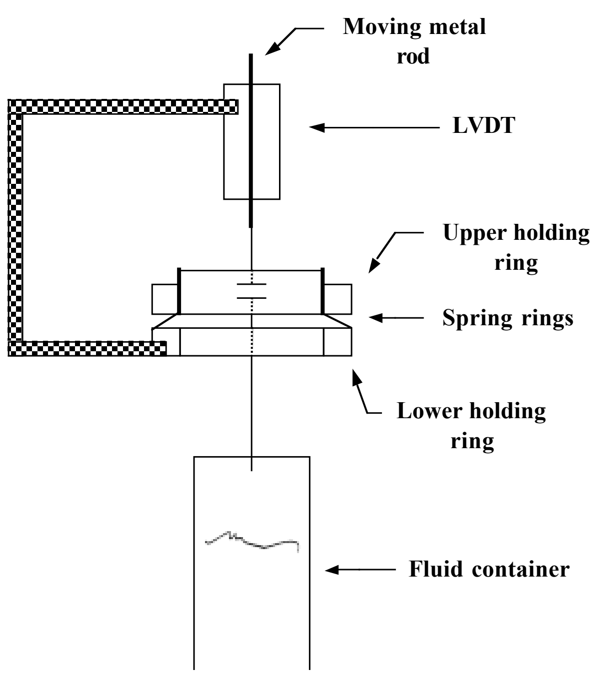


Figure B.1 Schematic of measurement assembly. The lower holding ring is held fixed with respect to the linear variable differential transfomer, LVDT. The upper holding ring and metal rod move in response to a change in mass of the fluid container. The response is dependent on the stiffness of the spring rings. A voltage is produced by the LVDT to conclude the measurement.

The samples displacement is measured with a linear variable differential transformer, LVDT. The LVDT has two leads wound around a central hole. In the central hole is placed a metal rod that couples the two leads and produces a voltage. Within a modest range, the voltage output of the transformer is linearly dependent on the position of the rod (figure B.2).

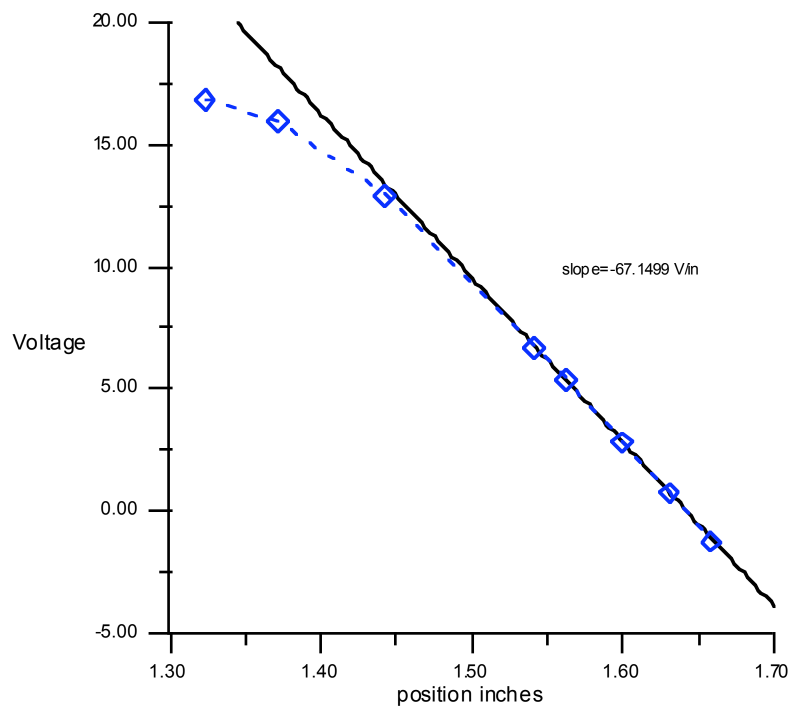


Figure B.2 LVDT ouput voltage versus position of the metal rod within the LVDT, calibration measurement. The position measurements are all made with a micrometer relative to a point fixed with respect to the LVDT. At zero voltage, the metal rod is centered vertically within the LVDT. The relationship between output voltage and position of the metal rod is linear between ±5V and ± .1 in.

The rod is securely fastened to the upper holding ring of the sample. The displacement of the upper holding ring is therefore directly proportional to the voltage produced by the transformer. Force is applied to the sample in the form of a weight. A container in which a measured amount of water is poured, is hung from the upper holding ring. The force acting on the upper holding ring is the mass of water times the acceleration due to gravity, Vg. Having measured both the force and displacement simultaneously, the stiffness can be calculated.

At an input voltage of 15 V, the transformer is initially calibrated and found to have a change in voltage with respect to a change in displacement of ~2560 V/m. Mass versus displacement is measured multiple times for the same approximate conditions fo both the cylindrical sample and reference (figures B.3 & B.4).

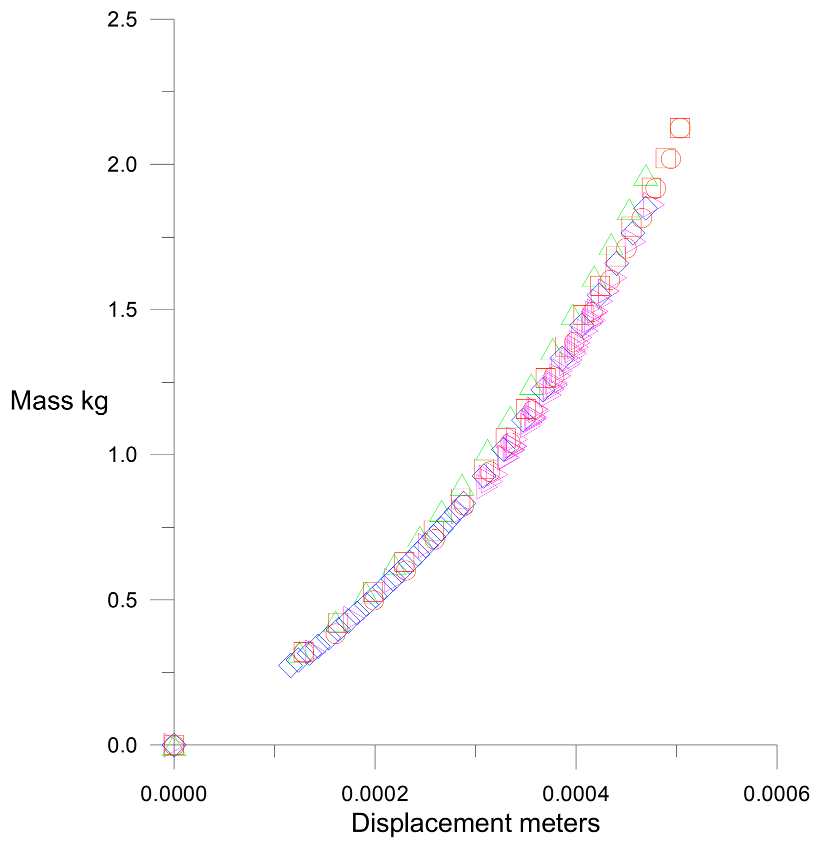


Figure B.3 Mass versus displacement for the cylindrical sample (all measurements). The stiffness is the ratio of mass to displacement times the acceleration due to gravity, the slope of this curve. The slope is changing indicating a stiffness that is not constant.

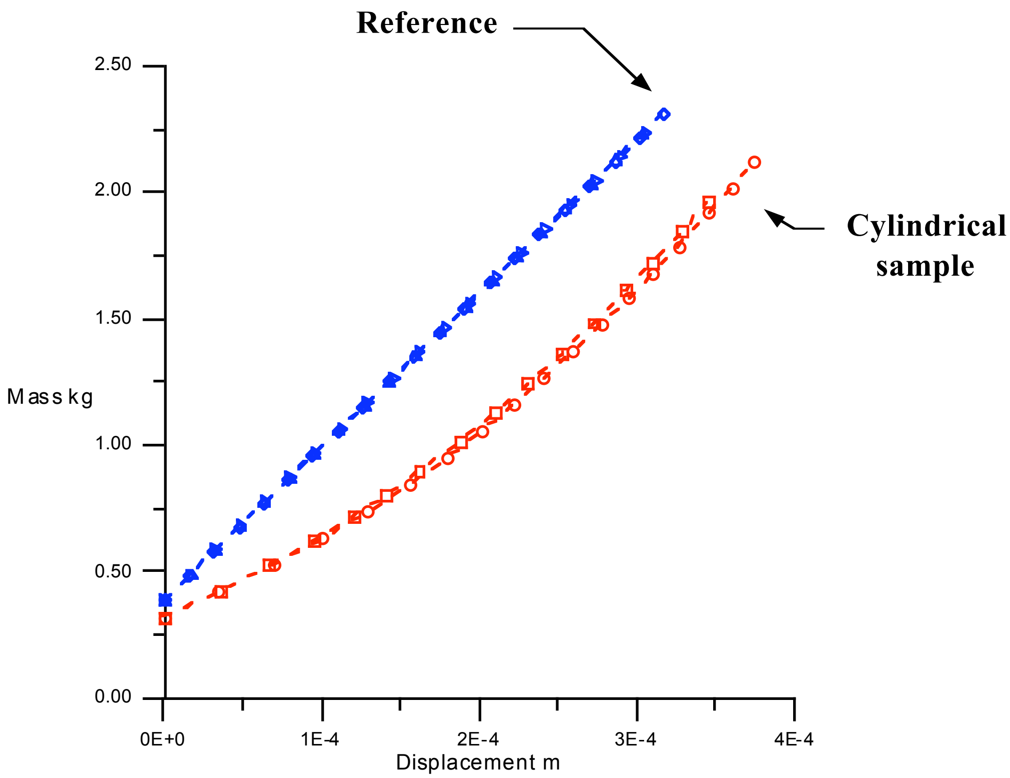


Figure B.4 Mass versus displacement for both the cylindrical sample and reference. The reference has a constant slope, therefore it has a contstant stiffness.

The stiffness of the sample and reference is calculated by dividing the change in mass by the change in position and multiplying by the acceleration due to gravity (figures B.5 & B.6). The stiffness of the reference is constant at ~59.5 kN/m. The stiffness of the cylindrical sample has a linear relationship with respect to the displacement,

$$k = 1.38 \times 10^8 x + 2.23 \times 10^4 \text{ N/m}, \quad \text{(B.1)}$$

and the mass,

$$k = 2.29 \times 10^4 m + 2.52 \times 10^4 \text{ N/m}. \quad \text{(B.2)}$$

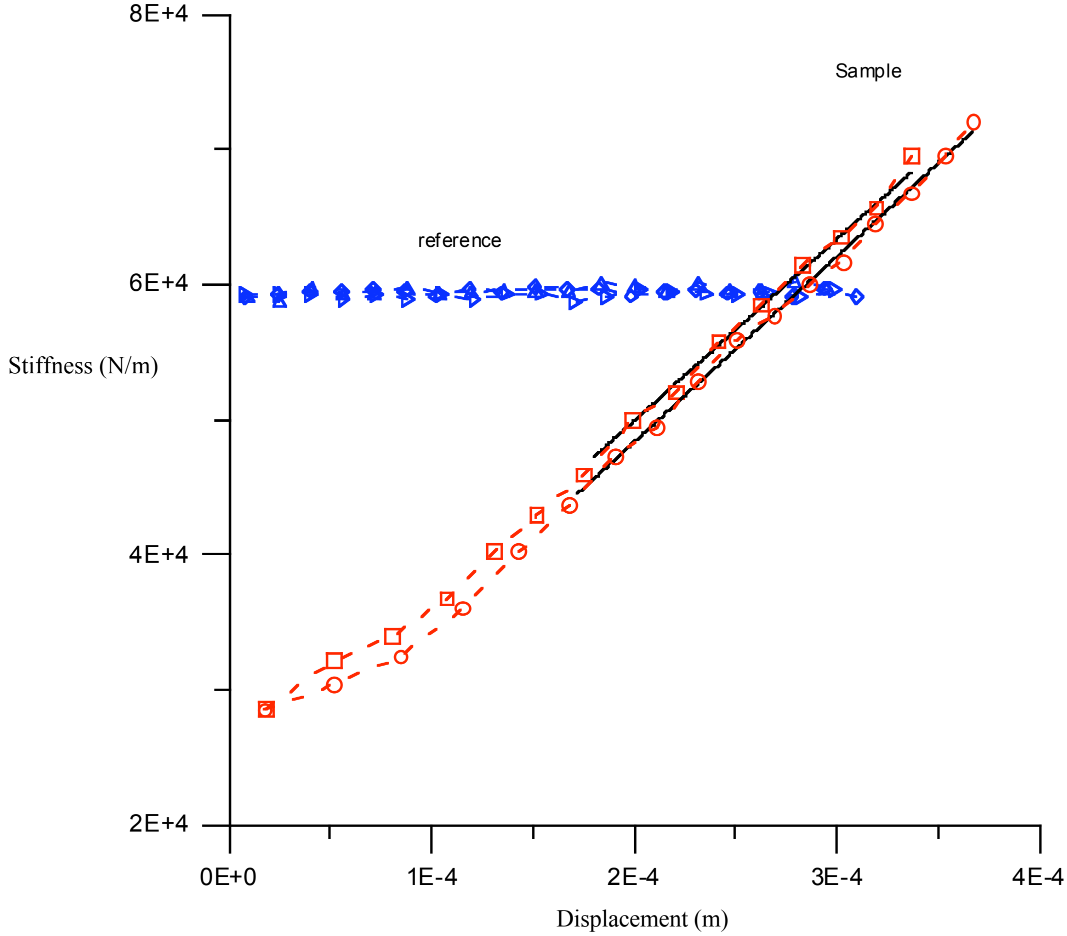


Figure B.5 Stiffness versus displacement for cylindrical sample and reference (2 measurements). The stiffness of the reference is nearly constant. The stiffness of the cylindrical sample is linear with respect to displacement over a range of ~.2 mm.

