USE OF PNEUMATIC PULSE STIMULUS FOR INDUCEMENT OF LARGE AMPLITUDE FLEXURAL WAVEFORMS IN WEB MATERIALS FOR THE PURPOSE OF LOCAL TENSION MEASUREMENT

By

CURTIS M. VICKERY

Bachelor of Science
Oklahoma State University
Stillwater, Oklahoma
1981

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1983

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
May, 1992

USE OF PNEUMATIC PULSE STIMULUS FOR INDUCEMENT OF LARGE AMPLITUDE FLEXURAL WAVEFORMS IN WEB MATERIALS FOR THE PURPOSE OF LOCAL TENSION MEASUREMENT

Thesis Approved:

R L Lowery
Thesis Adviser

James K. Good

Robert M. Worth

Richard L. Cummins

Thomas C. Collins
Dean of the Graduate College

ACKNOWLEDGMENTS

Work toward completion of a Doctor of Philosophy degree leads one down many paths. One becomes involved with ideas and areas of study perhaps not delved into before. At many points along this journey futility and frustration may intercede, challenging one's resolve. Responsibility of the researcher is to search out the various paths and obstacles that may present themselves, and endeavor to glean any relevant and useful information therein. Through repetition of this process, progress may be realized toward degree completion.

This study details experimental and analytical analysis of a new tension measurement system for the web producing and web handling industries. The project was provided through the Web Handling Research Center at Oklahoma State University. Support of the Center and industrial consortium members has been very much appreciated during project development. Interest in the effort has been substantial, which has been a real motivational factor. Consortium patience has been appreciated in that a number of students, working limited hours, have contributed to this overall effort. Special thanks go to Mobil Chemical Company in Shawnee, Oklahoma, and Mr. Art Fischer for their help in scheduling and performing industrial field tests that have been documented in this report.

Analytical investigation of this project allowed for study into many areas of science and engineering. This was an opportunity to examine a wealth of information, theory, and applications that were heretofore largely unknown to myself. I am greatly appreciative of the opportunity to pursue these studies and the subsequent computer modeling contained herein. Special thanks are extended to Mr. Rod McAbee and Mr. Jim Dalton of the CEAT computer center. Their help in acquiring quality plots and program listings for publication has enhanced this effort greatly.

Acknowledgment must be made to those who have been of great service during this project period. I would like to thank friends of the Stillwater community, especially Dr. William Chace

of Highland Park United Methodist Church, for their honesty and sincerity during times of need. I very much appreciate the attention and cooperation provided by my faculty committee consisting of Dr. Keith Good, Dr. Peter Moretti, Dr. Ron Delahoussay, and Dr. Richard Cummins. Their interest in the various project aspects has been gratifying. A good bit of energy was expended in preparation of this document. I would like to thank Mrs. Laura Johnson for assistance with graphics schematics provided in this study. Special thanks are extended to Ms. Charly Fries and Ms. Suzanne Spears for their efforts in bulk report formatting and preparation; it could not have been accomplished without their wonderful spirit and sense of purpose.

Finally, I would like to acknowledge those who have had the greatest effect on this researcher. These are individuals who serve to keep one focused and motivated even during times when feelings of futility and frustration are severe. My parents, Dr. and Mrs. Rollin Vickery, have been very supportive of my efforts and career decisions. I am grateful for the high regard that education was accorded during my upbringing. Above all, I would like to thank my major adviser, Dr. Richard Lowery, for his unending support. During times of self doubt, Professor Lowery would provide a much needed boost. When periods of complacency or overconfidence were present, Professor Lowery would always provide some new ideas or methods to ponder. Dr. Lowery is a storehouse of ideas, yet sight is always fixed on useful applications of such ideas. I will always retain the quiet sense of competence, purpose, and versatility that I have had the pleasure to experience in my years with Dr. Lowery here at Oklahoma State University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Tension measurement is a subject of interest to participants of the web producing and handling industries. A web-like material is manufactured and processed as continuous sheets that are stored in wound roll form. Web thickness is many orders of magnitude smaller than the web width. Web manufacturing and processing industries deal with many different web material properties and geometries such as paper, plastic, synthetic, or a combination of these basic materials in varying thicknesses and widths. Different size, thickness, and composition of webs create different problems with respect to web winding and handling. Web structural dynamics coupled to control system and winding dynamics may result in a quality or flawed product, depending on knowledge of forces exerted on the web during processing. This study deals primarily with measurement of web tensile forces. Tension in a web is an indication of the force required to draw the web through a process system and/or the compressive force applied to roll wraps during winding. Thus, knowledge of web tension is basic to web manufacturing and handling process control.

Average tension measurement across a roll span is commonly accomplished by use of load cells in conjunction with an idler roll. Thus, the number of pounds indicated by the load cell divided by the web width provides an average tension value. While this affects feedback information for gross process control, other more subtle problems also affect web handling and web winding processes. Factors such as web uniformity, alignment of web fibers, and tension distribution across a web span all affect web dynamics and roll quality. This research effort addresses the final item above, local tension measurement allowing for knowledge of tension distribution across a web span. This tension profile information could lead to process corrections or winder alignment corrections such that improvements in roll quality could be realized. A general overview of web tension and tension measurement has been provided in Chapter II.

Positive and negative features of present tension measurement methods have been highlighted in this section.

To be proposed and investigated in this report is a method of web tension measurement which may be used at discrete locations along a web span. This effort has been accomplished through the support and cooperation of the Web Handling Research Center at Oklahoma State University and associated WHRC Industrial Consortium members. The consortium members provided requirements for such a tension measurement system that have been adhered to during this investigation. One major requirement was that tension measurements had to be obtained without directly contacting the web material. The proposed scheme provides a sharp noncontacting stimulus to a web surface under test, creating traveling waves that propagate from the input disturbance location. Sensing and processing of these waves allows for determination of a tension indication.

Detailed in Chapter III is the chronology of development of the above stated proposed experimental tension measurement scheme from a single concept to an industry adapted tension measurement system. Section 3.1 outlines historical background of project development. System hardware development along with system automation and data acquisition activities have been presented in sections 3.2 and 3.3. Analysis of the system theoretical model and subsequent system refinements with regard to sampling and data processing has been detailed in sections 3.4 and 3.5. Finally, section 3.6 illustrates adaptation of the experimental system to a wide variety of web materials, thicknesses, and tension levels. Both laboratory and industrial field test results have been included in these discussions.

Analytical examination of the dynamic relations between the tension measurement system and web under test was performed. Desired from this analysis was to better understand why the experimental tension measurement system provided the particular shape of signals that were being sensed and processed. Qualitatively, signal shape was dependent on web tension, web thickness and flexural rigidity, and amplitude of the input web stimulus. Unknown were the process dynamics, that is, coupling that was occurring between the input web stimulus and web surface. The analytic study to be presented examines the input web stimulus and coupling to web structures through finite difference modeling procedures.

Chapter IV provides an examination of pulse propagation theory and the applicable fluid and thermodynamic properties involved in this analysis. To be modeled is a weak shock front such that the inviscid, unsteady state, nonisentropic, compressible flow equations are used. An examination of the equation set characteristics is provided to illustrate the interaction of the independent variables used in the analysis. A short discussion of integration schemes for evaluating these partial differential equations is included as support material. An integration scheme was selected whereupon the two dimensional inviscid, unsteady state, nonisentropic, compressible flow equations were cast in proper finite difference form with boundary conditions noted.

Web structure modeling is provided in Chapter V. Depending on thickness and stiffness, web materials may behave like plates or membranes. To a transverse stimulus, thin webs behave essentially like membranes while retaining a nonzero flexural rigidity. Thick web materials exhibit more of a plate-like response but still may experience very large deflections. Thus, webs may exhibit characteristics usually attributed to both membranes and plates. Static and dynamic versions of the linear membrane and plate equations are derived and boundary conditions are noted.

Modeling results are presented in Chapter VI. Graphical results of the pulse modeling are presented as a result of Chapter IV finite difference derivations. Pressure levels from pulse modeling have been coupled to the dynamic membrane and plate models given in Chapter V. Graphical results of membrane and plate deflection response to shock pulse input are presented and comparisons to experimental results are offered.

This study contains information from both fields of experimental mechanics and analytics. Of main interest to the Web Handling Research Center industrial consortium members was the experimental equipment and evaluation techniques. A viable and accurate tension measurement system would be a fine addition to the web handling and processing industry in efforts to improve their industrial processes. The analytics have been performed to gain insight into the mechanics of the proposed tension measurement system and to qualify system results. This study will endeavor to provide a complete examination of the experimental methods and analytical basis behind the success of the tension measurement system to date.

CHAPTER II

TENSION LITERATURE SURVEY

Knowledge of web tension is basic to web manufacturing and winding processes. Often, various web parameters are affected by web tension used during manufacture. Fluctuations in these parameters may result in a nonhomogeneous web coupled with roll quality defects in the finished product. Measurement of web tension has become the focus of many in the web handling industry as a means of improving web consistancy and roll quality. A short review of conventional web tension control systems will be presented to justify a research effort into tension measurement.

A typical web tension control system, as outlined by Ketterer [1], incorporates four factors into the process. These factors are nominal tension or set point, tension measured via sensors, error indication based on differences in set point and measured tension levels, and feedback of a correction signal. A thorough knowledge of tension control systems and web characteristics is essential such that a selected control system will have optimal sensitivity, response time, and operating range for the given web material to produce a uniform web product.

Nominal tension is typically applied to a web by driving one end of the web through a clutch system while inhibiting the other end of the web through a braking system. Torque sensors are used with the clutch and brake assemblies to guage the tension level. Engineers involved with these systems continue to develope clutch and brake mechanisms such that predictable and repeatable control may be realized.

Several different mechanisms are available to perform the clutch/braking process. Drum or disk brakes still enjoy wide use in the web handling industry [2, 3], which may be actuated typically through pneumatic or hydraulic means. Deeg [4] presented a short article outlining the positive and negative aspects of pneumatic/hydraulic actuated braking systems. Electromagnetic braking systems, illustrated in *Machine Design* magazine [5], allow for braking proportional to a controlled current applied to electromagnets. Also described in

Machine Design magazine were magnetic particle brakes [6] where a controllable magnetic field guides ferrous particles into the path of input and output shafts. Variable friction caused by these ferrous particles provided the torque control. Increasingly, shaft encoders are being utilized in clutch/braking control. Murray [7] outlined such a method where AC induction motors were used to vary torque to a driving roll, eliminating a clutch arrangement. In all of these cases, the clutch/brake torque control is designed to be sensitive not only to impulses that affect the system but also to inertial changes that affect tension due to the changing of roll radius during wind on or wind off operations.

Dynamic tension sensing is necessary to gauge effectiveness of the set point tension control system. Dancer arms or load cell idler rollers are commonly used for this purpose. Haggstrom [8] presented a basic tension sensing scheme where a load cell coupled idler roll was used. Shown in Figure 2.1, this method indicates the average force value that is applied to the idler roll by the web under tension. Different types of force transducers could be used in conjunction with this arrangement to achieve the tension proportional signal depending on sensitivity and range requirements.

Dancer arm rollers provide tension feedback information based on position. Typically, dancer arms are positioned according to a set point (nominal) tension level. Any subsequent rotation of the arm produces a change in signal output. This is shown schematically in Figure 2.2. Bak [9] outlined the use of dancer arm sensors on both the driving (clutch) portion and inhibitive (brake) portion of a web line for tension control. Critical to the use of either load cell or dancer arm idlers is the dynamics of the idler assemblies themselves. These dynamics must be known and compensated for to prevent biasing of the desired tension proportional output signal.

The last two aspects of a typical tension control system—derivation of an error indication from set point and sensor signal differences, and the feedback of the corrected signal to the clutch/brake assemblies—will not be examined in this report. Motivation for study of tension measurement is of interest. Even with improvements in tension control systems through more sophisticated sensors, quicker response times, and higher sensitivities, problems in web manufacturing still exist. Several researchers have provided papers regarding tension-related roll quality problem areas.

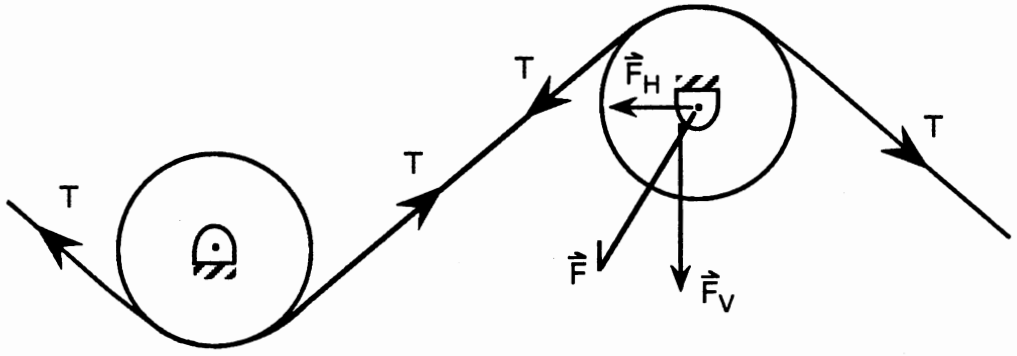


Figure 2.1. Idler Roller Load Cell Type Tension Sensor [8]

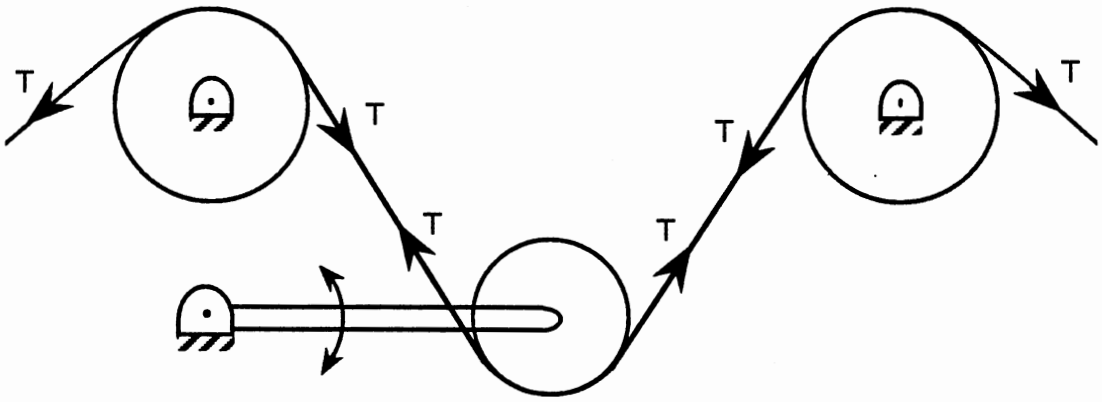


Figure 2.2. Dancer Arm Rotation Transducer Tension Sensor [9]

Hutzenlaub [10] presented a paper giving general conditions for good roll quality. When contact (nip) rolls were in use, tension in the web leading to the winder may be different than the resultant tension of the wound web. Factors such as roll diameter must be considered due to accumulation of pressure at the web core. Roisum [11] indicated that the easiest parameters to control in a winding scheme are torque, nip, and tension. These parameters may affect the web prior to, during and after the actual winding operation. Thus, knowledge of local web tension variation could be of benefit to winding operations. Ernst [12] indicated that inconsistent tension with respect to time and web span could lead to defects such as poor starts, offsets, dishing, interweaving, interlayer slippage, bursts, wrinkles, starring, and poor slitting. Dandan [13] also addressed such failure modes with respect to paper winding processes. Dandan acknowledged inherent fluctuations in average tension control due to eccentricities or other sources. Speculated was that the coincidence of tension surges with web product defects led to high instance of breaks. One solution to help alleviate such problems was to create as uniform a web product as possible, thus reducing the probability of a material flaw.

In these published papers, tension, applied either through clutch/brake assemblies or through nip contact rollers, has an effect on roll quality. Indicated also was concern for uniformity given by even distribution of web material and/or tension across a web span. Devices for measuring local or area tension have been proposed and marketed by researchers in the web handling and manufacturing industries. An examination of these devices and methods is an appropriate preface for the discussions to be presented in this report.

Noncontact wave generation in plate/membrane like materials has been examined for a number of years. Luukkala, Heikkila, and Surakka [14] examined plate resonances in this manner in 1971. The objective of the study was to generate a resonance condition in paper web material through coincidence effects, leading ultimately to determination of the paper material elastic parameters.

Coincidence occurs when the speed of sound in the atmosphere equals the speed of sound in the web, thus creating a resonance condition. Resonance conditions were achieved through variation of two quantities: the frequency of acoustical web excitation and the angle of incidence of the acoustical waves with respect to the web. Figure 2.3 shows the general

configuration for this analysis. Due to dispersive effects of the web material, plate phase velocity was also frequency dependent. Plate modeling techniques coupled with simplifying assumptions served to correlate the angle of incidence and phase velocity variables. Experimental tests used a set excitation frequency coupled to variation of the angle of incidence to generate phase velocity versus frequency data for the web material in both the machine direction and the cross direction. An estimate of the web material anisotropy was the final result of this study.

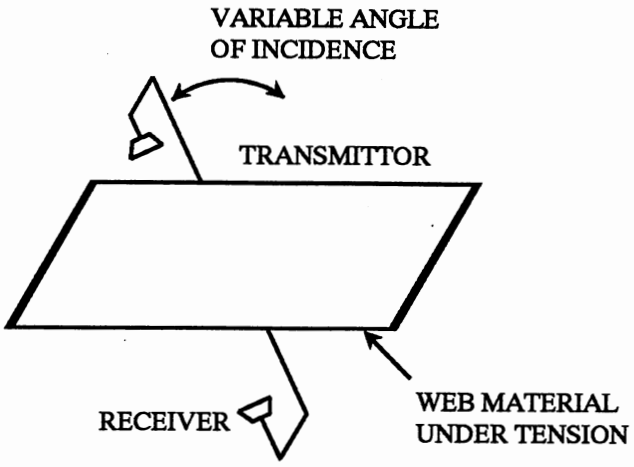


Figure 2.3. Plate Resonance Test Configuration [14]

A study involving this plate resonance technique as well as ultrasonic contacting techniques was conducted by Mann, Baum, and Habeger [15] in 1979. With respect to the aforementioned plate resonance method, noted was the practical limit of acoustical excitation to below 400 kHZ due to atmospheric attenuation. With this limitation, only relatively thick web materials could be tested such that the coincidence effect could occur within the excitation frequency limit. Ultrasonic methods were described as well for obtaining elastic parameters of webs and web stacks. Orthotropic wave theory was provided which related longitudinal and transverse web

phase speeds to material stiffness parameters. Experimenal procedures for stiffness determination were outlined and test case results were provided.

A recent study into experimental determination of paper elastic properties and anisotropic behavior was conducted by Olofsson, Molin, and Kyosti [16] in 1991. Laser holographic interferometry techniques were used to examine wave propagation in paper web materials. To generate elastic waves, a metal sphere was impinged on a static web surface, thus simulating an impulsive input. Hologram film was double exposed, first just prior to sphere contact and second at some programmable delay time after impact. Subsequent analysis of the film revealed contours of equal, out of plane displacements of the web surface. A desirable aspect of this research effort was the acquisition of interferograms that showed visually the magnitude, direction, and location of propagating waveforms from the impulsive source as a function of time. Web elastic parameters were then extracted from these data.

An on-line contacting ultrasonic method was developed by Baum and Habeger [17] in 1980. The plate resonance method, noncontacting in nature, was discarded by these researchers as being unfeasible in an on-line measurement situation. Again, phase velocities with respect to machine and cross directions were desired to gauge web anisotropy.

Ultrasonic transmitters and receivers were mounted on rotating wheels that made contact with the web. When the transmitter was coincident with the web, an ultrasonic burst was launched which traveled in all directions from the source. When the bursts were picked up by the receivers and corrected for web transport velocity, a time of flight value was then available from which to calculate phase velocity. Noted in the paper were the successes and areas of difficulty in implementing this method.

An updated version of the above system was reported by Habeger and Baum [18] in 1986. Improvements in their system included improved transducer design, synchronization, and signal sensing and interpretation. To improve synchronization, a transducer was used to provide a signal indicating when the transmitter, mounted on a rotating wheel, was some known angle of rotation away from the optimal web surface pulse launching site. When the ultrasonic pulse was received at machine direction and cross direction receivers, also mounted on synchronized wheels, digitizing and storage operations of the received pulse signals was performed. Zero

crossings of the digitized records were isolated and a time of flight value, corrected for the web transport velocity, was calculated. Haberger and Baum reported good results achieved with this on-line system with these improvements installed.

Habeger pursued use of ultrasonics for measurement of web elastic parameters as was outlined in a 1988 publication [19]. Static tests on web material coupons were performed to gauge amount of or lack of orthotropic symmetry in the material machine and cross directions. This led to calculations of shear coupling coefficients for the nonorthotropic cases.

Ultrasonic transducers were used to excite both longitudinal and transverse Lamb waves within the web test coupon. Phase velocity of these input waves was dependent on angle of transmission and reception of said input waves with respect to the material machine and cross direction orientation. An equation was used with the achieved phase velocity results such that planar specific stiffness coefficients could be derived. A nonorthotropic angle value was then obtained from viewing symmetry of the phase velocity squared and the calculated specific stiffness as the ultrasonic wave input angle varied from 0 degrees to 360 degrees. A nonzero nonorthotropic angle quantity indicated that nonzero shear coupling coefficients were present in subsequent material evaluation and handling.

In the above developments by Habeger and Baum, a web contacting method was used. Other contacting web tension indicators have been developed that use different stimulus/ transducing methods. The Norwegian Pulp and Paper Research Institute devised a simple force coupled device for tension measurement, as was reported by Hansen [20] in 1986. Contact of the device with a web would deflect a soft spring blade, which was used as input to an inductive displacement transducer. With appropriate pressure of the transducer head to the web material, Hansen reported that stable, repeatable results may be achieved from this device.

Compressed air was used instead of a spring for web deflection by the Scandev Invent Beetle, reported by Linna and Moilanen [21] in 1988. Reaction of the web under test to the compressed air input was correlated to web tension. In a test environment, the Scandev Invent Beetle required a permanent test stand and a spacing plate to properly set up the device prior to testing. Linna and Moilanen examined the positive and negative aspects of this system. In

general, results were quite sensitive with respect to initial transducer-to-web spacing and location with respect to rolls and web edges.

Another device that used compressed air as the web stimulus was developed by Russian researchers. This device, as reported by Walbaum and Lisnyansky [22], applied compressed air to webs that were being guided through the device by rollers. Web fluctuations created a pressure signal that was sensed by a microphone located in a central pneumatic chamber of the transducer head. The microphone output was then fed to a loudspeaker system, properly calibrated, to track the frequency of the web fluctuations, which was then correlated to web tension.

Industrial consortium members of the Web Handling Research Center, for various reasons, required noncontact as the bottom line for any tension measurement system. Due to possible damage to web coatings, scratching or denting of web surfaces, or damage to the web structure itself, it was felt that any contact other than random incident contact would be unacceptable.

Noncontacting web tension measuring devices to date generally rely on acoustical excitation coupled to time of flight wave propagation calculations. Aforementioned plate resonance experiments were likely the basis for these techniques. Researchers began to closely couple acoustic excitation to the web structure, forcing surface bending waves rather than zero order symmetrical or antisymmetrical Lamb-waves. A benefit of this was reduction of the excitation frequency such that atmospheric attenuation effects were reduced. A drawback, however, was the introduction of surrounding air loading effects on the wave propagation process.

The Altim Tensometer, developed by the Swedish paper industry and described by Meinander and Marttinen [23] in 1983, utilized the approach described above. A later paper by Marttinan and Luukkala [24] further explained the approach and theoretical basis. The Altim Tensometer produced a strong line excitation to a web by passing acoustical excitation through a long narrow slit. Microphones placed at intervals fore and aft of this acoustical slot monitored propagation of the resultant wave. Signal processing procedures, including correlation methods and compensation for air loading, were used to determine the time of flight value and hence the

web material phase speed. Tension values were then found through use of tabulated data relating tension to phase speed and web material basis weight.

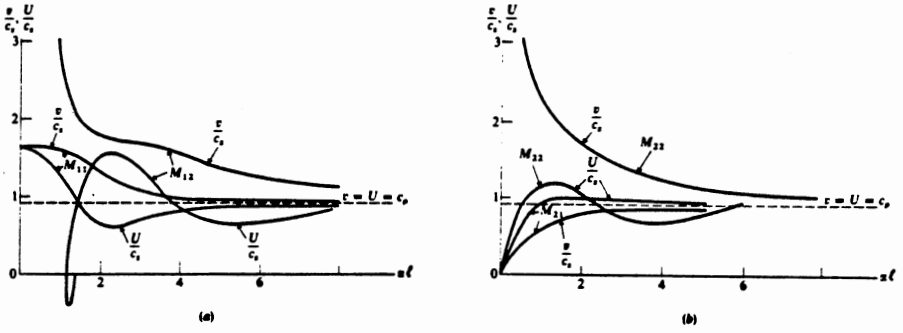
Optical detection of web response to acoustical excitation was used in the TENSCAN non-contacting tension measurement system, tested by Rye [25] in a 1988 publication. Time of flight indications were obtained through reflection of laser light to sensors, thus providing indication of passage of web flexural waves induced by an acoustical source. By laser configuration, the nominal distance between web surface and transducer head was measured and used as a correction factor in the calculations. Again, correlation techniques were used to obtain time of flight values, and thus the web phase velocity. Tension values were achieved through user input of web basis weight which was used in the data conversion process.

The noncontacting tension measurement systems outlined above have positive aspects for the web manufacturing/handling industry. One advantage is the development time already applied to the methods, over ten years, where many of the strengths and weaknesses could be noted and upgraded. All of the systems outlined above, however, have limits to their respective performance, depending on web stiffness, thickness, and tension level. The Web Handling Research Center supported development of a new tension measurement system that would be inexpensive and be applicable to a wide variety of web materials and tension levels. It was believed that such a system could aid the web manufacturing and web processing industries as a whole. Thus, basis for development of the tension measurement system to be chronicled in this report was established.

The experimental tension measurement system to be described in this report deals with interaction of a pneumatic shock wave with a web surface. From this interaction, flexural waveforms are generated in a web whereupon sensing of these flexural waveforms is used in the tension measurement process. Thus, many different mechanisms are involved in the proposed tension measurement system. Theory of wave propagation in elastic media is applicable to the process as is response of membranes and plates to impulsive inputs. Depending on the characterization of the web material as either a membrane or a plate, analysis complexity can be affected. Additionally, linear or nonlinear, static or dynamic membrane and plate models are available which certainly affect the simplicity or complexity of the analysis. A

literature survey is presented below to outline some of the available analysis techniques that have been used in similar situations. Of interest in this survey is an examination of literature concerning wave propagation in thin membranes and plates, interaction of shock waves with plate and shell structures, and static and dynamic response of plates and membranes to impulsive inputs. Due to the complexity of these types of problems, numerical methods have been used in many of these analyses. Presentation of synopses of applicable literature will provide some insights as to the method of analysis to be used in this study.

Lamb presented the general theory of wave propagation in elastic half spaces [26] and later in plates [27]. Relations for propagation speed of symmetric (longitudinal) and antisymmetric (shear) plate waves were presented, where simplifications could be applied based on the ratio of excitation wavelength to plate thickness. This theory, however elegant, is not as applicable to the current analysis due to high frequencies required of the input stimulus to provide correlation to theory. In the low frequency end of plate response, as provided graphically by Tolstoy [28] and reproduced in Figure 2.4, use of excitation with wavelength on the order of plate thickness results in zero order longitudinal and transverse waves being excited. For many web materials of thickness well below one-thousandth of an inch, ultrasonic input frequencies would be required to achieve this type of response. Web ultrasonic on-line test system and coupon research of Baum and Habeger outlined earlier utilized this approach.



Group and phase velocities for first two symmetric (M_{11}, M_{12}) and antisymmetric (M_{21}, M_{22}) modes of propagation of a free plate. Note the negative group velocity on M_{12}; this actually implies only that group and phase velocities are in opposite directions: energy flow has to be away from a source at all times.

Figure 2.4. Generalized Plate Response [28]

Due to lower frequency excitations being adapted for noncontacting web stimulus, bending waves will be the dominant rather than Lamb-type waves. Propagation speed of bending waves is then a major consideration in the web tension measurement system to be proposed. Wavelength of such bending wave disturbances will be much greater than the web thickness. Cohen and Berkal presented a theory of propagation of such large, sharp amplitude waves in membranes [29], which was later formulated by Cohen [30] for plates.

Cohen and Berkal [29] provided an in-depth presentation of longitudinal and transverse wave propagation in membranes. Propagating waves on a membrane surface were treated as a propagating curve where discontinuities or jump conditions could occur. Considered was the case of a membrane surface, initially at rest, being influenced by a shock wave. Compatibility relations for membrane deflection, and hence the strain and rotation, were established with respect to the jump conditions. Derived were the two modes of membrane wave propagation, shear-transverse (equiareal) and longitudinal (irrotational). Propagation speeds for the two wave types were:

$$V^2_{LONG.} = \frac{E\,h}{\rho\,(1-\nu^2)} \quad ; \quad V^2_{SHEAR} = \frac{1}{2}\,\frac{E\,h}{\rho\,(1+\nu)} \tag{2.1}$$

For a typical one mil (h = 0.001 in.) plastic web, specific density of approximately 0.9, Young's Modulus E on the order of [6 (10^5)] psi, and Poisson's Ratio ν of 0.3 may be assumed. These parameters yield shear and longitudinal wave propagation speeds of 4363 and 7374 ft/sec, respectively. These phase speeds meet expectations of the bulk Lamb-type material response to ultrasonic input.

Cohen [30] provided much of this same type of analysis with respect to plate wave propagation. Again, waves were treated as propagating surface curves, where discontinuities were allowed. Derived were propagation speeds of extensional waves (longitudinal and shear-transverse) and bending waves (bending, twisting, and kink). For extensional waves, compatibility relations for displacement were formulated for the jump conditions, which were then related to the plate strain conditions. Thus, speed of wave propagation for shear-transverse and longitudinal waves were given by:

$$V^2_{LONG.} = \frac{(1-\nu)\ E\ h}{\rho\ (1+\nu)\ (1-2\nu)} \quad ; \quad V^2_{TRANSVERSE} = \frac{1}{2}\frac{E\,h}{\rho(1+\nu)} \tag{2.2}$$

For comparative purposes, the classical propagation speeds for these modes was provided:

$$V^2_{\substack{LONG.\\CLASSICAL}} = \frac{E\ h}{\rho\ (1-\nu^2)} \quad ; \quad V^2_{\substack{TRANSVERSE\\CLASSICAL}} = \frac{1}{2}\frac{E\,h}{\rho(1+\nu)} \tag{2.3}$$

In the case of bending waves, a similar analysis was performed such that propagation speeds for bending and twisting modes were given as:

$$V^2_{BEND.} = \frac{1}{2}\ \frac{E\ h}{\rho\ (1-\nu^2)} \quad ; \quad V^2_{TWISTING} = \frac{1}{2}\frac{E\,h}{\rho(1+\nu)} \tag{2.4}$$

Note that $V_{BEND.}$ and $V_{TWIST.}$ equal the classical values of Equation (2.3). For the web parameters of the preceding paragraph, the bending propagation speed is approximately 5214 ft/sec. Again, this phase speed is substantially higher than will be viewed with respect to large amplitude flexural waveforms to be detailed in this study.

Russian researchers have examined the interaction of shock waves with various structures. Two publications of such work are cited below. Babaev, Kubenko, and Krishtalev studied interaction of a shock wave with a deformable cylindrical shell [31]. Kubenko and Moseenkov studied interaction of a weak shock wave on membranes which separated two acoustical half spaces [32]. Some methods of analysis were common to these two research works.

Basis for this analysis were equations of motion of the flexible structure for the directional displacement components considered and a potential function to describe motion of fluid at the shell structure surface. Compatibility relations were used in each case at the fluid-structure interface to provide an impenetrability criteria. Transform of equations to the Laplace domain with respect to time and Bessel function domain with respect to space was used with assumed series solutions. Thus, formulation of equations for pressure fluctuations and structure displacements was accomplished. An unspecific procedure for inversion of the transformed equations was outlined in general and simplifications used for computational ease were noted. Figures showing pressure fluctuations for test cases were provided by the authors.

Analysis of plates and membranes may be accomplished through a variety of methods depending on the underlying assumptions adopted. Complications arise in analysis due to consideration of more arbitrary or general situations such as nonsymmetrical loadings or consideration of nonlinear factors such as in-plain strains. The most prevalent models in which to begin an analysis are the linear membrane equation, the nonlinear (Foppl) membrane equations, the linear (Kirchoff) plate equation, and the nonlinear (Von Karman) plate equations. In general, literature was not sought for the membrane or plate free vibration (eigenvalue) problem. These problems examine the homogeneous membrane/plate equation alone where resonant frequencies are of prime interest. The homogeneous plate solution is applicable, however, to the forced plate problem where it comprises a portion of the overall solution. Also, literature related to the Mindlin plate model, where rotary inertia and shear deformations are considered, was not actively collected due to added complexity of the model with respect to potential rewards possibly achieved when viewed with respect to thin web materials.

Energy methods were used to solve the large deflection Von Karman equations by Stippes in 1951 [33]. Using the Ritz method, a potential function for the plate was formulated based on total potential energy from internal and external forces acting on the plate. Infinite series solutions for the plate deflection and stress function over the two dimensional plate area was assumed. Substitution of these assumed solutions into the potential function allowed for minimization of the potential function with respect to the unknown series coefficients. Thus, a set of simultaneous equations could be solved for the series coefficients such that the displacement and stress function values could be generated.

A relatively simple approach to the solution of the Kirchoff small deflection plate equation was presented by Cadambe and Kaul [34] in 1955 based on procedures outlined by Timoshenko [35]. The procedure used the Moment Sum definition to split the fourth order Kirchoff linear plate equation into two second order partial differential equations. Lower order derivatives were indicated to be beneficial in numerical analysis. Plate flexural rigidity or membrane tension could be used as the structural stiffness constant in the resulting equations. Knowledge of deflections and moments at the plate boundaries was required. This method precluded the use of free edge boundary conditions due to free edge boundary conditions being based on third

order partial derivatives. Cadambe and Kaul outlined the formation of moment and displacement meshes and the general static solution procedure.

A widely referenced publication for the solution of the nonlinear Foppl membrane problem was presented by Shaw and Perrone in 1954 [36]. A similar paper was presented by Kao and Perrone in 1972 [37]. Nonlinear equations were derived using minimum potential energy principle, resulting in three equations where the in-plane deflections u and v and the transverse deflection w were coupled. Procedures were provided for the step-by-step evaluation of these coupled equations. Nonlinear relaxation techniques were used to solve the equation set and a procedure was described to adapt the method to general shaped membranes.

A method for initial value static analysis of plate mechanics was provided by Al-Khaiat and West [38] in 1986 and Al-Khaiat [39] in 1988. Using the Kirchoff small deflection plate model, a combination of finite difference formulation and trapezoidal rule integration was used to solve for lateral plate displacement. For a plate lying in an x-y plane, the scheme called for restructuring of the Kirchoff equation through retainment of partial derivatives with respect to x while replacing partial derivatives with respect to y by their finite difference formulations. Thus, an expression for the fourth partial derivative of deflection w with respect to x was achieved as a function of lower order derivatives at adjacent grid points. Trapezoidal rule was then used to formulate equations for the third, second, and first partial derivatives of w with respect to x and lastly an equation in w. Use of these formulated partial derivatives with an iterative evaluation allowed for slow convergence over the problem grid. Plate deflections were obtained through solution of simultaneous equations involving plate boundary conditions. Al-Khaiat presented results of the method for uniform pressure loading of square plates, with and without in-plane loads, for comparisons to exact results.