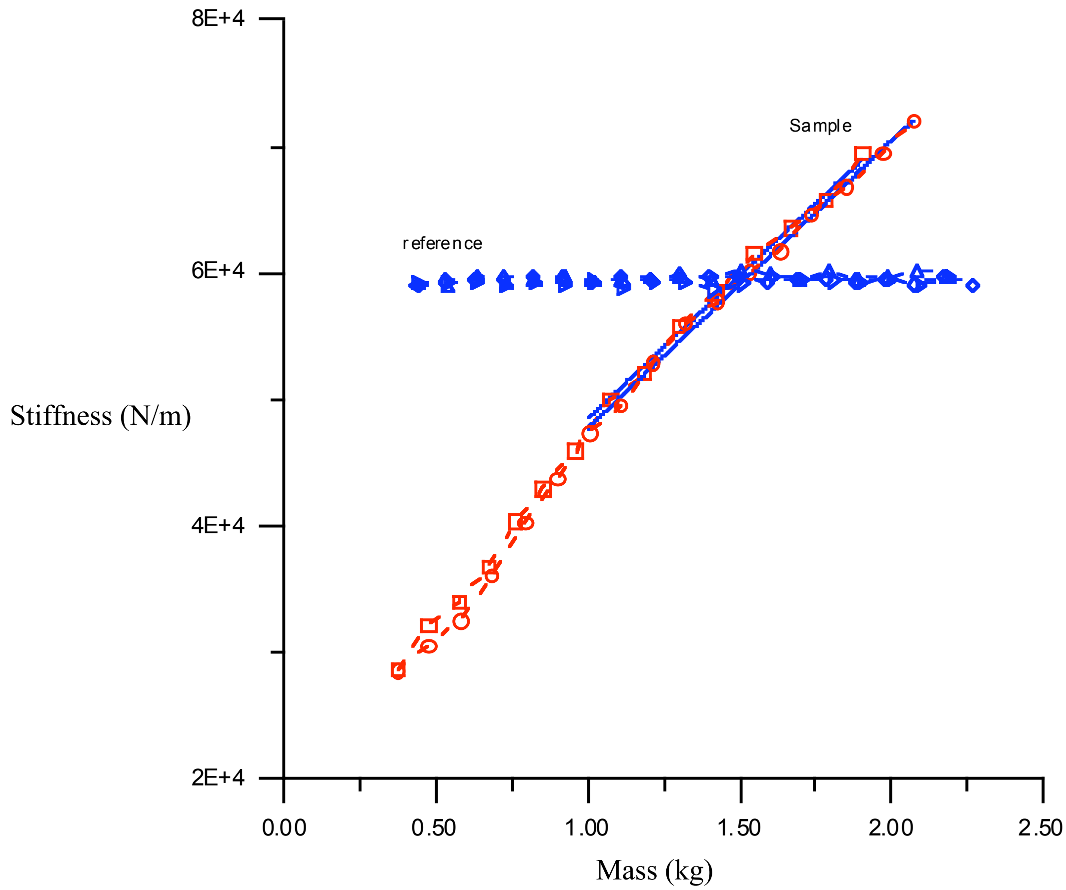


Figure B.6 Stiffness versus mass for cylindical sample and reference. The reference stiffness is independent of applied force between .5 and 2.0 kg. The cylindrical sample stiffness increases linearly between 1.0 and 2.0 kg.

Appendix C

COMPUTER MODEL

Introduction

The following program is written in C++. The program could be compiled to run on any computer with any operating system supporting a C++ compiler. The program calculates the energy lost due to both viscous effects, Elossv, and resistive contact line motion, Eloss, in dependence on frequency of sinusoidal deformation, freq. The program consists of a main routine, main(), and three subroutines, guess(), setsvv(), and vol(). Its 9 input arguments are equilibrium contact angle in degrees, *dcont*, contact angle hysterisis, *cont_hyst*, initial radius of fluid volume, *Ri*, initial gap separation, *orig_ht*, amplitude of deformation, *Amp*, velocity of the contact line at the contact angle hysterisis, *vm*, constant *b* controlling slope of sticking regime, dry stiffness scaling factor, *k1dry*, and slope of slipping regime in deg s/m, *a*.

Within the main() routine, a moving contact line and changing contact angle are described for multiple frequencies between .001 and 100 Hz. Each frequency occurs over 1/4 of a cycle and is broken up into 1000 time steps. During each of these increments, the change in gap separation is calculated. Due to conservation of volume, the deformation results in a change in contact angle. As the contact angle moves from its equilibrium state, the probability of the radius changing increases.

The energy lost is calculated and summed over every time step. At the end of the time loop the stiffness and energy stored due to contact line movement, and the energy lost due to viscous effects are calculated.

The following is the basic structure of the program

Subroutines()

main()

I. Loop over frequencies, .001 Hz to 100 Hz.

- A. Loop over quarter cycle with increment dt
  1. Change in contact angle, dc, radius, R
  2. Calculate Eloss
- B. Calculate k1, Estor, Elossv, Atten

II. Output data

Code

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>
#define small .000000001
#define pi 3.14159265
#define rho .001
#define mslwmirpl .6

//GUESS ROUTINE USED TO DETERMINE NEW CONTACT ANGLE
double guess(double pre, double pre_pre, double dif, double pre_dif)
        {
        double slope, predicted, differ;
        if (!(differ = dif - pre_dif)) differ = small;
        slope = dif * (pre - pre_pre)/differ;
        while (1)
                {
                predicted = pre - slope;
                if (predicted < 0)
                        {
                        printf("Error: predicted %e, pre %f pre_pre %f dif %f pre_dif
        %f\n", predicted, pre, pre_pre, dif, pre_dif);
                        exit(-2);
                        }
                if (predicted < pi/2.0) break;
                slope /= 2.0;
                }
        return predicted;
        }

double svvRsq, svvhsq;          // pi*R*R*h and pi*h*h*R

// SUBROUTINE TO SET VOLUME COEFFICIENTS

void setsvv(double R, double h)
{
svvhsq =   pi * h * h * R;
svvRsq =   pi * h * R * R;
}

long num_vol_calls;
//VOLUME SUBROUTINE USING EQ. II.20

double vol(double contact_angle)
        {
        double beta, sinbeta;
        ++num_vol_calls;
        if (!(beta = pi/2.0-contact_angle)) return svvRsq;
        sinbeta = sin(beta);
        return svvRsq - svvhsq * .5 * (beta - cos(beta)*sinbeta)/(sinbeta*sinbeta);
        }

//MAIN ROUTINE

int main(int narg, char *argv[])
{

//  All units in millimeters, kilograms, seconds unless otherwise noted.

double xm, X, xcap, xini, x1;
int it, i, flag;
double Amp, orig_ht, dhdt, Uh;
double new_ht;
double k1, k2 = 59459.3;                    //k1, stiffness of sample. k2, stiffness of reference in kN/m
double volume_cur, volume_pre, Area, Areai, R, Ri, dR;
double cont_an_pre, dif, dif_pre, can_pre_pre;
double surface_tension = .07275, n = .0000010019;            //Surface tension and viscosity of water
```

```
double contact_angle1, contact_angle2, dc, dc_prob;
double dcont, cont_hyst;
double Elost, Estor, Elossv, Eacc;
double cap_pres, cap_pres_i, dFcap, Atten, argatten, attenwrs, Atten_no_v,       Atten_no_c;
double deg_xm, deg_xini, met_per_turn;
double freq, t, wr = 10000000.0;
double scale, sum;
FILE *outdat,*outmat,*outmat2;

if (narg < 9)
        {
        printf("\n\n\t\tInput argumants less than required\n\n\t\tnarg = %d\n\n",narg -                    1);
        printf("\t\tusage: visc_men dcont cont_hyst Ri orig_ht Amp dR\n");
        return 1;
        }

outmat2 = fopen("attmatt6.dat","w");
outmat = fopen("attmat.dat","w");
outdat = fopen("att.dat","w");

//Command line arguments

dcont = atof(argv[1]);              //equilibrium contact angle, degrees
cont_hyst = atof(argv[2]);          //contact angle hysterisis, degrees
Ri = atof(argv[3]);                 //radius of fluid film, mm
orig_ht = atof(argv[4]);            //gap height, mm
Amp = atof(argv[5]);                //change in gap height, mm
double vm = atof(argv[6]);     //sudden change in radius, mm
double b = atof(argv[7]);           //power value controlling trans. between sicking and slipping
double k1dry = atof(argv[8]); //initial dry stiffness above ref stiffness
double a = atof(argv[9]);           //slope velocity/radian above vm


printf("\n\n");
fprintf(outdat,"freq,Atten,attenwrs,Atten_no_v,Atten_no_c,Kave/k2,Eave,Eave2,Radi        us,cont_ang2\n");

//          To create a new surface, add the appropriate lines,
//          1.          for (variable...
//          2.          printf(,variable)
//          3.          fprintf(outmat,,variable)

//for(r = 8; r > 0; r = r - .1)
//for(a = a; a > a - .005; a = a - .005)
//for(b = 80; b > 0; b = b - 1)
//for(cont_hyst = 20; cont_hyst > -0.1; cont_hyst = cont_hyst - .25)
//  {
  contact_angle1 = dcont * pi/180;              //Convert to radians
  R = Ri;
  setsvv(R,orig_ht);                            //Set up volume constants
  volume_pre = vol(contact_angle1);



//                              THE FREQUENCY LOOP

#define start_freq .00001
for(freq = start_freq; freq <= 100; freq = freq * 2)
        {
        double qcycle = 1.0/(4.0*freq);
        double dt = qcycle/1000;
        contact_angle2 = contact_angle1;
        cont_an_pre = contact_angle2;
        double prev_delta_ca = .0001;
        R = Ri;
        new_ht = orig_ht;
        Elost = 0;
        sum = 0;
        for(t = 0; t <= qcycle; t += dt)                //Value needed to scale Amp because
                                //of finite approximation to dhdt
                {
```

```
                        dhdt = Amp * 2 * pi * freq * cos(2 * pi * freq * t);
                        sum += dhdt*dt;
                        }
            scale = Amp/sum;

//                                  THE TIME LOOP

            for(t = 0; t <= qcycle; t += dt)
                        {
                        if ((dc = contact_angle2 - contact_angle1) > 0)
                                    {
                                    if (dc < (cont_hyst * pi/180))
                                                dR = dt * vm * pow(180 * dc/(pi * cont_hyst),b);
                                    else
                                                dR = ((dc - cont_hyst * pi/180)/a + vm) * dt;
                                    R = R + dR;
                                    }
                        Elost = Elost + 4 * pi * surface_tension * (cos(contact_angle1) -
                                    cos(contact_angle2))*R*dR;
                        fprintf(outmat2,"%f\t%f\t%f\t%f\n",t,log10(freq),R,dc*180/pi);
                        dhdt = Amp * 2 * pi * freq * cos(2 * pi * freq * t);
                        new_ht -= dhdt*dt*scale;
                        setsvv(R, new_ht);                      //Set up volume constants
                        dif_pre = volume_pre - vol(contact_angle2);
                        double sv_ca_in = cont_an_pre = contact_angle2;
                        contact_angle2 += prev_delta_ca;
                        it = 0;
                        do
                                    {
                                    volume_cur =  vol(contact_angle2);
                                    if (!(dif = volume_pre - volume_cur)) break;
                                    can_pre_pre = cont_an_pre;
                                    cont_an_pre = contact_angle2;
                                    contact_angle2 = guess(cont_an_pre,can_pre_pre,dif,dif_pre);
                                    dif_pre = dif;
                                    it = it + 1;
                                    }
                        while(dif_pre > small || dif_pre < -small);
                        prev_delta_ca = contact_angle2 - sv_ca_in;
                        }//End of time loop

            Areai = pi * Ri * Ri;
            Area = pi * R * R;
            cap_pres_i = 4 * surface_tension * cos(contact_angle1)/orig_ht;
            cap_pres = 4 * surface_tension * cos(contact_angle2)/new_ht;
            dFcap = Areai * cap_pres_i - Area * cap_pres;
            k1 = dFcap/Amp + k1dry * k2;


//                      Calculate energy stored and energy lost

            Estor = dFcap * dFcap / (2 * k2);
            Uh = Amp * freq;
            Elossv = 3*pi*n*Ri*Ri*Ri*Ri*Uh/(2*orig_ht*orig_ht);
            Atten = (Elost + Elossv)/(2 * pi * (k2 * Amp * Amp/2 + Estor));
            Atten_no_v = Elost/(2 * pi * (k2 * Amp * Amp/2 + Estor));
            Atten_no_c = (Elossv)/(2 * pi * (k2 * Amp * Amp/2));
            argatten = wr*wr * tan(Atten)/(wr*wr-freq*freq);
            attenwrs = atan(argatten);

            fprintf(out,"%f,%f,%f,%f,%f,%f\n",freq,Atten,Kave/k2,Eave,Rave, cont_ang_ave * 180/pi);
            fprintf(outmat,"%f\t%f\t%f\t%f\n",log10(freq),a,Atten_no_v,k1);
            fprintf(outdat,"%f,%f,%f,%f,%f,%f,%f,%f,%f,%f\n",freq,Atten,attenwrs,
            Atten_no_v,Atten_no_c,k1/k2,Elost,Estor,R,contact_angle2 * 180/pi);
            }//End of frequency loop
fclose(outmat2);
fclose(outmat);
fclose(outdat);
return 0;
}
```

Appendix D

DATA AQUISITION PROGRAM

This program is written in basic and requires the Hi-Tech Basic and DOS operating systems.

```
1!   RE-STORE "Q2da"
2!
3!  THE FOLLOWING IS USED TO LOAD THE NATIONAL INSTRUMENTS AT-GPIB
4!  DRIVER AND FOLLOWING COMMAND SETS COM1 TO THE RS232
6!    LOAD BIN "/HTB386/GPIBN;BO AT-GPIB BA 2C0 IN 11"
7!    LOAD BIN "SERIAL;DRIVER 1 DISABLE"
8!  THE MAJORITY OF THIS PROGRAM WAS WRITTEN BY JOSEF PAFFENHOLZ
9!  MODIFICATIONS MADE BY ROB ANDZIK (6/18/90) AND OLIVER BOYD (6/10/94)
11   COM /Nicdata/ REAL Vrange2,Period,Freq,Clkdiv
14   COM /Filename/User$[5],Path$[20],Date$[10],INTEGER Nr
15   COM /Flag/ INTEGER Flag,Storeflag$[15],Stackingflag$[15]
16   COM /Set_up/ INTEGER Setflag
17   COM /Ct/ REAL Ct_output,@Rs232
18   COM /Mirmove/ REAL Ab,Bc,Cd,De,Ef,Fa
20   COM /Infom/Ident$[3],Samp$[20],Satur$[20],Comm$[80],Prmir$[1]
21   COM /Extra/Fate$[4]
22   COM /Zeros/Nr$[3]
24   User$="Q2"
25   ASSIGN @Wavetek TO 709
26   ASSIGN @Rs232 TO 11
27   CALL Init_rs232
28   CALL Init_motor
29   CALL Look_for_date
30   Nr=0
31   Ghj=1
33   Ab=-154000
34   Bc=-259000
35   Cd=-149000
36   De=-162000
37   Ef=-233000
38   Fa=-195000
39   Prmir$="a"
40   Path$="/"&User$&"/DAT/"&Date$
41   OUTPUT @Rs232;"SC1"
42   OUTPUT @Rs232;"ST1"
43  ON ERROR CALL Read_dir
44  CREATE DIR Path$
45  OFF ERROR
46   Storeflag$="STORAGE ON"
47   C$=CHR$(255)&"K"
48   CLEAR SCREEN
49   PRINT "YOUR OPTIONS: (1)PLAY WITH WAVETEK"
50   PRINT "              (2)PLAY WITH MIRRORS"
51   PRINT "              (3)CONFIGURE ANALOGIC"
52   PRINT "              (4)TAKE MEASUREMENTS"
53   PRINT ""
54   PRINT ""
55   PRINT ""
56  IF Ghj=1 THEN
58   PRINT " **** NOTE ****"
59   PRINT ""
60   PRINT " If measurement has been interupted, you must"
61   PRINT "    make sure that the variable prmir$(line 39),"
62   PRINT "    initially mir a, has the value of the"
63   PRINT "    present mirror"
64   PRINT ""
65   PRINT "present mirror should be: ",Prmir$
67   Ghj=2
68  END IF
69   INPUT I
70   IF I=1 THEN GOSUB Wavetek
71   IF I=2 THEN GOSUB Mirrors
72   IF I=3 THEN CALL Config_ana
73   IF I=4 THEN CALL Set_up
74   GOTO 48
75 Wavetek:!
76   OUTPUT @Wavetek;"R"
```