A publication by Jones [40] utilized the approximate analysis of nonlinear plates first introduced by Berger [41] in 1954. Jones' interest was the application of the Berger method to membranes in an attempt to simplify the analysis. Foppl nonlinear membrane equations are achieved through use of the Von Karman plate equations with zero flexural rigidity. Zero flexural rigidity reduced the equations such that the membrane deflection was shown to be proportional to a Prandtl stress function over the membrane area. This stress function was

derived from the membrane shape, edge conditions, and pressure loading. Results were provided in this paper for different membrane shapes but only using a uniform pressure loading stimulus.

Nerantzaki and Katsikadelis [42] solved the nonlinear Von Karman plate equations through use of Green's function to convert the integral expression for potential energy over the plate area to line integrals over the plate boundary. Thus, line integral expressions were derived for the plate deflection and stress function. Boundary element methods were used to numerically evaluate these integrals for the deflection and stress function values. Stern [43] provided a very complete article on boundary value methods with respect to plate vibration in 1979. It was believed these methods were not as applicable to the modeling to be performed in this report.

Collocation methods have been applied to plate vibration and deflection problems. Collocation procedures involve the specification of plate deflection at a certain number of points whereupon some error between the calculated deflection and the true deflection may occur between these collocation points. Burgess and Mahajerin [44] provided a plate deflection analysis where fictitious loads external to a plate boundary were applied to enforce boundary conditions via the collocation approach. In this analysis, the plate under study was considered to be a portion of an infinite plane where upon the solution to the Kirchoff plate equation for the actual loading was obtained. Fictitious loads were placed on an expanded boundary a distance S from the actual plate boundary. Collocation procedures were used to obtain the strengths of these loads such that the required boundary conditions were met. Hence, fictitious loads would then be a solution to the homogeneous Kirchoff plate equation. Burgess outlined a method of breaking up a plate into a number of subregions and allowing the actual desired load to be considered constant over the incremental area. Examination of the influence of the actual loads and fictitious loads on each plate subregion was performed iteratively until sufficient convergence occurred. Presented were a number of examples including uniform load, patch load, and point load configurations.

A publication by Bauer detailed dynamic analysis of plate vibrations [45]. Similar papers by Chandrasekharappa and Srirangarajan [46, 47] used this approach with a slight modification to examine dynamic response of plates to pulse excitations.

Large deflection Von Karman equations were used in this method with the introduction of a time function f(t) such that time varying plate deflection and stress function, respectively, could be expressed through use of separation of variables as:

$$\text{Displacement:} \qquad w(x,y,t) = h\ f(t)\ g(x)\ h(y) \tag{2.5}$$

$$\text{Stress Function:} \qquad F(x,y,t) = F^{*}(x,y)\ f^{2}(t) \tag{2.6}$$

Boundary conditions were used to obtain generic form of the deflection with respect to the x and y space variables. For example, simply supported plate boundaries allow for the deflection expression (2.5) to be written as shown in Equation (2.7) where plate deflection is zero at plate boundaries $x = \pm a$ and $y = \pm b$.

$$w(x,y,t) = h\ f(t)\ \cos\left(\frac{\pi\ x}{a}\right)\ \cos\left(\frac{\pi\ y}{b}\right) \tag{2.7}$$

Equation (2.7) with Equation (2.6) were used to formulate the large deflection plate equations for dynamic analysis:

$$W[w,F] = \nabla^{4}w + \frac{\rho\, h}{D}\ \frac{\partial^{2}w}{\partial t^{2}} - \frac{p_z}{D}$$
$$- \frac{h}{D}\left\{\frac{\partial^{2}F}{\partial y^{2}}\frac{\partial^{2}w}{\partial x^{2}} + \frac{\partial^{2}F}{\partial x^{2}}\frac{\partial^{2}w}{\partial y^{2}} - 2\,\frac{\partial^{2}F}{\partial x\,\partial y}\frac{\partial^{2}w}{\partial x\,\partial y}\right\} \tag{2.8}$$

$$\nabla^{4}F^{*} = -\frac{\pi^{4} E\, h^{2}}{2\, a^{2} b^{2}}\left\{\cos\left(\frac{2\,\pi\, x}{a}\right) + \cos\left(\frac{2\,\pi\, y}{b}\right)\right\} \tag{2.9}$$

$$\ddot{f}(t) + \omega^{2}\ f(t) + \varepsilon\ \omega^{2}\ f^{3}(t) = P(t) \tag{2.10}$$

P(t) is a user defined load variation with respect to time. Ritz Galerkin method was used to obtain the ω and ε parameters through integration of the product W [w,F] g(x) h(y) over the plate area.

In application of this method, the time function f(t) is broken into two functions:

$$f(t) = r(t) + s(t) \tag{2.11}$$

such that s* (t) is a solution to:

$$\ddot{s}(t) - \omega^{2}\ s(t) = P(t) \tag{2.12}$$

which is a linearized version of the original differential Equation (2.10). Thus, the sum of homogeneous and particular solutions leads to the effective differential equation to solve:

$$\ddot{r}(t) + \omega^2\ r(t) + \varepsilon\ \ \omega^2\,(r(t) + s^*(t))^3 = 0 \tag{2.13}$$

Bauer provided results for a step input pressure function and an exponentially decaying pressure pulse. Chandrasekharappa and Srirangarajan provided the same examples using slightly different numerical evaluation techniques.

Fourier collocation expansion techniques were used by Nagaya to examine the dynamic response of plates and membranes to transient loads [48, 49]. Desired was a method of analysis which could be applied to an arbitrarily shaped plate or membrane for any general transient load. Fourier collocation expansion involved conversion of the linear plate/membrane governing equation to the Laplace domain. Bessel functions of the first and second kind along with coefficients to enforce boundary conditions and coefficients to describe the particular solution were then part of the transformed governing equation. This methodology was alluded to earlier in this survey with regard to shock-structure interaction [31, 32].

Plate/membrane boundaries were broken into segments whereupon appropriate boundary conditions were imposed. Curved boundaries were broken up into many segments to reduce error in the calculations. Resulting were equations for plate deflection, slope, and moment in terms of Fourier coefficients and Bessel functions. Matrices were formed from said coefficients for the number of series terms considered, where upon the Fourier coefficients could be obtained through solution of simultaneous equations. Plate deflection was obtained through Laplace transform inversion integral and residue theorem. Nagaya presented examples of a parabolic membrane and an elliptical plate response to a uniform pressure-exponentially time decaying input load. A good deal of numerical computation expertise seemed necessary to perform the required Laplace transform inversions to obtain desired deflection values.

A dynamic analysis of the Mindlin plate model was performed by Assadi-Lamouki and Krauthammer [50]. Noted earlier, Mindlin plate model considers effects of shear deformations and rotary inertia. This model could possibly be suited to thick heavy webs that experience large amplitude deflections, and thus a synopsis of the methodology is provided below. For

thin webs, however, the added complexity of the analysis due to inertial and shear components made this model not as attractive to this study.

In this analysis, plate deflections u, v, and w were written in terms of out-of-plane slope functions β_x and β_y :

$$u = -z\,\beta_x(x,y,t) \;;\; v = -z\,\beta_y(x,y,t) \;;\; w = w_{AVG}(x,y,t) \tag{2.14}$$

where $\beta_x = \partial w/\partial x - \gamma_{zx}$; $\beta_y = \partial w/\partial y - \gamma_{zy}$; and γ_{zx} and γ_{zy} are shear angles of the plate cross section.

In-plane strains were assumed to vary linearly through the plate thickness and out-of-plane strains were assumed constant throughout the thickness:

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = z \begin{Bmatrix} -\dfrac{\partial \beta_x}{\partial x} \\ -\dfrac{\partial \beta_y}{\partial y} \\ -\dfrac{\partial \beta_x}{\partial y} - \dfrac{\partial \beta_y}{\partial x} \end{Bmatrix} \tag{2.15}$$

$$\begin{Bmatrix} \gamma_{zy} \\ \gamma_{zx} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial w}{\partial y} - \beta_y \\ \dfrac{\partial w}{\partial x} - \beta_x \end{Bmatrix} \tag{2.16}$$

Elasticity laws were used with Equations (2.15) and (2.16) to obtain the corresponding stress conditions.

Rotary inertia consideration led to second derivatives of β_x, β_y and w with respect to time that became part of the moment and shear equilibrium equations for a plate differential element. Thus, equations were formed to couple displacements, moments, and shear forces at each plate grid point. Assadi-Lamouki and Krauthammer provided the solution technique and stability criteria for this method with some numerical results.

Outlined have been a few examples of plate and membrane analysis in both linear and nonlinear, static and dynamic cases. Of most interest were methods relating to transverse loading and deflection, where rotary shear and twisting could be neglected. Publications involving finite difference techniques were also desired for application with shock pulse modeling to be performed. Some techniques which were believed not as applicable to the

present study, such as finite element methods and boundary element methods, were not researched extensively. Examples of applicable publications include a very informative finite element analysis of transient wave propagation in plates by Sansalone, Carino, and Hsu [51]. Combination of finite difference, finite element, and Laplace transform methods were used in a dynamic plate analysis by Beskos and Leung [52]. Niemi and Pramila provided a finite element examination of transverse vibrations of a moving membrane in a surrounding fluid [53], which has direct application to the web handling scenario. A wealth of information is available for study. The objective of this literature review was to briefly outline some of the available analysis and evaluation methods that have been proposed and applied to membrane and plate models. Some of these methods will be used in this report in modeling of web response to a shock wave input.

CHAPTER III

DEVELOPMENT OF TENSION MEASUREMENT SYSTEM

3.1 Background of Experimental Tension Measurement System

Ideas behind the tension measurement system to be described in this report were visualized by Dr. Richard Lowery through interaction with the Web Handling Research Center consortium members. Desired by the web handling industry was a means of local tension measurement that would be versatile and low in costs. This tension measurement system would be required to be noncontacting and small in size physically so as to adapt to a large number of industrial plant settings where space limitations could be a factor. Additionally, the device would be required to work with a variety of different web materials of varying thicknesses and over a wide range of tension loads. Achieving all of these objectives would be beneficial to the web handling industry when compared to the devices presently available for local tension measurement.

Visualized by Lowery was some device to input an impulse-like disturbance to a web specimen, producing longitudinal, shear, or flexural waveforms in the web that could be sensed. Intuitively, speed of waveform propagation is proportional to web tension. Thus the beginnings of the project were oriented toward generation of an impulse-like disturbance in a web and the subsequent sensing and interpretation of flexural waveforms produced.

To provide an impulse input to a web specimen, a pneumatic pulsing system was designed and fabricated by project associate Nutter [54]. Shown in Figure 3.1-1, the pneumatic pulser was essentially a regulated pressure source used with a motor driven gating mechanism. The gating mechanism was a flat disk with a hole drilled in one spot. When the hole in the disk aligned with the pressure source outlet, a pulse of air was discharged through the pulser tube. At this point, the pressure pulse from this gating action was akin to a raised cosine pulse due to overlap of the two holes, which is a nonlinear function of area versus time during the overlap interval. This

pulse made a "wooshing" sound as the motor drove the disk through one cycle. Unsteady, compressible flow of the pressure pulse along the pulser tube length resulted in steepening of the pressure pulse front. Thus, the end result of the process was a "snapping" sound as the steepened pressure front reached the pulser tube exit. Another project associate, Ahn [55], performed experiments to determine optimal length and size of tubing to achieve the loudest, sharpest snapping sound for the rotary pulsing system. The shock front produced through this means was used as the impulsive input stimulus to web materials under test.

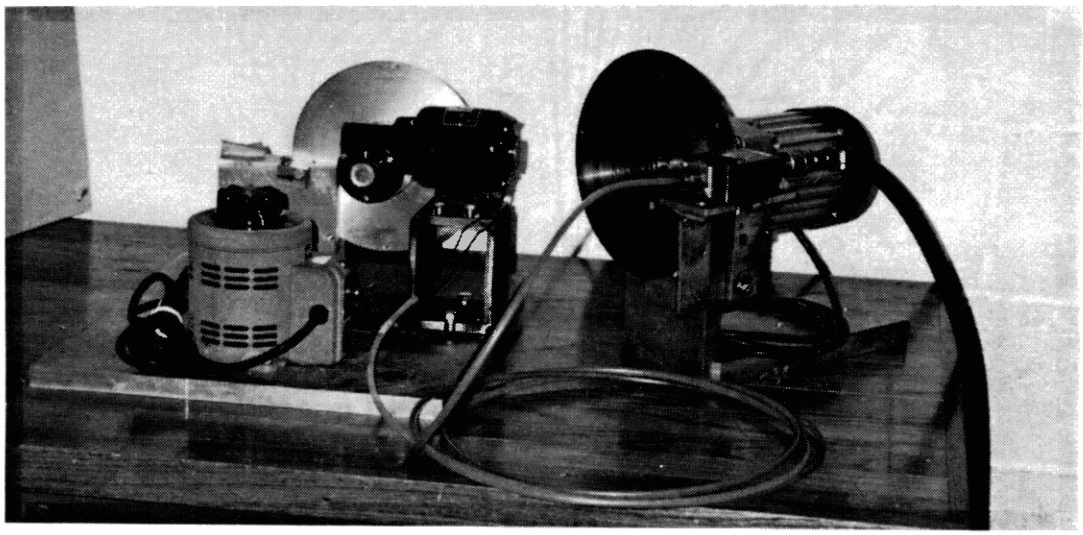


Figure 3.1-1. Rotary Pulsers—Original Version (Left) and Updated Version (Right)

Web tension was related to wave propagation speed in web materials by two notable models, the in-vacuo and air loaded membrane deflection equations. Project associate Lee [56] provided a literature review and theoretical examination of these models in reference to web tension application. Sensing of wave propagation speed was then implemented in the experimental system such that pneumatic pulse generated waves could be analyzed.

Project associate Bradley [57] developed the pulse input/waveform sensing arrangement first used in experimental tension measurement tests. This transducer head, pictured in Figure 3.1-2, contained two sensing microphones such that two signals would be available for viewing. As web tension varied, variation in arrival time of the generated waveforms at the sensing microphones could be captured with readout instruments. This first test system was situated on the WHRC Roisum Machine, where static and dynamic tests were performed. A ball screw platform for transducer head mounting was obtained such that tension profile tests could be performed.

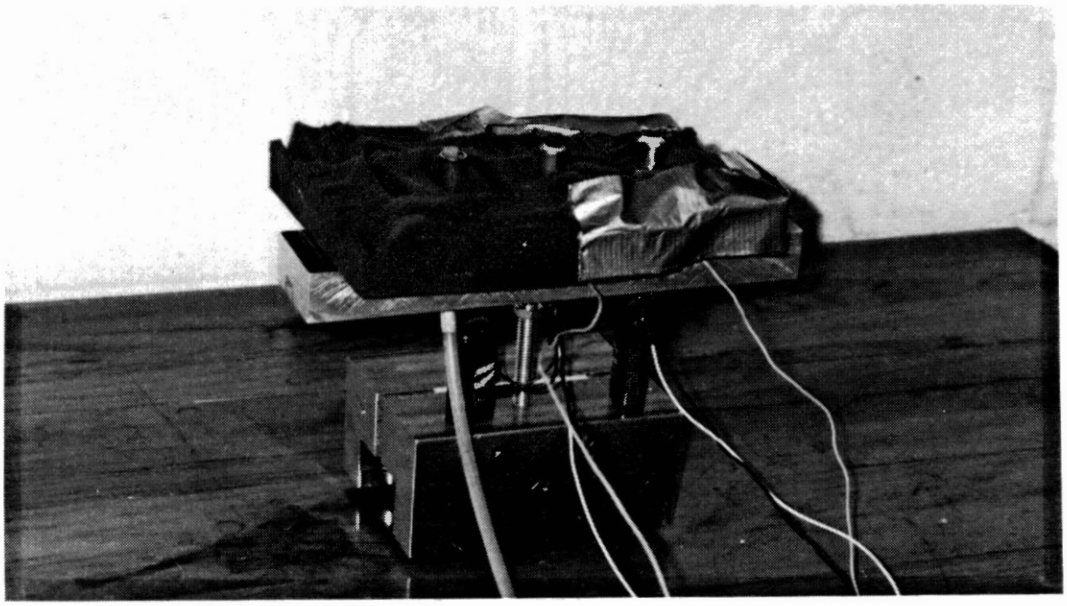


Figure 3.1-2. Original Bradley Transducer Head [57]

Bradley performed extensive tests on different web materials at different tension levels. Examined were captured signal shapes, amplitudes, and time of flight information. Passive signal conditioning methods were first used to provide more easily interpreted signals. The pneumatic

impulse, a broad band signal, provided a good deal of high frequency noise that was picked up by the transducer head electric microphones such that low pass filtering was used to reduce noise effects. Bradley also experimented with an operational amplifier diode clamping circuit to pass positive signal peaks while clamping off any negative peaks.

Preliminary automation of the system was accomplished by project associate David Magee . Magee incorporated an IBM compatible computer and Metrabyte DAS50 A-D board into the tension measurement system for data acquisition and analysis purposes. Five records from the above mentioned diode clamping circuit outputs were obtained by a computer program and averaged together. Figure 3.1-3 is an example of a typical trace from this particular procedure. An analysis program would then search out the peak values of the two independent signal sequences. A time of flight value was then achieved through knowledge of the analog to digital sample rate and the index numbers corresponding to the peak amplitudes. Tension values were then calculated using the in-vacuo membrane and ribbon models.

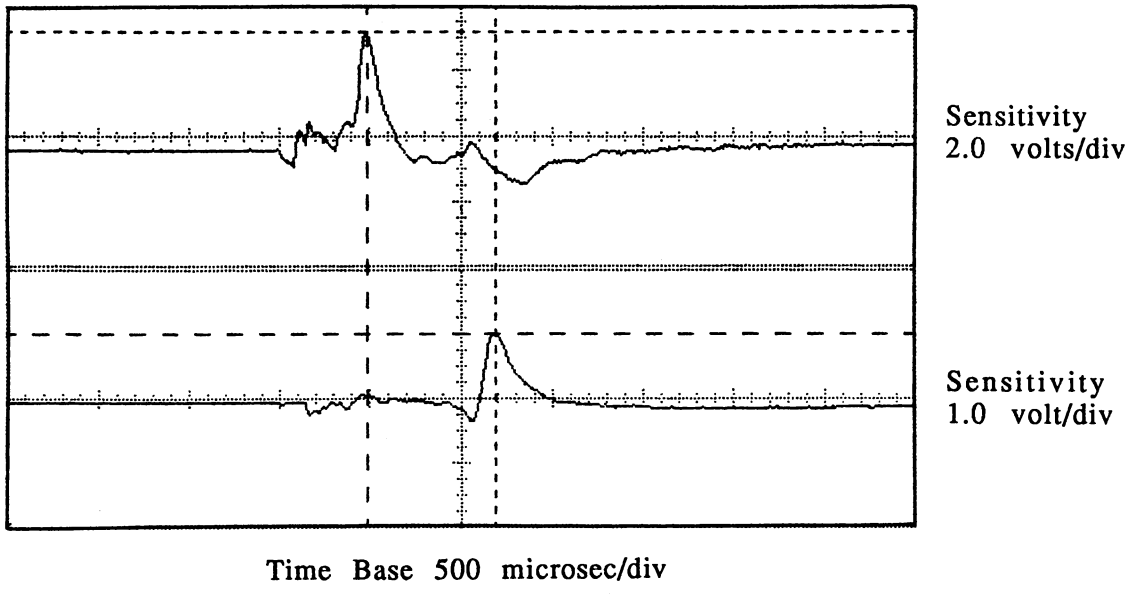


Figure 3.1-3. Typical Signal From Bradley Tension Tests [57]

Several questions arose from the Bradley preliminary studies that needed resolution. Some of these questions are outlined below. Research by Ahn revealed negative going signals as a result of pulser tube impulsive shock front input to a web under test. With the same microphone sensors in use, Bradley chose to clamp off the negative portion of the waveform. A contradiction appeared here in the handling of the sensed signals. A hypothesis would be needed to explain what the "proper" shape of the system microphones signals should look like. Mechanics of the pulser-to-web interaction was unresolved. The Bradley experimentation showed that higher amplitude waveforms could be obtained in a web when tension was relatively higher. This seemed to contradict the expected result that one could derive from string vibration theory.

The WHRC Roisum Machine did not possess a good system for web tension control. This allowed for uncertainty in average tension being applied to a web test specimen Thus, some uncertainty remained regarding some of the waveform and numerical data obtained during testing by project associates as outlined above.

Presented has been a brief background history of the tension measurement system that is the topic of this report. This has been presented to give a perspective of the number of researchers involved in this total effort to benefit the web handling industry. These preliminary efforts provided a viable basis for achieving the objectives outlined earlier. Further refinement of the experimental procedures, both sensing and signal processing, were needed to improve the system performance.

3.2 Field Testing of Initial Ideas

Development of the method for discrete localized tension measurement by this researcher began at this point. Projected was field testing of the system in an industrial setting once familiarity with operation and behavior of the system had been achieved and necessary fixturing had been fabricated. Mobil Chemical Company in Shawnee, Oklahoma, was to be the industrial test site, where plastic web materials were manufactured using tension levels from 0.2 to 0.8 pli. To better control tension levels in the laboratory, the experimental apparatus was transferred from the WHRC Roisum machine to a static frame arrangement. Transferred also was the ball screw platform such that tension profiles could be obtained. Plastic web materials were utilized at low

tension levels to adequately simulate conditions that would exist at the Mobil Chemical web manufacturing facility.

An HP 54501A digital oscilloscope was used to readout signals achieved from the new experimental laboratory test configuration. An example of a raw signal obtained in this manner is given in Figure 3.2-1. Note the major regions of influence indicated in the figure. Region 1 is dominated by high frequency noise due to the airburst or "snap" of the pneumatic shock wave. Region 2 is the pressure fluctuation due to the response of the web to the impulsive input. Region 3 is the aftermath of the major web pulse passage which is dominated by low frequency web flutter. Region 2 is the area of interest in the tension measurement scheme. Of interest was to determine the relation of the speed of this generated waveform with respect to tension and also to observe how the waveform shape changed with tension.

Figure 3.2-2 is a replication of Figure 3.2-1 with some time intervals inscribed. Time period T1 represents the interval for the airburst signal to travel from the fore (upstream) microphone to the aft (downstream) microphone. With the microphones two inches apart, the velocity of this airburst signal was:

$$C_{air} = (2 \text{ in.})/(160(10^{-6}) \text{ sec}) = 1.250\ (10^{4}) \text{ in./sec} \qquad (3.1)$$

This value of C_{air} is close to the typical speed of sound in air. Time period T2 is the interval between the airburst and when the web pulse passes the upstream microphone. Again, for two-inch spacing between the pulser tube and the upstream microphone, the speed of the generated waveform could be approximated by:

$$C_{web} = (2 \text{ in.})/(390\ (10^{-6}) \text{ sec}) = 5.128\ (10^{3}) \text{ in./sec} \qquad (3.2)$$

Finally, time period T3 corresponds to the time interval by which the web waveform travels from the upstream to the downstream microphone. With two inch spacing, this propagation speed is found to be:

$$C_{web} = (2 \text{ in.}) / (760(10^{-6}) \text{ sec}) = 2.632(10^{3}) \text{ in./sec} \qquad (3.3)$$

Note the difference in propagation speed as given by the C_{web} values for regions 2 and 3. Time difference T2 was dependent on both the pulse and web response, that is, coupling of the input pneumatic pulse to the web. Contrasted to this was time differential T3 due to waveform

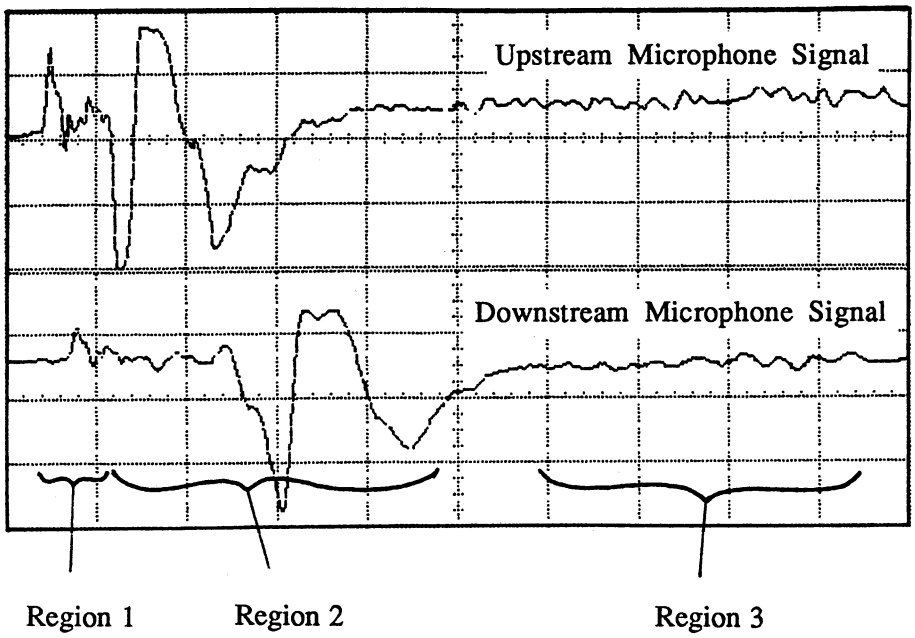


Figure 3.2-1. Typical Web Response to Pulsing System With Main General Time Intervals Marked

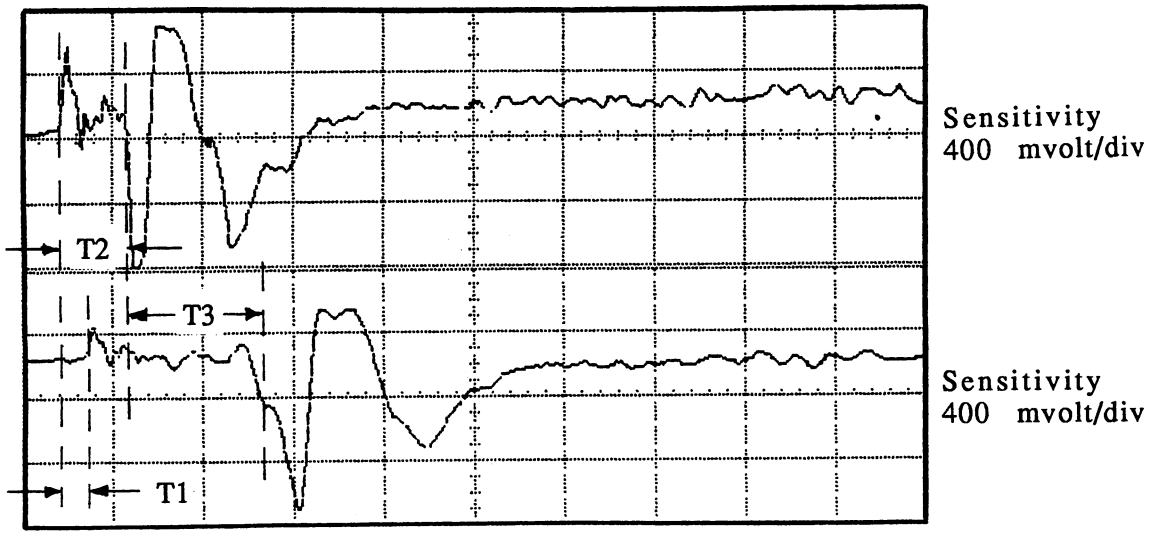


Figure 3.2-2. Reproduction of Figure 3.2-1 With Specific Time Intervals Marked

propagation in the web alone. This large difference in propagation speed, nearly a two-to-one ratio, illustrates complexity of the coupling involved in this experimental system.

In preparation for field testing, new transducer heads were prepared and signal filtering was implemented. At the Mobil Chemical plant, web thicknesses of 0.0005 to 0.0015 inches were typical with web widths running 9 to 10 feet. Web transport velocity would be 450 to 500 feet per minute. Care would be required in housing of the system pulser tube and sensing microphones such that intermittent web contact would not scratch or snag the web, possibly causing a break. To reduce inherent system noise, filtering was used to reduce the effect of airburst noise and low frequency web flutter.

Two transducer heads were fabricated of balsa wood, one short and one long transducer head or "shoe." Figure 3.2-3 shows the long shoe positioned on the static laboratory stand. Critical dimensions of these transducer heads were the spacing between the pulser tube and the microphone sensors. Two-inch spacing was used with the short shoe and four-inch spacing was used with the long shoe. Figures 3.2-4 and 3.2-5 are oscilloscope traces of raw signals obtained using these transducer heads on thin plastic web material at low tension. Isolating the major web pulse, the frequency band of these signals varied from roughly 1000 to 2000 Hz. Again, high frequency airburst noise was present as was the low frequency trailing web flutter.

Bandpass filtering was used to reduce high frequency noise and eliminate low frequency web flutter as well as any microphone DC biases. Second order Butterworth coefficients were used in the filter design, which was realized by a noninverting operational amplifier circuit. Filter specifications are provided in Figure 3.2-6. Filter bandwidth limits were selected based on observations such as was outlined for Figures 3.2-4 and 3.2-5. Figure 3.2-7 is an example of a bandpass filtered signal achieved with the long transducer head. As can be seen, the signal has been cleaned up giving the major web pulse component an accentuated appearance. Time interval measurement was aided by this conditioning such that gross estimates of web tension could be made. At this point field testing was in order to observe real world response of the system.

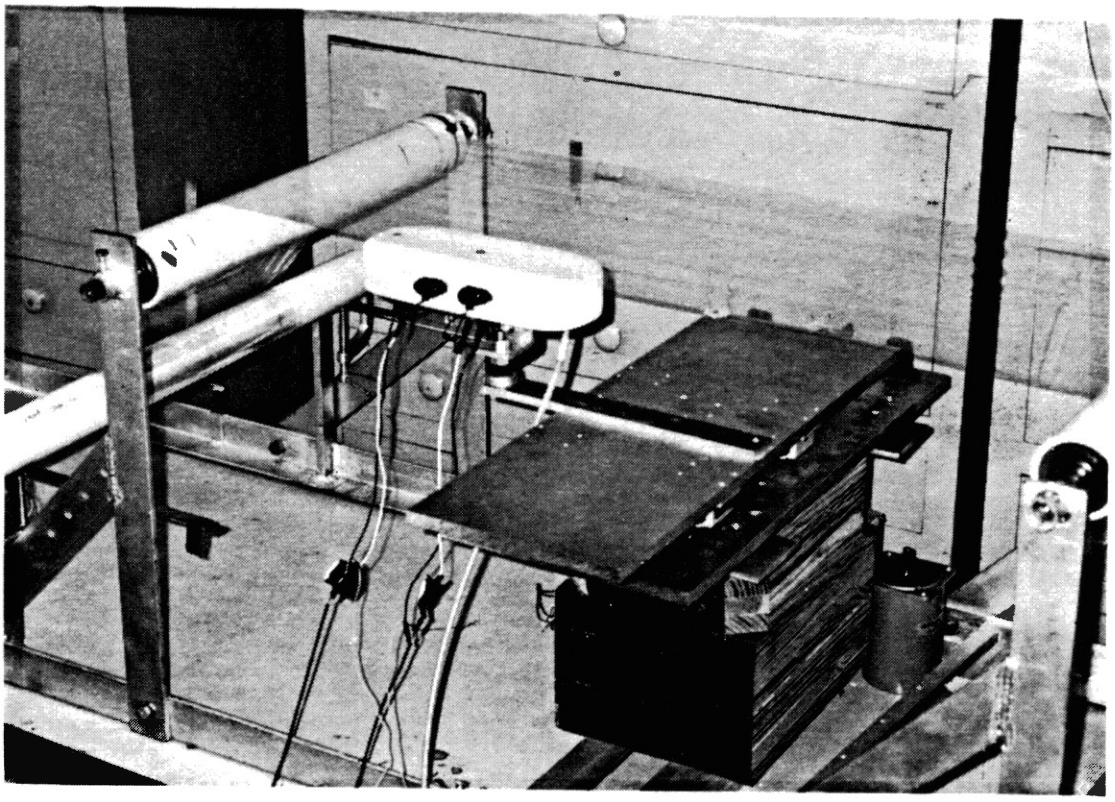


Figure 3.2-3. Long Transducer Head in Use on Laboratory Test Stand

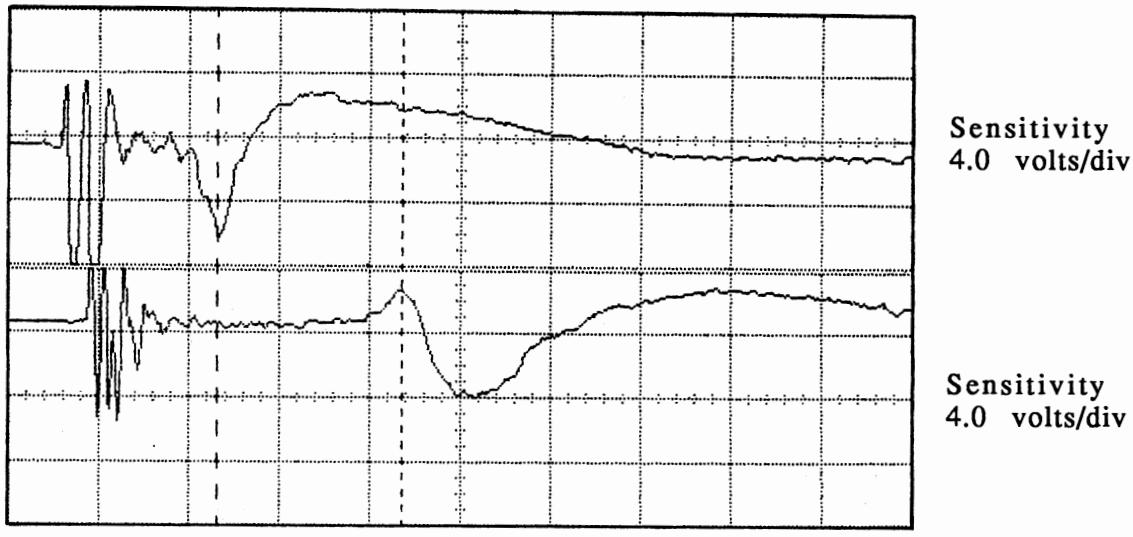


Figure 3.2-4. Raw Signal Acquired Through Use of Short Transducer Head

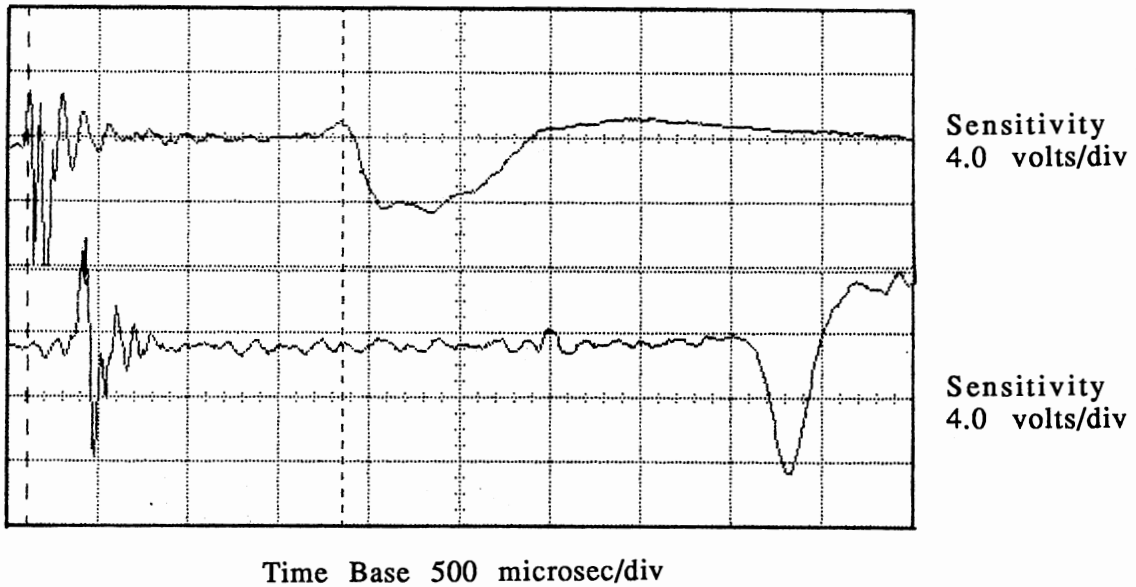
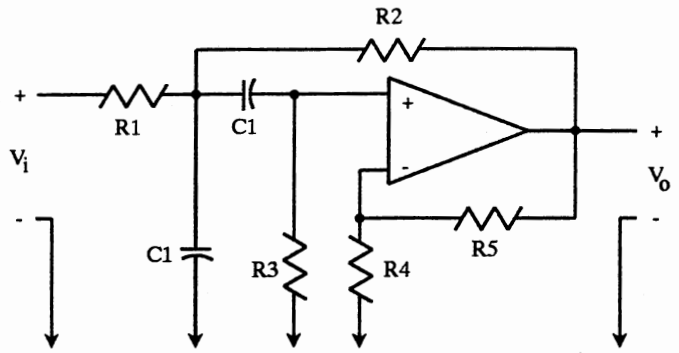


Figure 3.2-5. Raw Signal Acquired Through Use of Long Transucer Head of Figure 3.2-3



$$H_s(j\omega) = \frac{K\ B\ \omega_c\ (j\omega)\ /\ Q}{(j\omega)^2 + (j\omega)\ \omega_c\ /\ Q + \omega_c^2}$$

K = Gain; Q = Quality Factor

B = Butterworth Coefficient

C1 Selected $\approx 10/f_c\ \mu F$

$$R1 = 2\ Q/K\ C1\ \omega_c$$

$$R2 = \frac{2}{\left[-\frac{1}{Q} + \sqrt{\left(\frac{K}{Q} - \frac{1}{Q}\right)^2 + 8}\right] \omega_c\ C1}$$

$$R3 = \frac{1}{\omega_c^2\ C1^2}\left(\frac{1}{R1} + \frac{1}{R2}\right)$$

$$R4 = R5 = 2\ R3$$

Figure 3.2-6. Bandpass Filter Specifications [58]

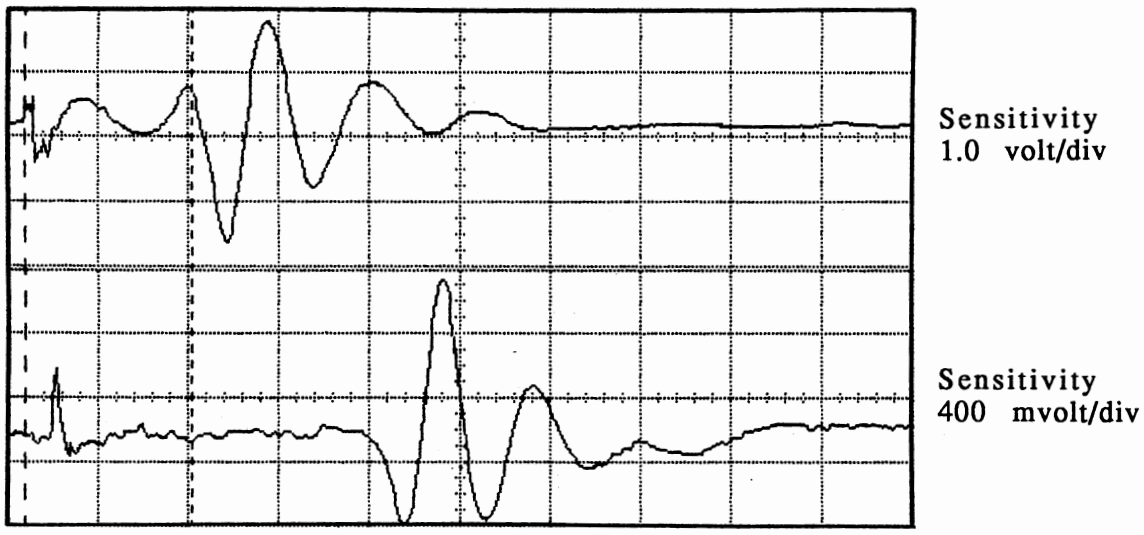


Figure 3.2.7. Filtered Signal Acquired Through Use of Long Transducer Head

The first field tests at the Shawnee, Oklahoma, Mobil Chemical facility were performed on February 15, 1990. Test system consisted of the rotary pulser, long transducer head, filter circuitry, and an HP 54501A digital oscilloscope. Figure 3.2-8 is an example of a typical waveform obtained where the associated winding frame load cell indicated 81 pounds tension over a 120-inch web width. Figure 3.2-9 is a similar result at 88 pounds tension. Comparing the two figures, the time interval given by the oscilloscope cursors varied with tension. These results are based on load cell supplied average tension and not necessarily the exact tension at the test location. Generally, results of this field test were considered positive in that strong signals were obtained from the pneumatic pulse/microphone sensing system under conditions of variable and often substantial air gap between the transducer shoe and web. Questions remained regarding the shape of the microphone signals and proper interpretation of these signals.

A study was performed to assess the validity of signals obtained from the tension measurement system with respect to actual perceived web motion. Figures 3.2-8 and 3.2-9 exhibit common web waveform characteristics. The upstream microphone displayed basically a positive leading lip followed by a dominant downward (negative) pulse followed by an upward (positive) rebound pulse. The downstream microphone exhibits nearly an inversion of this waveform. Microphone polarity was tested such that an increase in pressure (compression) corresponded to a positive voltage output whereas a decrease in pressure (vacuum) corresponded to a negative voltage output. Microphone response knowledge could allow for interpretation of the acquired signals to formulate a general hypothesis regarding web response to the pneumatic input pulses.

Web air loading during response to the pneumatic impulse inputs appeared to influence the shape of acquired waveforms. In the analysis of Figures 3.2-8 and 3.2-9, a point to remember is that the pulser tube and sensing microphones were all housed in the transducer head assembly on a common web side. Considering upstream microphone signals, the positive waveform lip prior to the dominant downward pulse indicates a small compressive load prior to a dominant vacuum load. Following this vacuum load is a positive rebound, or compressive load.

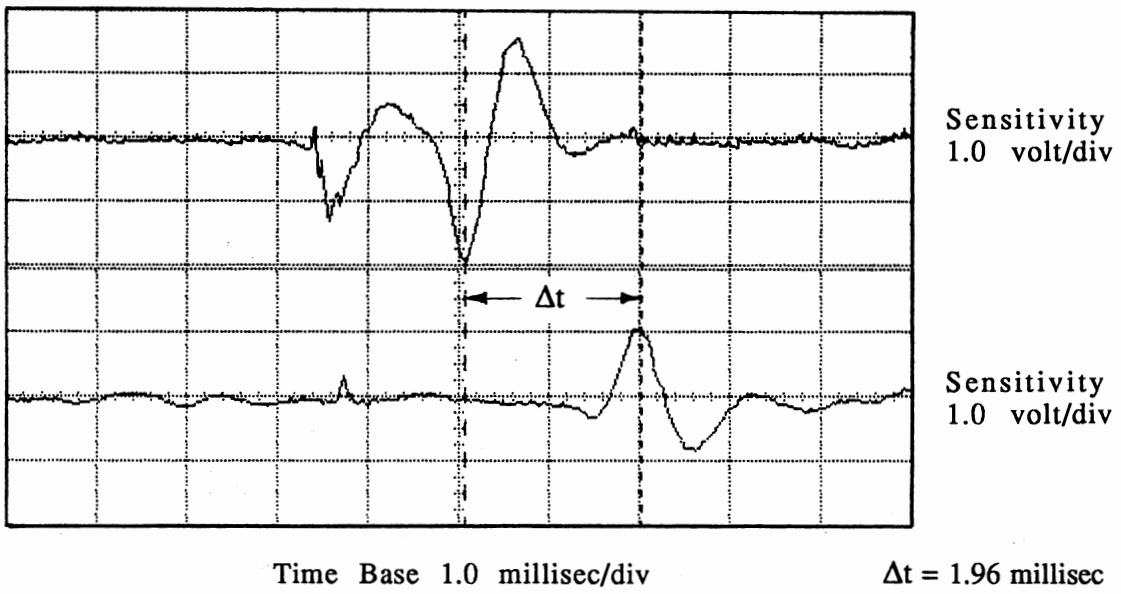


Figure 3.2-8. Trace From Shawnee Test for 81-Lb Tension Over a 120-In. Web Span

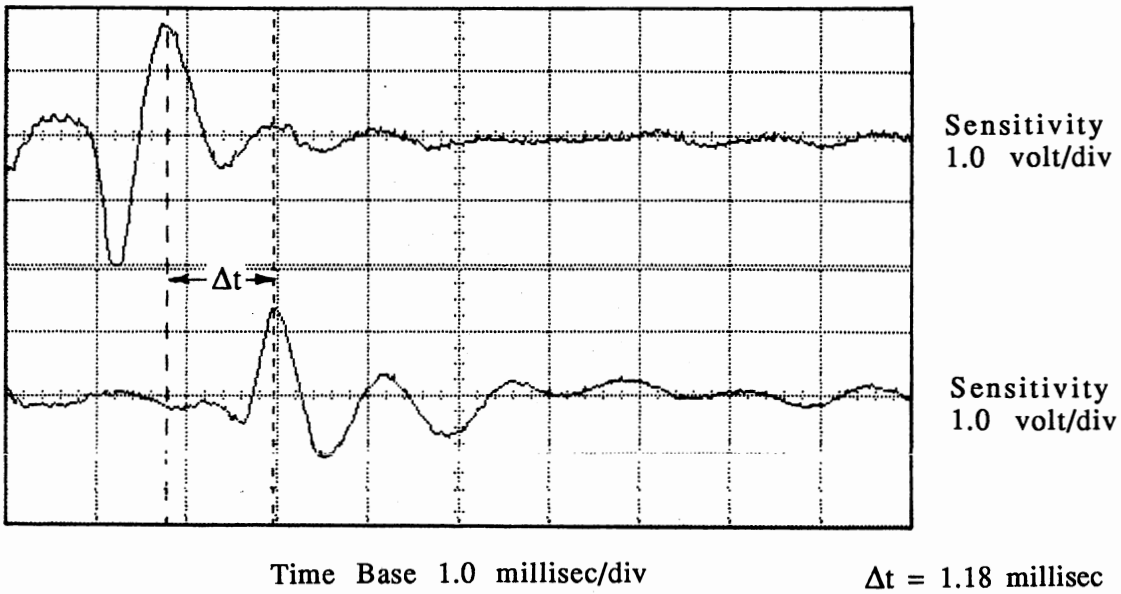


Figure 3.2-9. Trace From Mobil Test for 88-Lb Tension Over a 120-In. Web Span

Figure 3.2-10 is an illustration of the qualitative hypothesis formulated to explain this general waveform characteristic. It was believed that air residing atop the web would be displaced outward from the impulsive input center point. This air mass would cause the initial compressive lip that was described above (Figure 3.2-10b). Next, the major web response to the impulse input would force the web away from the microphone, resulting in the relative vacuum condition (Figure 3.2-10c). The rebound condition would then provide the final compressive load (Figure 3.2-10d). Hence, signals captured via the sensing microphones appear to be dependent on pneumatic input pulse-to-web coupling, the web structure mechanics, and the surrounding air load effects on the web dynamic response.

The hypothesis outlined above explains the shape of a typical waveform captured by the upstream microphone and oscilloscope. It does not explain, however, the "inversion" evident in signals from the downstream microphone. Signals were obtained through use of the long transducer shoe, which was rather flat and smooth to avoid web damage upon contact. It was believed that air from the pneumatic pulse and air drawn into the system by the moving web boundary layer was being trapped by the flat shoe surface. A new transducer head was designed and fabricated, as shown in Figure 3.2-11. This design retained the smooth surface of the prior shoe but provided a means of dispersion of air between the pulser tube outlet and the sensing microphones.

3.3 Refinement of Transducers and Signal Processing

Coincident with development of signal generation and sensing aspects of the project was development of data acquisition and signal processing aspects. Thus far, data have been presented as oscilloscope traces. Time of flight information, subsequently referred to as a "delta t" (Δt) value, was obtained by setting oscilloscope cursors at the "best" positions with respect to the signals obtained from the fore and aft microphones. Thus, a delta t value obtained by this method could be somewhat arbitrary depending on the waveform shape being examined. Desired was a method to eliminate the "eyeball" approach and automate the process. An HP 9816 computer was available for use with the HP 54501A digital oscilloscope for initial development in these areas.

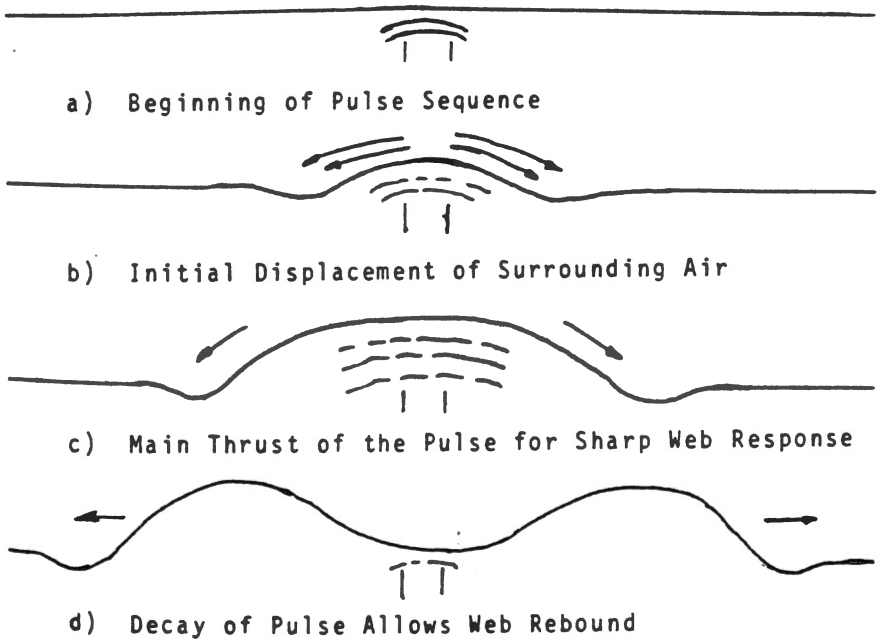


Figure 3.2-10. Illustration of Hypothesis to Explain Shape of Experimental Signals

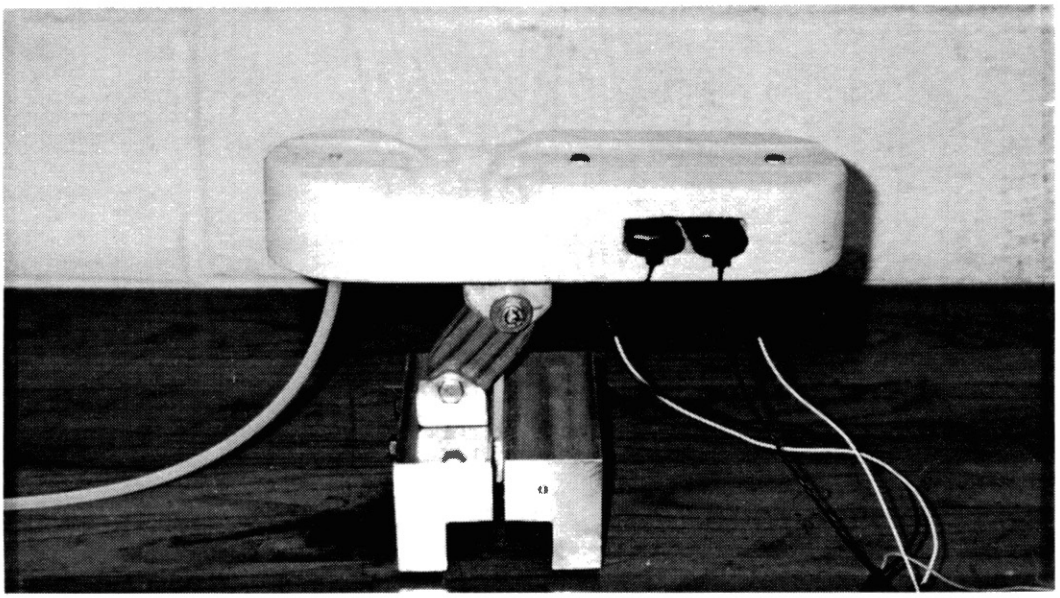


Figure 3.2-11. Revised Transducer Head Design

Remote programming of the HP 54501A digital oscilloscope was accomplished through use of the HP 9816 computer. This allowed for continued digital oscilloscope use, which was a very good readout instrument. Thus, various oscilloscope settings were programmed and data records were obtained, whereupon an HPIB/IEEE488 bus connection allowed for transfer of captured signal data to the HP 9816 computer. This method proved to be adequate in that trigger levels, amount of pre- or post-trigger, and selection of single or averaged records could be programmed and subsequent waveform results noted. Once the digitized waveform record was transferred to the computer, various signal processing schemes could be initiated.

Cross correlation was utilized for determination of the time of flight, Δt value for a given test. Captured by instrumentation was one complete web pulse cycle as follows. Airburst pneumatic shock "snap" was used as the oscilloscope trigger source, whereupon oscilloscope post-trigger facility was used to exclude this airburst noise from the final record. Web pulse cycle completion was indicated by the downstream microphone signal. With the cross correlation method, the correlation starting point was somewhat arbitrary. To achieve time of flight information, sufficient record shifts for complete signal overlap had to be performed. Equation (3.4) is the cross correlation function used in this analysis where T_S is the x and y channel sample period [59].

$$\widehat{R_{xy}}(rT_s) = \frac{1}{N-r} \sum_{n=1}^{N-r} x_n y_{n+r} \; ; \; r = 0, 1, 2, \ldots, m \tag{3.4}$$

Figure 3.3-1 is an example of a tension test waveform captured through the automated data acquisition system with the transducer head design of Figure 3.2-11. Again, "inversion" of the web signal occurred between fore and aft microphone sensing. Figure 3.3-2 is the cross correlation function achieved from the waveforms of Figure 3.3-1. Due to the waveforms being somewhat inverted with respect to each other, one would expect the maximum negative correlation value to represent the point of maximum overlap of the two signals. From Figure 3.3-2, the index corresponding to the point of maximum negative correlation occurred at shift number 202. Delta t value was then found through knowledge of the shift number and oscilloscope sampling rate. From Figure 3.3-1, the oscilloscope acquired 512 samples in 5 milliseconds such that the sample period T_S was found by:

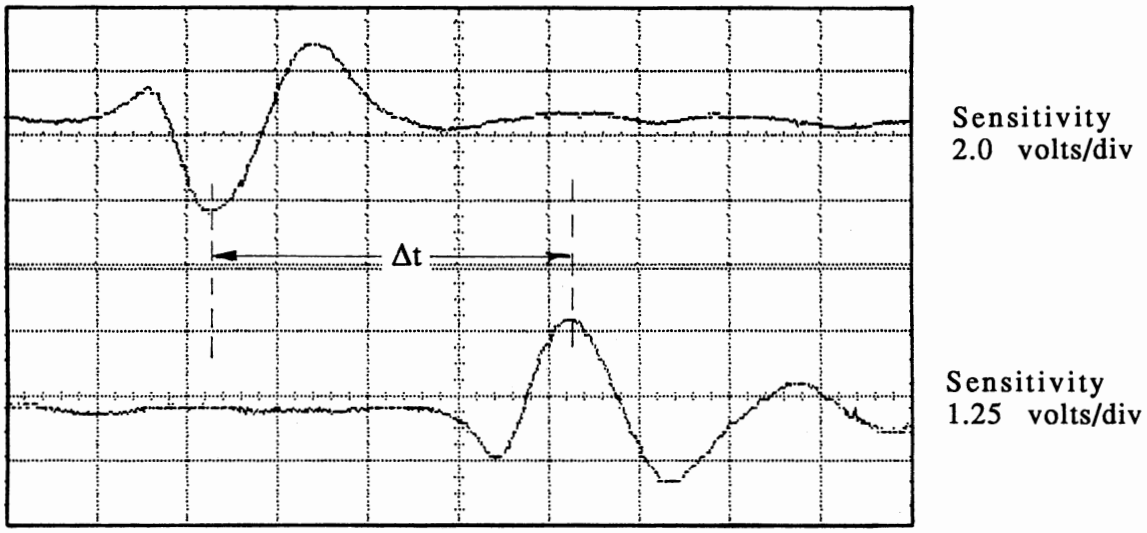


Figure 3.3-1. Typical Signal Captured by Automated System

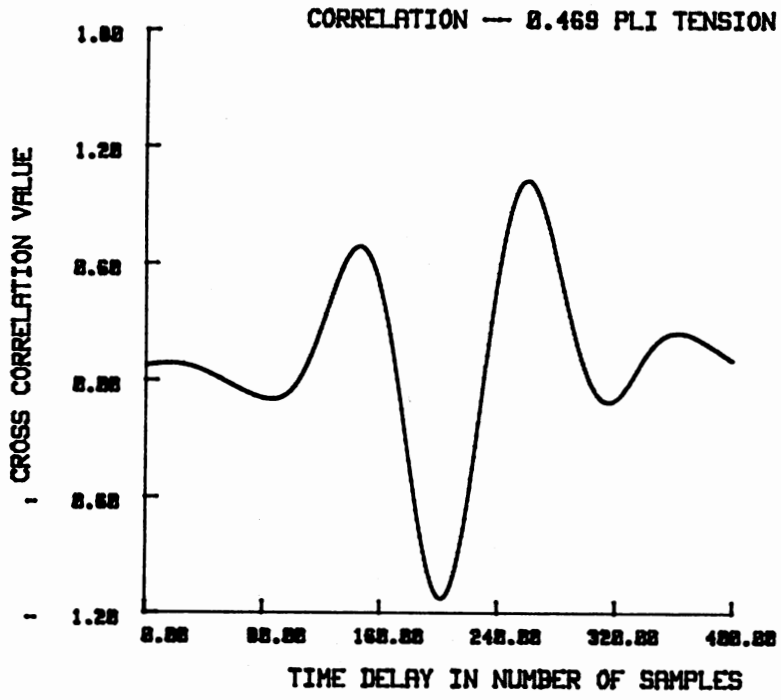


Figure 3.3-2. Cross Correlation Function Corresponding to Waveforms of Figure 3.3-1

$$T_S = 5.0\ (10^{-3})\ \text{sec}/512\ \text{samples} = 9.7656(10^{-6})\text{sec/samples} \quad (3.5)$$

The delta t value was then found by:

$$\Delta t = (202\ \text{shifts})\ (9.7656(10^{-6})\ \text{sec/shift}) = 1.9726\ (10^{-3})\ \text{sec} \quad (3.6)$$

Examination of Figure 3.3-1 provides for a quick estimate of this delta t value:

$$\Delta t \approx (4\ \text{divisions})\ (500\ (10^{-6})\ \text{sec/division}) = 2.0\ (10^{-3})\ \text{sec} \quad (3.7)$$

Thus, cross correlation procedures have aided in determination of time of flight information for this tension measurement system under development. This computational improvement, the new transducer head, and an improved web traverse fixture were then taken back to Shawnee Mobil plant for additional field tests.

The second round of tension measurement system field testing at Mobil Chemical was performed March 22, 1990. Test objectives were to again examine signals in the industrial environment, to assess performance of transducing and signal conditioning subsystems, and to verify automated data acquisition/data processing capability. Added to the system hardware was an upgraded web traverse to allow arbitrary transducer head positioning along the web span. This new traverse was comprised of a rigid length of square tubing along which a roller bearing transducer head mount would glide. This addition eliminated vibrational effects that led to large, inconsistent air gaps between the web and transducer head that were experienced in the first field test sequence.

Conditions for this second sequence of tests were as follows. A plastic web of 1.25 mil thickness and 0.91 specific density was being processed at roughly 74 pounds tension across a 10 foot wide web span. A bowed roller was in use at the test position, which would affect the inherent air gap between the web and transducer head. Figure 3.3-3 is an example of a captured signal with the transducer head located near the web midspan and slightly touching the web. Figure 3.3-4, by contrast, is an example of a signal achieved with the transducer head near the web edge, where a weaker signal was available due to a substantial air gap, approximately 1/2 inch, between the transducer shoe and web. Of interest was to see if variable signal strength, caused by the bowed roller, would affect time of flight delta t computation.

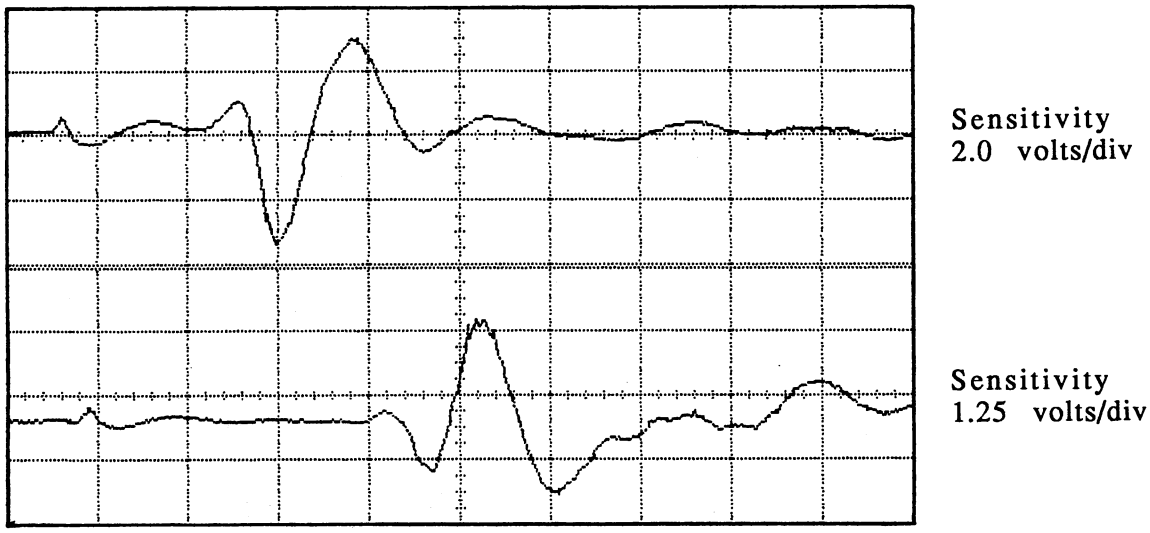


Figure 3.3-3. Signals From Mobil Chemical Test With Transducer Head Near Web Midspan and Slightly Touching Web

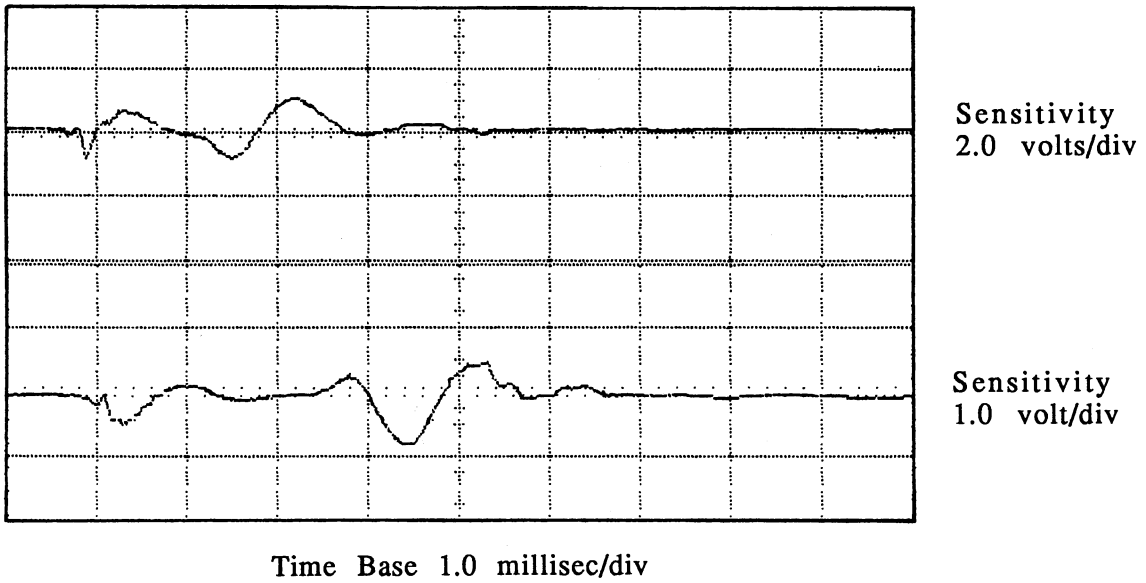


Figure 3.3-4. Signals From Mobil Chemical Test With Transducer Head Near Edge of Web With Large Airgap Present

The automated data acquisition system was thoroughly tested in this industrial environment. Figure 3.3-5 is a trace captured by the oscilloscope under program control. Oscilloscope Averaging Mode was used such that four records were averaged and displayed. Figure 3.3-6 is the cross correlation function achieved from this data set. For this particular case, maximum negative correlation value occurred at shift location 230, yielding a delta t value of:

$$\Delta t = (230 \text{ shifts})\ (9.7656\ (10^{-6})\ \text{secs/shift}) = 2.246\ (10^{-3})\ \text{secs} \qquad (3.8)$$

This method of delta t derivation proved to be quite insensitive to signal strength. Values obtained were very consistent with respect to values estimated from oscilloscope traces.

Success in signal processing aspects of the tension measurement system was achieved at this point, but transducer head performance was still in question. In the effort to create a smooth transducer head surface to avoid web damage, the wave travel mechanism had been adversely affected, resulting in the "inversion" of a propagating flexural waveform in travel from the upstream to downstream sensor. Trapped air was considered a source of the problem, hence the transducer head redesign as was shown in Figure 3.2-11. When this inversion problem continued, a new hypothesis was needed. It was believed the transducer shoe surface, being large with respect to the microphone sensors, was somehow causing an impedance change in the web, thus affecting the traveling web waveform phase.

A transducer head was designed and fabricated which utilized many features of the Bradley transducer head (Figure 3.1-2) while also providing protection against both web snags and acoustic reflections. Figure 3.3-7 is a schematic of this new transducer head, where the airfoil shaped floats and foam padding were intended to diffuse flexural wave and acoustical noise reflections. Metal tubes served to position the pulser tube and microphones while having minimal effect on the web during the wave propagation process. A new set of band pass filter parameters was implemented at this time. Filter center frequency was shifted to 1700 Hz with bandwidth limits roughly 1000 to 4000 Hz. With these changes laboratory trial tested, a field test was scheduled for industrial environment verification.

Mobil Chemical industrial testing was resumed on April 10, 1990. Test objectives were to test performance of the new transducer head design. Figure 3.3-8 is a trace obtained through the

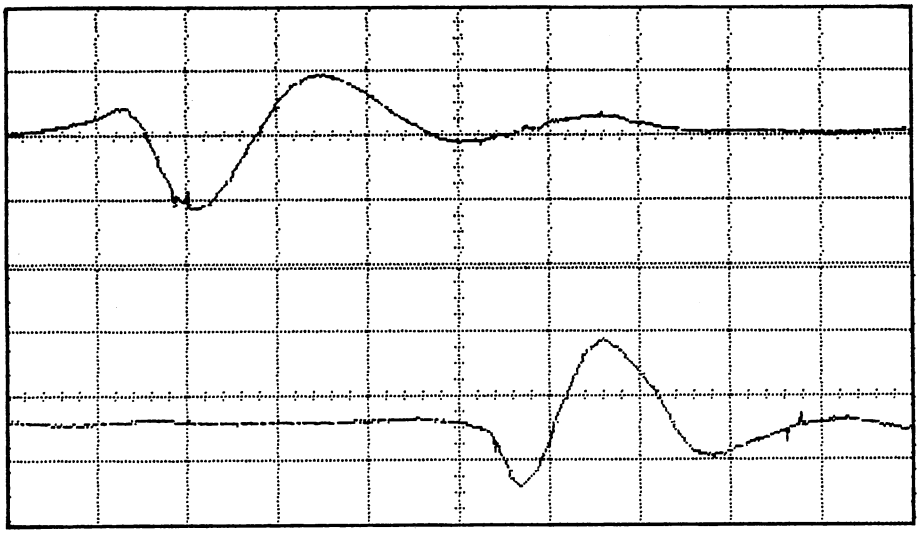


Figure 3.3-5. Signals From Mobil Chemical Tests Captured Through Automated Data Acquisition System

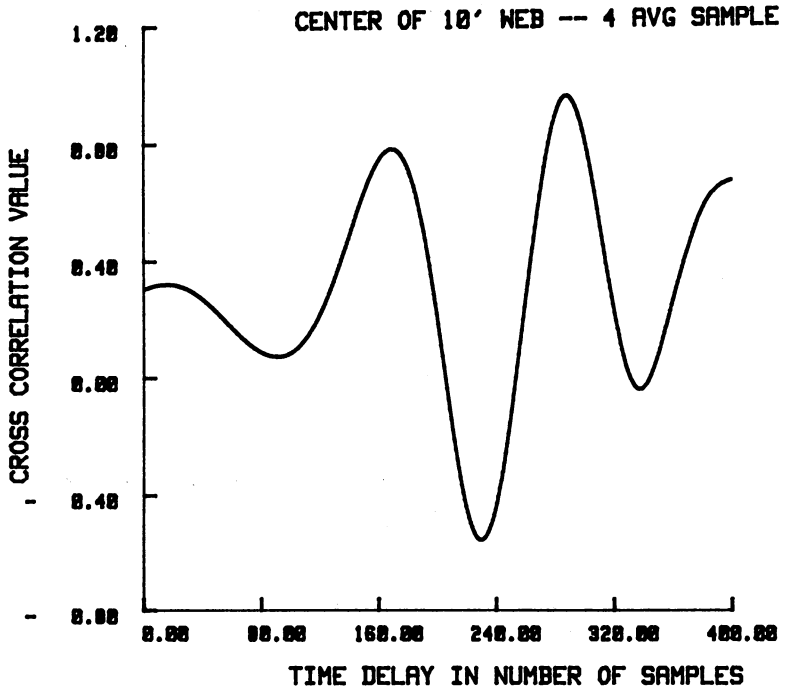


Figure 3.3-6. Cross Correlation Function Corresponding to Waveforms of Figure 3.3-5

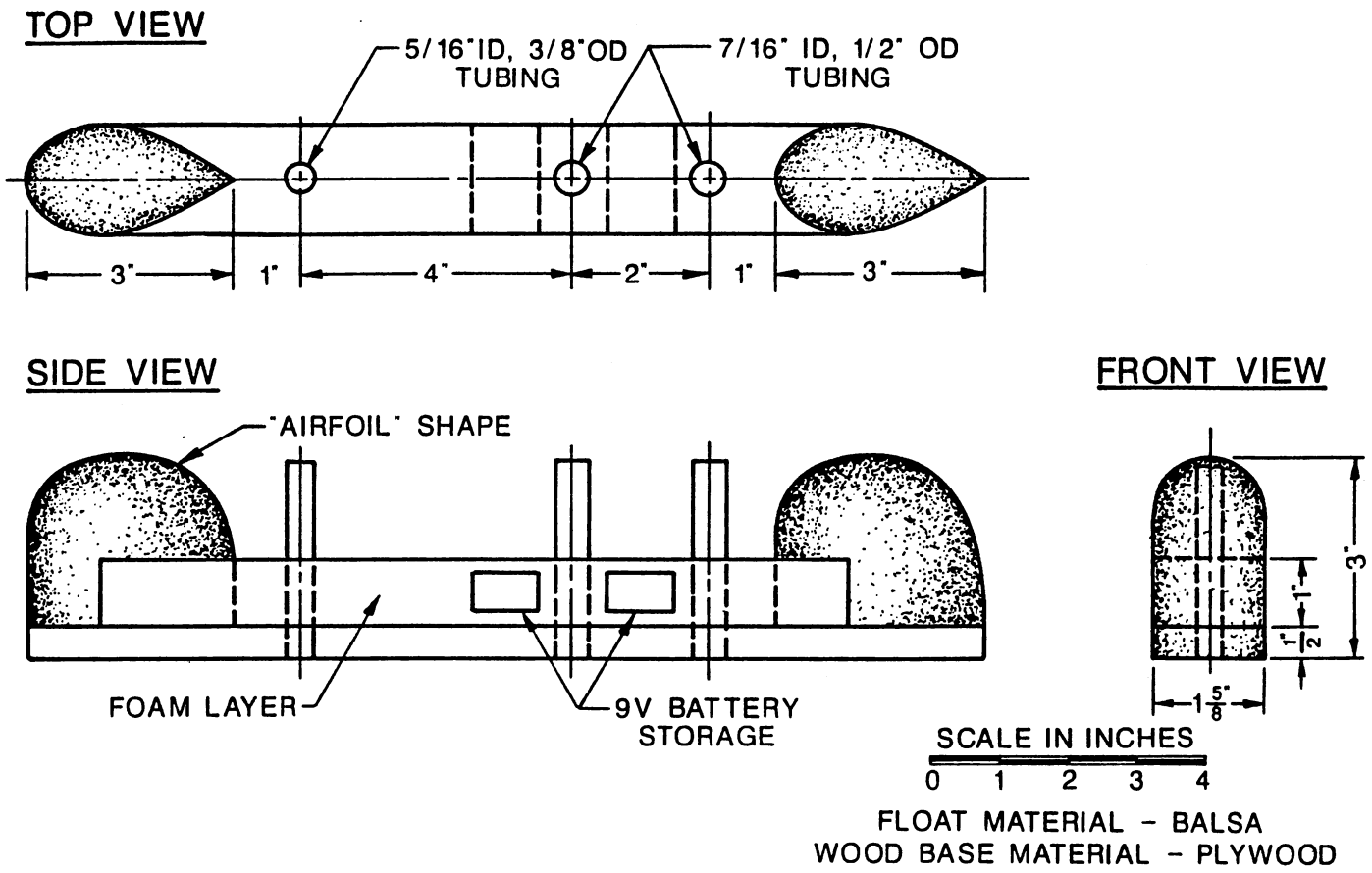


Figure 3.3-7. New Transducer Head Design to Alleviate Problem of Waveform Inversion Behavior

HP 9816 computer controlling program. Both upstream and downstream microphones exhibit the same characteristics in that an initial upward lip is followed by a dominant downward pulse. Web rebound follows the dominant downward pulse. Oscilloscope cursor positioning provided for an estimated delta t value, which yielded 1.01 milliseconds. Figure 3.3-9 is the cross correlation function corresponding to the Figure 3.3-8 trace. Since the two signals were of the same polarity, the maximum positive correlation value is of interest, which occurred at shift number 109. Delta t for this case is then:

$$\Delta t = (109 \text{ shifts}) (9.7656 (10^{-6}) \text{ secs/shift}) = 1.06 (10^{-3}) \text{ secs} \quad (3.9)$$

The sample rate used in this case could have been increased while still capturing the necessary information on the scope screen. Figure 3.3-10 is an example from this test sequence whereupon a faster sample rate was used, where oscilloscope cursors provide an approximate delta t value of 0.928 milliseconds. Figure 3.3-11 is the corresponding cross correlation function. Maximum correlation value occurred at shift number 262 such that delta t in this case is given by:

$$\Delta t = (262 \text{ shifts}) (3.906 (10^{-6}) \text{ secs/shift}) = 1.02 (10^{-3}) \text{ secs} \quad (3.10)$$

Correlation process resolution was increased by using a higher sample rate. A tradeoff, however, was that an increased sample rate implied that more correlation shifts would be necessary to insure complete signal overlap and thus a longer data processing period.