```
77   CLEAR SCREEN
78   INPUT "AMPLITUDE =?",Amp
79   INPUT "FREQUENCY =?",Freqw
80   O_freq=Freqw*8192
81   OUTPUT @Wavetek;"C0F"&VAL$(O_freq)&"A"&VAL$(Amp)&"I"
82   RETURN
83 Mirrors:!
84   CLEAR SCREEN
85   REAL J
86   Ch=2
87   INPUT "By step(1) or by mirror(enter)?",Ch
88   IF Ch=1 THEN
89     INPUT "HOW MANY STEPS?",J
90   ELSE
91     INPUT "Which Mirror to go to (lower case)?",Mir$
92     IF Prmir$="a" THEN
93        IF Mir$="a" THEN J=0
94        IF Mir$="b" THEN J=Ab
95        IF Mir$="c" THEN J=Ab+Bc
96        IF Mir$="d" THEN J=-Fa-Ef-De
97        IF Mir$="e" THEN J=-Fa-Ef
98        IF Mir$="f" THEN J=-Fa
99     END IF
100    IF Prmir$="b" THEN
101       IF Mir$="b" THEN J=0
102       IF Mir$="c" THEN J=Bc
103       IF Mir$="d" THEN J=Bc+Cd
104       IF Mir$="e" THEN J=-Ab-Fa-Ef
105       IF Mir$="f" THEN J=-Ab-Fa
106       IF Mir$="a" THEN J=-Ab
107    END IF
108    IF Prmir$="c" THEN
109       IF Mir$="c" THEN J=0
110       IF Mir$="d" THEN J=Cd
111       IF Mir$="e" THEN J=Cd+De
112       IF Mir$="f" THEN J=-Bc-Ab-Fa
113       IF Mir$="a" THEN J=-Bc-Ab
114       IF Mir$="b" THEN J=-Bc
115    END IF
116    IF Prmir$="d" THEN
117       IF Mir$="d" THEN J=0
118       IF Mir$="e" THEN J=De
119       IF Mir$="f" THEN J=De+Ef
120       IF Mir$="a" THEN J=-Cd-Bc-Ab
121       IF Mir$="b" THEN J=-Cd-Bc
122       IF Mir$="c" THEN J=-Cd
123    END IF
124    IF Prmir$="e" THEN
125       IF Mir$="e" THEN J=0
126       IF Mir$="f" THEN J=Ef
127       IF Mir$="a" THEN J=Ef+Fa
128       IF Mir$="b" THEN J=-De-Cd-Bc
129       IF Mir$="c" THEN J=-De-Cd
130       IF Mir$="d" THEN J=-De
131    END IF
132    IF Prmir$="f" THEN
133       IF Mir$="f" THEN J=0
134       IF Mir$="a" THEN J=Fa
135       IF Mir$="b" THEN J=Fa+Ab
136       IF Mir$="c" THEN J=-Ef-De-Cd
137       IF Mir$="d" THEN J=-Ef-De
138       IF Mir$="e" THEN J=-Ef
139    END IF
140    Prmir$=Mir$
141    IF Gl=1 THEN Gl=Gl+1
142  END IF
143    OUTPUT @Rs232;"ST0"
144    WAIT .1
145    OUTPUT @Rs232;"A1.5 V3 D"&VAL$(J)&" G"
146    OUTPUT @Rs232;"ST1"
```

```
147    RETURN
148   END
149!*********************************************************************
150 SUB Init_rs232
151  CONTROL 11,0;1
152  CONTROL 11,13;9600
153  CONTROL 11,14;3
154  CONTROL 11,5;3
155  CONTROL 11,12;16+32+128
156 SUBEND
157!*********************************************************************
158 SUB Init_motor
159  COM /Ct/ REAL Ct_output,@Rs232
160  OUTPUT @Rs232;"8LD3"
161  OUTPUT @Rs232;"A3"
162  OUTPUT @Rs232;"K"
163 SUBEND
164!*********************************************************************
165 SUB Look_for_date
166  COM /Filename/User$,Path$,Date$,INTEGER Nr
167  COM /Extra/Fate$
168  D$=DATE$(TIMEDATE)
169  M$=D$[4,6]       !MONTH
170  D$=D$[1,2]       !DAY
171  D$=TRIM$(D$)
172  Add_zero(D$)
173  SELECT M$
174  CASE "Jan"
175    M$="01"
176  CASE "Feb"
177    M$="02"
178  CASE "Mar"
179    M$="03"
180  CASE "Apr"
181    M$="04"
182  CASE "May"
183    M$="05"
184  CASE "Jun"
185    M$="06"
186  CASE "Jul"
187    M$="07"
188  CASE "Aug"
189    M$="08"
190  CASE "Sep"
191    M$="09"
192  CASE "Oct"
193    M$="10"
194  CASE "Nov"
195    M$="11"
196  CASE "Dec"
197    M$="12"
198  END SELECT
199  CALL Add_zero(M$)
200  Fate$=M$&D$
201  Date$=M$&D$&"_"
202 SUBEND
203 !*********************************************************************
204 SUB Set_up
205   COM /Nicdata/ REAL Vrange2,Period,Freq,Clkdiv
206   COM /Set_up/ INTEGER Setflag
207   COM /Filename/User$,Path$,Date$,INTEGER Nr
208   COM /Mirmove/ REAL Ab,Bc,Cd,De,Ef,Fa
209   COM /Infom/Ident$[3],Samp$[20],Satur$[20],Comm$[80],Prmir$[1]
210   COM /Extra/Fate$
211   COM /Ct/ REAL Ct_ouput,@Rs232
212   DIM Response$[1],Accept$[20]
213   DIM Freqa(30)
214   DIM Frtyp$[4]
215   INTEGER Num
216   ASSIGN @Rs232 TO 11
```

```
217  ASSIGN @Wavetek TO 709
218  Anafig$="N"
219  OUTPUT @Wavetek;"R"
220  CLEAR SCREEN
221  IF Prmir$="a" THEN
222    GOSUB Info
223    ELSE
224    REAL Mo
226    PRINT ""
227    PRINT ""
228    PRINT "                    LASER LIGHT MUST BE AT MIRROR A."
229    PRINT ""
230    PRINT "        It is currently on mirror ",Prmir$
231    PRINT ""
232    PRINT "      Now moving to a, are you ready to play?"
234    IF Prmir$="b" THEN Mo=-Ab
235    IF Prmir$="c" THEN Mo=-Bc-Ab
236    IF Prmir$="d" THEN Mo=-Cd-Bc-Ab
237    IF Prmir$="e" THEN Mo=Ef+Fa
238    IF Prmir$="f" THEN Mo=Fa
239    OUTPUT @Rs232;"ST0"
240    WAIT .1
241    OUTPUT @Rs232;"A1.5 V3 D"&VAL$(Mo)&" G"
242    OUTPUT @Rs232;"ST1"
243    INPUT "    **  Press return to continue  **",Dummy
245    GOSUB Info
246  END IF
247!The variable H corresponds to the frequency
249  FOR H=1 TO Num
250   Freq=Freqa(H)
251   Out_freq=Freq*8192
252   Amp_s=.04*LOG(Freq)+Amp
254   OUTPUT @Wavetek;"C0F"&VAL$(Out_freq)&"A"&VAL$(Amp_s)&"I"
255   IF H=1 THEN GOSUB Start_up
256   GOSUB Start
257  NEXT H
258  GOSUB End
259 Info:!
260    CLEAR SCREEN
261    ASSIGN @Ff TO "c:\q2\prg\q2dr.inf";FORMAT ON
262    INPUT "Y-intercept ( = 3.6050 for 2.8 fringes at current gains)? ",Amp
263    INPUT "Type of frequency range (low,test,manu)? ",Frtyp$
264    IF Frtyp$="test" THEN
265      Num=1
266      Freqa(1)=10
267      OUTPUT @Ff;Fate$,VAL$(Nr+1),VAL$(Num*6),"xxx"
268      GOTO 293
269    END IF
270    IF Frtyp$="manu" THEN
271      INPUT "How many frequencies (max=21)? ",Num
272      FOR I=1 TO Num
273      PRINT "Frequency #"&VAL$(I)&"?"
274      INPUT K
275      Freqa(I)=K
276      NEXT I
277    ELSE
278      ASSIGN @Ffile TO "q2da.fre";FORMAT ON
279      ENTER @Ffile;Num
280      PRINT Num
281      FOR I=1 TO Num
282      ENTER @Ffile;Freqa(I)
283      PRINT Freqa(I)
284      NEXT I
285    END IF
286    INPUT "Three letter identifier? ",Ident$
287    PRINT TABXY(1,30),"Capital X is a variable which you supply."
288    INPUT "Sample type (X glass slides, rock type, glass beads)?",Samp$
289    INPUT "Saturant (none, water, methanol, etc.)?",Satur$
290    PRINT TABXY(1,30),"sampXX, X.Xmm sl., Xx prestress, and other comments"
291    INPUT "Comments (within double quotes)?",Comm$
```

```
292     OUTPUT @Ff;Fate$,VAL$(Nr+1),VAL$(Num*6),Ident$,Samp$,Satur$,Comm$
293     RETURN
294 Start:!
295 !The variable M corresponds to the mirror at a given frequency
296    FOR M=1 TO 6
297    GOSUB Ttl
298    GOSUB Output_ttl
299    IF M=1 THEN GOTO 308
300    OUTPUT @Rs232;"ST0"
301    WAIT .1
302    IF M=2 THEN OUTPUT @Rs232;"A1.5 V3 D"&VAL$(Ab)&" G"
303    IF M=3 THEN OUTPUT @Rs232;"A1.5 V3 D"&VAL$(Bc)&" G"
304    IF M=4 THEN OUTPUT @Rs232;"A1.5 V3 D"&VAL$(Cd)&" G"
305    IF M=5 THEN OUTPUT @Rs232;"A1.5 V3 D"&VAL$(De)&" G"
306    IF M=6 THEN OUTPUT @Rs232;"A1.5 V3 D"&VAL$(Ef)&" G"
307    OUTPUT @Rs232;"ST1"
308    PRINT "ATTEMPT #"&VAL$(M)&" AT FREQ:"&VAL$(Freq)&"("&VAL$(H)&") using WaveTek voltage: "&VAL$(Amp_s)
309    WAIT 20
310    CALL Read_data
311    NEXT M
312    OUTPUT @Rs232;"ST0"
313    WAIT .1
314    OUTPUT @Rs232;"A1.5 V3 D"&VAL$(Fa)&" G"
315    OUTPUT @Rs232;"ST1"
316    RETURN
317 Ttl:!
318    IF Freq<.0625 THEN
319       Clkdiv=2
320       GOTO 339
321    END IF
322    IF Freq<.125 THEN
323       Clkdiv=4
324       GOTO 339
325    END IF
326    IF Freq<.25 THEN
327       Clkdiv=8
328       GOTO 339
329    END IF
330    IF Freq<.5 THEN
331       Clkdiv=16
332       GOTO 339
333    END IF
334    IF Freq<1 THEN
335       Clkdiv=32
336       GOTO 339
337    END IF
338     Clkdiv=64
339    RETURN
340 Output_ttl:!
341    Ttlport=&H3BC
342    Divisor=256-Clkdiv
343    OUT Ttlport,Divisor
344    RETURN
345 Start_up:!
346       CLEAR SCREEN
347       INPUT "CONFIGURE ANALOGIC DIGITIZER ?, Y/N [N]",Anafig$
348       IF Anafig$="Y" OR Anafig$="y" THEN CALL Config_ana
349       !
350       IMAGE 16/,30A,18A
351       CLEAR SCREEN
352       BEEP 240,.1
353       BEEP 440,.1
354       Setflag=0
355       RETURN
356 End:!
357     QUIT
358 SUBEND
359 !*****************************************************************
360 SUB Read_dir
```

```
361   COM /Filename/User$,Path$,Date$,INTEGER Nr
362   DIM Cat$(1:400)[80],File$(1:400)[14]
363   CAT "/"&User$&"/DAT/"&Date$ TO Cat$(*);NO HEADER,COUNT Nr
364   ERROR SUBEXIT
365 SUBEND
366 !*******************************************************************
367 SUB Read_data
368   COM /Nicdata/ REAL Vrange2,Period,Freq,Clkdiv
369   COM /Filename/User$,Path$,Date$,INTEGER Nr
370   COM /Flag/ INTEGER Flag,Storeflag$[15],Stackingflag$[15]
371   COM /Tmdt/Datetime$[20],Time$[20]
372   COM /Set_up/ INTEGER Setflag
373   INTEGER N
374   ON KEY 0 CALL Not_used
375   ON KEY 1 CALL Not_used
376   ON KEY 2 CALL Not_used
377   GOTO 379
378   PAUSE
379   ON KEY 3 LABEL Storeflag$,2 CALL Storeflag
380   ON KEY 4 CALL Not_used
381   ON KEY 5 CALL Not_used
382   ON KEY 6 CALL Not_used
383   ON KEY 7 CALL Not_used
384   ON KEY 8 LABEL "STORED NR:"&VAL$(Nr) CALL Not_used
385   ON KEY 9 CALL Not_used
386   Nad=715
387   ASSIGN @Nic TO 715
388   OUTPUT Nad;"FLDDLM(3)=5"   !SET THE FEILD DELIMITER TO COMMA
389   OUTPUT Nad;"FLDLEN(3)=10"  !SET THE FIELD LENGTH FOR NUMERIC PRECISION=10
390   OUTPUT Nad;"OMODE=1"       !SET FOR DATA TRANSFER ONLY
391   OUTPUT Nad;"FORMAT=1"      !SET FOR AN ASCII FORMAT
392   OUTPUT Nad;"LINLEN=80"     !SET LINE LENGTH TO 80
393   OUTPUT Nad;"NPTS"          !FIND OUT HOW MANY POINTS IN RECORD
394   ENTER @Nic;N
395   ALLOCATE INTEGER Data(1:N),REAL Stackdata(1:N)
396  !
397  CALL Dataacq(Data(*),N)
398  !
399   ON KEY 3 LABEL Storeflag$,5 CALL Storeflag
400 Cont: IF Storeflag$="STORAGE ON" THEN CALL Store_data(Data(*),N)
401     Setflag=1
402     BEEP 523.25,.2
403     BEEP 1046.5,.2
404     BEEP 783.99,.25
405 SUBEND
406 !*******************************************************************
407 SUB Config_ana
408 !THIS PROGRAM WILL CONFIGURE THE D6100 FOR Q DATA AQUISITION
409   DIM Command$[100],Darm$[4],Filter$(1:4)[12],Ch$[1],Trig_source$[10],Junk$(50)[20]
410   ASSIGN @Ana TO 715
411   !
412   CLEAR SCREEN
413   OUTPUT @Ana;"RESET"
414   INPUT "TO QUIT NOW PRESS 1, Else enter",K
415   IF K=1 THEN GOTO 479
416   DISP "INITIALIZING ANALOGIC: PRESS F2 WHEN ANALOGIC IS READY"
417   BEEP
418   PAUSE
419   DISP
420   OUTPUT @Ana;"DISARM"                  !SEND DARM COMMAND
421   !
422   FOR I=1 TO 4
423     Ch$=VAL$(I)                 !PUT THE VALUE OF I INTO A CHARACTER STRING
424     Filter$(I)="FILTER("&Ch$&")=1"        !COMBINE FILTER INTO ONE STRING
425     OUTPUT @Ana;Filter$(I)               !SEND FILTER COMMANDS TO ANA
426   NEXT I                               !FOR ALL FOUR CHANNELS
427 !
428 !      THE FOLLOWING COMMANDS CAME FROM "MODEL 6100 UNIVERSAL
429 !   WAVEFORM ANALIZER   VOLUME II: COMMAND INDEX"  pF7C-1 TO F7C-145
430 !   THE COMMANDS ARE LISTED ALPHABETICALLY
```