At this point in the tension measurement system development the need for versatility and speed with respect to automation, program handling, data acquisition, and data analysis called for a new approach. An IBM compatible computer was acquired for use with the Metrabyte A-D board used by Magee during the Bradley project development. These changes in data acquisition methodology would allow for faster programming changes and faster signal processing capability compared to the HP 9816 computer. Meanwhile, project associates were developing a new pulser system where a solenoid valve would gate the pneumatic pressure pulses. Driven by an adjustable clocking source, such a system could be designed to pulse at faster rates than were presently available with the rotary valve pulser. Looking toward the need to measure tension profiles for wide web spans at Mobil Chemical, a stepper motor assembly was integrated into the traverse subsystem for computer control of transducer head location. Finally, an analysis was performed

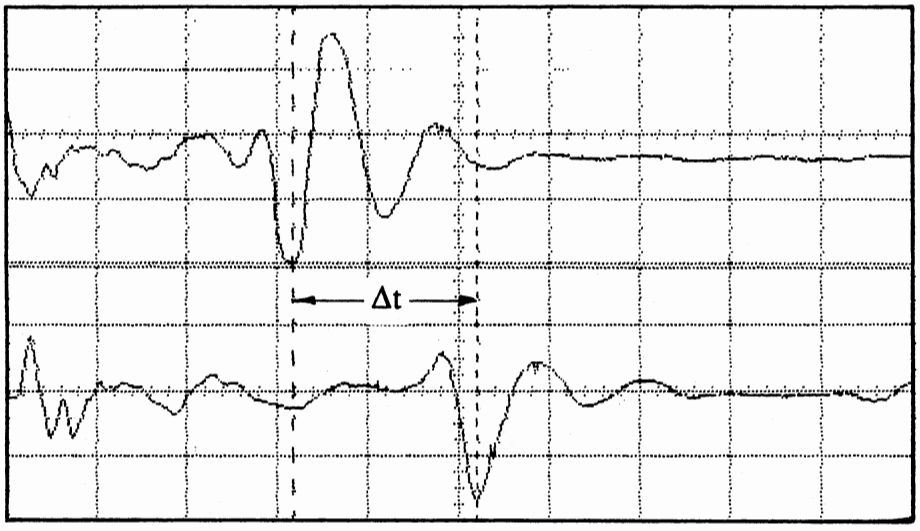


Figure 3.3-8. Traces From Mobil Chemical Test With Automated System Using a Relatively Slow Sample Rate

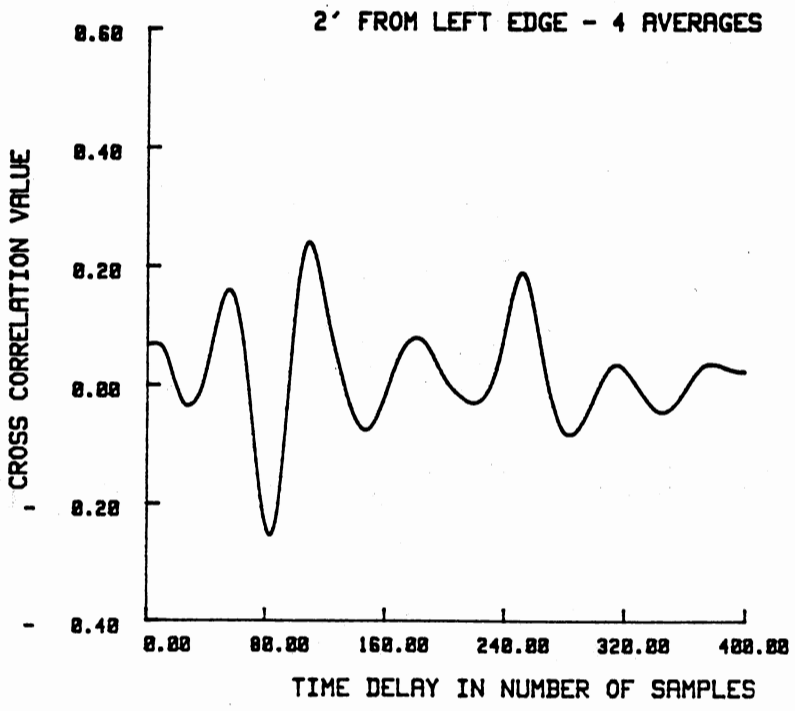


Figure 3.3-9. Cross Correlation Function Corresponding to Waveforms of Figure 3.3-8

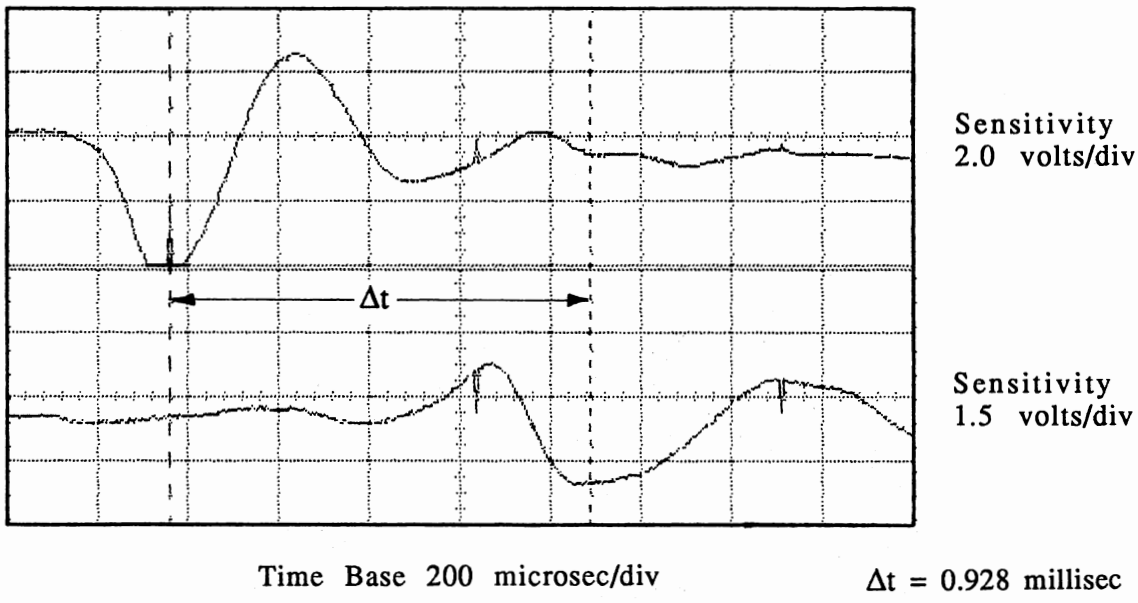


Figure 3.3-10. Traces From Mobil Chemical Test With Automated System Using a Relatively High Sample Rate

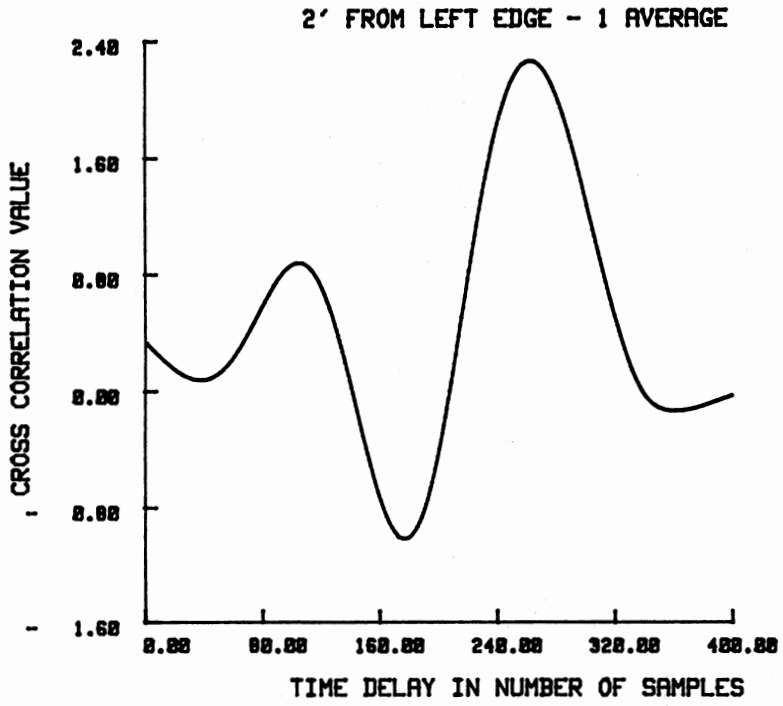


Figure 3.3-11. Cross Correlation Function Corresponding to Waveforms of Figure 3.3-10

of the conversion of acquired experimental data to tension indicators. New data sampling, assembly, and analysis techniques were to be implemented to help improve the tension values derived from this experimental method.

3.4 Analysis and Implementation of System Model

To this point, tension measurement system discussion has revolved around obtaining a time of flight, or delta t value. As was indicated earlier in this study, the ribbon equation model was to be employed for conversion of experimental data to tension indications. The ribbon equation is essentially a corrected version of the vibrating membrane wave equation. Uniform tension applied to membrane boundaries and differential stress analysis results in the linear membrane equation below based on notation of Figure 3.4-1:

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = \frac{1}{C^2_{membrane}} \frac{\partial^2 w}{\partial t^2} \; ; \; C_{membrane} = \sqrt{\frac{T}{\rho_{membrane}}} \tag{3.11}$$

where w is membrane transverse deflection. Thus, speed of wave propagation in a membrane is proportional to the square root of the membrane tension in a vacuum condition.

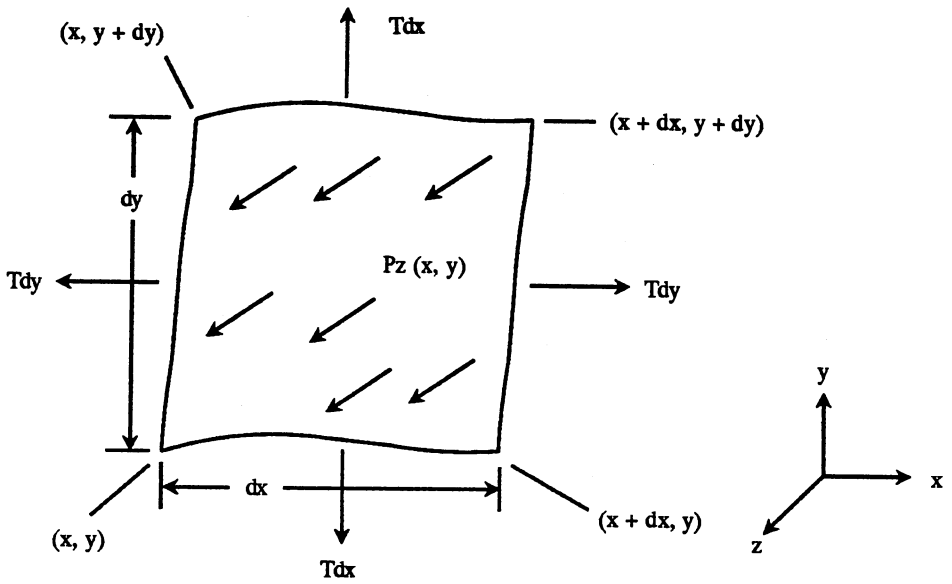


Figure 3.4-1. Notation for Linear Membrane Equation (3.11) [60]

To correct the membrane equation for air loading, a frequency dependent air loading force is added to the base relation (3.11). An assumed solution to the resulting partial differential equation coupled with compatibility relations between air particles and membrane surface allow for derivation of the dispersion relation for a membrane vibrating in air [61]:

$$\omega^2 = \frac{T\ K^2_{membrane}}{\rho_{membrane} + \frac{2\ \rho_{air}}{\sqrt{K^2_{membrane} - K^2_{air}}}} \tag{3.12}$$

where $K_{air} = \omega/C_{air}$ and $K_{membrane} = \omega/C_{membrane}$ are wave numbers for air and membrane in vacuo, and ω is the membrane excitation frequency.

A basic assumption in membrane deflection equation derivation is that of zero flexural rigidity. Many web materials exhibit a non zero flexural rigidity, and thus have characteristics of plates. Additionally, webs typically experience tension in one direction only whereas the membrane derivation assumes equal tension along all boundaries. The ribbon equation model, however, was believed to be a good basis for tension measurement data conversion. For a web application, the ribbon equation becomes:

$$T = C^2_{web}\left(\rho_{web} + \frac{2\ \rho_{air}}{\sqrt{K^2_{web} - K^2_{air}}}\right) \tag{3.13}$$

where $K_{web} = \omega_{web}/C_{web}$ and $K_{air} = \omega_{web}/C_{air}$. Lee [56] and Bradley [57] provide additional background material regarding the adaptation of this equation to the proposed tension measurement system. Quantities ρ_{web}, ρ_{air}, and C_{air} would be considered constant with variables being web phase velocity C_{web} and the induced flexural wave characteristic frequency $\omega_{web} = 2\ \pi\ f_{web}$.

Common to the in-vacuo membrane model and the fluid loaded membrane model is the speed of sound in each medium. This quantity was to be derived experimentally through the time of flight information. From the project beginnings, Equation (3.11) was used with time of flight information to arrive at a crude tension estimate via:

$$T = \rho_{web}\, C^2_{web} = \rho_{web}\left(\frac{X}{\Delta t}\right)^2 \tag{3.14}$$

where X is the distance between transducer head sensing microphones. Originally desired was to excite the web at such a high frequency that the correction term in Equation (3.13) would be

negligible. If this were the case, then Equation (3.13) would degenerate to Equation (3.14). The impulsive pneumatic input idea was partially based on this concept of high frequency stimulus due to the impulse function theoretically having infinite bandwidth. Problems due to dispersive and coupling effects, however, forced use of the ribbon equation with the frequency dependent correction factor.

Good laboratory tension estimates were achieved using cross correlation time of flight values and an assumed characteristic frequency value in the ribbon equation. Field testing at Mobil Chemical was performed using this arrangement, where a characteristic frequency value was selected based on actual oscilloscope signal traces. Results of these tests were not as accurate as was anticipated. Subsequent experimentation led to the belief that further adjustment of filtering, sampling process, record averaging, and characteristic frequency selection would not sufficiently improve the tension indication quality. A more in depth ribbon equation analysis was performed to examine the effect of web characteristic frequency on ribbon equation tension values.

Equation (3.13) was written in general terms including speeds of sound C_{web} and C_{air} and wave numbers K_{web} and K_{air}. To relate this equation to the measurement system parameters and measured quantities, Equation (3.13) may be rewritten as:

$$T = \left(\frac{X}{\Delta t}\right)^2 \left(\rho_{web} + \frac{2\ \rho_{air}}{(2\pi f_{web}) \sqrt{\left(\frac{\Delta t}{X}\right)^2 - \left(\frac{1}{C_{air}}\right)^2}} \right) \tag{3.15}$$

where X = distance between sensing microphones (inches), Δt = experimental time of flight value (seconds), f_{web} = characteristic frequency of propagating flexural wave (Hz), ρ_{air} = air density ($lbm/in.^3$), and ρ_{web} = web area density ($lbm/in.^2$). The nature of this equation is somewhat difficult to visualize due to the dependence of tension on two variables, Δt and f_{web}. From experimentation, coupling of these two variables was hard to quantify because a tension increase causes the induced flexural wave to speed up, causing pressure fluctuations sensed by system microphones to occur more quickly, creating an apparent increase in web characteristic frequency. One must decide how much of an apparent ribbon equation tension value increase may be attributed to a phase velocity change and how much may be attributed to a characteristic frequency

change. Other factors affect the experimentally viewed web response such as pneumatic pulse sharpness, coupling aspects, and web mechanics aspects. Since little control had become apparent with respect to the experimental system coupling and mechanics aspects, the objective for enhanced system performance then involved determination of the relative Δt and f_{web} variables contribution of the to the overall tension indications produced.

Sensitivity of a multivariable function F to a variable q_i may be expressed as a vector with components $\partial F/\partial q_i$ such that a perturbation in a variable δq_i results in a perturbation in the function value δF [62]. This perturbation δF_{q_i} due to δq_i may be approximated by:

$$\delta F_{q_i} = F(q_1,q_2,..,q_i+\delta q_i,..,q_n) - F(q_1,q_2,..,q_i,..,q_n) \cong \frac{\partial F}{\partial q_i}\delta q_i \tag{3.16}$$

For an n variable function, vector representation may be used to express the perturbation in F due to n component perturbations:

$$\delta F \cong \left\{\frac{\partial F}{\partial q_1}, \frac{\partial F}{\partial q_2}, \ldots, \frac{\partial F}{\partial q_i}, \ldots \frac{\partial F}{\partial q_n}\right\} \begin{Bmatrix} \delta q_1 \\ \delta q_2 \\ \vdots \\ \delta q_i \\ \vdots \\ \delta q_n \end{Bmatrix} \tag{3.17}$$

This result is often expressed as the uncertainty of a function measurement indication due to uncertainty in each of the individual function variable measurements. Maximum uncertainty is achieved through summation of absolute component uncertainty values [63]:

$$U_F = \left|\frac{\partial F}{\partial q_1}\delta q_1\right| + \left|\frac{\partial F}{\partial q_2}\delta q_2\right| + \cdots + \left|\frac{\partial F}{\partial q_i}\delta q_i\right| + \cdots + \left|\frac{\partial F}{\partial q_n}\delta q_n\right| \tag{3.18}$$

Applied to the ribbon equation, maximum uncertainty in a tension value due to uncertainty in the f_{web} and Δt variables is given by:

$$U_T = \left|\frac{\partial T}{\partial f_{web}}\delta f_{web}\right| + \left|\frac{\partial T}{\partial \Delta t}\delta \Delta t\right| = U_T^{f_{web}} + U_T^{\Delta t} \tag{3.19}$$

Evaluating the indicated partial derivatives of the ribbon Equation (3.15) provides:

$$\frac{\partial T}{\partial f_{web}} = \frac{-X^2 \rho_{air}}{\pi \Delta t^2 f_{web}^2 \sqrt{\left(\frac{\Delta t}{X}\right)^2 - \left(\frac{1}{C_{air}}\right)^2}} \tag{3.20}$$

$$U_T^{f_{web}} = \frac{x^2 \rho_{air}\, \delta f_{web}}{\pi f_{web}^2\, \Delta t^2 \sqrt{(\frac{\Delta t}{x})^2 - (\frac{1}{C_{air}})^2}} \tag{3.21}$$

$$\frac{\partial T}{\partial \Delta t} = \frac{-\rho_{air}}{\pi f_{web}\, \Delta t \left[\left(\frac{\Delta t}{X}\right)^2 - \left(\frac{1}{C_{air}}\right)^2\right]^{3/2}} - \frac{2x^2 \rho_{web}}{\Delta t^3} - \frac{2x^2 \rho_{air}}{\pi f_{web}\, \Delta t^3 \sqrt{\left(\frac{\Delta t}{X}\right)^2 - \left(\frac{1}{C_{air}}\right)^2}} \tag{3.22}$$

$$U_T^{\Delta t} = \frac{\rho_{air}\, \delta \Delta t}{\pi f_{web}\, \Delta t\, ((\frac{\Delta t}{x})^2)^{3/2}}$$

$$+ \frac{2x^2 \rho_{web} \delta \Delta t}{\Delta t^3} + \frac{2x^2 \rho_{air}\, \delta \Delta t}{\pi f_{web}^2\, \Delta t^2 \sqrt{(\frac{\Delta t}{x})^2 - (\frac{1}{C_{air}})^2}} \tag{3.23}$$

Four variables are present in the above relations: f_{web}, Δt, δf_{web}, and $\delta \Delta t$. Typical values of f_{web} and Δt, consistent with experimental test results, will be used such that variation in tension may be examined due to δf_{web} and $\delta \Delta t$ variations.

Figure 3.4-2 is a plot of web tension versus characteristic frequency f_{web} using five discrete values of flexural wave time of flight Δt. Web density 3.25 (10^{-5}) lbm/in.^{2} was used in this example, which is appropriate for a 0.9 specific density plastic web material 0.001 inches (1 mil) thick. Time of flight values used were selected based on typical results obtained using the transducer head of Figure 3.3-7. The trace corresponding to $\Delta t = 0.001$ seconds will be selected for the following error analysis due to this trace well representing the range of laboratory tension levels examined thus far. Uncertainty in variables f_{web} and Δt will be presented as percentages of the assumed values. A 10 percent uncertainty, for example, would be expressed as $\delta f_{web} = 0.1\, f_{web}$ and $\delta \Delta t = 0.1\, \Delta t$.

Affect of error in characteristic frequency f_{web} was examined through plotting of the $U_T^{f_{web}}$ function of Equations (3.19) and (3.21). Figure 3.4-3 is the result of the above mentioned normalization process, where discrete values of $\delta f_{web} = 0.04\, f_{web}$, $0.08\, f_{web}$, $0.12\, f_{web}$, $0.16\, f_{web}$, and $0.20\, f_{web}$ were considered. In the range of typical characteristic frequency, 1200 to 2200 Hz, a 20 percent error in the assignment of or measurement of characteristic frequency f_{web}

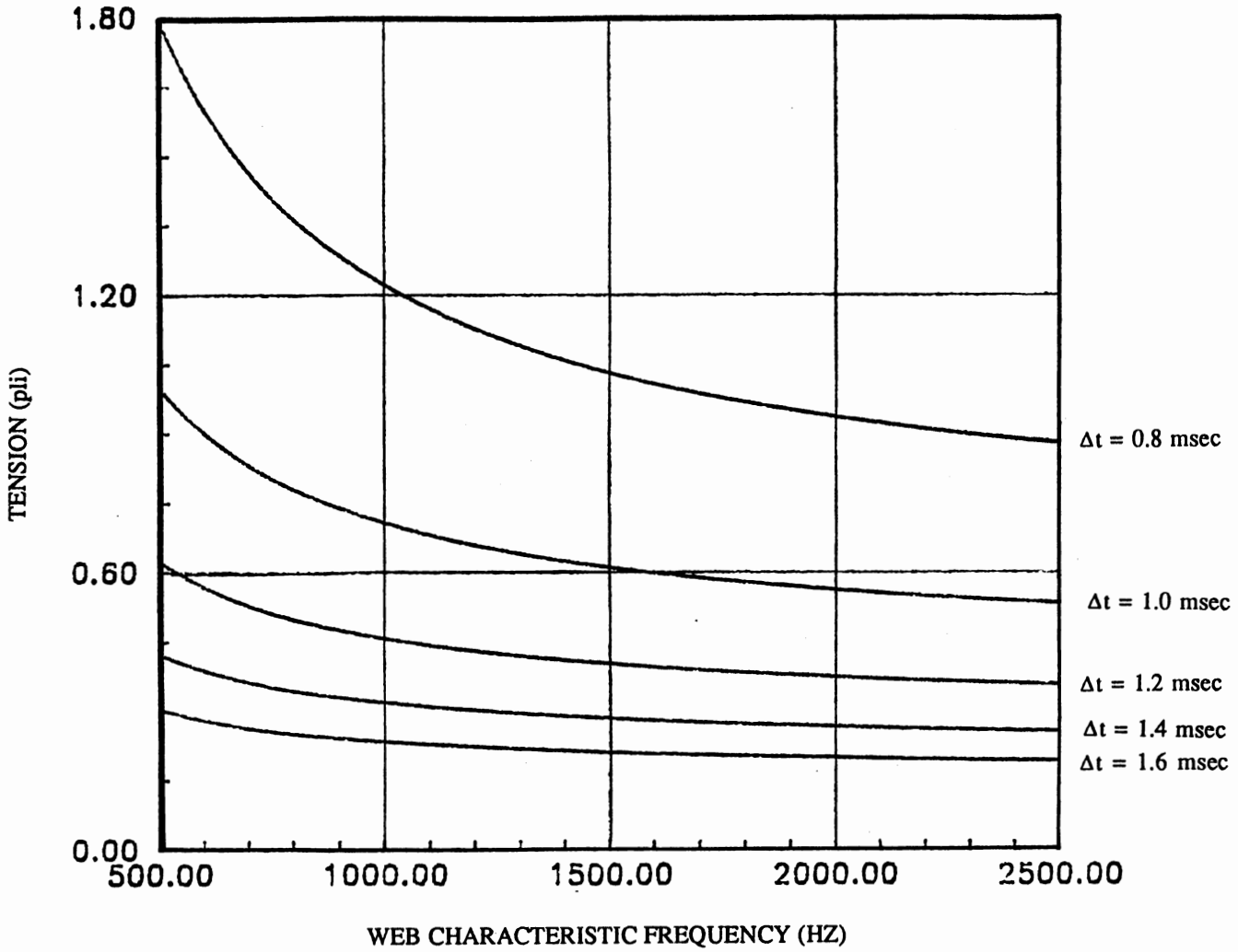


Figure 3.4-2. Ribbon Equation Tension as a Function of Web Characteristic Frequency for Discrete Δt Values

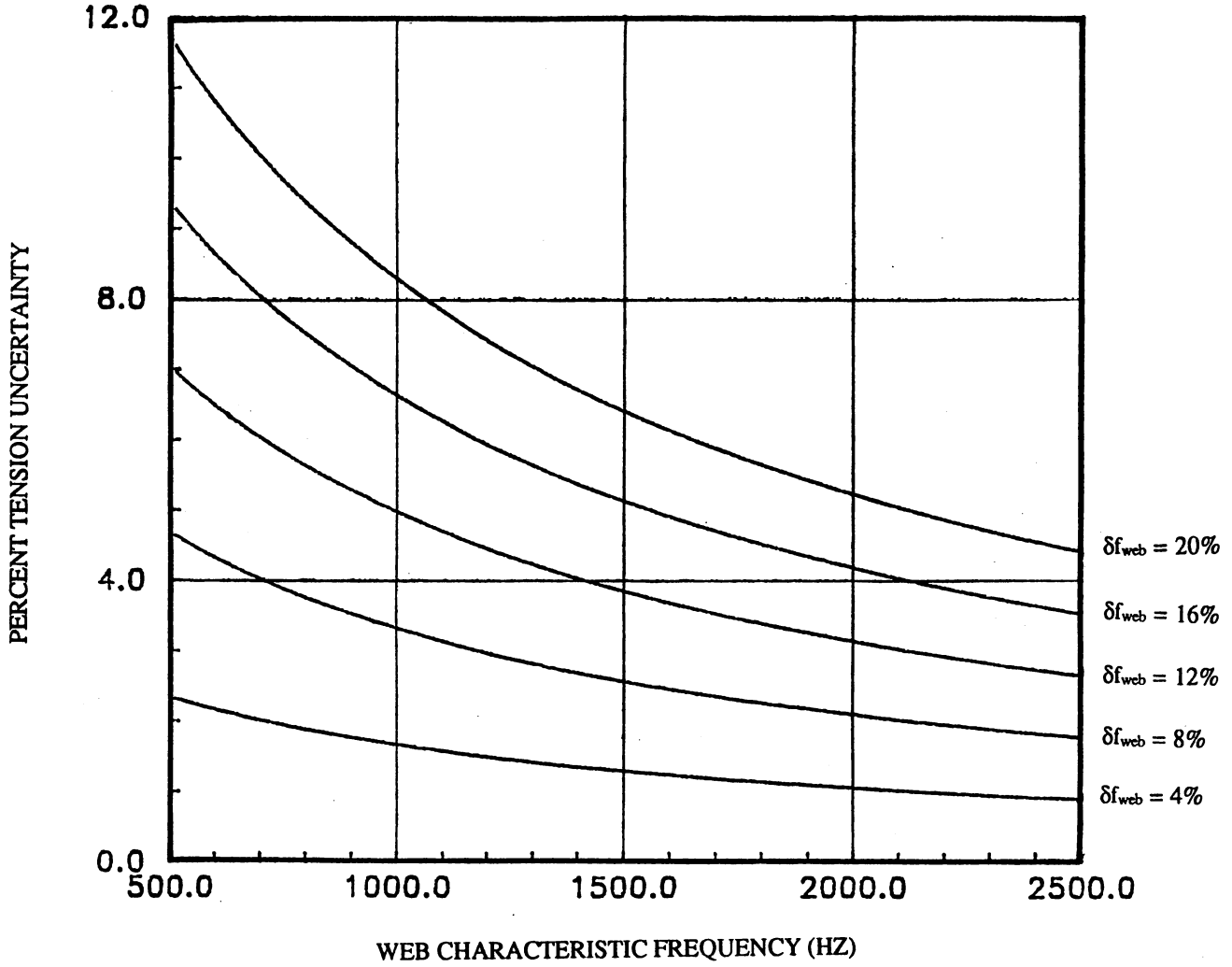


Figure 3.4-3. Percent Uncertainty in Tension Due to Uncertainty in Web Characteristic Frequency

results in a 5 to 7 percent tension value error. As can be seen, a tension error of less than 4 percent could be expected with adequate, within 10 percent, flexural wave characteristic frequency measurement.

Figure 3.4-4 is a similar plot which provides percent uncertainty in tension as a result of error in the time of flight Δt variable. This plot was achieved through evaluation of the $U_T^{\Delta t}$ function of Equations (3.19) and (3.23). Discrete values of time of flight error $\delta\Delta t = 0.04\ \Delta t$, $0.08\ \Delta t$, $0.12\ \Delta t$, $0.16\ \Delta t$, and $0.20\ \Delta t$ were considered. As can be seen in the figure, uncertainty in tension due to error in the Δt value is substantial. A 20% Δt error results in a 50 percent tension error in the 1200 to 2200 Hz characteristic frequency range. Clearly, an accurate measurement of time of flight is necessary for accurate tension indications.

Cross correlation analysis outlined earlier was found to be beneficial in determination of an experimental time of flight value. This Δt value was obtained by examining the cross correlation data for the maximum value, noting the shift index associated with this maximum value, and multiplying this shift index times the data acquisition system sample period per channel. Correlation techniques are of great service in these types of applications due to their ability to smooth out noisy data sequences. Thus, if the point of maximum correlation is regarded as the optimal time of flight value, then uncertainty in this process could be considered to be ± 1 shift or ± 1 sample period as the maximum correlation value is located.