```
431 !
432   OUTPUT @Ana;"TRIG"            !SET TRIGGER SOURCE TO
433   OUTPUT @Ana;"TRGSRC=7"        !TO EXT. TRIGGER
434   OUTPUT @Ana;"TRGLEV=2"        !SET TRIG. LEVEL TO TWO VOLTS
435   OUTPUT @Ana;"TRGM=1"          !SET MODE TO NORMAL
436 !
437 !
438 !       THESE LINES TAKEN OUT TO SPEED UP THE ARMING PROCESS
439 !
440 !
441 !
442 !
443 !
444   OUTPUT @Ana;"BUFR"            !SELECT BUFFER MENU
445   OUTPUT @Ana;"INPSEL=2"        !SELECT BUF.A2
446   OUTPUT @Ana;"REC=1"           !MODE OFF
447   OUTPUT @Ana;"INPSEL=3"        !SELECT BUF.A3
448   OUTPUT @Ana;"REC=1"           !MODE OFF
449   OUTPUT @Ana;"INPSEL=4"        !SELECT BUF.A4
450   OUTPUT @Ana;"REC=1"           !MODE OFF
451   OUTPUT @Ana;"TMBSEL=2"        !TIMEBASE B
452   OUTPUT @Ana;"INPSEL=2"        !SELECT INPUT CHANNEL 2
453   OUTPUT @Ana;"REC=2"           !MODE ON
454   OUTPUT @Ana;"TMB"             !SELECT TMB MENU
455   OUTPUT @Ana;"TMBSEL=1"        !SELECT TMB A
456   OUTPUT @Ana;"NPTS=16384"      !SET #OF POINTS TO 16384
457   OUTPUT @Ana;"PERSRC=3"        !SELECT EXTCLK TTL
458   OUTPUT @Ana;"TMBSEL=2"        !SELECT TMB B
459   OUTPUT @Ana;"NPTS(2)=16384"   !SET #OF POINTS TO 16384 FOR 2
460   OUTPUT @Ana;"PERSRC=3"        !SELECT EXTCLK TTL
461   OUTPUT @Ana;"TRCSRC(1)=BUF.A1"!SELECT BUF.A1 AS UPPER SOURCE
462   OUTPUT @Ana;"X"               !SELECT X MENU
463   OUTPUT @Ana;"XSCL(1)=1X1/4"   !SET X SCALE TO 1/4
464   OUTPUT @Ana;"XOFF(1)=.7678"   !ADJUST TRACE TO START AT RIGHT
465   OUTPUT @Ana;"TRCSRC(2)=BUF.B2"!SELECT BUF.B2 AS LOWER SOURCE
466   OUTPUT @Ana;"XSCL(2)=1X1/4"   !SET X SCALE TO 1/4
467   OUTPUT @Ana;"XOFF(2)=.7678"   !ADJUST TRACE TO START AT RIGHT
468   OUTPUT @Ana;"TRIG"            !SELECT TRIG MENU
469   OUTPUT @Ana;"HLDOFF=1"        !SET HOLD OFF TO NONE
470   OUTPUT @Ana;"MARK"            !SELECT MARK MENU
471   OUTPUT @Ana;"TRACE=2"         !SELECT TRACE 2
472   OUTPUT @Ana;"MARKER=2"        !SELECT BASELINE
473   OUTPUT @Ana;"BLINE=2"         !TURN BASE LINE ON
474   OUTPUT @Ana;"MARKER=3"        !SELECT CROSSHAIR
475   OUTPUT @Ana;"CROSS=2"         !SET CROSSHAIR MODE TO CENTER
476   OUTPUT @Ana;"TMB"             !SELECT TIMEBASE DISPLAY
477   OUTPUT @Ana;"INTEN=1"
478   OUTPUT @Ana;"LOCAL"           !GOTO LOCAL MODE
479 !
480 SUBEND
481 !********************************************************************
482 SUB Store_data(INTEGER Data(*),N)
483   COM /Nicdata/ REAL Vrange2,Period,Freq,Clkdiv
484   COM /Filename/User$,Path$,Date$,INTEGER Nr
485   COM /Flag/ INTEGER Flag,Storeflag$[15],Stackingflag$[15]
486   COM /Tmdt/Datetime$,Time$
487   COM /Zeros/Nr$[3]
488   INTEGER Resol,Types,Mrows,Ncols,Imagf
489   REAL Number,Resolution
490   DIM Name$[25]
491   Resol=16
492   Nr=Nr+1
493   PRINT Nr
494   Nr$=VAL$(Nr)
495   CALL Add_zero(Nr$)
496   Filename$="/"&Date$&Nr$
497   Ptspercycle=8192/Clkdiv     !ASSUMES 8192 POINTS PER CYCLE
498   PRINT "THE FILE IS BEING STORED AT PATH"&Path$&Filename$&".DTA"
499   PRINT ""
500   Datetime$=DATE$(TIMEDATE)
```

```
501  Time$=TIME$(TIMEDATE)
502  !
503  CREATE Path$&Filename$&".DTA",0  !CREATE A DOS FILE
504  ASSIGN @File TO Path$&Filename$&".DTA";FORMAT OFF
505  Types=0                    !0 FOR PC'S
506  Mrows=N                    !NUMBER OF ROWS IN MATRIX
507  Ncols=1                    !NUMBER OF COLUMNS IN MATRIX
508  Imagf=0                    !IMAGINARY FLAG
509  Number=N                   !REAL VERSION OF N
510  Resolution=Resol           !REAL VERSION OF RESOL
511  !
512  !     OUTPUT DATA MATRIX
513  !
514  Name$="qdata"
515  CALL Head(@File,Types,Mrows,Ncols,Imagf,Name$)
516  OUTPUT @File;Data(*)
517  !
518  !     OUTPUT VRANGE2
519  !
520  Name$="vrange2"
521  CALL Head(@File,Types,Mrows,Ncols,Imagf,Name$)
522  OUTPUT @File;Vrange2
523  !
524  !     OUTPUT FREQUENCY
525  !
526  Name$="freq"
527  CALL Head(@File,Types,Mrows,Ncols,Imagf,Name$)
528  OUTPUT @File;Freq
529  !
530  !     OUTPUT PTSPERCYCLE
531  !
532  Name$="ptspercycle"
533  CALL Head(@File,Types,Mrows,Ncols,Imagf,Name$)
534  OUTPUT @File;Ptspercycle
537  !
538  Flag=1
539 Endsub: SUBEND
540 !*******************************************************************
541 SUB Add_zero(I$)
542  COM /Zeros/Nr$
543  IF (VAL(I$)<10) AND (POS(I$,"0")=0) THEN I$="0"&I$
544  IF I$=Nr$ AND (VAL(I$)<100) THEN I$="0"&I$
545 SUBEND
546 !*******************************************************************
547 SUB Not_used
548  BEEP
549 SUBEND
550 !*******************************************************************
551 SUB Storeflag
552  COM /Flag/ INTEGER Flag,Storeflag$[15],Stackingflag$[15]
553  SELECT Storeflag$
554  CASE "STORAGE OFF"
560    Storeflag$="STORAGE ON"
570  CASE "STORAGE ON"
580    Storeflag$="STORAGE OFF"
590  END SELECT
600  Flag=1
610 SUBEND
620 !******************************************************************
630 SUB Stackingflag
631  COM /Flag/ INTEGER Flag,Storeflag$[15],Stackingflag$[15]
632  SELECT Stackingflag$
633  CASE "STACKING OFF"
634    Stackingflag$="STACKING ON"
635  CASE "STACKINGFLAG ON"
636    Stackingflag$="STACKING OFF"
637  END SELECT
638  Flag=1
640 SUBEND
650 !******************************************************************
```

```
660 SUB Dataacq(INTEGER Data(*),N)
670 !DATA OUTPUT FROM ANALOGIC 6100
680   COM /Nicdata/ REAL Vrange2,Period,Freq,Clkdiv
690   DIM A$[99],Pause_flag$[1]
700   Pause_flag$="N"
710   ON KEY 5 LABEL "READING" CALL Not_used
720   ON KEY 3 LABEL "PAUSE ASAP",2 GOSUB Asap
730   Nad=715
740   ASSIGN @Nic TO 715
750   !
760   OUTPUT Nad;"FLDDLM(3)=5"   !SET THE FIELD DELEMETER TO A COMMA
770   OUTPUT Nad;"FLDLEN(3)=10"  !SET THE FIELD LENGTH FOR NUMERIC PRECISION=10
780   OUTPUT Nad;"OMODE=1"       !SET FOR DATA TRANSFER ONLY
790   OUTPUT Nad;"FORMAT=1"      !SET FOR AN ASCII FORMAT
800   OUTPUT Nad;"LINLEN=80"     !SET LINE LENGTH TO 80
810   OUTPUT Nad;"TRGSEL=2"      !SET THE TRIGGER TO THE ARM TRIGGER
820   OUTPUT Nad;"TRGSRC=7"      !SET THE SOURCE TO BE EXTERNAL
830   OUTPUT Nad;"HLDOFF=9"      !TURN HOLD ON UNTIL ARMED
840   OUTPUT Nad;"CLRERR"        !CLEAR THE ERROR STATUS BIT
850   OUTPUT Nad;"CLRAQU"        !CLEAR THE AQUISITION COMPLETE BIT
860   OUTPUT Nad;"CLRKEY"        !CLEAR THE KEY CODE BIT
870   OUTPUT Nad;"ERRSRQ=1"      !TURN OFF ERROR STATUS BYTE
880   OUTPUT Nad;"KEYSRQ=1"      !TURN OFF KEY SERVICE REQUEST
890   OUTPUT Nad;"OUTSRQ=1"      !CLEAR BIT 7 OFF STATUS BYTE
900 Start: OUTPUT Nad;"CLRSRQ"   !CLEAR THE SERVICE REQUEST BYTE
910        OUTPUT Nad;"AQUSRQ=2" !TURN ON THE AQUISITION COMPLETE BYTE
920        OUTPUT Nad;"ARM"      !ARM THE DIGITIZER
930     REPEAT
940       OUTPUT Nad;"SRQ"       !REQUEST THE SERVICE REQUEST STATUS BYTE
950       ENTER @Nic;Service     !GET STATUS BYTE
960     UNTIL Service>=19        !WAIT TILL DATA ACQD OR ERROR
970     ON KEY 5 LABEL "TRANSFER DATA" CALL Not_used
1380       OUTPUT Nad;"BUF.B2?"     !TELL 6100 TO DUMP HEADER IN ASCII
1390       ENTER @Nic;A$            !GET THE HEADER OF CH 2
1400    Count=0
1410    I=0
1420    REPEAT                     !DECOMPOSE THE HEADER
1430      I=I+1
1440      IF A$[I;1]<>"," THEN      !NOT START OF A FIELD
1450        GOTO Cnt
1460      ELSE
1470        Count=Count+1
1480      END IF
1490      SELECT Count
1500      CASE 8
1510    Vrange2=VAL(A$[I+1;1]&A$[I+2;1]&A$[I+3;1]&A$[I+4;1]&A$[I+5;1]&A$[I+6])
1520      END SELECT
1530 Cnt: UNTIL Count=8
1540    OUTPUT Nad;"FLDDLM(3)=7"    !SET THE FIELD DELIMITER TO A LINE FEED
1550    OUTPUT Nad;"FLDLEN(3)=10"   !SET THE FIELD LENGTH FOR NUMERIC PREC.=10
1560    OUTPUT Nad;"FORMAT=4"
1570    ALLOCATE INTEGER A2(1:N)
1610    OUTPUT Nad;"SRC(2)"
1620    ENTER @Nic USING "W";A2(*)
1630    PRINT "2"
1640    OUTPUT Nad;"LOCAL"          !RESTORES LOCAL CAPABILITY TO ANALOGIC
1660    MAT Data=A2
1670    IF Pause_flag$="Y" THEN
1680      BEEP
1690      PAUSE
1700      Pause_flag$="N"
1710    END IF
1720    GOTO End
1730 Asap: Pause_flag$="Y"
1740       RETURN
1750 End:!
1760 SUBEND
1770 !******************************************************************
1780 SUB Head(@File,INTEGER Types,Mrows,Ncols,Imagf,Name$)
1810   INTEGER Length
```