Percent uncertainty in experimental tension results may be evaluated through use of the above cross correlation uncertainty criteria. For example, section 3.3 provided two test examples in which the new transducer head of Figure 3.3-7 was used. These tests used sample periods of: Case 1—9.7656 (10^{-6}) sec/sample (Equation (3.9)) and Case 2—3.906 (10^{-6}) sec/sample (Equation (3.10)). Uncertainty term $U_T^{\Delta t}$ of Equation (3.23) may be evaluated using $\Delta t = 1.06$ msec, $\delta\Delta t = 9.7657\ (10^{-6})$ sec in Case 1 and $\Delta t = 1.02$ msec, $\delta\Delta t = 3.906\ (10^{-6})$ sec in Case 2. Assuming a characteristic frequency of 1700 Hz and web density of $3.25\ (10^{-5})$ lbm/in.^{2}, the nominal tension and uncertainty results for these cases are:

$$\text{Case 1: } T = 0.4436 \pm 0.009526 \text{ pli } (2.15\%) \tag{3.24a}$$

$$\text{Case 2: } T = 0.4853 \pm 0.004349 \text{ pli } (0.90\%) \tag{3.24b}$$

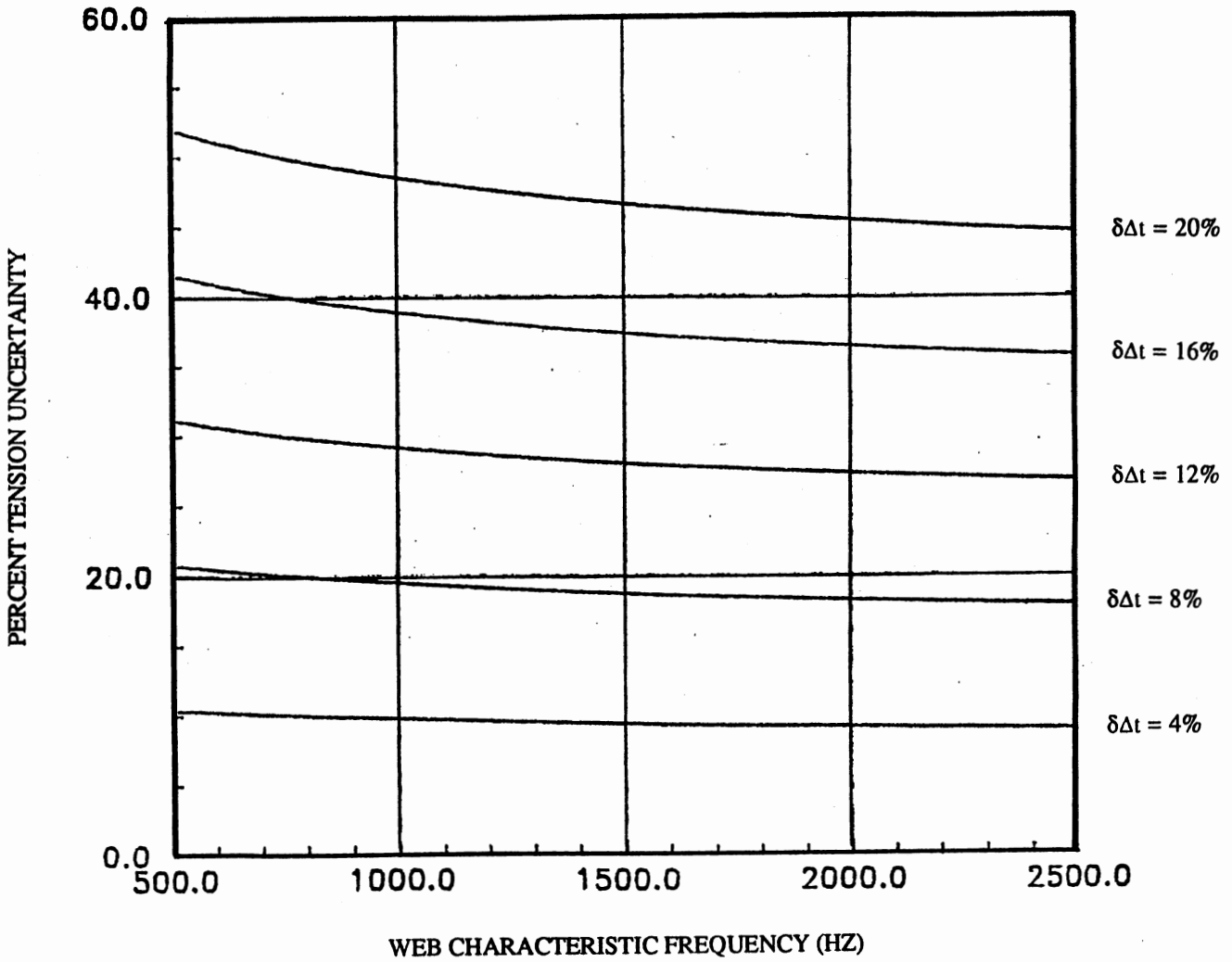


Figure 3.4-4. Percent Uncertainty in Tension Due to Uncertainty in Time of Flight

These results indicate that if the cross correlation process is regarded as the most accurate means of achieving a time of flight value, then uncertainty in time of flight and subsequent uncertainty in the produced tension indication may be held to a very low value.

The above sensitivity/uncertainty analysis was performed due to problems in obtaining accurate tension values from the experimental system. Time of flight information alone was not sufficient to provide for accurate, repeatable tension results. Uncertainty plots, Figures 3.4-3 and 3.4-4, reveal some noteworthy results. From Figure 3.4-4, percent error in tension due to error in Δt is substantial, but does not vary greatly with characteristic frequency f_{web}. Thus, the method used to experimentally derive Δt had provided tension indications close to the expected tension levels fairly independently of f_{web}. Figure 3.4-3, however, shows a more rapid rate of change of tension error due to error in f_{web}, especially in low tension cases where f_{web} is nominally low. Development of an experimental procedure to derive flexural waveform characteristic frequency would likely result in more accurate, repeatable tension indications.

Preliminary procedure formulation to obtain web characteristic frequency had begun prior to performance of the above uncertainty analysis. Fast Fourier Transform algorithms were tried with already established computer programs. These programs would acquire a data record, compute time of flight through cross correlation, compute characteristic frequency through Fourier transform, and finally produce a tension indication through the ribbon equation model. What seemed to be a fairly straightforward assignment was found to have some subtle difficulties to overcome. Tradeoffs were necessary in the implementation of both the cross correlation and Fourier transform algorithms with respect to resolution, handling of records, and processing time.

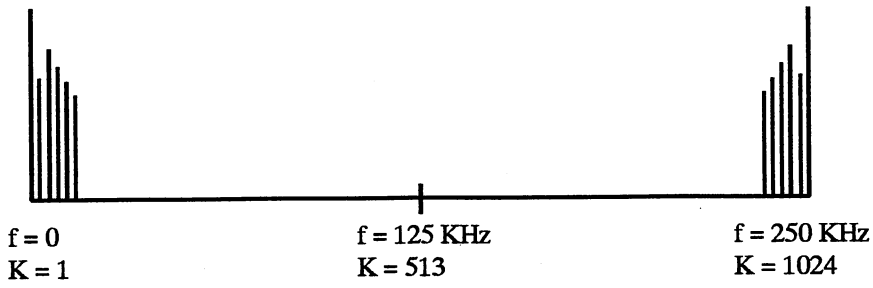
As was indicated in section 3.3, a high sample rate (short sampling period) was desired with respect to the cross correlation computation. Increased sample rate improved resolution of the time of flight computation at the expense of increased processing time. With respect to a discrete Fourier transform, a high sample rate implies a broad frequency band being examined. For a practical record length Fourier transform, the result is very poor resolution with respect to the characteristic frequency computation. Typical from oscilloscope traces presented in this report was web characteristic frequency that ranged from roughly 1000 to 2500 Hz at tension levels of roughly 0.5 to 0.8 pli for one mil thick plastic web material. Thus, the Nyquist sample rate of

5000 Hz per channel would provide minimal coverage of the anticipated signal bandwidth. This is contrasted to a sample rate on the order of 200,000 Hz per channel used in the correlation analysis.

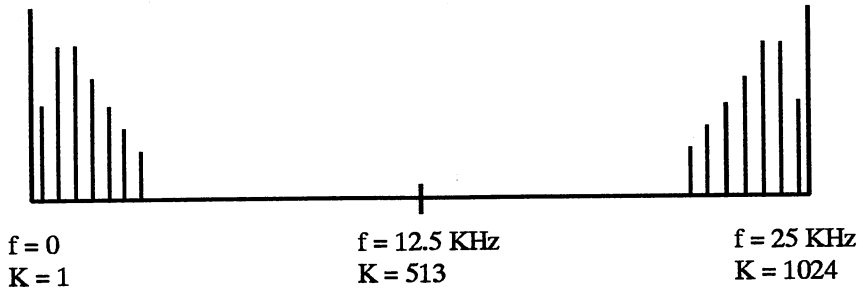
A decimation in time FFT algorithm was used in frequency domain analysis [64]. Record lengths were required to be some power of two. Examination of achievable resolution provided some interesting results. Sampling rates used in the cross correlation were selectable from 100 to 400 kHz per channel, with rates 200 to 250 kHz providing the best compromise between resolution and processing time for typical web and tension conditions given above. For a 1024 point FFT, the achievable resolution would then be $\Delta f = 195.3$ Hz for a 200 kHz sample rate and $\Delta f = 244.1$ Hz for a 250 kHz sample rate. Thus, only a few increments of Δf would span the entire anticipated input signal bandwidth, given as roughly 1000 to 2500 Hz. This is illustrated in Figure 3.4-5a. Clearly, sample rate reduction was needed to accomplish a measure of frequency domain resolution.

Consider a sample rate reduction from a 400 - 500 kHz sample rate for two channels down to 50 kHz sample rate for two channels, or 25 kHz per channel. For a 1024 point FFT, a resolution of $\Delta f = 24.4$ Hz would be available. With this change, illustrated in Figure 3.4-5b, the input signal bandwidth could be spanned by roughly 100 increments of Δf. A sample rate around 50 kHz would allow for adequate frequency domain resolution but would change the approach to microphone signal sampling.

A straightforward approach to signal data record assembly was used with respect to the cross correlation analysis. User selected sampling rate determined record length, the number of samples to be collected per pneumatic input pulse. If averaging was desired, then records were added together point for point before beginning correlation computations. Sample rate selected determined the number of necessary cross correlation shifts to perform to ensure proper signal overlap. Typically, the entire time interval of interest, from the beginning of the leading web pulse to the tail of the trailing web pulse, was less than two milliseconds using the transducer head of Figure 3.3-7. If for example a slower sample rate of 25 kHz per channel were to be used, then one would only be able to achieve fifty samples of useful information per channel in the two millisecond window described. One could increase the sample rate somewhat to acquire more



(a) Rate 100•Signal BW



(b) Rate 10•Signal BW

Figure 3.4-5. Effect of Sample Rate on Finite Bandwidth Signal

useful data points, but the desired resolution, on the order of 25 Hz to 75 Hz, would be sacrificed. In any case, several signal data records would need to be assembled so that sufficient Fourier Transform points would allow for the desired frequency domain resolution.

Discrete Fourier transform theory [65] allows for a transformation of a periodic sequence $\tilde{x}(n)$ to its frequency domain equivalent $\tilde{X}(k)$ through the transform pair:

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) \exp[-j\, 2\pi kn/N] \tag{3.25}$$

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) \exp[j\, 2\pi kn/N] \tag{3.26}$$

$\tilde{X}(k)$ is also a periodic sequence, and is symmetric about the midpoint of the span k = N/2 or equivalently $\omega = \pi$. If the sequence of interest x(n) is of finite duration and is M samples long, that is, transient in nature, then a "periodic" signal may be formed by placing these transient records front to back such that:

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n + rM) = x(n \text{ modulo } M) \tag{3.27}$$

Thus, the discrete Fourier transform $\tilde{X}(k)$ may be computed for this assembled sequence through use of Equation (3.25).

Experimental system frequency domain resolution dictated the need to acquire multiple records for processing. Thus, several records were placed front to back in a computer array prior to Fourier transform computation, as is illustrated in Figure 3.4-6. This assembly, creating a pseudoperiodic sequence, would serve as an averaging mechanism. Furthermore, a discrete frequency spectrum would result from such a sequence, aiding in subsequent selection of a characteristic frequency value. A tradeoff in accuracy versus data acquisition time again had to be dealt with. More assembled records would provide better results due to more of the critical data being present in the data array. Only one record was available per pulse of the pneumatic pulser, however, such that more records assembled meant more record assembly time prior to processing.

A workable solution to the sampling tradeoff described above was formulated and the zero packing method [59] was used to double the Fourier transform resolution. A pseudoperiodic

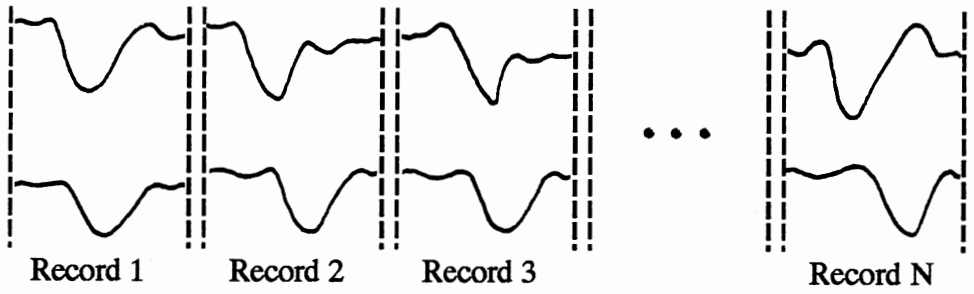


Figure 3.4-6. Illustration of Back-to-Back Signal Records

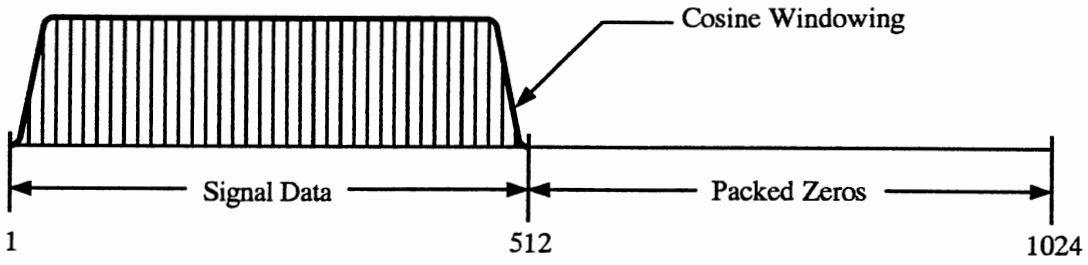


Figure 3.4-7. Illustration of Windowed, Augmented Data Array

record was constructed from five to seven individual records, each containing from 100 points (five records) to 75 points (seven records) for 512 total points. Ordinarily, the frequency increment is the sample rate divided by the record length N. By augmenting the record with N (512) zeros, the record becomes 2N points long, 1024 points in this case. Desired resolution was then found by division of the sample rate per channel, on the order of 25 to 30 kHz, by 1024 points. A cosine windowing function was used on the record data portion prior to FFT processing. Figure 3.4-7 is an illustration of the completed record.

Cross spectral density function between system microphone signals was the actual Fourier transform result computed. Bendat and Piersol [59] provided the basic procedure for computing this spectral function. Essentially, a complex record is formed:

$$z_n = x_n + jy_n \;\; ; \;\; n = 0, 1, 2,..., N - 1 \tag{3.28}$$

where x_n is constructed from upstream microphone signals and y_n is constructed from downstream microphone signals. The Cooley-Tukey decimation in time FFT procedure resulted in a transformed Z(k) vector. Transformed X(k) and Y(k) vectors were found through:

$$X(k) = \left(Z(k) + Z^*(N-k)\right) / 2 \;\; ; \;\; k = 0, 1, 2,..., N-1 \tag{3.29}$$

$$Y(k) = \left(Z(k) - Z^*(N-k)\right) / j2 \;\; ; \;\; k = 0, 1, 2,..., N-1 \tag{3.30}$$

where * indicates a complex conjugate. The cross spectral density magnitude estimate was then found through:

$$\widehat{G_{xy}}(f_k) = \frac{2\,T_s}{N} \; X^*(k)\; Y(k) \;\; ; \;\; T_s = \text{Sample Period} \tag{3.31}$$

where f_k is the frequency associated with the index k, found through f_k = (k)(Sample Rate per Channel)/N. Once calculated, the cross spectral density magnitude function dominant peak was selected as the web characteristic frequency.

Field tests of this tension measurement system version at Mobil Chemical were conducted on January 18, 1991. Objectives of these tests were to examine performance of the system with the new FFT analysis capability. Additionally, both the rotary pulser and new solenoid pulser were to be tested for comparative purposes. Web parameters included web volume density 56.5 lbm/ft^3 and thickness 0.69 (10^{-3} in.), resulting in an area density of 2.248 (10^{-5}) lbm/in.^{2}. Nominal

average web tension was approximately 78 pounds over a 120 inch web span for 0.65 pli. Several tests were performed, including a continuous scan mode in which the transducer head would scan back and forth across the web, writing Δt, f_{web}, and tension indications to an output file. Stationery tests were performed to gauge variation of tension indications over a period of time. As support, oscilloscope traces were used to provide estimates of the f_{web} and Δt values.

A single point test using the new solenoid pulser was performed to view the cross correlation and Fourier transform functions obtained from test data. Figure 3.4-8 displays the signals from this test which were obtained through digital oscilloscope computer programming. Sharper pulses were achieved with the solenoid pulser in comparison to those achieved through the rotary pulser. Oscilloscope cursors provide a web characteristic frequency estimate, approximately 2100 Hz.

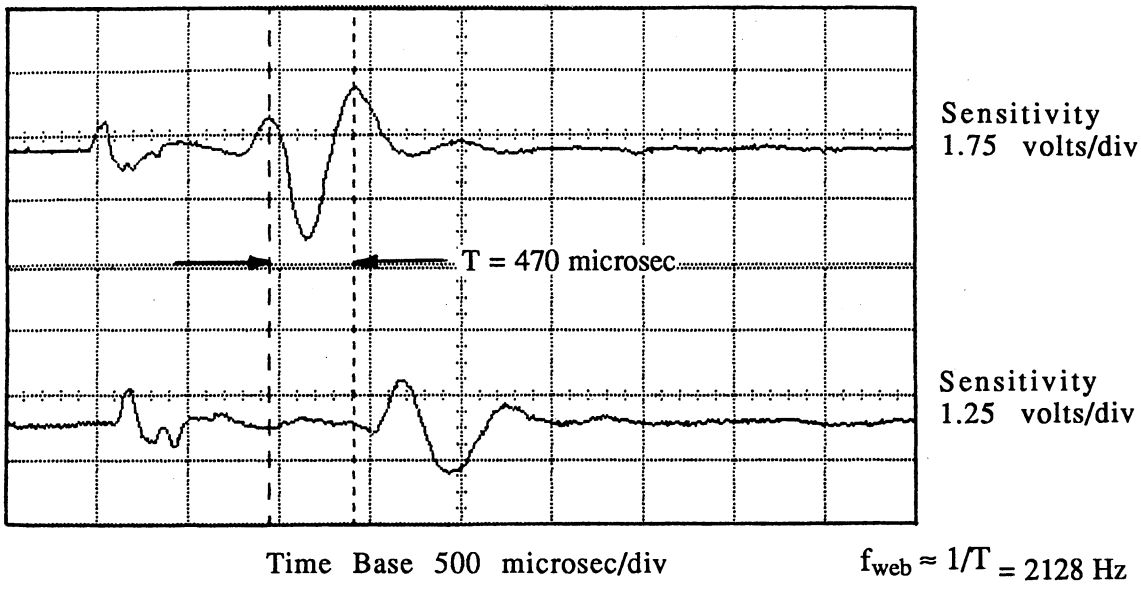
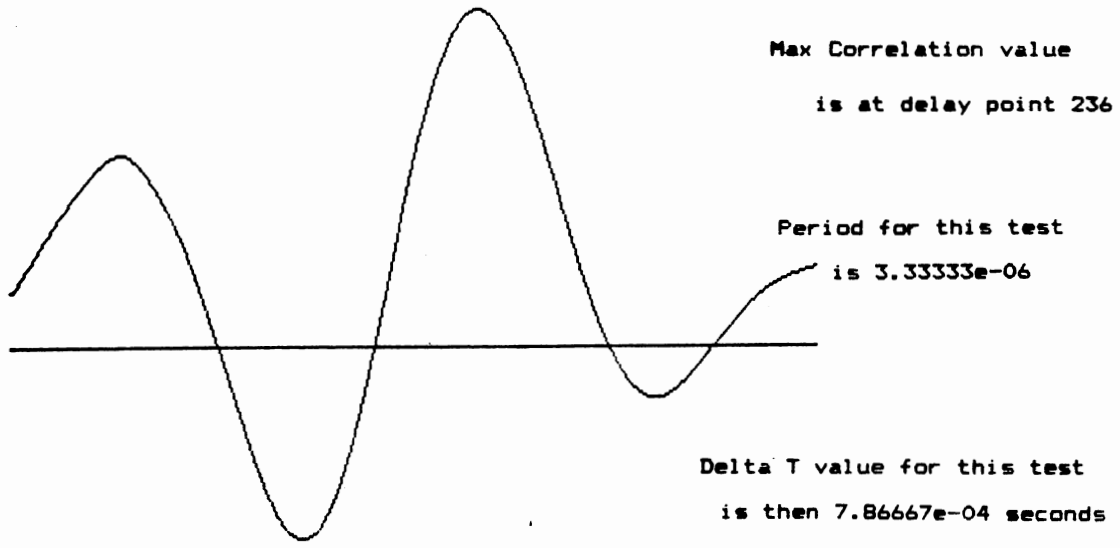
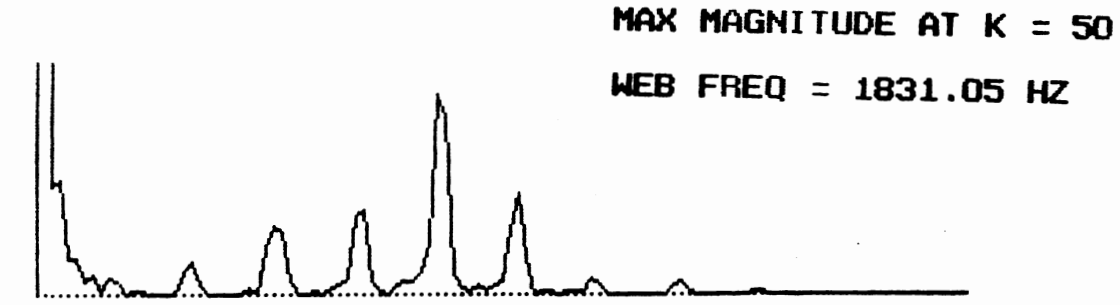


Figure 3.4-8. Trace From Mobil Chemical Test of Waveform With Rough Estimate of Characteristic Frequency Marked

Figures 3.4-9a and 3.4-9b are the cross correlation and cross spectrum functions from this test, respectively. Delta t value was found to be 0.78667 milliseconds from Figure 3.4-9a. Characteristic frequency was found to be 1831 Hz from Figure 3.4-9b. Resultant tension value was computed as 0.706 pli. This particular computer program did not account for web transport



(a) Cross Correlation Function



SAMPLING RATE = 75000 HZ
FOR 1024 PT FFT: FREQUENCY INCREMENT = 36.6211 HZ

(b) Fourier Transform of the Waveform of Figure 3.4-8

Figure 3.4-9. Signal Processing Results for Waveform of Figure 3.4-8

velocity, which in this case would result in approximately a 7 percent error in the tension indication. If this correction were applied, then the indicated tension would be 0.662 pli, which would correlate fairly well to the nominal tension obtained through the winding system load cell.

Figure 3.4-10 is the graphical summary from a stationery test in which the transducer head was located at the web midspan. This graphical summary is normally used to display tension profiles, where the delta X values indicate relative spacing of test locations across the web span. In this case, however, the transducer head was stationery and nine tests were selected. Dashed in Figure 3.4-10 is the average indicated tension level, 0.68280 pli. One can see variation of tension above and below this average level during this tension versus time test. Tension load cell would fluctuate from 76 to 80 pounds during a test such as this, which would correspond to an average tension variation of 0.63 to 0.67 pli. Web transport velocity was accounted for by this computer program, aiding tension indication accuracy. All of the indicated tension values are within 16% of the average value such that no "bad" or far out values were processed. Performance of the solenoid pulser, which provided a sharp strong pulse, had helped system response in this regard.

The tension measurement system performed well in this set of field tests. Some problems remained with regard to the sampling procedure. For each test point, cross correlation data were taken at one time whereupon the sampling rate was changed and Fourier transform sampling was performed. Computer memory constraints would not allow for all sampling to be performed at once. Winding systems have inherent eccentricities, causing tension variation with time. Effect of the split sampling procedure was to possibly compute a time of flight value from a different tension condition from which a characteristic frequency value was computed. Thus, indicated tension values had the possibility of being "hybrid" and not the actual tension at some one specific point in time. For slowly varying tension conditions, this would not be a great concern, but one would prefer to acquire all necessary data at one time if possible. This problem would prove hard to overcome during remaining system development activity.

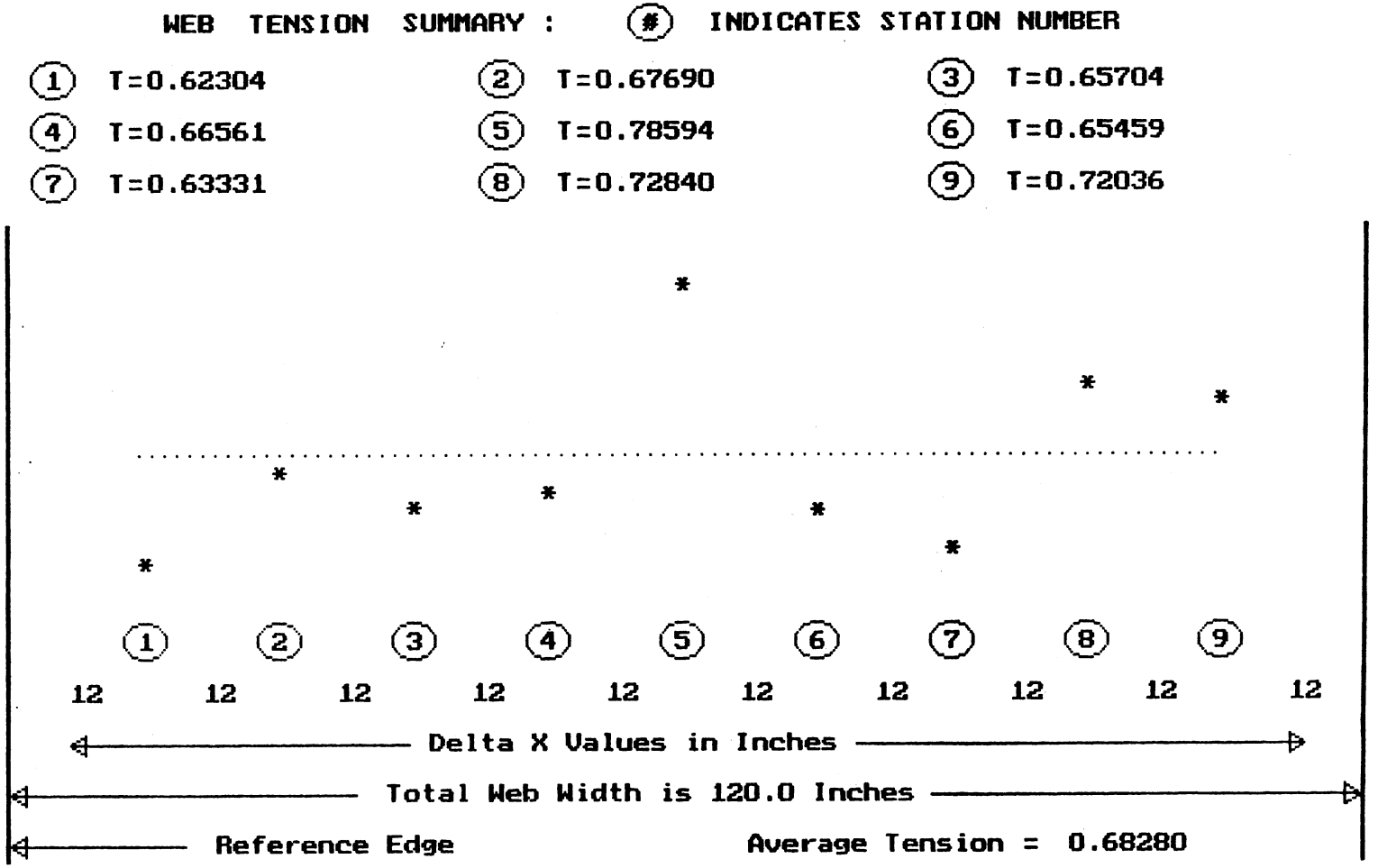


Figure 3.4-10. Stationary Test at Midspan of 10-Ft Wide Web

3.5 Implementation of Adjustable Tension Measurement System Response

The experimental web tension measurement system described thus far in this report had demonstrated creditable results on thin plastic web material. One objective was to make this system applicable to many web materials. An adjustable system response would be needed to provide this enhancement and thus obtain optimal system performance when arbitrary web materials were in use.

Outlined in section 3.4 was development of a method of acquiring a characteristic frequency of induced flexural waveforms in web materials. Increasingly, it became apparent that the system should be "tuned" in order to obtain the best characteristic frequency value. Analog bandpass filtering was already in use, serving to tune the system to one set of filter characteristics. If, however, the "nominal" web characteristic frequency was located sufficiently away from the filter center frequency, then inconsistent, unrepeatable results would plague the system computed tension indications. An attempt was made to determine whether or not filtering was unduly affecting the signals being processed.

Underdamped second order systems are oscillatory in nature. By selecting a second order bandpass filter center frequency coincident with a "nominal" web characteristic frequency, would the resulting filter output exhibit any oscillatory effects. In other words, would the filter with a signal of frequency content in close proximity to its natural frequency cause the filter to "ring"? Butterworth coefficients were used in the filter design for a filter damping ratio of 0.707. Little overshoot would then be expected in response to an impulsive input. Figure 3.5-1 is an analog filter time domain response to a periodic square wave pulse input of duration 1 millisecond. No oscillation problems are apparent from examination of this figure. Subsequent frequency domain tests, using periodic and one-shot square pulse inputs of varying time duration, revealed no adverse effect by the filter on frequency domain response. Figure 3.5-2 is an example of frequency domain filter response to a 500 microsecond one-shot square pulse. From this figure, DC components have been attenuated through bandpass filtering but the zero crossings remain at intervals of 2000 Hz, as is predicted from theory for a pulse of 500 microseconds duration.

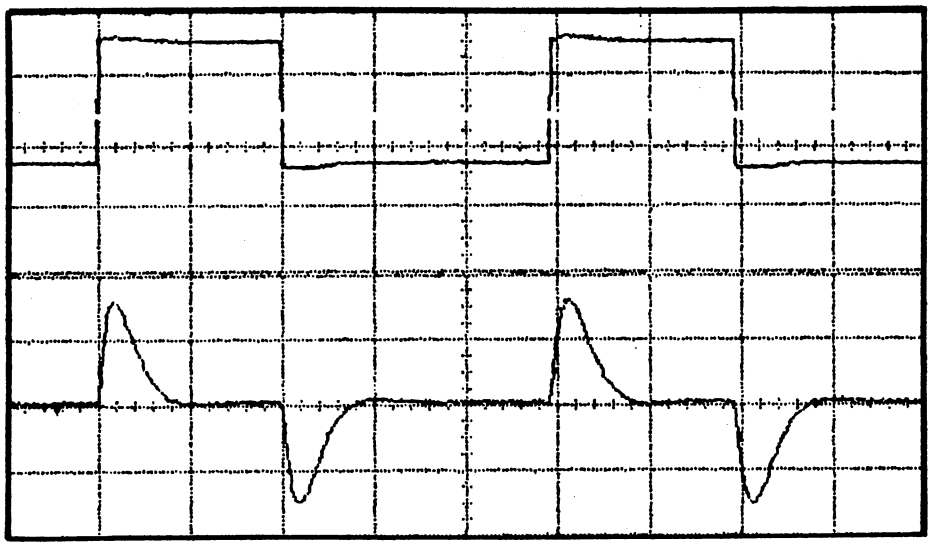


Figure 3.5-1. Response of Analog Band-Pass Filter to Square Pulse Train Signal

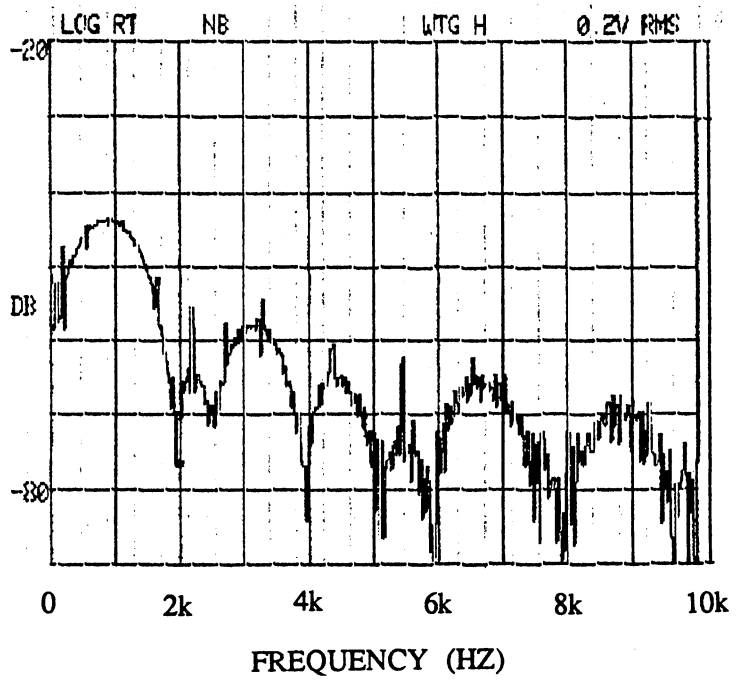


Figure 3.5-2. Spectrum of Transient Response of Analog Band Pass Filter to Square One-Shot Pulse

$$H(z) = H_a(s)\Big|_{s=\frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}} \tag{3.35}$$

Application of Equation (3.35) to the analog transfer function (3.32) resulted in the digital transfer function (3.36), which could be expressed as a recursive relation (3.37) where Y is an output sequence and X is an input sequence.

$$H(z) = \frac{2\,T_s K\,\omega_c\,(1 - z^{-2})}{z^{-2}(4Q-2T_s\omega_c+T_s^2\omega_c^2Q) + z^{-1}(-8Q+2T_s^2\omega_c^2Q) + (4Q+2T_s\omega_c+T_s^2\omega_c^2Q)} \tag{3.36}$$

$$\begin{aligned}\left(1.0 + \pi\,T_s\,BW+(\pi\,f_c\,T_s)^2\right) Y_n &= 2.0\left(1.0 - (\pi\,f_c\,T_s)^2\right) Y_{n-1} \\ &+ \left(-1.0 + \pi\,T_s\,BW - (\pi\,f_c\,T_s)^2\right) Y_{n-2} + \left(\pi\,K\,T_s\,BW\right)\left(X_n - X_{n-2}\right)\end{aligned} \tag{3.37}$$

Adaptation of digital filtering to the experimental tension measurement system was accomplished through programming of Equation (3.37). One benefit of the digital filtering scheme was the elimination of the analog filters and associated hardware. Due to low microphone signal voltage levels, however, an operational amplifier noninverting amplifier circuit was retained to boost the raw microphone signals. Equation (3.37) allowed for user selectable filter center frequency f_C, filter bandwidth BW, and filter gain K. Decisions regarding values of these parameters could be made based on the filtered waveform displayed on the computer CRT screen.

Field tests at the Shawnee Mobil Chemical facility were conducted on September 3, 1991, to test different filtering methods with the solenoid pulser on two different web lines. Analog bandpass filters were prepared with three different sets of filter parameters. Digital filtering parameters, as indicated earlier, could be altered at any time. Of interest was to test the indicated tension sensitivity to filtering parameters f_C and BW.

Three sets of analog bandpass filters were used in these tests with parameters: (a) f_C = 1000 Hz, BW = 1500 Hz; (b) f_C = 1500 Hz, BW = 1000 Hz; and (c) f_C = 2000 Hz, BW = 2000 Hz. The first web line tested was processing 1.1 mil plastic web at approximately 0.7 pli tension. For this moderate tension level, the analog filter with f_C=1500 Hz, BW=1000 Hz seemed to provide the best results. This conclusion of "best" was based on the signal as displayed by the oscilloscope and on the time of flight, characteristic frequency, and tension values produced. Interestingly, this filter was more closely tuned to the prevailing characteristic frequency while possessing the narrowest bandwidth.

Digital filtering was tested in this industrial environment. Default filter parameters, f_c = 1000 Hz, BW = 1500 Hz, K = 2.0, provided an initial filtered waveform display, as illustrated in Figure 3.5-3. As shown in the figure, the user could select a number of options to position the waveform, change the sampling rate, or change the filter parameters.

Figure 3.5-4 is a tension profile graphical display achieved with f_c = 1500 Hz, BW = 2000 Hz, and a 600 kHz sample rate. This relatively high sample rate provided for high time domain resolution and adequate frequency domain resolution, where 60 kHz sample rate was used in acquiring Fourier analysis data. Tension variations shown in Figure 3.5-4 were the result of variation in Δt only due to no variation in calculated characteristic frequency through the nine test positions. Average tension, 0.71792 pli, compared very favorably with the nominal average tension of approximately 0.7 pli.

A second web line was tested, where 0.45 mil plastic was being processed at approximately 0.2 pli tension. Such a low tension level had not been tested extensively in the laboratory. Generally, lower tension such as this would result in a lower value of characteristic frequency coupled with a higher time of flight value. A bowed roller was present at the test site, which presented problems in setting of transducer head spacing with respect to the web. Figure 3.5-5 is an oscilloscope trace of a typical signal from this test location. These signals were taken from the operational amplifier noninverting amplifier circuit output prior to sampling. High frequency signal noise was due to an electrostatic coating processor located adjacent to the transducer head location. Combination of the low tension, bowed roller spacing concerns, and close proximity to a high frequency noise source would severely test the tension measurement system performance.

A tension profile test for the above described test situation was performed using default digital filter parameters f_c =1000 Hz, BW=1500 Hz. Sampling was performed at 500 kHz and 50 kHz for the cross correlation and Fourier analyses, respectively. Figure 3.5-6 is the graphical summary of this test. Excessive air gap, approximately 1/2 inch, was noted at stations 8 and 9 of this test. Poor characteristic frequency values were achieved at stations 8 and 9 due to this air gap, thus biasing the reported tension values at these stations and the resultant average tension. From the figure, average tension was given as 0.25258 pli. If stations 8 and 9 values were discarded, then the average tension for stations 1 through 7 would equal 0.23391 pli, which

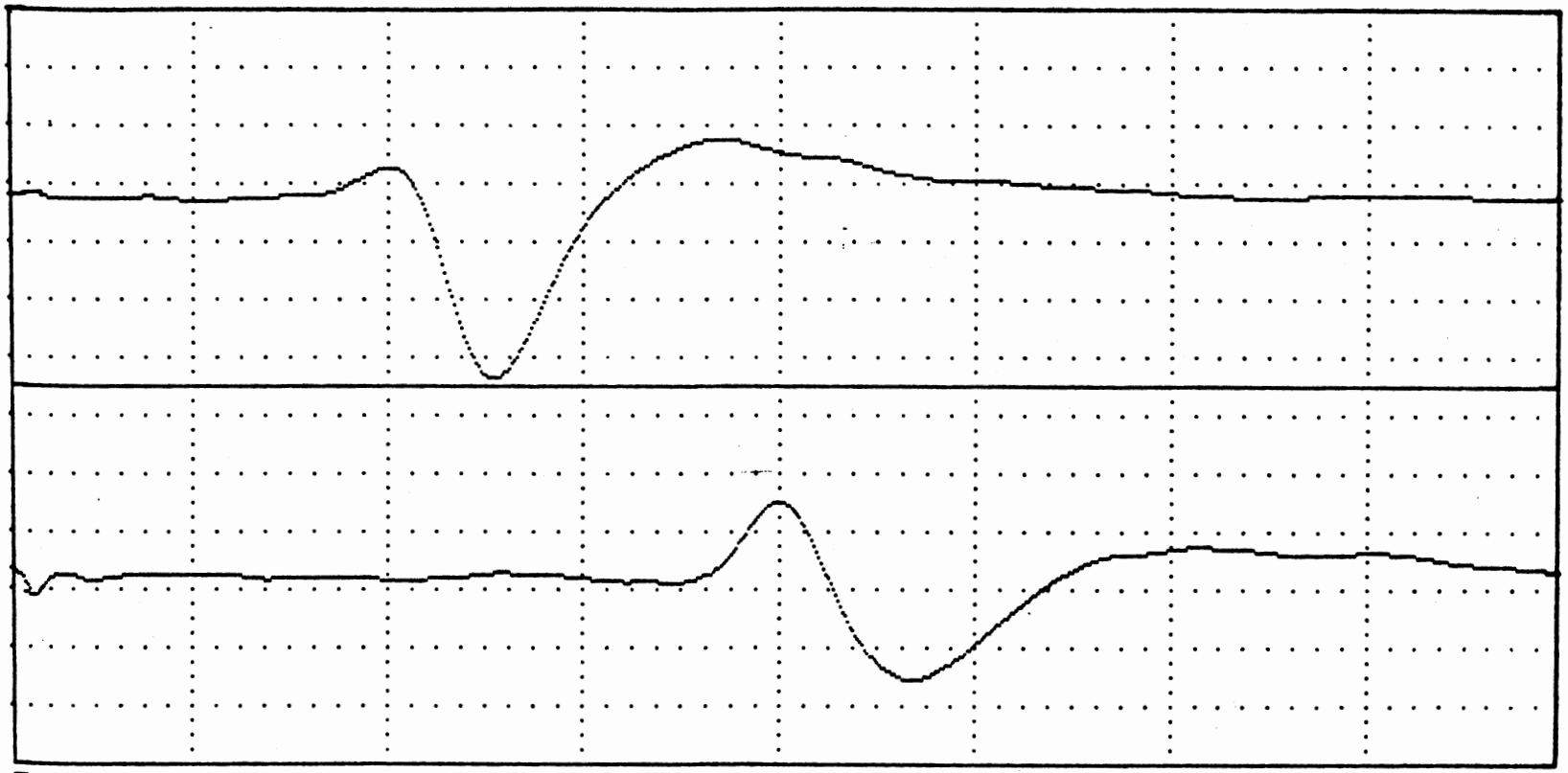


Figure 3.5-3. Display of Waveform for Initial Default Digital Filter Parameters

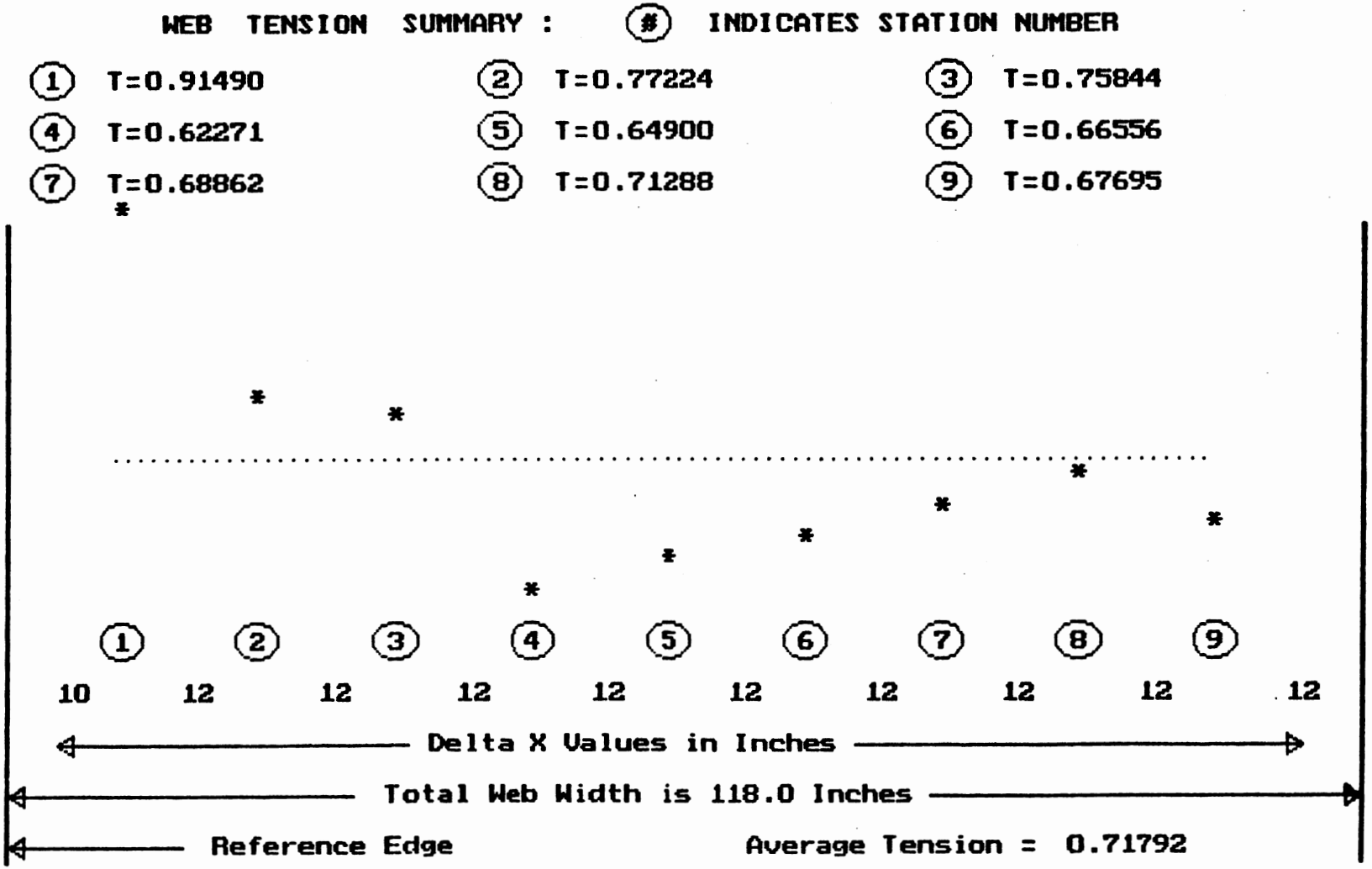


Figure 3.5-4. Tension Profile—Mobil Chemical Web Line 1 Filter Parameters:
f_c = 1500 Hz, BW = 2000 Hz

would provide good agreement with the approximate average tension of 0.2 pli as provided by the winding system load cell.

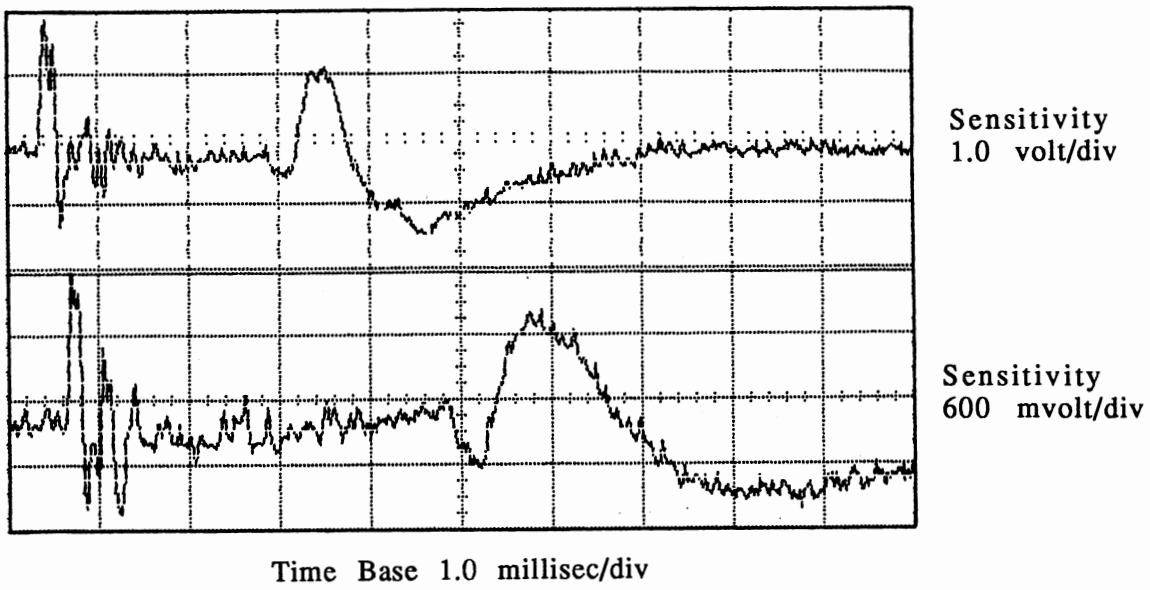


Figure 3.5-5. Typical Trace From Mobil Chemical Test on Low Tension Web Line

Field tests described above showed that the tension measurement system was capable of producing very creditable tension indications. Using analog filtering methods, system performance was shown to be quite sensitive to the filter parameters such that proper tuning of the system was needed to insure best performance. Digital filtering methods allowed for easy filter parameter changes which helped to alleviate some of the sensitivity problems. Once tuned, the digital filtering methods provided extremely good results in some severe test conditions. Remaining to be performed, however, were tests of web materials of various thicknesses and compositions. Of interest would be to observe the system response to thicker materials, where web flexural rigidity would be more of a factor in the generation of flexural waveforms.

3.6 Adaptation to Arbitrary Web Materials and Tensions

During the introduction to section 3.5, mention was made of the need to adapt the experimental web tension measurement system to a wide variety of web materials and tension

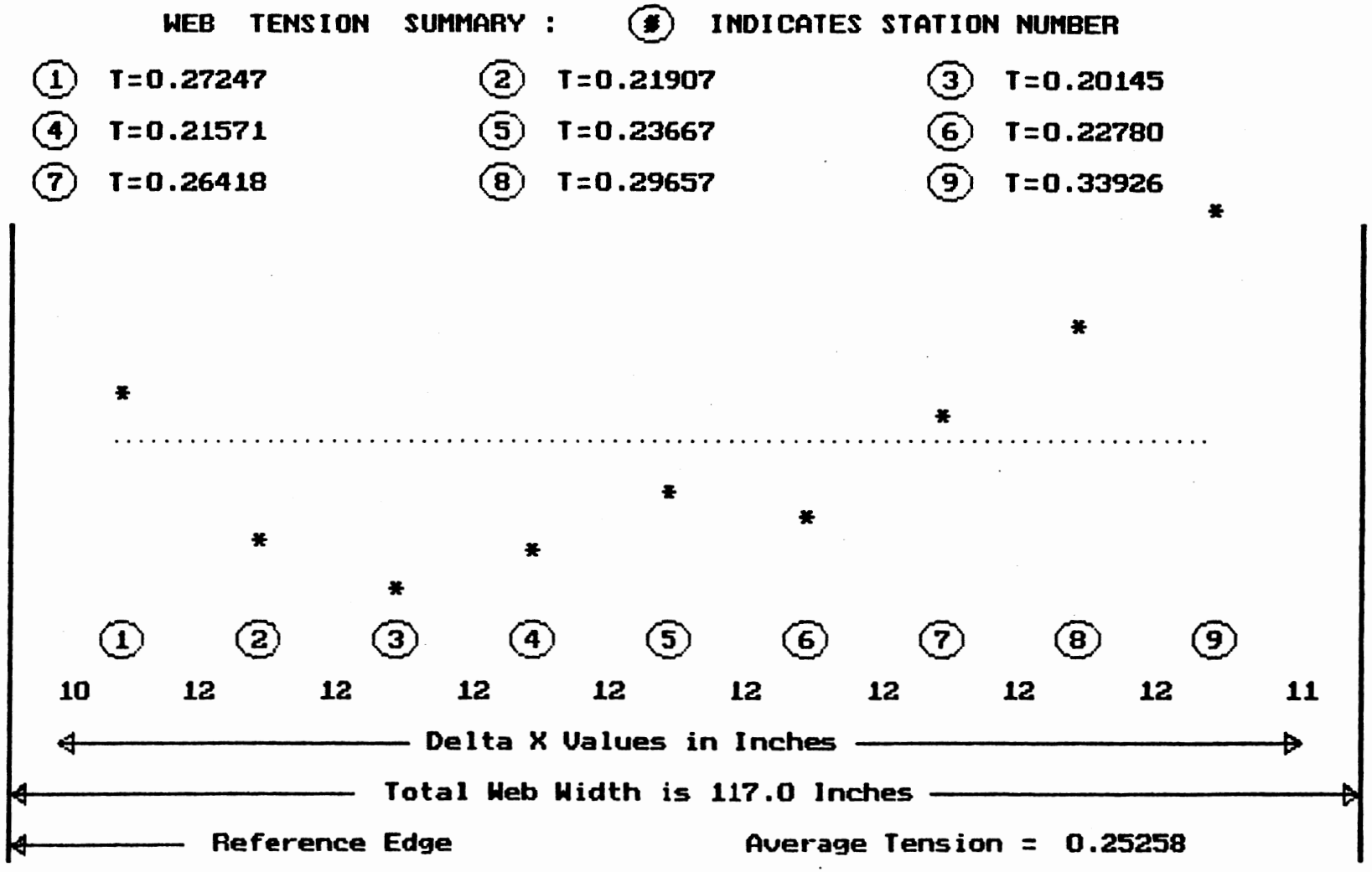


Figure 3.5-6. Tension Profile—Mobil Chemical Web Line 2 Filter Parameters: f_c = 1000 Hz, BW = 1500 Hz

levels. This section will be devoted to particular test cases performed in the laboratory where varying web material and web tension has been observed. Experiments with signal processing methods have been performed to gauge advantages or disadvantages of various methods with respect to resolution and accuracy of results.

Section 3.5 detailed the process of configuring the tension measurement system for variable response depending on user specified filtering parameters. This proved to be a valuable asset during preliminary laboratory testing of arbitrary web materials. Problems became apparent due to web anisotropies. An example of this was observed while testing a synthetic material named Nomex, supplied by Dupont Corporation. Nomex and other web materials often contain "tight" spots and "soft" spots such that signals sensed by the system microphones might look radically different at these locations. Even more perplexing was that a tight spot and a soft spot might be as little as 1/2 inch apart along a web span. Some different signal processing procedures were developed to gauge performance with respect to these types of problems.

During tension measurement system development, cross correlation function revealing the relation of signals obtained from the fore (upstream) and aft (downstream) microphones was calculated by a brute force approach. Discrete cross correlation function, Equation (3.4), was calculated as per the definition, resulting in rather lengthy processing time. Good results were obtained which precluded tampering with this procedure during prior development phases. Fourier transform methods may be applied to cross correlation computations as an alternate approach. Processing time savings may be realized if fast Fourier transform algorithms are thus used.

Cross correlation calculations have thus far been viewed as purely a time domain operation. Correlation functions are related to spectral density functions through the Wiener-Khinchin relations [94]. For cross correlation defined as:

$$R_{xy}(\tau) = \lim \frac{1}{T} \int_0^T x(t)\, y(t+\tau)\, dt \tag{3.38}$$

The two-sided spectral density function is related by [95]

$$S_{xy}(\omega) = \int_{-\infty}^{\infty} R_{xy}(\tau)\, e^{j\omega\tau}\, d\tau = F.T.[R_{xy}(\tau)] \tag{3.39}$$

where F.T. indicates Fourier transform. Thus, computing the cross spectrum of x and y data records provides a record which may be inversely transformed to provide the cross correlation function:

$$R_{xy}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S_{xy}(\omega)\, e^{j\omega\tau}\, d\omega = F.T.^{-1}[S_{xy}(\omega)] \tag{3.40}$$

This may be alternately expressed as the Fourier Transform pair [95]:

$$R_{xy}(\tau) \Leftrightarrow X^{*}(\omega)\, Y(\omega) \tag{3.41}$$

Bendat and Piersol [59] provided a set of procedures to follow to utilize this approach. Due to the circular nature of the Fourier transform, recommended was the padding of N sample data sequences with N zeros and then performing 2N point discrete Fourier transforms (FFTs). This would insure decoupling of subsequent cross correlation function values. After performing the complex multiplication of Equation (3.41), inverse Fourier transform is performed. Lastly, recommended was use of a multiplicative scale factor N/(N-r) for - N < = r < = N - 1 to finally obtain $\widehat{R}_{xy}(rh)$ values. A similar procedure was recommended by Stearns and David [94] but using instead a scale factor 1/(N-r) at the end of the process.

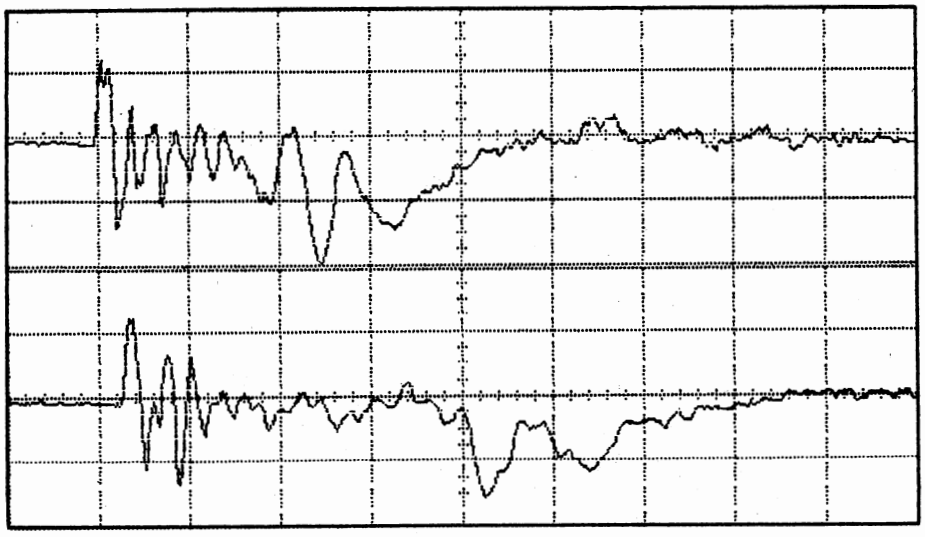
Some of the ideas outlined above had been used in calculation of web characteristic frequency. Thus, application of these ideas in cross correlation context was not extremely difficult. The only drawback to the method was the limited number of data vector sizes available, where a power of 2 was required. Vector sizes 512 and 1024 were most applicable for sample rates typically used in data acquisition. This procedure was assembled in program WH1FC, which is listed in Appendix E.

During evaluation of the characteristic frequency value from sensed microphone signals, several records were acquired and placed front to back in a vector. Fourier transform of this pseudoperiodic record contained benefits of record averaging and a resultant discrete frequency spectrum from which a dominant spike could be identified. Disadvantages included additional

III have been incorporated into these tests. Recall from section 3.5 that operational amplifier noninverting amplifier circuits were retained to boost the raw microphone signals voltage levels. In these upcoming presentations, oscilloscope traces will be referred to as "raw" traces even though they are acquired at output of these operational amplifier circuits and not directly from the sensing microphones. These signals are raw in that no deliberate frequency domain modification has been implemented with possible exception of inherent operational amplifier bandwidth limitations. Digital filtered computer CRT screen dumps will also be presented to show effects of filtering on the raw signals. Amplitude of these displays is arbitrary depending on the gain value selected. Thus, gain will not be specifically detailed in these discussions. Rotary pulser was initially used in these tests whereupon the newer solenoid pulser was implemented roughly halfway through the tests. One test objective is to provide a final presentation to the web manufacturing and web handling industries who may consider use of these ideas in their own industrial facilities. Another objective is to determine whether or not any firm statements can be made with respect to system performance and applicability.

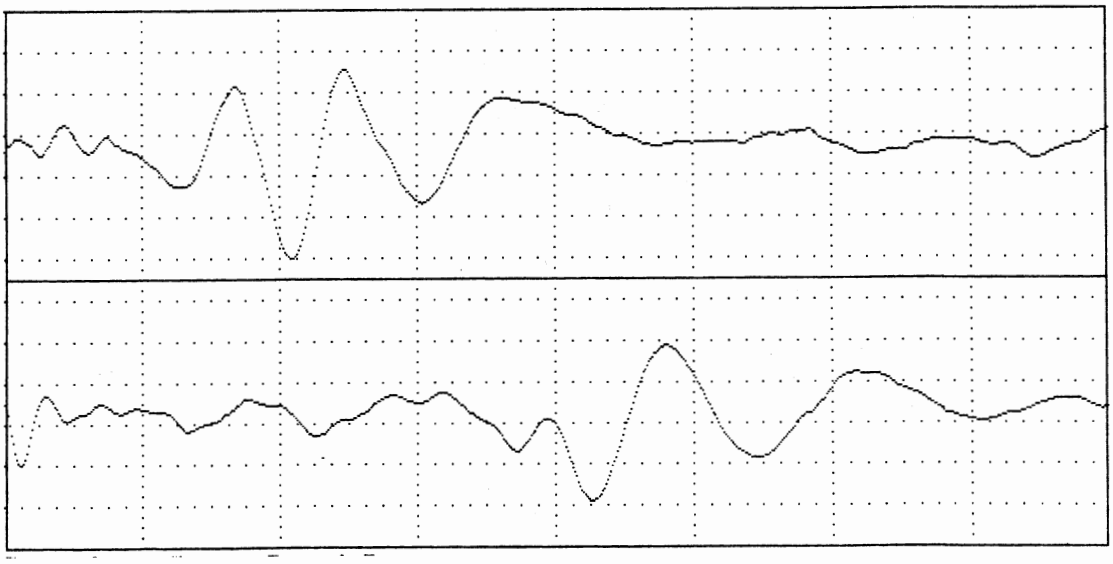
Test A considers tension measurement system use on 2 mil thick Type 410 Nomex material which possessed a relatively slick surface. Area density for this web material was 5.8 (10^{-5}) lbm/in.^{2} and applied average tension was 0.72 pli. A single point test was performed using signal processing procedures given in section 3.5 (program WH1DF of Appendix E). Figure 3.6-1a is the raw oscilloscope view of the test signal and Figure 3.6-1b is the digital filtered version with filter center frequency 2000 Hz and bandwidth 2500 Hz. From this figure, the web flexural pulse is quite evident in the top trace (upstream microphone) but degrades somewhat during propagation to the bottom trace (downstream microphone). From oscilloscope estimations, Δt and f_{web} are approximately 0.9 milliseconds and 3200 Hz, respectively. Figure 3.6-2a is the cross correlation function and Figure 3.6-2b is the spectral density function for this test, where data were collected at 400 and 50 kHz sample rates for these two procedures, respectively. From the figures, Δt = 0.915 msec and f_{web} = 2514.6 Hz. Tension indication from these data was 0.718 pli using the in-vacuo equation (Equation (3.14)) and 0.870 pli using the ribbon equation (Equation (3.15)).

Judging system accuracy is difficult from such a single point test. This single point program WHIDF allows for viewing of cross correlation and frequency domain results such that tuning



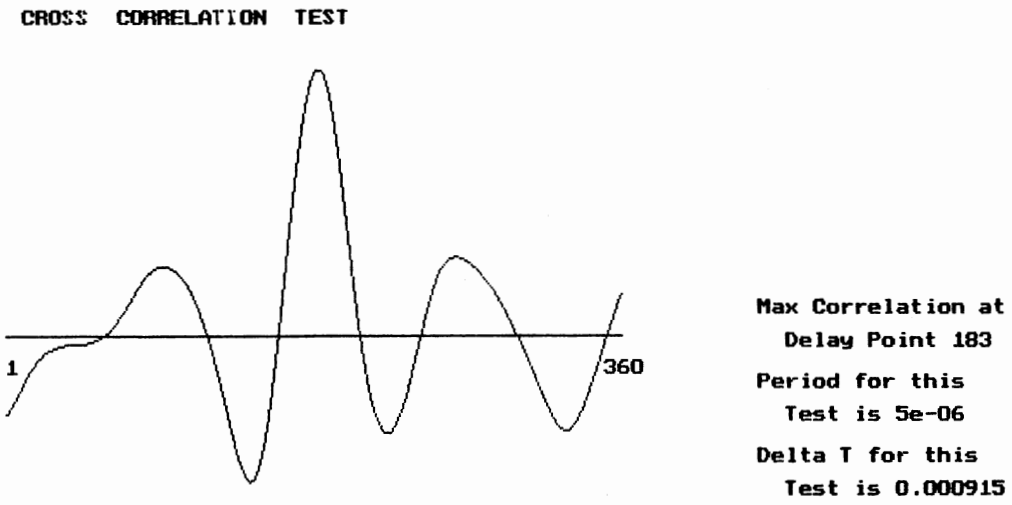
Time Base 500 microsec/div

(a) Raw Unfiltered Signal

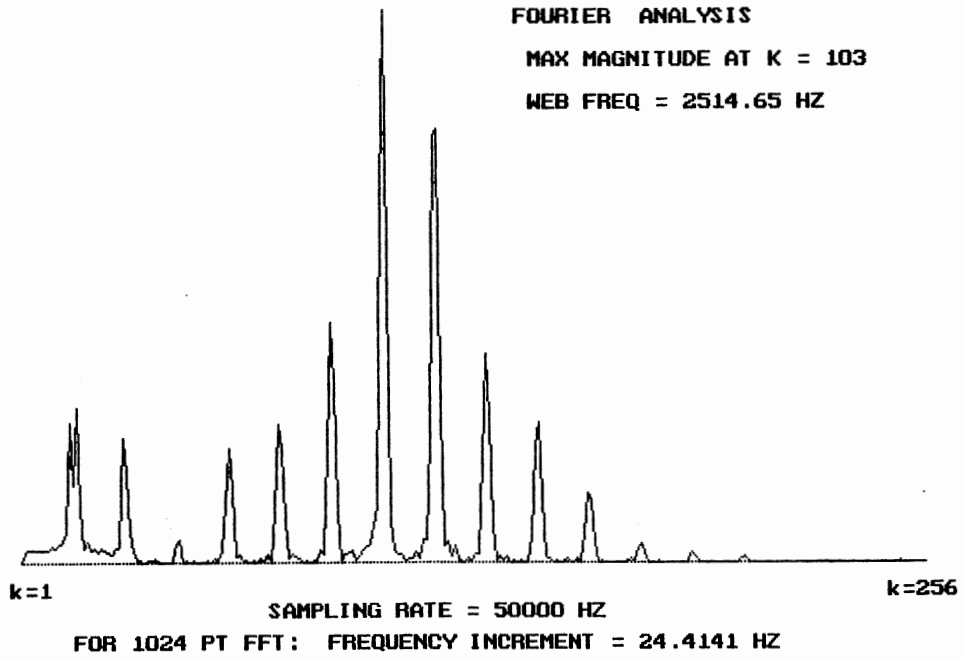


(b) Digital Filtered Signal

Figure 3.6-1. Test A Signals—2 Mil Type 410 Nomex Material



(a) Cross Correlation Function



(b) Spectral Density Function

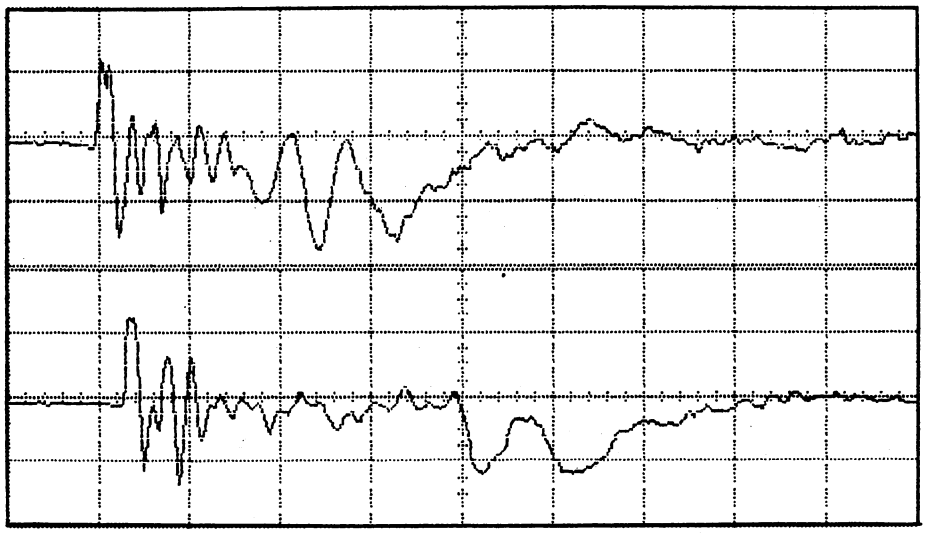
Figure 3.6-2. Test A Signal Processing Results

may be adjusted and retried. An optimal set of filtering parameters may be found by such a trial and error process. Results obtained here will be used as a basis for comparison to fast correlation and single record spectral density signal processing methods.

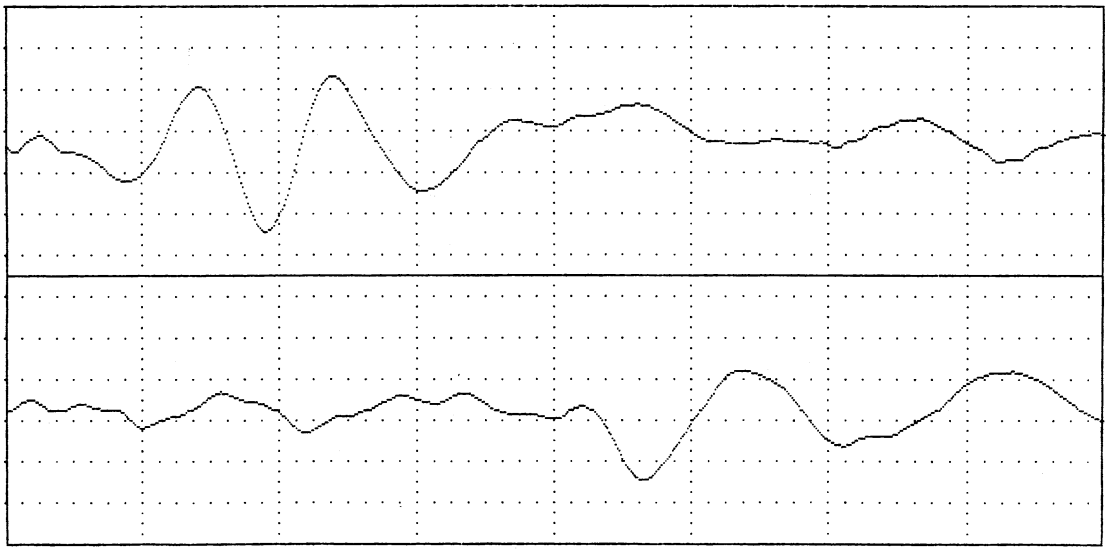
Test B was performed for the same test conditions as Test A but using signal processing methods described earlier in this section. Program WH1FC (Appendix E) was used with filter parameters center frequency f_c = 2000 Hz and bandwidth BW = 2500 Hz. A higher sample rate of 500 kHz was used in this test compared to Test A. Figures 3.6-3a and 3.6-3b are raw oscilloscope and digital filtered signals, respectively, acquired in this test, which maintain the character of those presented in Test A. Figure 3.6-4a and 3.6-4b are the fast cross correlation and single record spectral density functions, respectively. Results of signal processing, Δt = 0.924 msec and f_{web} = 2734.4 Hz, are very similar to that obtained in Test A. Tension results are 0.704 pli using in-vacuo and 0.840 pli using ribbon equation data conversions

Fast correlation procedure, graphically presented in Figure 3.6-4a, has provided a trace quite different from that produced using standard cross correlation, as in Figure 3.6-2a. Much higher oscillations of the cross correlation function appear, which may result in a problem in proper shift point selection for Δt determination. From Figure 3.6-4a, the solid horizontal line represents zero where the cross correlation function swings positive and negative about this line. Two positive peaks at shift locations 231 and 322 and two negative peaks at shift locations 185 and 296 are quite close in amplitude absolute value. Fortunately in this case the proper peak was maximum, yielding the correct Δt value. General implementation of the fast correlation procedure was taken from Bendat and Piersol [59]. Specified was a scaling multiplier of the form N/(N-r), where N is the number of record data samples and r is the shift index. This multiplier tends to greatly amplify cross correlation values for low shift numbers compared with the original definition of Equation (3.4). This scaling factor was changed to that of the original definition and that recommended by Stearns and David [94], 1/(N-r). This change helped alleviate these oscillatory problems in later tests.