```
1820   Length=LEN(Name$)+1
1830   OUTPUT @File;Types,Mrows,Ncols,Imagf,Length,Name$
1840 SUBEND
```

Appendix E

DATA REDUCTION PROGRAM

The following material and program was writtten by Rainer Moerig based on a prior Matlab version written by Ivan Getting. The data reduction program is written in Fortran and requires any computer and operating system supporting a fortran compiler.

Remarks

Q2DR.EXE

Q2DR.EXE is an executable Fortran porgram. The name Q2DR means **Q2 D**ata **R**eduction. To some extent Q2DR.EXE is "optimized" for the Q2 experiment.

Differences between Q2DR and qdr2.m (matlab)

Q2DR.EXE is based on the matlab program qdr2.m. In principle all features of the matlab program qdr2.m (including the channel1 fit) are contained in Q2DR.EXE. In addition Q2DR.EXE offers running modes (see: Q2DR.EXE running modes) in which the guesses for the fit-function of channel2 are done automatically.

Old data acquisition program QDANLMAT

The Basic program QDANLMAT was used to control the measurement. Two channels are measured and the data (2-byte integers) are transfered to the computer. QDANLMAT converts the 2-byte integers to 8-byte reals. A similar conversion (2-byte integers to 4-byte integers) is done for other stored parameters. This blows up the size of each stored data file to approximately 262,000 bytes (simply by storing the channel data as 4-byte integers (readable by matlab), would reduce the file size by a factor of 2).

Q2 requirements

Q2 only needs channel2 as 2-byte integers (= 16 bits = resolution of analogic) and a few of the many stored parameters. Therefore, QDANLMAT was modified.

New data acquisition program Q2DA

Q2DA (**Q2 D**ata **A**cquisition) is basically the same program as QDANLMAT, but only channel2 is transfered to the computer. The data are **not** converted. They are stored as they come in from the analogic as 2-byte integers. The parameters VRANGE2, FREQUENCY, and PTSPERCYCLE are stored (VRANGE2 is unnecessarily stored). Each data file has a size of approximately 33,000 bytes.

Q2DA reads the frequencies to be measured from the file Q2DA.FRE. Q2DA writes information such as date, # of first file, # of files to be reduced, 3 character identifier, sample type (20 char.), saturant (20 char.), and comments (50 char.) to the file Q2DR.INF. These information are read by Q2DR.EXE.

When the data acquisition is finished, a QUIT command in Q2DA transfers control back to DOS (necessary to run Q2DA in Q2RUN.BAT).

Q2RUN.BAT

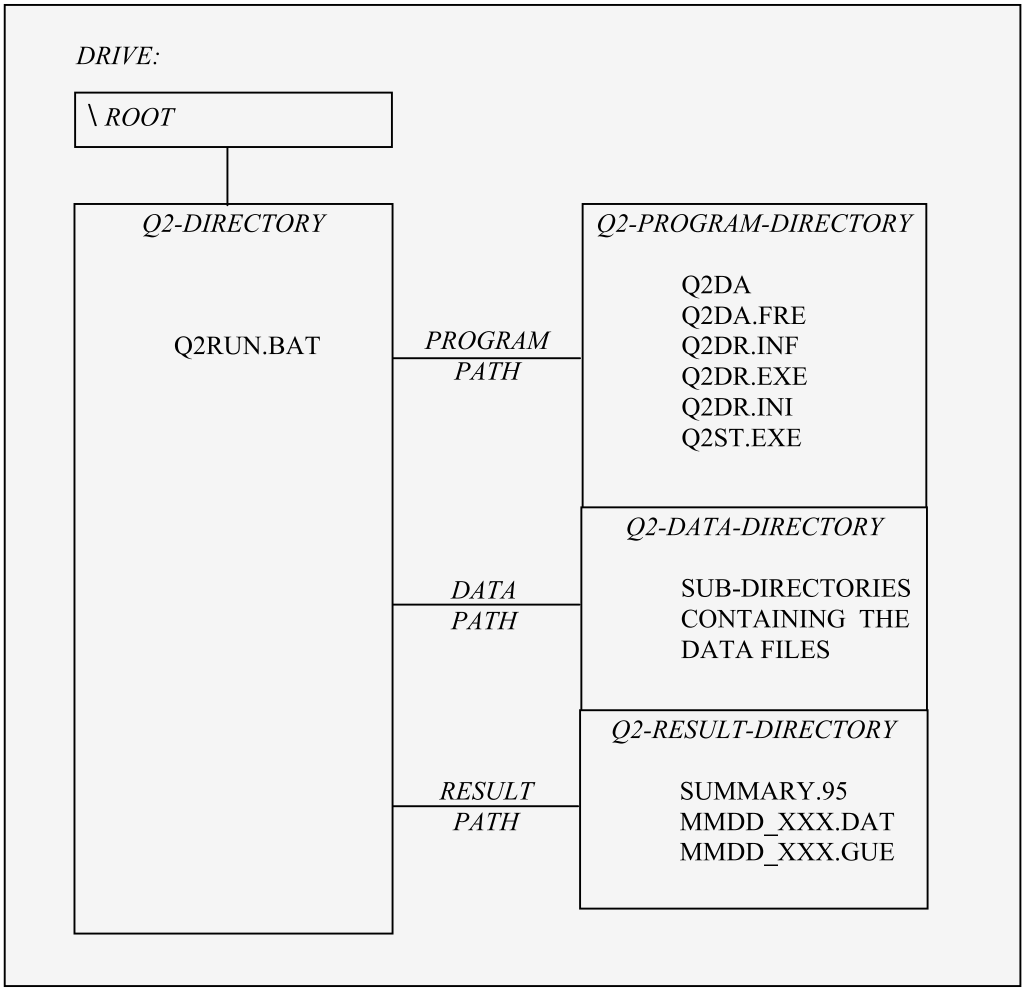
Q2RUN.BAT is the file that has to be executed to take a measurement with a subsequent data reduction.

Q2RUN.BAT calls HTB.BAT. Q2DA has to be loaded and started manually. When the measurement is started (option 4 in Q2DA), control is given to HTB executing Q2DA. At the end of the data acquisition the QUIT command returns control to DOS, that means to Q2RUN.BAT. Q2RUN.BAT now calls Q2DR.EXE. If Q2DR.EXE runs in an automatic mode (see: Q2DR.EXE running modes), the guesses and the data reduction are done automatically, otherwise the guesses have to be made manually (similar to qdr2.m).

Q2RUN.BAT can be called from any directory because the Q2-directory containing Q2RUN.BAT is given in the path of the Q2-autoexec files.

Directory structure

The Figure below shows a possible (current, May 1995) Q2-directory structure. The files in this directory structure are described below:



Q2-DIRECTORY:

Q2RUN.BAT is the batch file to run a Q2 experiment with subsequent data reduction

Q2-PROGRAM-DIRECTORY:

Q2DA is the Q2 Data Acquisition program (Basic)

Q2DA.FRE contains the number of frequencies and the frequencies to be measured. This file is read by Q2DA.

Q2DA.INF contains information such as date, file number, sample, saturant etc. This file is created by Q2DA. Q2DR.EXE reads the information to reduce the right data files and to update the summary.95 file.

Q2DR.EXE is the Q2 Data Reduction program (Fortran)

Q2DR.INI contains the Q2DR.EXE running mode, the drive, the program path, the data path, and the result path.

Q2ST.EXE is a program to change the Q2DR.EXE running mode, the drive, or the paths. Because Q2ST.EXE checks your input for validity, it is recommended to make changes in Q2DR.INI by running Q2ST.EXE.

Q2-DATA-DIRECTORY:

contains the data files in sub-directories.

Q2-RESULT-DIRECTORY:

SUMMARY.95 is a summary file containing information about the measurements done. This file is updated by Q2DR.EXE.

MMDD_XXX.DAT is the result file. MMDD is the date, XXX is a 3 character identifier. This file contains the data file number, the frequency, the parameters A1 through A7, the standard deviation, and the number of iterations done in the Levenberg-Marquardt algorithm. A -1 indicates that convergence was not met.

MMDD_XXX.GUE is the guess file. It contains the data file number, the frequency, and the guesses for the parameters A1 through A7.

Q2DR.EXE Running Modes

Q2DR.EXE can be run in three different running modes. They are manu, auto, and auma.

manu

Manu means manual. The guesses for the parameters A1 through A7 have to be made manually by fitting the fit-function to the measured signal.

auto

Auto means automatic. The guesses are done automatically. If no convergence is reached in the Levenberg-Marquardt algorithm, the guessing and the reduction procedures are repeated once again with a slightly different guessing method. Irrespectively whether or not convergence is reached, after this second run, the results are stored and the next file is processed.

auma

Auma means automatic/manual. The guesses are done first automatically. If no convergence is reached for a file, the same second run is done as in the auto mode.

If all files are correctly reduced, the program terminates. If some files are incorrectly reduced, the program waits. By pressing ENTER to continue, the program switches into the manu mode. The files incorrectly reduced in the automatic mode can be reduced now once again by making the guesses manually.

In the result file the data resulting from the unsuccessful automatic run are replaced by the data resulting from the manual run.

Q2DR source code files

Q2DR.EXE is built up by the 7 Fortran source code files:

Q2MAIN.FOR
Q2DATA.FOR
Q2GUESSA.FOR
Q2GUESSM.FOR
Q2CH1FIT.FOR
Q2CH2FIT.FOR
Q2RESULT.FOR

These Fortran source code files can be changed by using any editor. The modified files must be saved as ASCII files. The extension FOR is necessary because the Fortran compiler uses this extension to recognize a file as a source code file. To create the executable Q2DR.EXE file, the source code files must be compiled and linked. This must be done with the Microsoft Fortran Powerstation Compiler (FL32) and Linker (LINK32) because graphics- and run-time-routines called in the source code files are special tools of Microsoft Fortran.

The following table lists the content of the different source code files (compare flow diagram):

source code file	**contents**
Q2MAIN.FOR	main program Q2DR and subroutine INFO
Q2DATA.FOR	subroutines FILES, Q2DATA, and SCALE
Q2GUESSA.FOR	subroutines GUESSA and POLY2
Q2GUESSM.FOR	subroutines GUESSM, START, GRAPHICSMODE, WINDOS, MINMAX, BACKGRD, SIGFIT, DRAW, COPY, CURSOR, ITEST, WRONG, ACTUG5, and ENDPROGRAM
Q2CH1FIT.FOR	subroutines CHN1FIT and QR
Q2CH2FIT.FOR	subroutines CHN2FIT, MARQ, HESSMAT, and GAUSSJ
Q2RESULT.FOR	subroutine RESULTS

Q2MAIN.FOR

```
program q2dr
      implicit double precision (a-h,o-z)
      character fname*80,fgues*80,fsfit*80,fssfit*80,fsumm*80
      character dat*4,txt*3,sampl*20,satur*20,comm*80
      character drive*2,path1*80,path2*80,path3*80
      common/fna/fgues,fsfit,fssfit,fsumm
      common/inf/dat,txt,sampl,satur,comm
      common/dri/drive,path1,path2,path3
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *           ppc,resol
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
      common/mod/garray(126,9),parray(126,11)
      common/gra/igraph,jgraph

      call info(nr1,nr2,imode)

      modus=1
c.....Automatic mode
      if(imode.eq.1.or.imode.eq.2) then
        do 1 i=nr1,nr2
        call files(i,fname,modus)
   77   call guessa(fname)
        call chn1fit(fname,stddev1)
        call chn2fit(fname,stddev1,stddev2,indic)
        if(method.eq.1) goto 77
        call results(i,nr1,nr2,stddev1,stddev2,indic,imode)
    1   continue
      end if

      if(imode.eq.2) then
        if(nrn.eq.0) then
          stop
        else
          print*,'press ENTER to continue'
          read(*,*)
          ifile=1
          igraph=nrnot(ifile)
          jgraph=nrnot(nrn)
        end if
      end if

c.....Manual mode
      if(imode.eq.0.or.imode.eq.2) then
        if(imode.eq.0) then
          igraph=nr1
          jgraph=nr2
        end if
        modus=0
   88   if(imode.eq.2) then
          nr1=nrnot(ifile)
          nr2=nrnot(ifile)
        end if
```

```
      do 2 i=nr1,nr2
      call files(i,fname,modus)
      call guessm(fname,i)
      garray(i,1)=dble(i)
      garray(i,2)=freq
      do 2 j=1,7
      garray(i,j+2)=guess(j)
  2   continue

      if(imode.eq.2) then
        if(ifile.ne.nrn) then
          ifile=ifile+1
          goto 88
        else
          imode=-2
          ifile=1
        end if
      end if

      modus=1
 99   if(imode.eq.(-2)) then
        nr1=nrnot(ifile)
        nr2=nrnot(ifile)
      end if

      do 3 i=nr1,nr2
      call files(i,fname,modus)
      do 4 j=1,7
      guess(j)=garray(i,j+2)
  4   continue
      call chn2fit(fname,stddev1,stddev2,indic)
      call results(i,nr1,nr2,stddev1,stddev2,indic,imode)
  3   continue

      if(imode.eq.(-2)) then
        if(ifile.ne.nrn) then
          ifile=ifile+1
          goto 99
        else
          imode=-3
          call results(i,nr1,nr2,stddev1,stddev2,indic,imode)
        end if
      end if
    end if

    stop
    end
******************************************************************
    subroutine info(nr1,nr2,imode)
    implicit double precision (a-h,o-z)
    character fgues*80,fsfit*80,fssfit*80,fsumm*80
    character dat*4,txt*3,sampl*20,satur*20,comm*80,cmode*4
    character drive*2,path1*80,path2*80,path3*80,path*80,sfile*80
    common/fna/fgues,fsfit,fssfit,fsumm
    common/inf/dat,txt,sampl,satur,comm
    common/dri/drive,path1,path2,path3
    common/con/pi,pi2,w
    common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)

c.....The data acquisition program 'Q2DA' writes information
c     to the file 'Q2DR.INF'. These informations are read in.
    open(1,file='q2dr.inf')
    rewind(1)
    read(1,'(a4)')dat
    read(1,'(i3)')nr1
    read(1,'(i3)')nr
    nr2=nr1+nr-1
    read(1,'(a3)')txt
    if(imode.eq.1) write(*,'(1x,a4,1x,i3,1x,i3,1x,a3)')dat,nr1,nr2,txt
```

```
      if(txt.ne.'xxx'.and.txt.ne.'XXX') then
        read(1,'(a20)')sampl
        read(1,'(a20)')satur
        read(1,'(a80)')comm
        if(imode.eq.1) then
          write(*,'(1x,2a20)')sampl,satur
          write(*,'(1x,a80)')comm
        end if
      end if
      close(1)

c.....read MODE, DRIVE, and PATHS from file 'Q2DR.INI'
c
c.....Q2DR.EXE runs automatically (auto), manually (manu), or
c     first automatically and then manually (auma).
      open(1,file='q2dr.ini')
      rewind(1)
      read(1,'(a4)')cmode
      if(cmode.eq.'auto') imode=1
      if(cmode.eq.'manu') imode=0
      if(cmode.eq.'auma') imode=2
      if(imode.lt.0.or.imode.gt.2)
     *stop 'Error: MODE (auto, manu, or auma) is not defined (Q2DR.INI)'
c.....drive
      read(1,'(a2)')drive
c.....paths
      read(1,'(a80)')path
      path1=drive//path(1:len_trim(path))
      read(1,'(a80)')path
      path2=drive//path(1:len_trim(path))
      read(1,'(a80)')path
      path3=drive//path(1:len_trim(path))
      read(1,'(a80)')sfile
      close(1)

c.....The guesses are written to 'fgues'.
      fgues=path3(1:len_trim(path3))//dat//'_'//txt//'.gue'

c.....The parameters resulting from chn1fit are written to 'fsfit'.
c     In this Q2-version neither the channel1-fit is done nor the
c     results are stored.
      fsfit=path3(1:len_trim(path3))//'sfit.dat'

c.....The parameters resulting from chn2fit are written to 'fssfit'.
      fssfit=path3(1:len_trim(path3))//dat//'_'//txt//'.dat'

c.....The info data are written to 'fsumm'.
      fsumm=path3(1:len_trim(path3))//sfile(1:len_trim(sfile))

      pi=dble(4.)*datan(dble(1.))
      pi2=dble(2.)*pi

      method=0

      return
      end
```

Q2DATA.FOR

```
*******************************************************************
      include 'flib.fi'
*******************************************************************
      subroutine files(i,fname,modus)
      include 'flib.fd'
      integer*4 length,handle
      character fname*80,pfad*80,nr*3,ext*4
      character dat*4,txt*3,sampl*20,satur*20,comm*80
      character drive*2,path1*80,path2*80,path3*80
      common/inf/dat,txt,sampl,satur,comm
```