Fourier transform of the single record sample, Figure 3.6-4b, provided a very good result in this case. Effective sample rate for the single record was 25 kHz for two channels where 64 data points and 64 zeros resulted in a 128 point record. Note differences in resolution for this

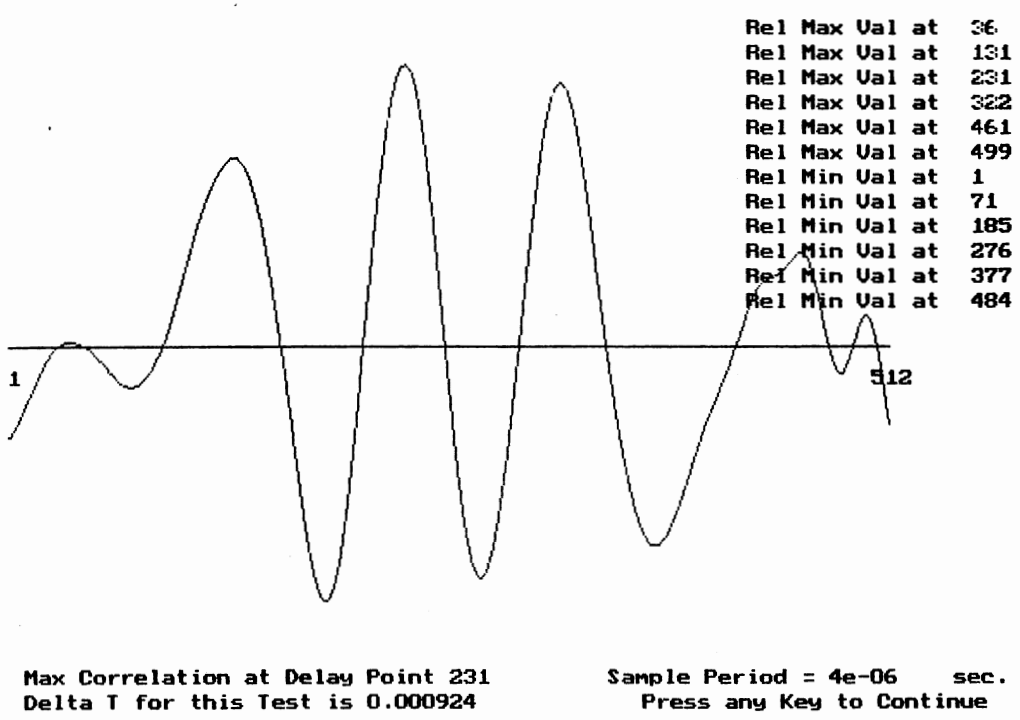


(a) Raw Unfiltered Signal

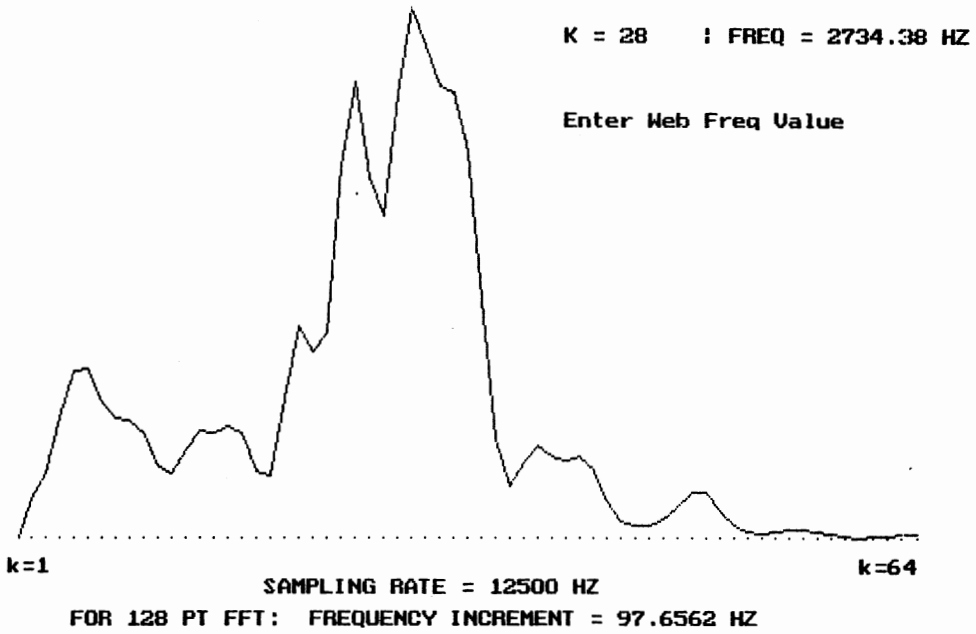


(b) Digital Filtered Signal

Figure 3.6-3. Test B Signals—2 Mil Type 410 Nomex Material



(a) Cross Correlation Function



(b) Spectral Density Function

Figure 3.6-4. Test B Signal Processing Results

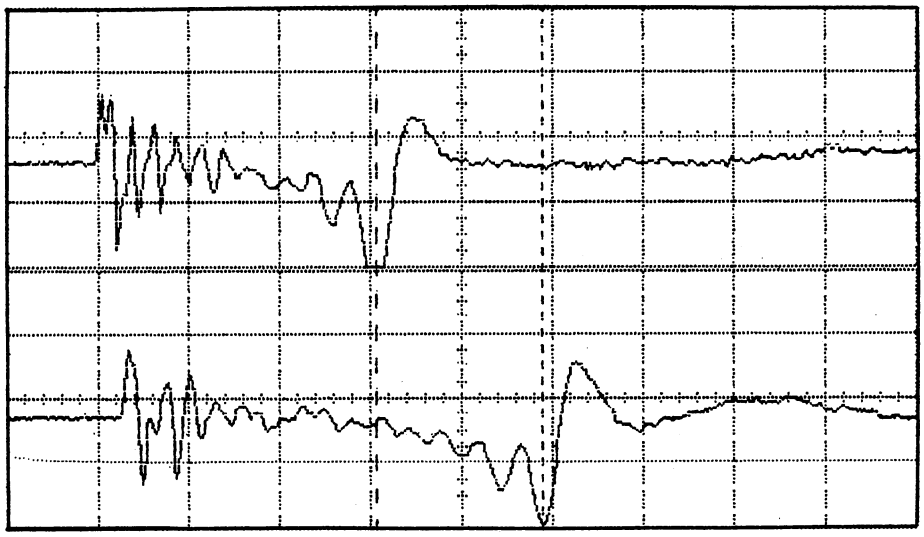
approach compared to that in Figure 3.6-2b. The pseudoperiodic record has four times the resolution, but sampling time length is much greater. The single record transform allows for fast processing, but is more of a hit-or-miss type procedure. One should test repeatability of this single record approach for each possible application to insure adequate performance.

Floppy disk material was examined in this test series, where 3 mil thickness and 1.67 (10^{-4}) lbm/in.^{2} area density made this material rather dense and stiff. For this test, applied average tension was 2.21 pli. A single point test was performed followed by a profile test to judge overall performance of the system for this situation.

Figures 3.6-5a and 3.6-5b are raw oscilloscope and filtered signals, respectively, for the single point floppy disk material tension test, using program WH1DF. Filter parameters selected were center frequency 2500 Hz and bandwidth 3000 Hz with a 400 kHz sample rate. From the oscilloscope trace it is evident that no problems were encountered in generating flexural waveforms in this material at a higher tension level than has heretofore been tested. Estimated Δt and f_{web} from this oscilloscope trace are 0.91 msec and 2940 Hz, respectively. The filtered waveforms exhibit great similarity between upstream and downstream detection, which is always a virtue with respect to the cross correlation process.

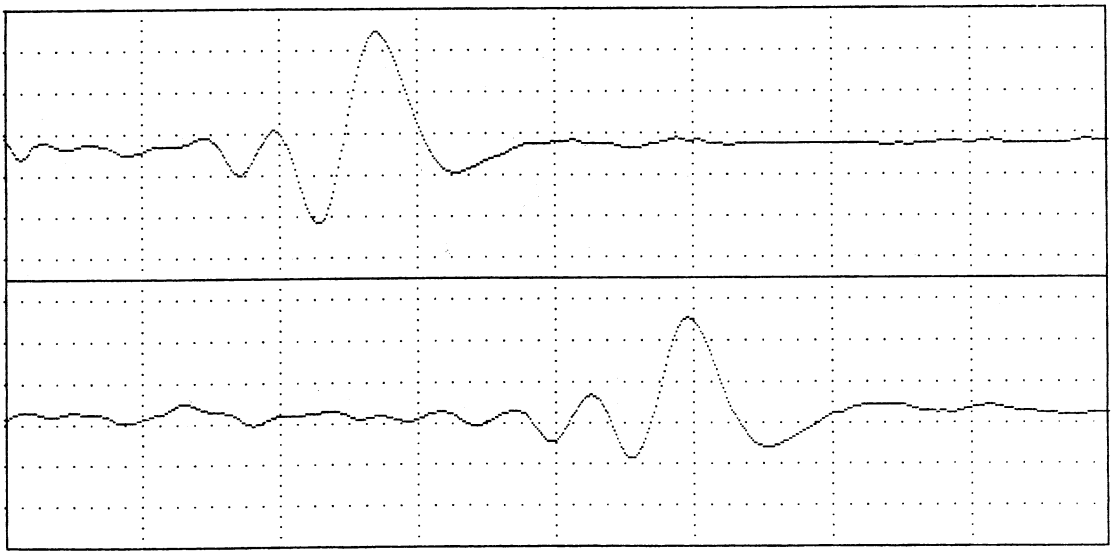
Figures 3.6-6a and 3.6-6b provide the cross correlation function and pseudoperiodic spectral density function for this test. From the cross correlation function, an unmistakable peak at 180 shifts provides a Δt value of 0.900 milliseconds. Fourier transform provides the nice triangle type spectral envelope, culminating in an f_{web} value of 2246.1 Hz. Tension results were 2.136 pli from the in-vacuo model and 2.315 pli from the ribbon model.

Test D was performed on this same material to achieve a tension profile. Program WH3DF (listed in Appendix F) was used, which utilizes the same signal processing techniques as program WH1DF used earlier. Filter parameters were kept the same as in Test C but sample rate was increased to 500 kHz. The transducer head was located 1 inch from the floppy disk material edge whereupon four tests were performed at 1-inch intervals to traverse the 5-3/8 inch web span. Table 1 provides time of flight, characteristic frequency, and estimated in-vacuo tension indications at the four test stations. Figure 3.6-7 is the graphical summary of the tension test providing ribbon equation tension indications. From Table 1, lowest Δt values at center test



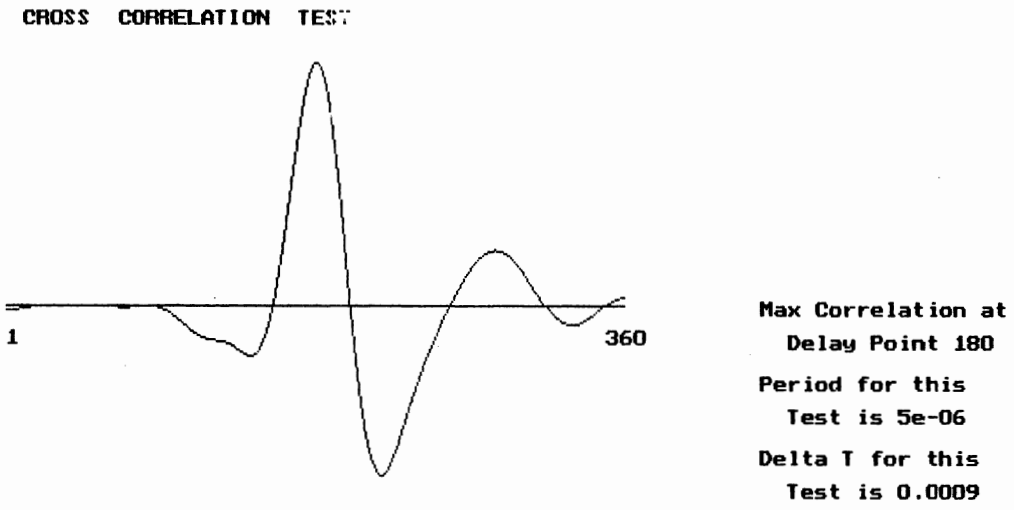
Time Base 500 microsec/div

(a) Raw Unfiltered Signal

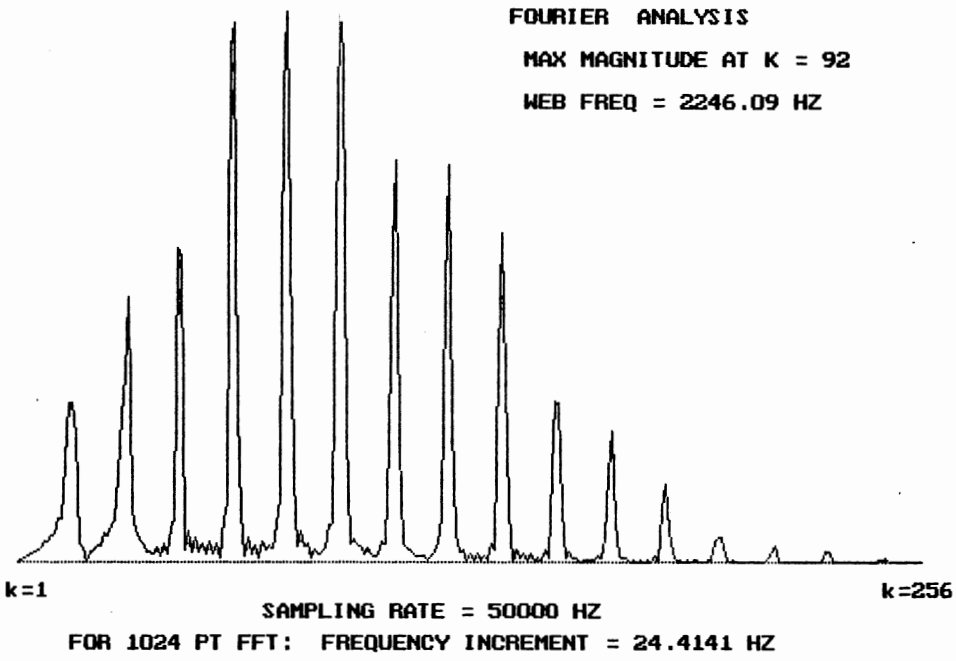


(b) Digital Filtered Signal

Figure 3.6-5. Test C Signals—3 Mil Floppy Disk Material



(a) Cross Correlation Function



(b) Spectral Density Function

Figure 3.6-6. Test C Signal Processing Results

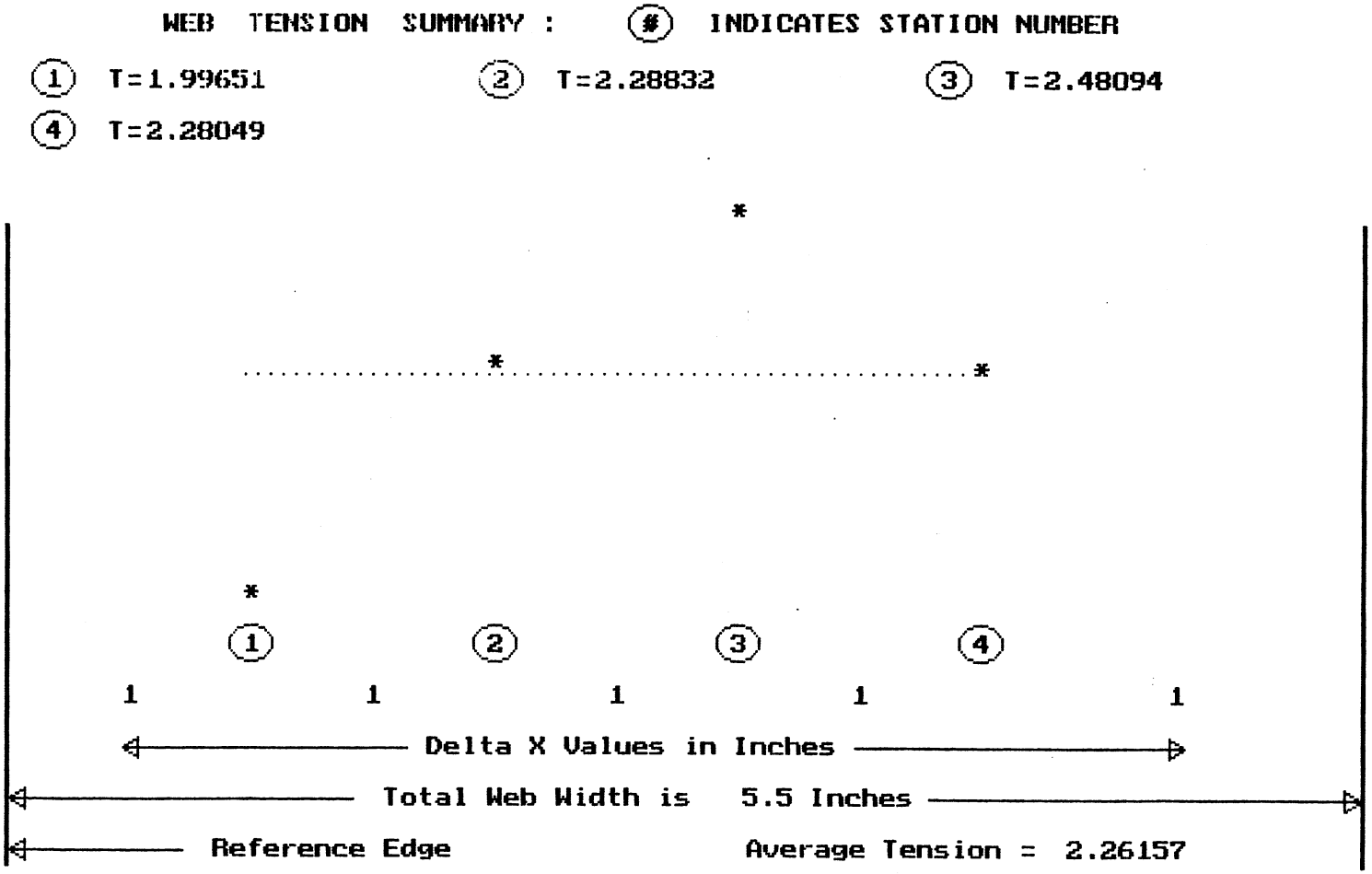


Figure 3.6-7. Test D - Floppy Disk Material Tension Profile Summary

locations imply greater tension, which is confirmed in Figure 3.6-7. Average tension indicated is 2.262 pli which is quite favorable in comparison to the 2.21 pli average applied tension.

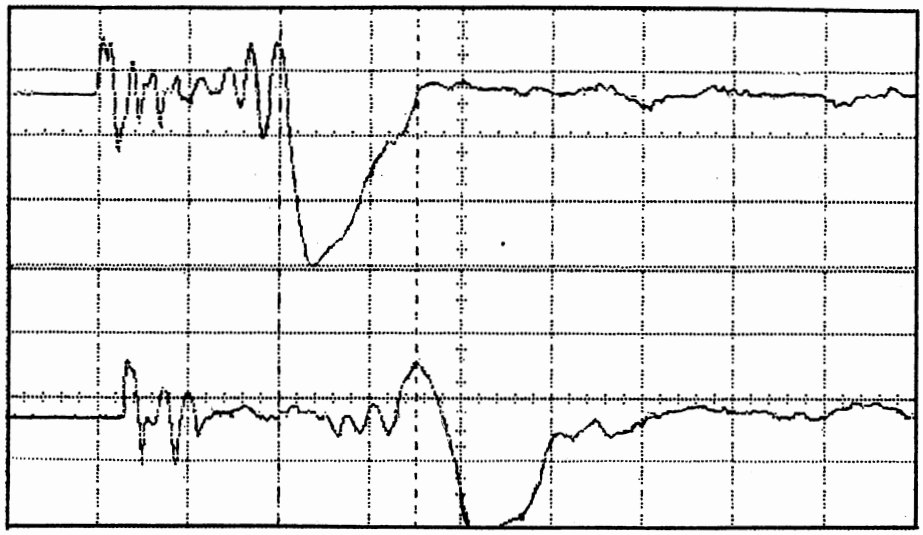
TABLE 1

TIME OF FLIGHT, CHARACTERISTIC FREQUENCY, AND IN-VACUO TENSION FOR TEST D TENSION PROFILE

Station	Δt (msec)	f_{web} (Hz)	$T_{IN\text{-}VACUO}$ (pli)
1	0.972	1928.7	1.832
2	0.904	2319.3	2.118
3	0.876	1928.7	2.255
4	0.912	1684.6	2.081

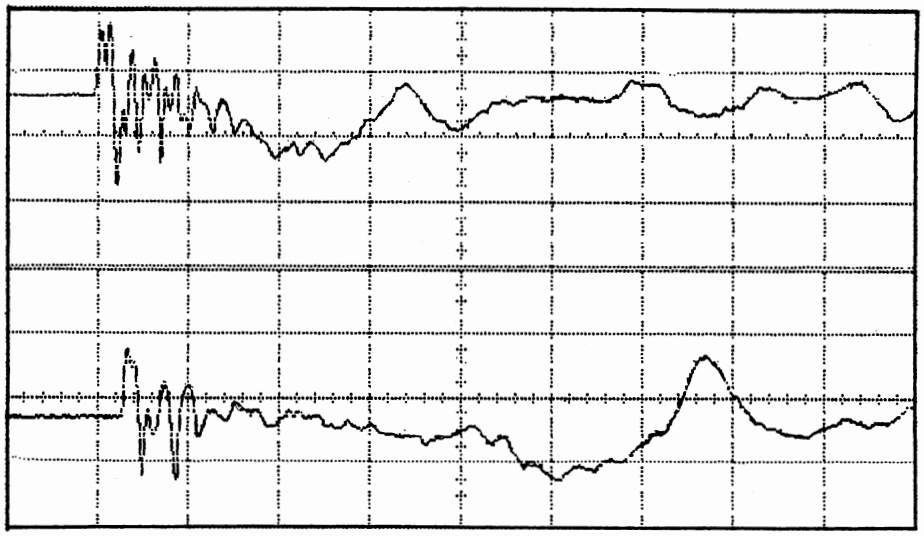
Program WH3DF was again used to perform a tension profile test of 5 mil thick type 411 Nomex material, Test E of this series. Type 411 Nomex has a courser surface finish and is somewhat softer than the Type 410 Nomex which was tested earlier. Area density for the material was 5.5 (10^{-5}) lbm/in.^{2} whereupon 0.85 pli average tension was applied. Figures 3.6-8a and 3.6-8b are oscilloscope traces taken one inch from either edge of the eight-inch wide web, corresponding to stations 1 and 7 of the upcoming tension profile. This illustrates the type of signal change that one may experience when traversing a web span. Notice the dramatic changes that have occurred in signal amplitude, time of flight, and characteristic frequency. Thus, this serves as a real motivator for trying different filtering parameters on different web locations during preliminary set up tests.

Figure 3.6-9 is the filtered trace corresponding to the location where Figure 3.6-8a was obtained, where filter center frequency 2000 Hz, bandwidth 2500 Hz, and sample rate 500 kHz were used. As indicated above, seven tests were performed in spanning the web. Table 2



Time Base 500 microsec/div

(a) Raw Signal Near Left Edge



Time Base 500 microsec/div

(b) Raw Signal Near Right Edge

Figure 3.6-8. Test E Signals—5 Mil Type 411 Nomex Material

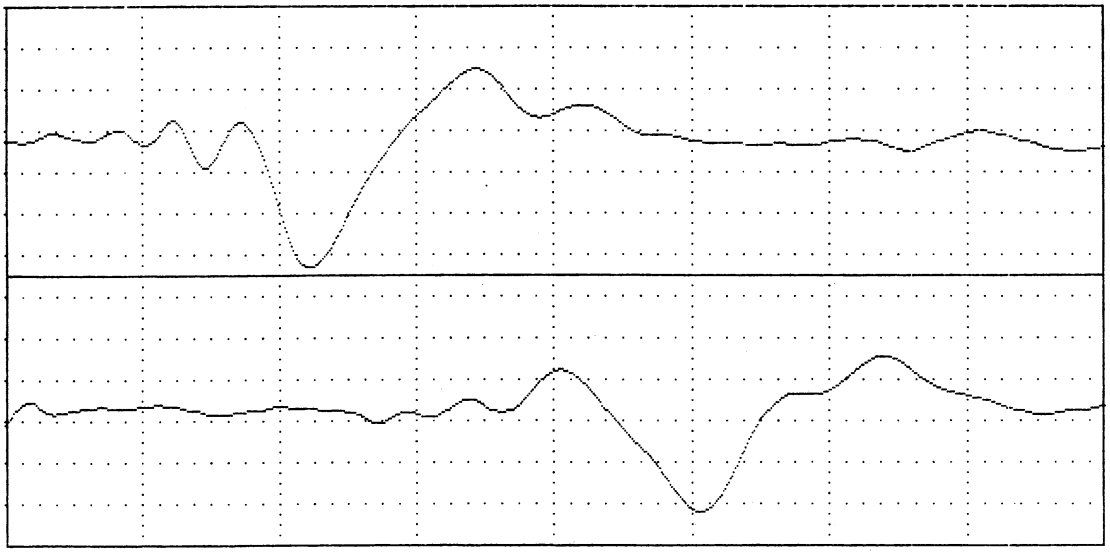


Figure 3.6-9. Digital Filtered Signal Corresponding to Location of Figure 3.6-8a

TABLE 2

TIME OF FLIGHT, CHARACTERISTIC FREQUENCY, AND IN-VACUO TENSION FOR TEST E TENSION PROFILE

Station	Δt (msec)	f_{web} (Hz)	$T_{IN\text{-}VACUO}$ (pli)
1	0.884	1098.6	0.729
2	1.020	1098.6	0.548
3	0.816	1464.8	0.856
4	1.000	1464.8	0.575
5	0.996	2197.3	0.580
6	1.424	1831.0	0.281
7	1.181	1098.6	0.412

provides the raw information for each station and Figure 3.6-10 is the graphical summary for the test. Oscilloscope traces of Figure 3.6-8 imply a tension gradient across the web, which is verified by the profile plot. This soft surfaced Nomex, however, does provide for some unpredictable responses. In particular, station 2 appears to be a "soft" spot in the web resulting in a high time of flight value. Average tension indicated in Figure 3.6-10, 0.789 pli, is within 8% of the applied average tension, which is quite good for this type of material.

At this point the solenoid pulser was implemented into testing. Some heavier materials were to be examined which would likely need a bit more drive to excite. A stronger, sharper pulse was available with the solenoid pulser compared to the rotary pulser. Additionally, pulse width and repetition rate could be varied, making this a more versatile instrument.

Test F was performed on 3 mil thick clear plastic web material. This material was fairly stiff and heavy, as evidenced by a 1.61 (10^{-4}) lbm/in.^{2} area density value. For this 13-inch wide web, 18.6 pounds were applied for an average applied tension of 1.43 pli. Test location was one inch from a web edge. Figure 3.6-11a shows the raw oscilloscope trace and Figure 3.6-11b shows the digital filtered trace for this test, where filter parameters were center frequency 1500 Hz, bandwidth 2000 Hz, sampled at 400 kHz. Notice striking similarity between the filtered and raw waveforms, which is an indication of proper filter parameter selection. Fast correlation and single record spectral density were performed through program WH1FC. Figure 3.6-12a and 3.6-12b are the cross correlation and spectral density functions, respectively. Critical findings were Δt = 1.165 msec and f_{web} = 1172 Hz, resulting in tension values of 1.229 and 1.386 pli from the in-vacuo and ribbon models, respectively. In this case, these alternative methods of signal processing have provided quite reasonable results.

A tension profile was acquired using conventional signal processing practice of program WH3DF, with results noted as Test G of this series. Initial position for this test was identical to the location just examined in Test F. Similarly, filter parameters were kept equal: f_c = 1500 Hz, BW = 2000 Hz and 400 kHz sample rate. Table 3 provides the Δt and f_{web} statistics from this test along with the in-vacuo tension estimates.

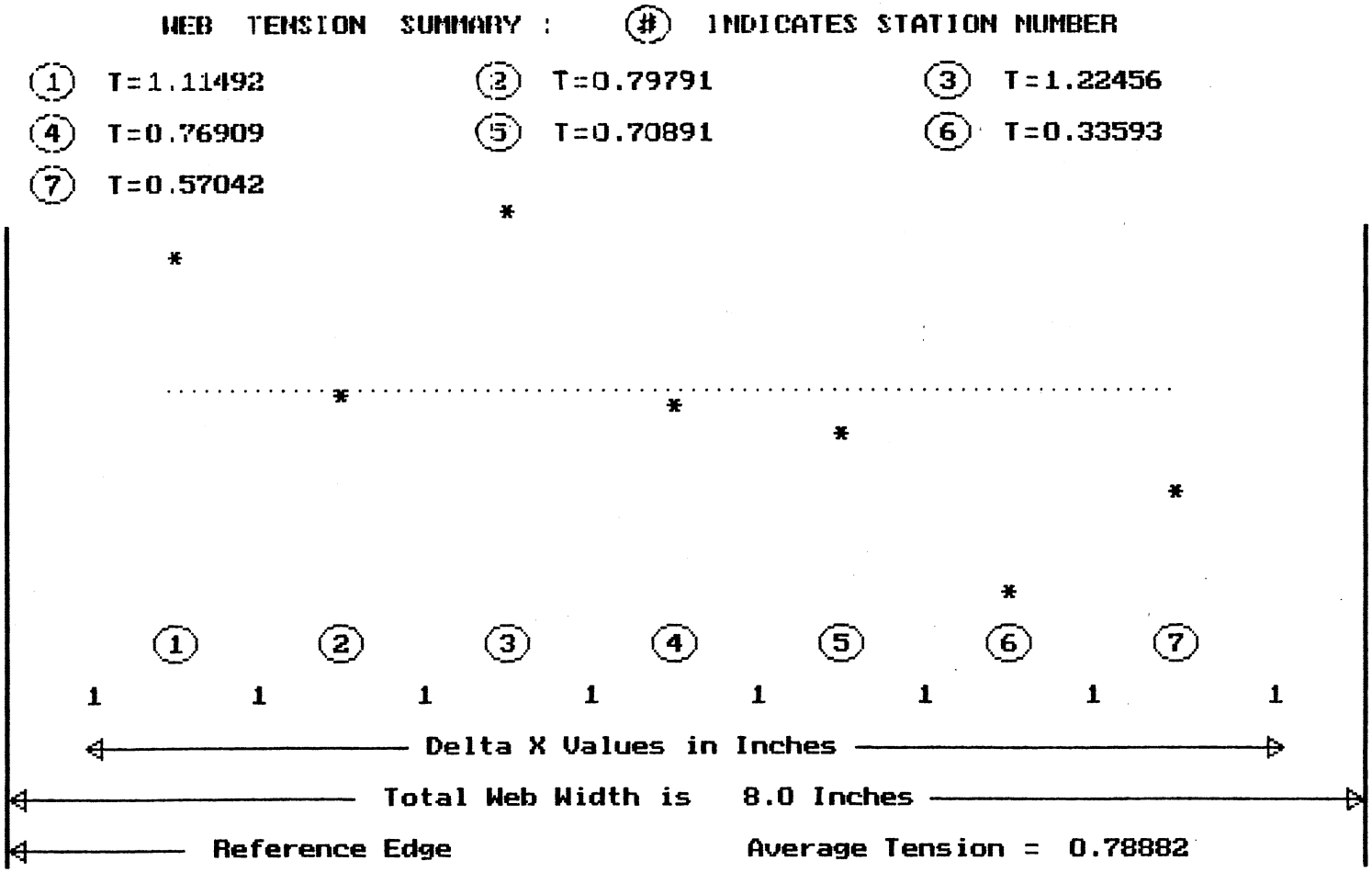
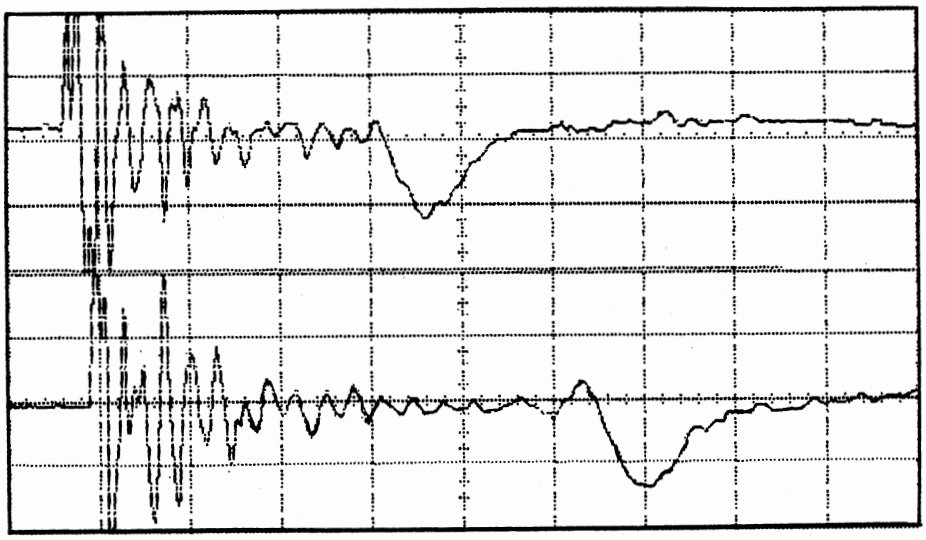
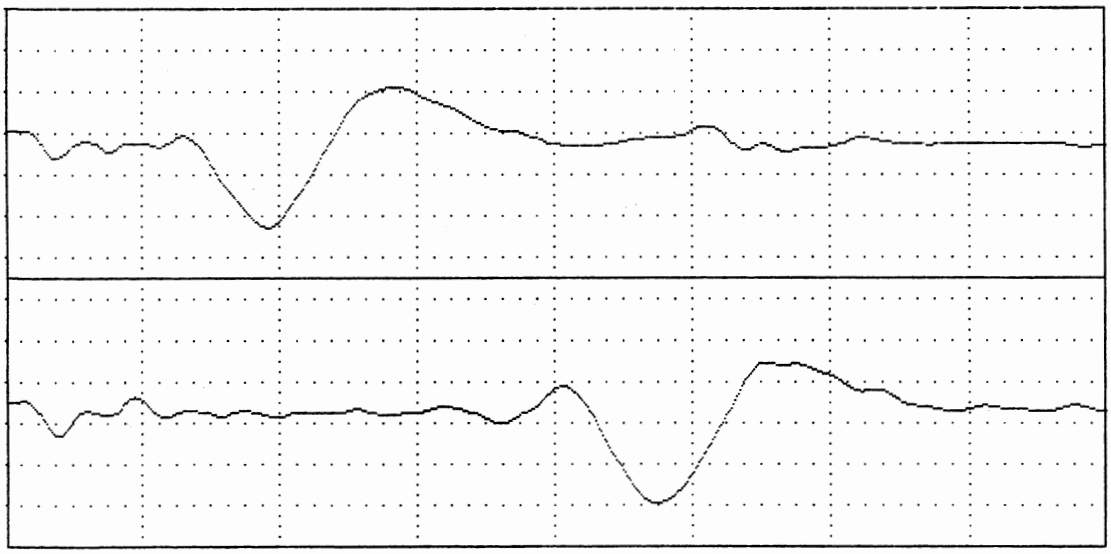


Figure 3.6-10. Test E - 5 mil Nomex Tension Profile Summary



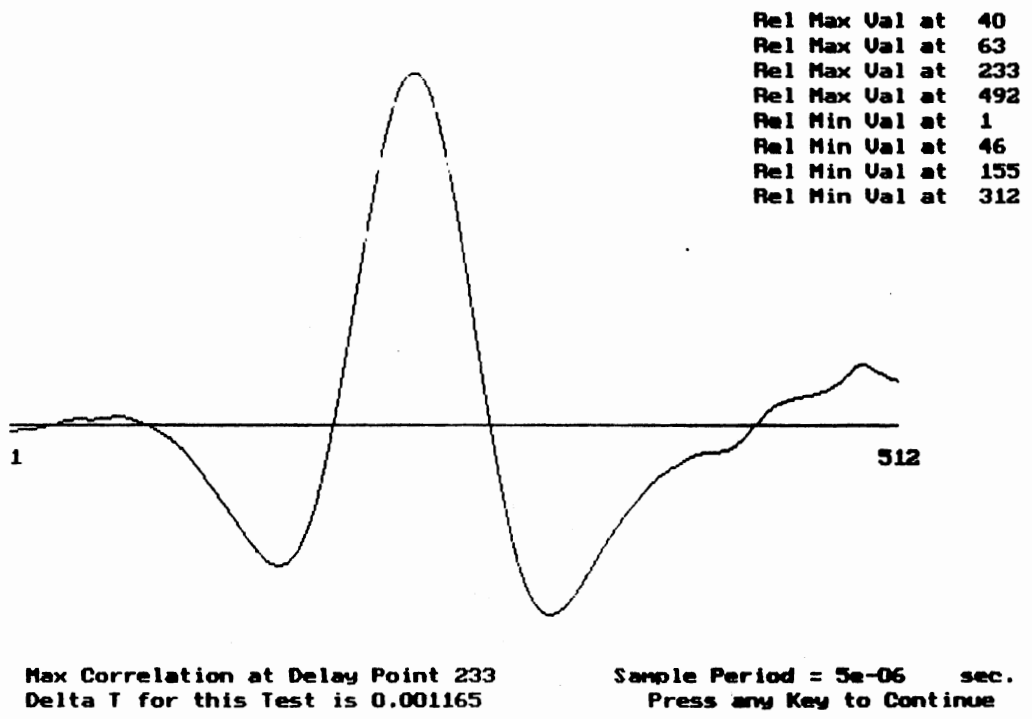
Time Base 500 microsec/div

(a) Raw Unfiltered Signal

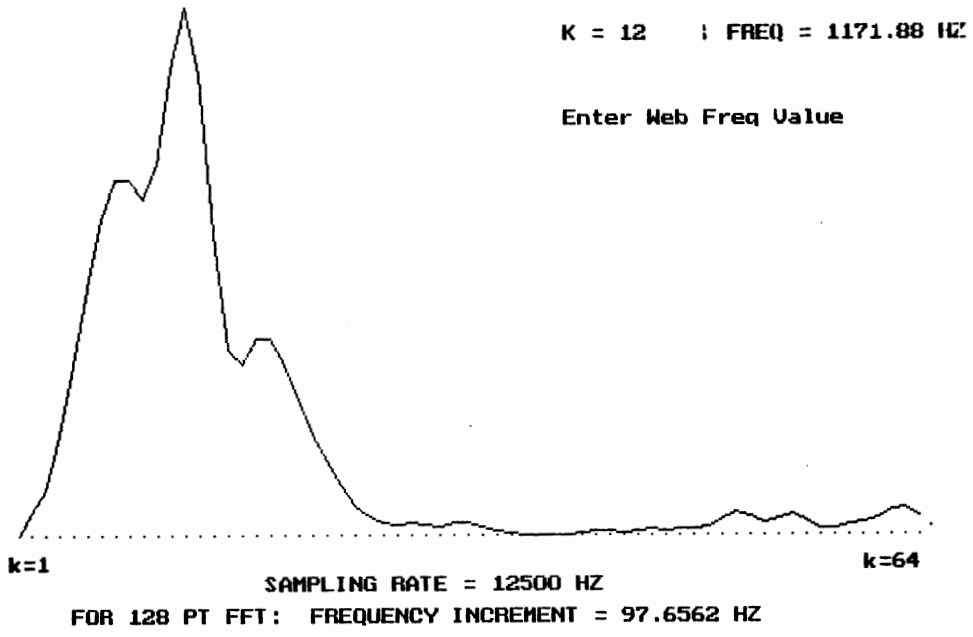


(b) Digital Filtered Signal

Figure 3.6-11. Test F—3 Mil Clear Plastic Material



(a) Cross Correlation Function



(b) Spectral Density Function

Figure 3.6-12. Test F Signal Processing Results

TABLE 3

TIME OF FLIGHT, CHARACTERISTIC FREQUENCY, AND IN-VACUO TENSION FOR TEST G TENSION PROFILE

Station	Δt (msec)	f_{web} (Hz)	$T_{IN\text{-}VACUO}$ (pli)
1	1.150	830.1	1.262
2	1.150	805.7	1.262
3	1.170	1220.7	1.219
4	1.165	1220.7	1.229
5	1.155	1220.7	1.251
6	1.155	830.1	1.251

The tension profile summary is provided in Figure 3.6-13, which indicates an average tension of 1.437 pli. Notice the lack of variation of the indicated tension levels about the average tension level. Maximum variation occurs at station 2 and is less than 5%. This heavier, stiffer web material allowed for very even set up on the laboratory test stand and thus a very even tension distribution was possible.