```
      common/dri/drive,path1,path2,path3
      record / file$info / info

      pfad=path2(1:len_trim(path2))//dat//'_\'
      ext='.dta'

      if(i.lt.10) then
        nr='00'//char(i+48)
      elseif(i.ge.10.and.i.lt.100) then
        i1=i/10
        i2=mod(i,10)
        nr='0'//char(i1+48)//char(i2+48)
      elseif(i.ge.100) then
        i1=i/100
        i2=mod(i,100)
        i3=mod(i2,10)
        i2=i2/10
        nr=char(i1+48)//char(i2+48)//char(i3+48)
      end if
      fname=pfad(1:len_trim(pfad))//dat//'_'//nr//ext

c.....Check whether data file exists
      handle=file$first
      length=getfileinfoqq(fname,info,handle)
      if (handle.eq.file$last.or.handle.eq.file$error) then
        print 10,fname
        print*,'FILE does not exist ! '
        stop
      end if
      if (info.length.eq.0) then
        print 10,fname
        print*,'File is empty ! '
        stop
      end if
   10 format(1x,a80)

      call q2data(fname,modus)

      call scale

      return
      end
*******************************************************************
      subroutine q2data(fname,modus)
      implicit double precision (a-h,o-z)
      character fname*80,vnamei(20)*1,vname*20
      integer*2 inum(10),idat
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *           ppc,resol
      common/con/pi,pi2,w

c.....CHANNEL1 and VRANGE1 (and several other unnecessary parameters)
c     are not stored by QDANLMAT
      rn=16384.d0
      vrange1=1.d0
      clkdiv=1.d0
      resol=16.d0

      if(modus.eq.1) print 90,fname(1:len_trim(fname)),'   loading '
   90 format(1x,a28,a11)

      open(99,file=fname,form='binary')
      rewind(99)

      do 1 k=1,4

      do 2 i=1,5
      read(99)inum(i)
    2 continue
```

```
      do 3 i=1,inum(5)
      read(99)vnamei(i)
    3 continue

      do 4 i=1,inum(5)-1
      vname(i:i)=vnamei(i)
    4 continue

      if(k.eq.1) then
        do 5 j=1,inum(3)
        do 5 i=1,inum(2)
        read(99)idat
        qdata(i,j+1)=dble(idat)
    5   continue
      else
        read(99)realnr
      end if

      if(k.eq.2) then
        vrange2=realnr
      elseif(k.eq.3) then
        freq=realnr
      elseif(k.eq.4) then
        ppc=realnr
      end if

    1 continue

      close(99)

      w=pi2/ppc

      return
      end
*********************************************************************
      subroutine scale
      implicit double precision (a-h,o-z)
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
      dimension scal(2)
      reso=dble(2.)**resol
      scal(1)=vrange1/reso
      scal(2)=vrange2/reso
      do 1 i=1,int(rn)
      do 1 j=1,2
      qdata(i,j)=qdata(i,j)*scal(j)
    1 continue
      return
      end
```

Q2GUESSA.FOR

```
*********************************************************************
      subroutine guessa(fname)
      implicit double precision (a-h,o-z)
      character fname*80
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(128)
      dimension turn(256,2),pturn(3,2),delta(2),it1(2),fit1(2)
      dimension izero(16384)

      print 90,fname,'   guessing'
   90 format('+',a28,a11)

c.....Maximum/Minimum
      qmax=-1.d20
      qmin=1.d20
      do 1 i=1,int(rn)
```

```
      qmax=dmax1(qmax,qdata(i,2))
      qmin=dmin1(qmin,qdata(i,2))
    1 continue

c.....Guesses g3 and g4
      guess(3)=(qmax-qmin)/2.d0
      guess(4)=(qmax+qmin)/2.d0

c.....Arcsin of scaled signal
      do 2 i=1,int(rn)
      arg=(qdata(i,2)-guess(4))/guess(3)
      if(dabs(arg).gt.1.d0) arg=dsign(1.d0,arg)
      qdata(i,3)=dasin(arg)
    2 continue

c.....Quadratic function fit to arcsin of signal
      call poly2(kiend,m)

c.....Zeros in derivative
      prodold=1.d0
      nzero=0
      do 3 i=1,kiend-1
      prodnew=qdata(i,6)*qdata(i+1,6)
      if(i.gt.1) prodold=qdata(i-1,6)*qdata(i,6)
      if(prodnew.le.0.d0.and.prodold.gt.0.d0) then
        nzero=nzero+1
        if(dabs(qdata(i,6)).le.dabs(qdata(i+1,6))) then
          izero(nzero)=i
          if(izero(nzero-1).eq.i) izero(nzero)=i+1
        else
          izero(nzero)=i+1
          if(izero(nzero-1).eq.(i+1)) izero(nzero)=i+2
        end if
      end if
    3 continue

c.....Turn around points and guess g2
      sg2=0.d0
      nturn=2*int(rn/ppc)
      do 4 i=1,nturn
      iturn=(2*i-1)*int(ppc)/4
      difmin=1.d20
      do 5 j=1,nzero
      qdata(j,8)=dabs(dble(izero(j)+m-iturn))
      difmin=dmin1(difmin,qdata(j,8))
    5 continue
      do 6 j=1,nzero
      if(dabs(difmin-qdata(j,8)).le.1.d-20) then
        if(j.gt.2) then
          if(qdata(izero(j-1),5)*qdata(izero(j+1),5).gt.0.d0) then
            turn(i,1)=qdata(izero(j),4)
            turn(i,2)=qdata(izero(j),5)
          elseif(qdata(izero(j+1),5)*qdata(izero(j+2),5).gt.0.d0) then
            turn(i,1)=qdata(izero(j+1),4)
            turn(i,2)=qdata(izero(j+1),5)
          elseif(qdata(izero(j-1),5)*qdata(izero(j-2),5).gt.0.d0) then
            turn(i,1)=qdata(izero(j-1),4)
            turn(i,2)=qdata(izero(j-1),5)
          else
            turn(i,1)=qdata(izero(j),4)
            turn(i,2)=qdata(izero(j),5)
          end if
        else
          turn(i,1)=qdata(izero(j),4)
          turn(i,2)=qdata(izero(j),5)
        end if
      end if
    6 continue
      sg2=sg2+dble(iturn)-turn(i,1)
    4 continue
```

```
      guess(2)=pi2*sg2/dble(nturn)/ppc

c.....first part of signal
      istart=2*int(turn(1,1))-1-m
      if(-qdata(istart,6).ge.dble(0.)) then
        pf=1.d0
      else
        pf=-1.d0
      end if
      slope=pf

c.....unfold signal
      vorz=1.d0
      ja=1
      st=0.d0
      do 7 i=1,nturn
      do 8 j=ja,nzero
      st=st+pf
      if(dabs(turn(i,1)-qdata(izero(j),4)).lt.1.d-20) then
        if(i.eq.1) then
          if(pf*qdata(izero(1),5).gt.0.d0) st=st-pf
          turn(i,2)=st*pi+(-1)**st*turn(i,2)
          if(turn(i,2).lt.0.d0) then
            vorz=-1.d0
            turn(i,2)=vorz*turn(i,2)
          end if
          pf=-pf
          ja=j
          goto 7
        else
          st=st-dble(2.)*pf
          turn(i,2)=st*pi+(-1)**st*turn(i,2)
          turn(i,2)=vorz*turn(i,2)
          pf=-pf
          ja=j
          goto 7
        end if
      end if
    8 continue
    7 continue

c.....check for 2pi-jumps
      frac=dble(.75)
      if(method.eq.0) then
        do 30 i=3,nturn-2
        hope=dble(2.)*turn(i,2)-turn(i-2,2)
        diff=hope-turn(i+2,2)
        if(dabs(diff).gt.frac*pi2) then
          idiff=int(diff/pi2)
          if(idiff.eq.0) idiff=int(dsign(dble(1.),diff))
          do 31 j=i+2,nturn
          turn(j,2)=turn(j,2)+dble(idiff)*pi2
   31     continue
        end if
   30   continue
      else
        diff=turn(3,2)-turn(1,2)
        if(dabs(diff).gt.frac*pi2) then
          difsum=0.d0
          do 23 i=3,5,2
          difsum=difsum+dabs(turn(i+2,2)-turn(i,2))
   23     continue
          difsum=difsum/dble(2.)
          if(difsum.le.dabs(diff)) then
            turn(1,2)=turn(1,2)+dsign(1.d0,diff)*pi2
          else
            do 9 i=3,nturn,2
            turn(i,2)=turn(i,2)-dsign(1.d0,diff)*pi2
    9       continue
          end if
```

```
      end if
      diff=turn(4,2)-turn(2,2)
      if(dabs(diff).gt.frac*pi2) then
        difsum=0.d0
        do 24 i=4,6,2
        difsum=difsum+dabs(turn(i+2,2)-turn(i,2))
 24     continue
        difsum=difsum/dble(2.)
        if(difsum.le.dabs(diff)) then
          turn(2,2)=turn(2,2)+dsign(1.d0,diff)*pi2
        else
          do 10 i=4,nturn,2
          turn(i,2)=turn(i,2)-dsign(1.d0,diff)*pi2
 10       continue
        end if
      end if

      do 11 k=0,1
      do 11 i=3+k,nturn-2,2
      np=0
      sx=0.d0
      sx2=0.d0
      sy=0.d0
      sxy=0.d0
      do 12 j=1+k,i,2
      np=np+1
      sx=sx+turn(j,1)
      sx2=sx2+turn(j,1)*turn(j,1)
      sy=sy+turn(j,2)
      sxy=sxy+turn(j,1)*turn(j,2)
 12   continue
      den=np*sx2-sx*sx

      if(dabs(den).lt.1.d-300) goto11

      sl=(np*sxy-sx*sy)/den
      bl=(sx2*sy-sx*sxy)/den
      hope=sl*turn(i+2,1)+bl
      diff=hope-turn(i+2,2)
      if(dabs(diff).ge.frac*pi2) then
        idiff=int(diff/pi2)
        if(idiff.eq.0) idiff=int(dsign(dble(1.),diff))
        do 13 j=i+2,nturn,2
        turn(j,2)=turn(j,2)+dble(idiff)*pi2
 13     continue
      end if
 11   continue

c.....may be there is a 2pi-jump between each two turn around points
      sum=0.d0
      do 17 i=3,4
      do 17 j=i,nturn,2
      sum=sum+dabs(turn(j,2)-turn(j-2,2))
 17   continue
      sum=sum/(nturn-2)
      if(sum.gt.frac*pi2) then
        dif1=turn(3,2)-turn(1,2)
        vorz1=dsign(1.d0,dif1)
        dif2=turn(4,2)-turn(2,2)
        vorz2=dsign(1.d0,dif2)
        vorz=vorz1*vorz2
        if(dabs(dif1).gt.frac*pi2.and.dabs(dif2).gt.frac*pi2.and.
     *                         vorz.gt.0.d0) then
          do 18 i=3,4
          do 18 j=i,nturn,2
          turn(j,2)=turn(j,2)-vorz1*dble(int((j-1)/2))*pi2
 18       continue
        end if
      end if
    end if
```

```
c.....quadratic function fit to unfolded turn around points
      dn=dble(nturn/2)
      do 19 i=1,2
      sx=0.d0
      sx2=0.d0
      sx3=0.d0
      sx4=0.d0
      sy=0.d0
      syx=0.d0
      syx2=0.d0
      do 20 j=i,nturn,2
      x=turn(j,1)
      y=turn(j,2)
      x2=x*x
      x3=x2*x
      x4=x2*x2
      sx=sx+x
      sx2=sx2+x2
      sx3=sx3+x3
      sx4=sx4+x4
      sy=sy+y
      syx=syx+y*x
      syx2=syx2+y*x2
   20 continue
      det=dn*(sx4*sx2-sx3*sx3)-sx4*sx*sx-sx2*sx2*sx2+
     *                            dble(2.)*sx3*sx2*sx
      cdet=syx2*(sx3*sx-sx2*sx2)+syx*(sx3*sx2-sx4*sx)+
     *                           sy*(sx4*sx2-sx3*sx3)
      pturn(1,i)=cdet/det
      cdet=syx2*(sx2*sx-dn*sx3)+syx*(dn*sx4-sx2*sx2)-
     *                           sy*(sx4*sx-sx3*sx2)
      pturn(2,i)=cdet/det
      cdet=syx2*(dn*sx2-sx*sx)+syx*(sx2*sx-dn*sx3)+
     *                           sy*(sx3*sx-sx2*sx2)
      pturn(3,i)=cdet/det
   19 continue

c.....guesses g1, g6 and g7
      guess(1)=(pturn(1,1)-pturn(1,2))/pi2
      guess(6)=(pturn(2,1)+pturn(2,2))/dble(2.)
      guess(7)=(pturn(3,1)+pturn(3,2))/dble(2.)

c.....guess g5
      pia1sin=pi*guess(1)*dsin(w+guess(2))+guess(6)+guess(7)
      if(slope.ge.0.d0) then
        arg=qdata(1,3)-pia1sin
      else
        arg=dsign(dble(1.),qdata(1,3))*pi-qdata(1,3)-pia1sin
      end if
      guess(5)=arg

c.....correction of g5 (fitting first or second turn around point)
      do 21 j=1,2
      delta(j)=0.d0
      do 22 i=j,nturn,2
      iturn=(2*i-1)*int(ppc)/4-int(guess(2)*ppc/pi2)
      di=dble(iturn)
      fit=guess(3)*dsin(pi*guess(1)*dsin(w*di+guess(2))+guess(5)+
     *                 guess(6)*di+guess(7)*di*di)+guess(4)
      if(i.eq.1.or.i.eq.2) then
        it1(i)=iturn
        fit1(i)=fit
      end if
      delta(j)=delta(j)+dabs(qdata(iturn,2)-fit)
   22 continue
   21 continue
      delg5=dmax1(delta(1),delta(2))

      if(dabs(delg5-delta(1)).lt.1.d-20) then
```

```
        iwhere=1
      else
        iwhere=2
      end if
      arg0=(qdata(it1(iwhere),2)-guess(4))/guess(3)
      if(dabs(arg0).gt.1.d0) arg0=dsign(1.d0,arg0)
      phi0=dasin(arg0)
      arg1=(fit1(iwhere)-guess(4))/guess(3)
      if(arg1.gt.1.d0) arg1=dsign(1.d0,arg1)
      phi1=dasin(arg1)
      delg5=dabs(phi0-phi1)

      dit1=dble(it1(iwhere))
      argu1=pi*guess(1)*dsin(w*dit1+guess(2))+guess(5)+
     *          guess(6)*dit1+guess(7)*dit1*dit1
      argu2=w*pi*guess(1)*dcos(w*dit1+guess(2))+guess(6)+
     *                   dble(2.)*guess(7)*dit1
      argu3=w*w*pi*guess(1)*dsin(w*dit1+guess(2))-dble(2.)*guess(7)
      fitcur=-guess(3)*(dsin(argu1)*argu2*argu2+dcos(argu1)*argu3)

      vorz=(-1.d0)**(iwhere+1)
      if(qdata(it1(iwhere)-m,7).ge.0.d0.and.fitcur.ge.0.d0) then
        if(qdata(it1(iwhere),2).gt.fit1(iwhere)) then
          signum=-vorz
        else
          signum=vorz
        end if
      elseif(qdata(it1(iwhere)-m,7).le.0.d0.and.fitcur.le.0.d0) then
        if(qdata(it1(iwhere),2).gt.fit1(iwhere)) then
          signum=vorz
        else
          signum=-vorz
        end if
      elseif(qdata(it1(iwhere)-m,7).ge.0.d0.and.fitcur.le.0.d0) then
        delg5=pi-phi0-phi1
        signum=vorz
      elseif(qdata(it1(iwhere)-m,7).le.0.d0.and.fitcur.ge.0.d0) then
        delg5=pi-phi0-phi1
        signum=-vorz
      end if
      guess(5)=guess(5)+signum*delg5

      return
      end
*******************************************************************
      subroutine poly2(kiend,m)
      implicit double precision (a-h,o-z)
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *       ppc,resol

      n=int(ppc/16.d0)+1
      m=(n-1)/2
      kiend=int(rn)-2*m

      sx2=0.d0
      sx4=0.d0
      do 1 k=1,m
      x2=dble(k)*dble(k)
      sx2=sx2+x2
      sx4=sx4+x2*x2
    1 continue
      sx2=dble(2.)*sx2
      sx4=dble(2.)*sx4
      sn422=dble(n)*sx4-sx2*sx2

      do 2 k=m+1,int(rn)-m
      sy=0.d0
      syx=0.d0
      syx2=0.d0
```

```
      do 3 l=-m,m
      li=l+k
      sy=sy+qdata(li,3)
      syx=syx+dble(l)*qdata(li,3)
      syx2=syx2+dble(l)*dble(l)*qdata(li,3)
    3 continue

      ki=k-m
      qdata(ki,4)=dble(k)
      qdata(ki,5)=(sx4*sy-sx2*syx2)/sn422
      qdata(ki,6)=syx/sx2
      qdata(ki,7)=(dble(n)*syx2-sx2*sy)/sn422
    2 continue

      return
      end
```

Q2GUESSM.FOR

```
***********************************************************************
      include 'fgraph.fi'
***********************************************************************
      subroutine guessm(fname,i)
      implicit double precision (a-h,o-z)
      include 'fgraph.fd'
      integer*1 input
      integer*2 allo,run,color
      character fname*80
      common/mima/xmin,xmax,ymin,ymax,ch2min,ch2max
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
      common/para/ixa,ixb,icycle,mcycle
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *           ppc,resol
      common/gra/igraph,jgraph

      allo=0
      run=0
      call start(run)

      if(i.eq.igraph) then
        call graphicsmode()
        call backgrd
      end if
      call windos

      input=0
      do 1 while(input.ne.9)

      if(input.eq.0) then
        run=run+1
        call clearscreen($gclearscreen)
        call sigfit
        if(allo.eq.0) then
          color=12
          call draw(2,color,fname)
          call copy(allo,run)
        else
          call copy(allo,run)
        end if
        color=2
        call draw(4,color,fname)
      end if

      call cursor(input)

      if(input.eq.8) then
        call start(run)
        call windos
```

```
        allo=0
      end if

    1 continue

      if(input.eq.9) call copy(allo,0)

      if(i.eq.jgraph) call endprogram()

c     0:black  1:blue   2:green  3:cyan   4:red    5:magenta  6:brown   7:white
c     8:gray   9:light  10:light  11:light  12:light  13:light   14:yellow  15:bright
c              blue     green     cyan     red      magenta              white
      return
      end
*******************************************************************
      subroutine copy(allo,run)
      implicit double precision (a-h,o-z)
      include 'fgraph.fd'
      integer*1 buffer[allocatable](:)
      integer*2 error,status,allo,run
      integer*4 imsize
      common/mima/xmin,xmax,ymin,ymax,ch2min,ch2max
      if(run.eq.0) then
        deallocate(buffer,stat=error)
        return
      end if
      if(allo.eq.0) then
        if(run.gt.1) deallocate(buffer,stat=error)
        imsize=imagesize_w(xmin,ymax,xmax,ymin)
        allocate(buffer(imsize),stat=error)
        if(error.ne.0) then
          status=setvideomode($defaultmode)
          stop 'Error: insufficient memory'
        end if
        call getimage_w(xmin,ymax,xmax,ymin,buffer)
        allo=1
      else
        call putimage_w(xmin,ymax,buffer,$gpset)
      end if
      return
      end
*******************************************************************
      subroutine cursor(input)
      implicit double precision (a-h,o-z)
      include 'fgraph.fd'
      character t0*51,t1*51,t2*15,text(8)*3,ti*3,str*14,empty*14
      character in*10,intin*3
      integer*1     input
      integer*2     dummy2,ipos(8,2),row,col,rowstep,colstep
      record/rccoord/ cp
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
      common/para/ixa,ixb,icycle,mcycle
      data ((ipos(i,j),i=1,8),j=1,2) / 44,45,44,45,44,45,44,45,
     *                         4,4,30,30,56,56,82,82 /
      data text / 'A1=','A2=','A3=','A4=','A5=','A6=','A7=','#C=' /
      empty='              '
      colstep=3

      call settextposition(40,4,cp)
      dummy2=settextcolor(4)
      t0='0: display,          1-7: parameters,          '
      t1='8: # of cycles (#C),          9: next file      '
      call outtext(t0)
      call settextposition(40,55,cp)
      call outtext(t1)

      do 1 i=1,8
      row=ipos(i,1)
      col=ipos(i,2)
```

```
   call settextposition(row,col,cp)
   ti=text(i)
   call outtext(ti)

   if(i.lt.8) then
     write(str,'(f14.10)')guess(i)
   else
     write(str,'(i5)')icycle
   end if
   col=col+colstep
   call settextposition(row,col,cp)
   call outtext(str)
 1 continue

   call settextposition(42,4,cp)
   dummy2=settextcolor(9)
   t2='INPUT (0-9) :'
   call outtext(t2)
   in='          '
   do 2 while(len_trim(in).gt.1.or.ichar(in).lt.48.or.
  *                                   ichar(in).gt.57)
     call settextposition(42,19,cp)
     call outtext(empty)
     call settextposition(42,19,cp)
     read(*,'(a10)')in
     if(len_trim(in).gt.1.or.ichar(in).lt.48.or.
  *                                 ichar(in).gt.57) then
       row=42
       col=19
       call wrong(row,col)
     end if
 2 continue
   input=ichar(in)-48

   if(input.gt.0.and.input.lt.9) then
     call settextposition(42,4,cp)
     dummy2=settextcolor(4)
     call outtext(t2)

     row=ipos(input,1)
     col=ipos(input,2)
     dummy2=settextcolor(9)
     call settextposition(row,col,cp)
     ti=text(input)
     call outtext(ti)

     col=col+colstep
     call settextposition(row,col,cp)
     call outtext(empty)

     rowstep=(-1)**input
     row=row+rowstep
     call settextposition(row,col,cp)
     if(input.lt.8) then
       write(str,'(f14.10)')guess(input)
     else
       write(str,'(i5)')icycle
     end if
     call outtext(str)
     row=row-rowstep

     col=col+2
     call settextposition(row,col,cp)
     if(input.lt.8) then
       read(*,*)guess(input)
       if(input.ne.5) call actug5
     else
       icycle=-1
       do 3 while(icycle.le.0.or.icycle.gt.mcycle)
         read(*,'(a3)')intin
```

```
         call itest(intin,icycle)
         call settextposition(row,col,cp)
         if(icycle.le.0.or.icycle.gt.mcycle) call wrong(row,col)
 3      continue
      end if

      row=row+rowstep
      col=col-2
      call settextposition(row,col,cp)
      call outtext(empty)

    end if

    return
    end
**********************************************************************
    include 'flib.fi'
**********************************************************************
    subroutine wrong(row,col)
    include 'fgraph.fd'
    include 'flib.fd'
    integer*2       row,col
    integer*4       duration,frequency
    record/rccoord/ cp
    duration=500
    frequency=440
    do 1 i=1,6
    frequency=frequency+(i-1)*100
    call beepqq(duration,frequency)
  1 continue
    call settextposition(row,col,cp)
    call outtext('invalid input, repeat!')
    duration=1000
    call sleepqq(duration)
    call settextposition(row,col,cp)
    call outtext('                      ')
    call settextposition(row,col,cp)
    return
    end
**********************************************************************
    subroutine itest(in,ires)
    character in*3
    dimension ii(3)
    ind=0
    iend=len_trim(in)
    do 1 i=1,iend
    if(ichar(in(i:i)).lt.48.or.
   *   ichar(in(i:i)).gt.57) ind=1
  1 continue
    if(ind.eq.0) then
      ires=0
      do 2 i=1,iend
      ii(i)=ichar(in(i:i))-48
      ires=ires+ii(i)*10**(iend-i)
  2    continue
    end if
    return
    end
**********************************************************************
    subroutine actug5
    implicit double precision (a-h,o-z)
    common/con/pi,pi2,w
    common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
    common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
   *          ppc,resol
    pia1sin=pi*guess(1)*dsin(w+guess(2))+guess(6)+guess(7)
    arg=(qdata(1,2)-guess(4))/guess(3)
    if(dabs(arg).gt.1.d0) arg=dsign(1.d0,arg)
    arg=dasin(arg)
    slope=qdata(2,2)-qdata(1,2)
```

```
      if(slope.ge.0.d0) then
        arg=arg-pia1sin
      else
        arg=dsign(1.d0,arg)*pi-arg-pia1sin
      end if
      guess(5)=arg
      return
      end
*********************************************************************
      subroutine draw(nr,color,fname)
      implicit double precision (a-h,o-z)
      include 'fgraph.fd'
      character*9 strx,stry
      character fname*80
      integer*2 status,color,colo
      record/rccoord/ cp
      record/wxycoord/wxy
      common/mima/xmin,xmax,ymin,ymax,ch2min,ch2max
      common/para/ixa,ixb,icycle,mcycle
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol

      if(color.eq.12) then
        colo=7
        status=setcolor(colo)
        status=rectangle_w($gborder,xmin,ymax,xmax,ymin)
        status=setcolor(color)
      end if

      if(color.eq.2) then
        colo=7
        status=settextcolor(colo)
        call settextposition(3,3,cp)
        write(strx,'(e9.3)')xmin
        write(stry,'(e9.3)')ymax
        call outtext('('//strx//','//stry//')')
        call settextposition(37,98,cp)
        write(strx,'(e9.3)')xmax
        write(stry,'(e9.3)')ymin
        call outtext('('//strx//','//stry//')')
        call settextposition(3,91,cp)
        call outtext(fname(1:len_trim(fname)))
      end if

c.....nr=2 is measured signal, nr=4 is fit-function
c.....if you want to look at the measured signal only, activate the if statement below.
c.....The initial number of cycles displayed is icycle2 (subroutine start).
c     if(nr.eq.2) then
        status=setcolor(color)
        call moveto_w(qdata(ixa,3),qdata(ixa,nr),wxy)
c.....nrdisp is the displayed number of points per cycle for the fit-function
        nrdisp=128
        istep=1
        if(color.eq.2) istep=ixb/icycle/nrdisp
        do 1 i=ixa,ixb,istep
        status=lineto_w(qdata(i,3),qdata(i,nr))
    1   continue
c     end if

      return
      end
*********************************************************************
      subroutine sigfit
      implicit double precision (a-h,o-z)
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
      common/para/ixa,ixb,icycle,mcycle
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
```

```
      do 1 i=ixa,ixb
      di=dble(i)
      qdata(i,4)=guess(3)*dsin(pi*guess(1)*dsin(w*di+guess(2))+guess(5)+
     *                    guess(6)*di+guess(7)*di*di)+guess(4)
    1 continue
      return
      end
**********************************************************************
      subroutine start(run)
      implicit double precision (a-h,o-z)
      integer*2 run
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
      common/mima/xmin,xmax,ymin,ymax,ch2min,ch2max
      common/para/ixa,ixb,icycle,mcycle
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *        ppc,resol

      if(run.eq.0) then
        icycle1=1
        icycle2=2
        icycle=icycle2-(icycle1-1)
        ixa=(icycle1-1)*int(ppc)+1
        ixb=icycle2*int(ppc)
        mcycle=int(rn/ppc)
        ch2min=1.d300
        ch2max=-1.d300
        do 1 i=1,int(rn)
        qdata(i,3)=dble(i)
        ch2min=dmin1(ch2min,qdata(i,2))
        ch2max=dmax1(ch2max,qdata(i,2))
    1   continue
        guess(1)=3.d-6
        guess(2)=-.09d0
        guess(3)=(ch2max-ch2min)/2.d0
        guess(4)=(ch2max+ch2min)/2.d0
        guess(6)=0.d0
        guess(7)=0.d0
        pia1sin=pi*guess(1)*dsin(w+guess(2))+guess(6)+guess(7)
        arg=(qdata(1,2)-guess(4))/guess(3)
        if(dabs(arg).gt.1.d0) arg=dsign(1.d0,arg)
        arg=dasin(arg)
        slope=qdata(2,2)-qdata(1,2)
        if(slope.ge.0.d0) then
          arg=arg-pia1sin
        else
          arg=dsign(1.d0,arg)*pi-arg-pia1sin
        end if
        guess(5)=arg
      else
        icycle2=icycle-(icycle1-1)
        ixa=(icycle1-1)*int(ppc)+1
        ixb=icycle2*int(ppc)
      end if
      return
      end
**********************************************************************
      subroutine backgrd
      include 'fgraph.fd'
      integer*2 r,g,b
      integer*4 color,oldbgd
      rgb(r,g,b)=(#3f3f3f.and.(r.or.ishft(g,8).or.ishft(b,16)))
      color=rgb(63,63,63)
      oldbgd=setbkcolor(color)
      return
      end
**********************************************************************
      subroutine windos
      implicit double precision (a-h,o-z)
      include 'fgraph.fd'
```

```
      logical*2 switch/.TRUE./
      integer*2 status,maxx,maxy
      common/mima/xmin,xmax,ymin,ymax,ch2min,ch2max
      common maxx,maxy

      call minmax
      call setviewport(50,48,maxx-50,3*(maxy+1)/4-2)
      status=setwindow(switch,xmin,ymin,xmax,ymax)

      call settextwindow(1,5,60,130)

      return
      end
**********************************************************************
      subroutine minmax
      implicit double precision (a-h,o-z)
      common/mima/xmin,xmax,ymin,ymax,ch2min,ch2max
      common/para/ixa,ixb,icycle,mcycle
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol

      xmin=dble(ixa)
      xmax=dble(ixb)
      ymin=1.d300
      ymax=-1.d300
      do 1 i=ixa,ixb
      ymin=dmin1(ymin,qdata(i,2))
      ymax=dmax1(ymax,qdata(i,2))
    1 continue
      vorz=dsign(1.d0,ymin)
      ymin=(1.d0-vorz*5.d-2)*ymin
      vorz=dsign(1.d0,ymax)
      ymax=(1.d0+vorz*5.d-2)*ymax
      return
      end
**********************************************************************
      subroutine graphicsmode()
      include 'fgraph.fd'
      integer*2          modestatus,maxx,maxy
      record/videoconfig/ myscreen
      common             maxx,maxy
c
c     set highest resolution graphics mode
c
      modestatus = setvideomode( $xres256color )
      if(modestatus.eq.0) stop 'Error: cannot set graphics mode'
c
c     determine the minimum and maximum dimensions
c
      call getvideoconfig( myscreen )
      maxx = myscreen.numxpixels - 1
      maxy = myscreen.numypixels - 1
      return
      end
**********************************************************************
      subroutine endprogram()
c
c     ENDPROGRAM resets the screen before returning
c
      include 'fgraph.fd'
      integer*2 dummy
      dummy = setvideomode( $defaultmode )
      return
      end
```

Q2CH1FIT.FOR

```
******************************************************************
      subroutine chn1fit(fname,stddev1)
```

```
      implicit double precision (a-h,o-z)
      character fname*80
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *           ppc,resol
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(128)
      dimension x(5)

c.....Chn1fit is NOT done !
      print 10,fname,'   fitting '
   10 format('+',a28,a11)
      goto 20
c.....Chn1fit is NOT done !

      n=int(rn)
      m=5
      do 1 i=1,n
      xi=dble(i)
      wx=w*xi
      qdata(i,3)=dsin(wx)
      qdata(i,4)=dcos(wx)
      qdata(i,5)=1.d0
      qdata(i,6)=xi
      qdata(i,7)=xi*xi
      qdata(i,8)=qdata(i,1)
    1 continue

      call qr(m,n,x)

      sfit(1)=dsqrt(x(1)*x(1)+x(2)*x(2))
      if(x(1).lt.0.d0) sfit(1)=-sfit(1)
      sfit(2)=datan(x(2)/x(1))
      sfit(3)=x(3)
      sfit(4)=x(4)
      sfit(5)=x(5)

      sum=0.d0
      do 2 i=1,n
      xi=dble(i)
      fit=sfit(1)*dsin(w*xi+sfit(2))+sfit(3)+sfit(4)*xi+sfit(5)*xi*xi
      resid1=qdata(i,1)-fit
      sum=sum+resid1*resid1
    2 continue
      stddev1=dsqrt(sum/(n-5))

      print 11,fname,'   fitting  stddev1:',stddev1
   11 format('+',a28,a20,f7.4)

   20 return
      end
*****************************************************************
      subroutine qr(n,m,x)

c     n=number of unknowns x(i)
c     m=number of equations, m>=n
c     qdata(m,n+2)=model matrix
c     qdata(m,1)=vector of right side, original y(i)
c     x(n)=solution of qdata(m,n+2)*x(n)=qdata(m,1)

      implicit double precision (a-h,o-z)
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *           ppc,resol
      dimension x(n),p(5)

      do 1 k=1,n
      p(k)=dble(k)
    1 continue
      do 2 k=1,n
        r0=0.d0
        r1=dble(k)
```

```
      if(k+1.le.n) then
        do 3 i=1,m
        r0=r0+qdata(i,int(p(k))+2)*qdata(i,int(p(k))+2)
 3      continue
        r2=r0
        do 4 j=k+1,n
          r0=0.d0
          do 5 i=1,m
          r0=r0+qdata(i,int(p(j))+2)*qdata(i,int(p(j))+2)
 5        continue
          if(r0.le.r2) then
            r1=dble(j)
            r2=r0
          end if
 4      continue
      end if
      r0=p(k)
      p(k)=p(int(r1))
      p(int(r1))=r0
      do 6 i=1,m
      qdata(i,9)=qdata(i,int(p(k))+2)
 6    continue
      r1=0.d0
      do 7 i=1,m
      r1=r1+qdata(i,9)*qdata(i,9)
 7    continue
      if(dabs(r1).lt.1.d-50) then
        print*,'execution stopped'
        return
      end if
      qdata(int(p(k)),int(p(k))+2)=1.d0
      r0=0.d0
      do 8 i=1,m
      r0=r0+qdata(i,9)*qdata(i,8)
 8    continue
      x(int(p(k)))=r0/r1
      do 9 i=1,m
      qdata(i,8)=qdata(i,8)-x(int(p(k)))*qdata(i,9)
 9    continue
      if(k+1.le.n) then
        do 10 j=k+1,n
          r0=0.d0
          do 11 i=1,m
          r0=r0+qdata(i,9)*qdata(i,int(p(j))+2)
11        continue
          qdata(int(p(j)),int(p(k))+2)=r0/r1
          do 12 i=1,m
          qdata(i,int(p(j))+2)=qdata(i,int(p(j))+2)-
     *               qdata(int(p(j)),int(p(k))+2)*qdata(i,9)
12        continue
10      continue
      end if
 2 continue
    do 13 i=n,1,-1
      r0=0.d0
      if(i+1.le.n) then
        do 14 k=i+1,n
        r0=r0+qdata(int(p(k)),int(p(i))+2)*x(int(p(k)))
14      continue
        x(int(p(i)))=x(int(p(i)))-r0
      end if
13 continue

    return
    end
```

Q2CH2FIT.FOR

```
*****************************************************************
    subroutine chn2fit(fname,stddev1,stddev2,indic)
```

```
      implicit integer*4 (i-n)
      implicit double precision (a-h,o-z)
      character fname*80
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
      common/con/pi,pi2,w
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(128)
      dimension a(7),deltaa(7)

      print 10,fname,'   fitting '
   10 format('+',a28,a11)

      icount=0
      dlamb=1.d-3

      m=7
      n=int(rn)

      do 1 i=1,m
      a(i)=guess(i)
      deltaa(i)=dble(1.)
    1 continue

c.....Starting with a large ISTEP and reducing it while approaching convergence,
c     could speed up chn2fit. ISTEP is introduced but NOT changed during the run.
c      istep=int(ppc)/16
      istep=1
      istop=0
      do 2 while (istop.eq.0)
        call marq(a,deltaa,m,dlamb,istep)
        icount=icount+1
        indi=0
        if(dabs(deltaa(2)).lt.1.d-6.and.
     *                       dabs(deltaa(1)/a(1)).lt.1.d-4) then
          istep=1
          istop=1
          dlamb=dble(0.)
          call marq(a,deltaa,m,dlamb,istep)
        elseif(icount.ge.8) then
          istep=1
          istop=1
          indi=-1
          dlamb=dble(0.)
          call marq(a,deltaa,m,dlamb,istep)
        end if
    2 continue

      sum=0.d0
      do 3 i=1,n
      x=dble(i)
      sinarg=pi*a(1)*dsin(w*x+a(2))+a(5)+a(6)*x+a(7)*x*x
      qdata(i,1)=a(3)*dsin(sinarg)+a(4)
      resid2=qdata(i,2)-qdata(i,1)
      sum=sum+resid2*resid2
    3 continue
      stddev2=dsqrt(sum/(n-7))

      do 4 i=1,m
      ssfit(i)=a(i)
    4 continue

      if(icount.lt.8) then
      print 20,fname,'   fitted     ampl.:',ssfit(1),'   phase:',
     *                                    ssfit(2),icount
      indic=icount
      method=0
      else
      print 20,fname,' NOT fitted   ampl.:',ssfit(1),'   phase:',
     *                                    ssfit(2),indi
      indic=indi
```

```
      method=method+1
      end if
   20 format('+',a28,a20,f7.3,a9,f7.4,i3)

      return
      end
*****************************************************************
      subroutine marq(a,deltaa,m,dlamb,istep)
      implicit integer*4 (i-n)
      implicit double precision (a-h,o-z)
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
      common/con/pi,pi2,w
      dimension a(m),deltaa(m)
      dimension beta(7),hmat(7,7)

      n=int(rn)
      istop=0
      call hessmat(a,beta,hmat,m,dlamb,n,istep)
      do 1 while (istop.eq.0)
        chisq=0.d0
        do 2 i=1,n,istep
        chisq=chisq+qdata(i,1)*qdata(i,1)
    2   continue

        if(dabs(dlamb).lt.1.d-20) then
          istop=1
        else
          call gaussj(hmat,m,m,beta,1,1)
          do 3 i=1,m
          deltaa(i)=beta(i)
    3     continue

          do 4 i=1,m
          a(i)=a(i)+deltaa(i)
    4     continue
          chisqnw=0.d0
          do 5 i=1,n,istep
          x=dble(i)
          sinarg=pi*a(1)*dsin(w*x+a(2))+a(5)+a(6)*x+a(7)*x*x
          qdata(i,1)=qdata(i,2)-a(3)*dsin(sinarg)-a(4)
          chisqnw=chisqnw+qdata(i,1)*qdata(i,1)
    5     continue

          if(chisqnw.le.chisq) then
            dlamb=1.d-1*dlamb
            istop=1
          else
            dlamb=1.d1*dlamb
            call hessmat(a,beta,hmat,m,dlamb,n,istep)
          end if
        end if
    1 continue
      return
      end
*****************************************************************
      subroutine hessmat(a,beta,hmat,m,dlamb,n,istep)
      implicit integer*4 (i-n)
      implicit double precision (a-h,o-z)
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
      common/con/pi,pi2,w
      dimension a(m),beta(m),hmat(m,m)

      do 1 i=1,n,istep
      x=dble(i)
      sinarg=pi*a(1)*dsin(w*x+a(2))+a(5)+a(6)*x+a(7)*x*x
      ssa=dsin(sinarg)
      csa=dcos(sinarg)
      qdata(i,1)=qdata(i,2)-a(3)*ssa-a(4)
```

```
      arg=w*x+a(2)
      qdata(i,3)=a(3)*pi*csa*dsin(arg)
      qdata(i,4)=a(1)*a(3)*pi*csa*dcos(arg)
      qdata(i,5)=ssa
      qdata(i,6)=dble(1.)
      qdata(i,7)=a(3)*csa
      qdata(i,8)=qdata(i,7)*x
      qdata(i,9)=qdata(i,8)*x
    1 continue

      do 2 j=1,m
      beta(j)=dble(0.)
      do 3 k=1,n,istep
      beta(j)=beta(j)+qdata(k,1)*qdata(k,j+2)
    3 continue
      do 2 i=j,m
      hmat(i,j)=dble(0.)
      do 4 k=1,n,istep
      hmat(i,j)=hmat(i,j)+qdata(k,i+2)*qdata(k,j+2)
    4 continue
      hmat(j,i)=hmat(i,j)
      if(i.eq.j) hmat(i,j)=(dble(1.)+dlamb)*hmat(i,j)
    2 continue

      return
      end
*******************************************************************
*     W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling   *
*     Numerical Recipes, Cambridge University Press, 1986, p.28-29 *
*******************************************************************
      subroutine gaussj(a,n,np,b,m,mp)
      parameter (nmax=7)
      implicit integer*4 (i-n)
      implicit double precision (a-h,o-z)
      dimension a(np,np),b(np,mp),ipiv(nmax),indxr(nmax),indxc(nmax)
      do 11 j=1,n
      ipiv(j)=0
   11 continue
      do 22 i=1,n
      big=dble(0.)
        do 13 j=1,n
        if(ipiv(j).ne.1) then
          do 12 k=1,n
          if(ipiv(k).eq.0) then
            if(dabs(a(j,k)).ge.big) then
              big=dabs(a(j,k))
              irow=j
              icol=k
            end if
          elseif(ipiv(k).gt.1) then
            print*,'singular martix'
            return
          end if
   12       continue
        end if
   13   continue
      ipiv(icol)=ipiv(icol)+1

      if(irow.ne.icol) then
        do 14 l=1,n
        dum=a(irow,l)
        a(irow,l)=a(icol,l)
        a(icol,l)=dum
   14   continue
        do 15 l=1,m
        dum=b(irow,l)
        b(irow,l)=b(icol,l)
        b(icol,l)=dum
   15   continue
      end if
```

```
      indxr(i)=irow
      indxc(i)=icol
      if(a(icol,icol).eq.dble(0.)) then
        print*,'Singular matrix'
        return
      end if
      pivinv=dble(1.)/a(icol,icol)
      a(icol,icol)=dble(1.)
      do 16 l=1,n
      a(icol,l)=a(icol,l)*pivinv
   16 continue
      do 17 l=1,m
      b(icol,l)=b(icol,l)*pivinv
   17 continue
      do 21 ll=1,n
      if(ll.ne.icol) then
        dum=a(ll,icol)
        a(ll,icol)=dble(0.)
        do 18 l=1,n
        a(ll,l)=a(ll,l)-a(icol,l)*dum
   18   continue
        do 19 l=1,m
        b(ll,l)=b(ll,l)-b(icol,l)*dum
   19   continue
      end if
   21 continue
   22 continue
      do 24 l=n,1,-1
      if(indxr(l).ne.indxc(l)) then
        do 23 k=1,n
        dum=a(k,indxr(l))
        a(k,indxr(l))=a(k,indxc(l))
        a(k,indxc(l))=dum
   23   continue
      end if
   24 continue
      return
      end
```

Q2RESULT.FOR

```
*****************************************************************
      subroutine results(i,nr1,nr2,stddev1,stddev2,indic,imode)
      implicit double precision (a-h,o-z)
      character fgues*80,fsfit*80,fssfit*80,fsumm*80
      character dat*4,txt*3,sampl*20,satur*20,comm*80,resfil*12
      character str*37
      common/fna/fgues,fsfit,fssfit,fsumm
      common/inf/dat,txt,sampl,satur,comm
      common/dat/qdata(16384,9),rn,vrange1,vrange2,freq,clkdiv,
     *          ppc,resol
      common/par/method,guess(7),sfit(5),ssfit(7),nrn,nrnot(126)
      common/mod/garray(126,9),parray(126,11)
      dimension g(9),p(11)

      method=0

      if(imode.gt.(-2)) then
        if(i.eq.nr1) then
          ibeg=i
          open(1,file=fgues)
c         open(2,file=fsfit)
          open(3,file=fssfit)
          rewind(1)
c         rewind(2)
          rewind(3)
          indisum=0
          if(txt.ne.'xxx'.and.txt.ne.'XXX') then
            open(9,file=fsumm,access='append')
            resfil=dat//'_'//txt//'.dat'
```

```
            write(9,90)resfil,sampl,satur,comm
   90         format(1x,a12,6x,2a20,a80)
            close(9)
          end if
        end if

        write(1,10)i,freq,(guess(j),j=1,7)
c         write(2,20)i,freq,(sfit(j),j=1,5),stddev1
        write(3,30)i,freq,(ssfit(j),j=1,7),stddev2,indic

        if(indic.eq.(-1)) then
          indisum=indisum+1
          nrn=indisum
          nrnot(nrn)=i
        end if

        if(i.eq.nr2) then
          iend=i
          close(1)
c           close(2)
          close(3)

          str=' file(s) may be reduced incorrectly !'
          write(*,40)indisum,str
   40       format(1x,'É',72('Í'),'»',/,
     *            1x,'º',72(' '),'º',/,
     *            1x,'º',16x,i3,a37,16x,'º',/,
     *            1x,'º',72(' '),'º',/,
     *            1x,'È',72('Í'),'_')
        end if
      end if

      if(imode.eq.(-2)) then
        parray(i,1)=dble(i)
        parray(i,2)=freq
        do 1 j=1,7
        parray(i,j+2)=ssfit(j)
    1   continue
        parray(i,10)=stddev2
        parray(i,11)=dble(indic)
      end if

      if(imode.eq.(-3)) then
        open(1,file=fgues)
        open(3,file=fssfit)
        rewind(1)
        rewind(3)
        l=1
        k=nrnot(l)
        do 2 i=ibeg,iend
        read(1,10)integ1,(g(j),j=2,9)
        g(1)=dble(integ1)
        read(3,30)integ1,(p(j),j=2,10),integ2
        p(1)=dble(integ1)
        p(11)=dble(integ2)
        if(integ1.ne.k) then
          do 3 j=1,9
          garray(i,j)=g(j)
          parray(i,j)=p(j)
    3     continue
          do 4 j=10,11
          parray(i,j)=p(j)
    4     continue
        else
          l=l+1
          k=nrnot(l)
        end if
    2   continue

        rewind(1)
```

```
      rewind(3)

      do 5 i=ibeg,iend
      write(1,10)int(garray(i,1)),(garray(i,j),j=2,9)
      write(3,30)int(parray(i,1)),(parray(i,j),j=2,10),
     *                 int(parray(i,11))
 5    continue
      close(1)
      close(3)
    end if

  10 format(1x,i3,2f9.4,f10.5,2f8.3,f9.3,2e11.4)
c  20 format(1x,i3,7e12.5)
  30 format(1x,i3,2f9.4,f10.5,2f8.3,f9.3,2e11.4,f7.4,i3)

    return
    end
```