Test H was performed on paper, a 4 mil slick surfaced coated paper with area density 1.708 (10^{-4}) lbm/in.^{2}. Applied tension was 1.858 pli over the 12-inch web width. Program WH1DF was used at a sample rate 500 kHz with filter parameters f_c = 1000 Hz and BW = 1000 Hz. Figures 3.6-14a and 3.6-14b provide the raw oscilloscope trace and digital filtered trace, respectively, where the transducer head was located near the web center. Thicker, stiffer web materials at higher tension levels will not deflect as far as will thin webs, as one would expect. Solenoid pulser had some difficulty deflecting this paper web material. Notice in Figure 3.6-14a the sensitivity values of 400 millivolts per division. This is the first test case examined thus far where the need for this higher sensitivity level was required. Signal to noise ration has been lowered such that sensitivity to filtering becomes more pronounced.

Figure 3.6-14b illustrates maximum signal "windowing" that one may utilize. Here, the bare minimum of the flexural waveform signals is retained such that cross correlation may be achieved with minimal noise influence. Figure 3.6-15a is the cross correlation function corresponding to

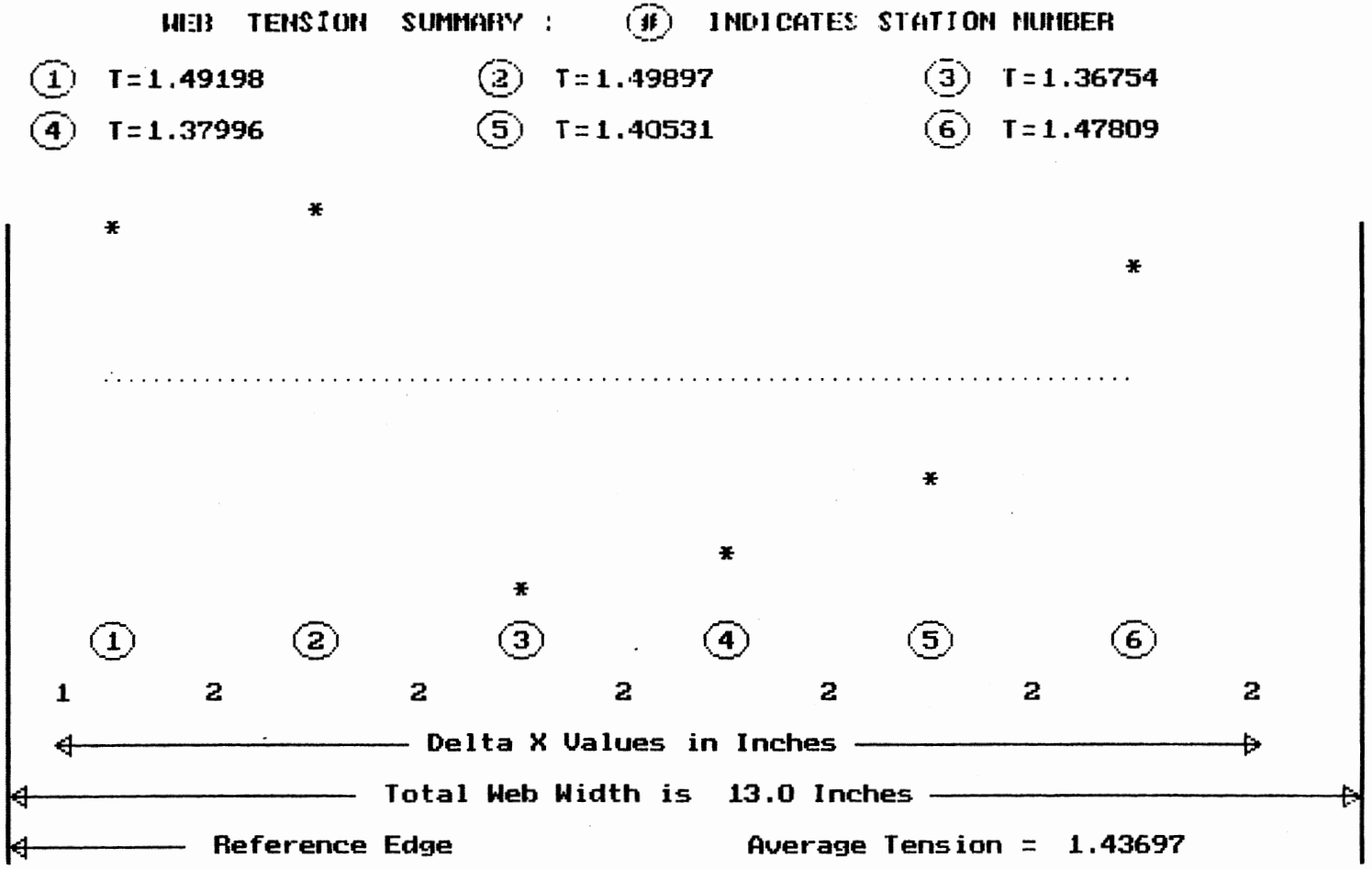
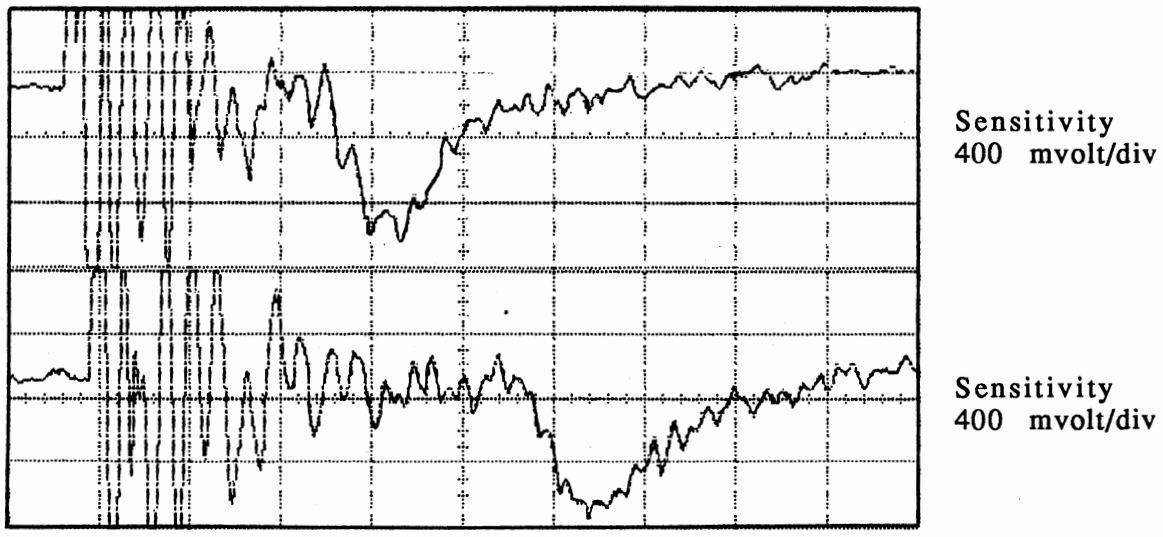
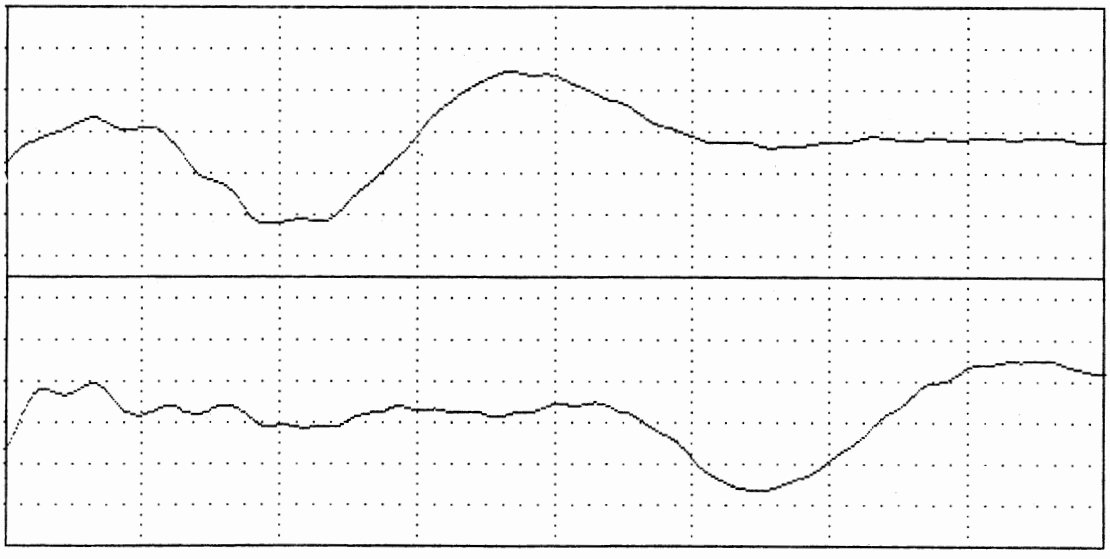


Figure 3.6-13. Test G—3 Mil Clear Plastic Tension Profile Summary



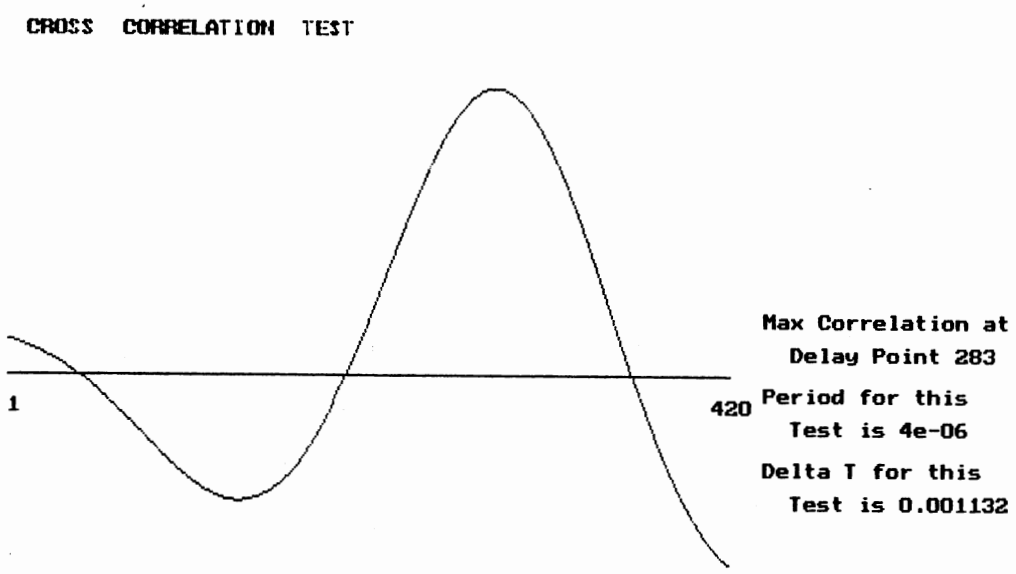
Time Base 500 microsec/div

(a) Raw Unfiltered Signal

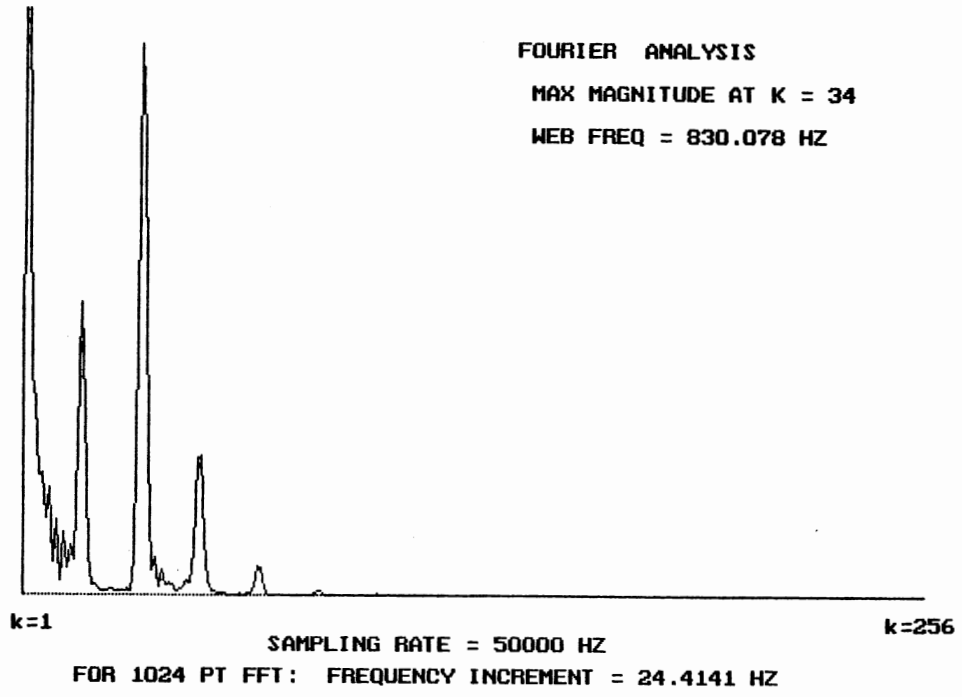


(b) Digital Filtered Signal

Figure 3.6-14. Test H Signals - 4 mil Coated Paper



(a) Cross Correlation Function



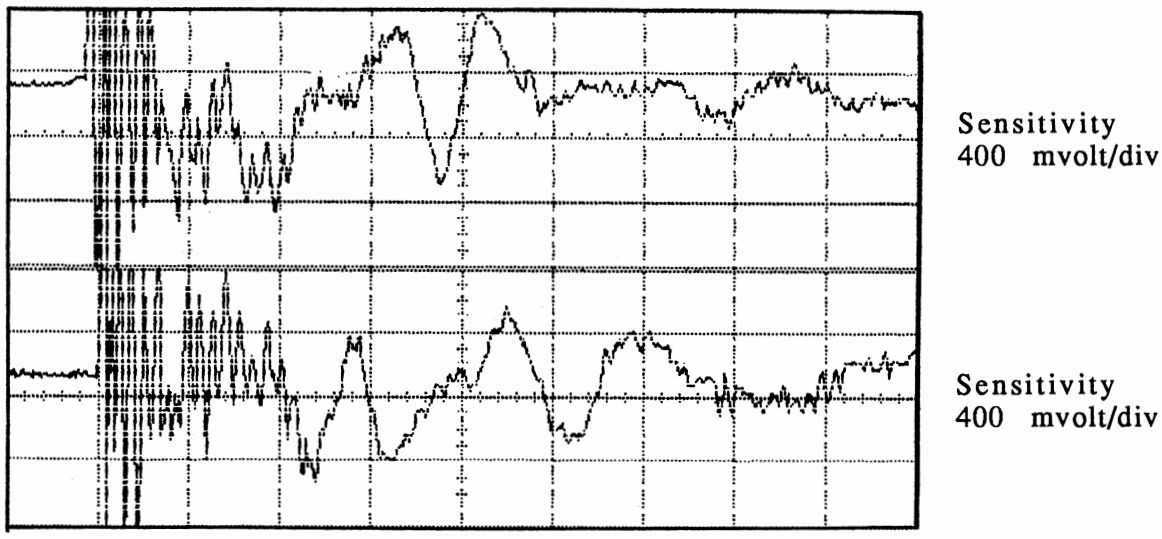
(b) Spectral Density Function

Figure 3.6-15. Test H Signal Processing Results

the waveforms of Figure 3.6-14b, where time of flight is given as 1.132 msec. This is the classical cross correlation function shape seen many times in earlier sections of this chapter. Figure 3.6-15b is the corresponding spectral density function. In this procedure, the average value of the assembled pseudoperiodic record is calculated and then subtracted from each term. At times, however, a DC component continues to show up in the Fourier transform, as can be seen in Figure 3.6-15b. In determination of characteristic frequency, this DC term is ignored, whereupon the spike at k = 34 was selected for an f_{web} value of 830.6 Hz. Converting this data revealed tension indications of 1.381 pli and 1.623 pli for in-vacuo and ribbon equations, respectively.

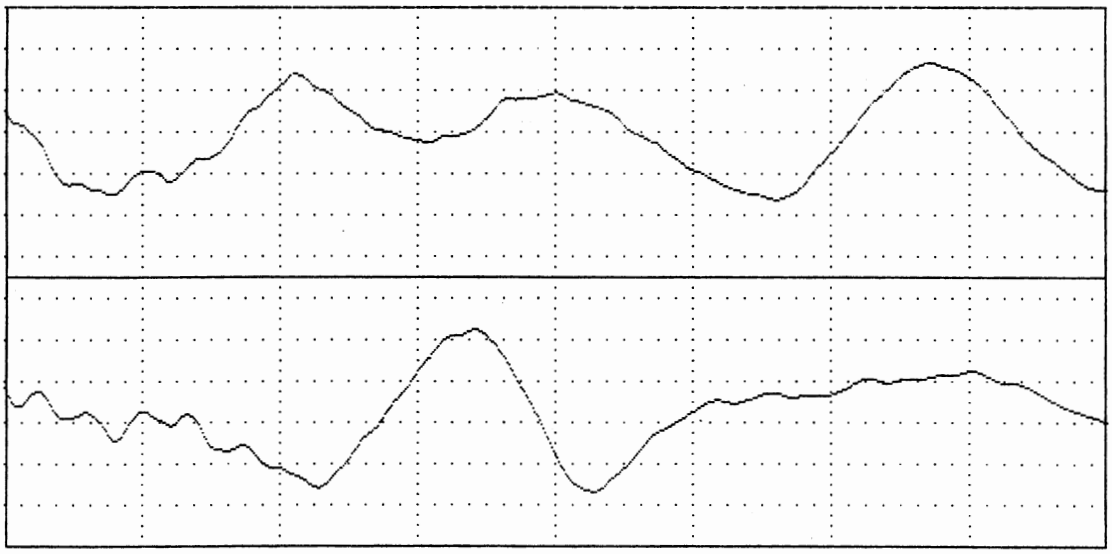
To further examine this paper web material, a tension profile, Test I, was measured. Default filter parameters were used, f_c = 1000 Hz, BW = 1500 Hz, with 400 kHz sample rate. Transducer head was moved from the web center to near one edge, where Figure 3.6-16a and 3.6-16b represent the raw oscilloscope and digital filtered signals, respectively. This characteristic is dramatically different from that observed in Test H, which was motivation for increased filter bandwidth on this test. Table 4 provides calculated variables for this profile test, where six tests were performed at 2-inch increments across the 12-inch web span. Figure 3.6-17 is the graphical summary for this test, where indicated average tension of 1.969 pli is in rather good agreement with the average applied tension of 1.858 pli. Note the very high tension value at station 6. Figure 3.6-18 is an oscilloscope trace at this location, where Δt may be estimated at approximately 0.8 msec, which lends support to the 0.835 msec value given in Table 4 and thus the high tension value. An additional single point test using program WH1FC was performed at this location, which yielded Δt = 0.725, f_{web} = 1269.5 Hz, $T_{IN\text{-}VACUO}$ = 3.367 pli, and T_{RIBBON} = 3.976 pli. It was believed that laboratory web set up was fairly uniform with maximum tension in the web midspan and less tension toward the web edges. This illustrates how difficult it really is to judge web tension distribution and how helpful this system may be as a measuring tool.

The last tests to be examined were performed on 10 mil type 411 Nomex material, which had an 8-inch span and area density of 1.08 (10^{-4}) lbm/in.^{2} upon which 0.97 pli average tension was applied. Note the light weight nature of this material. This was the thickest material tested, yet was a pliable material, lacking stiffness of some of the other materials tested.



Time Base 1.0 millisec/div

(a) Raw Unfiltered Signal



(b) Digital Filtered Signal

Figure 3.6-16. Test I Signals—4 Mil Coated Paper Material

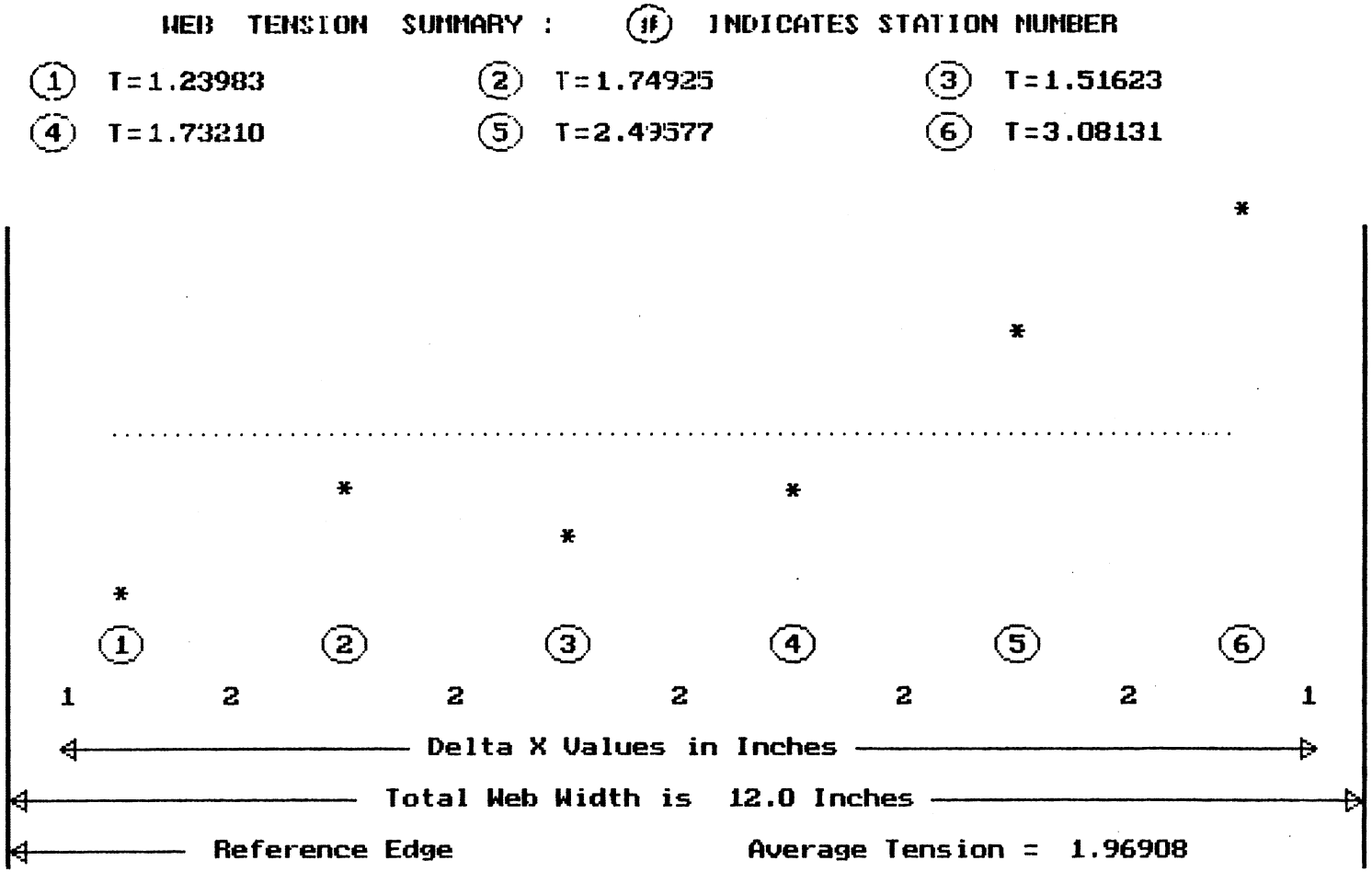


Figure 3.6-17. Test I—4 Mil Coated Paper Tension Profile Summary

TABLE 4

TIME OF FLIGHT, CHARACTERISTIC FREQUENCY, AND IN-VACUO TENSION FOR TEST H TENSION PROFILE

Station	Δt (msec)	f_{web} (Hz)	$T_{IN\text{-}VACUO}$ (pli)
1	1.275	927.7	1.088
2	1.085	927.7	1.503
3	1.160	927.7	1.315
4	1.090	927.7	1.490
5	0.920	927.7	2.091
6	0.835	927.7	2.538

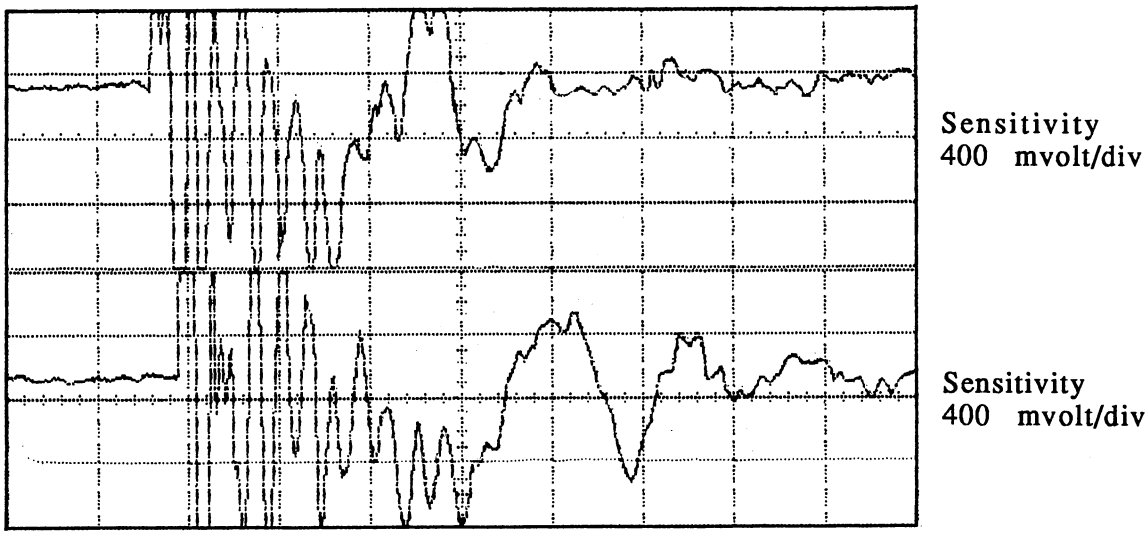


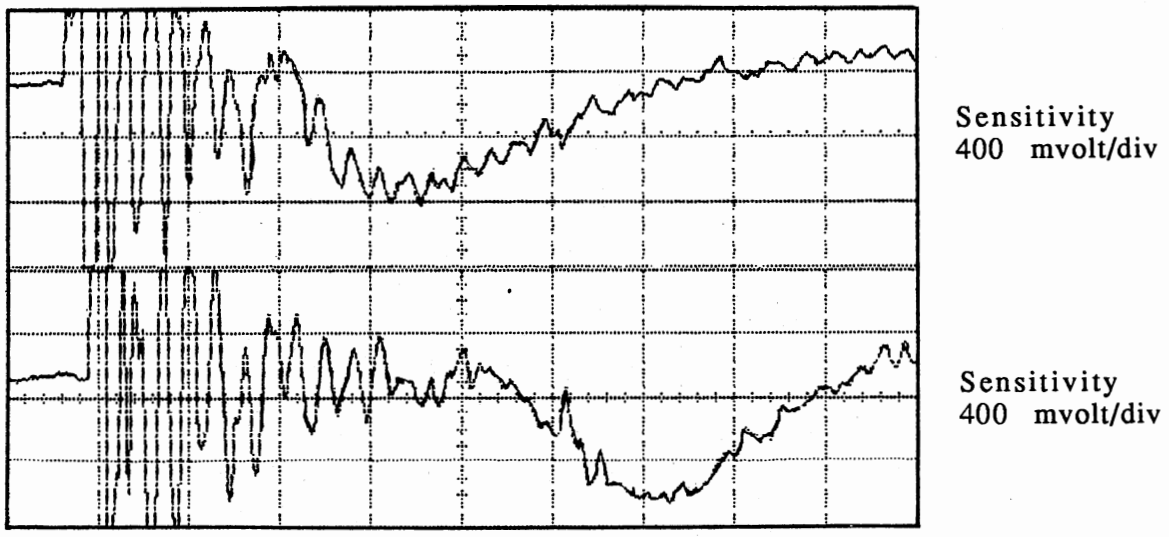
Figure 3.6-18. Raw Signal - Figure 3.6-17 Station 6

Test J is a single point test using program WH1FC with filter parameters f_c = 600 Hz, BW = 1000 Hz and sample rate 300 kHz. Figures 3.6-19a and 3.6-19b are the raw oscilloscope signals and digital filtered signals for this test, respectively, where the transducer head was located at web midspan. Visible is a lazy response, which predicated use of a low filter center frequency. Estimated from the raw oscilloscope signals are Δt and f_{web} values of 1.3 msec and 452 Hz, respectively. Figures 3.6-20a and 3.6-20b are cross correlation and spectral density functions for this waveform. From this figure, Δt = 1.33 msec and f_{web} = 293 Hz, resulting in tension indications of 0.629 pli for in-vacuo and 1.048 pli for ribbon models. Lack of frequency domain resolution may cause problems in cases such as this where characteristic frequency of the signal is low. These tension results are quite interesting and will be referred to later after examination of a tension profile obtained on this 10 mil Nomex material.

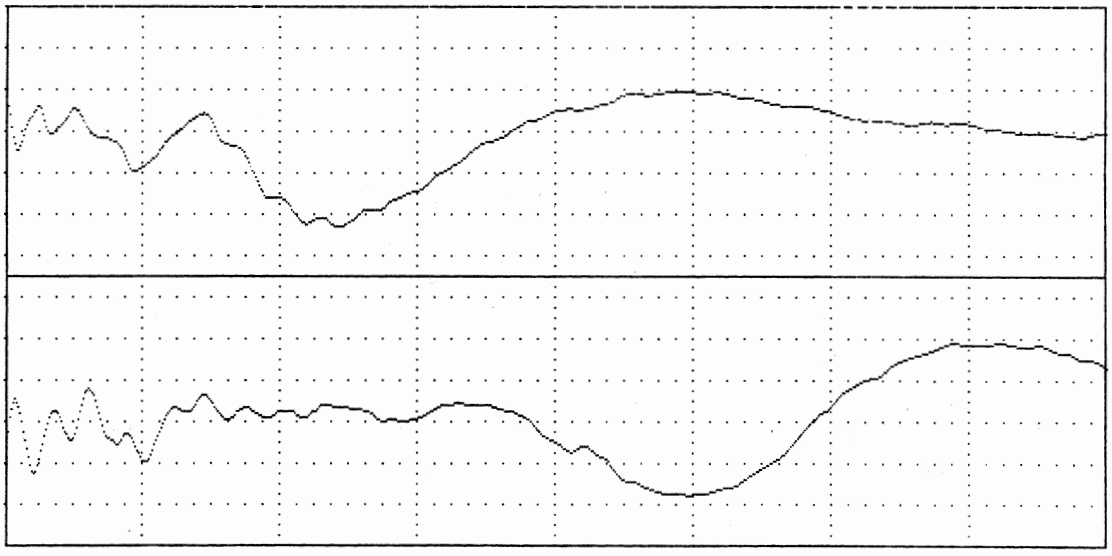
A tension profile was performed on the above Nomex material to assess response of the tension measurement system on this thick yet light, pliable material. Several attempts were made using various combinations of filter parameters, all yielding the same basic result, which was overly high values of ribbon equation tension. In-vacuo tension indications were more in line with the applied average tension, 0.97 pli. The profile to be described below illustrates this apparent air loading error for this particular material.

Test K was performed using filter parameters f_c = 600 Hz, BW = 1000 Hz, and 400 kHz sample rate. Transducer head was initially positioned 1 inch from the material edge, corresponding to Figure 3.6-21a. From the figure, voltage sensitivity is 1 volt per division, as has been typically used, yet peak to peak amplitude of generated flexural waveform signals is only approximately 1 volt. Filtered signal, Figure 3.6-21b, shows some similarity between upstream and downstream signals such that one would anticipate creditable system performance.

Figure 3.6-22 is the graphical summary from this test, with Table 5 providing time of flight, characteristic frequency, and in-vacuo information. Average tension indications are 1.713 pli from Figure 3.6-22 and 1.114 pli from Table 5 in-vacuo tension values. Time of flight information alone would appear to provide for better tension measurements in this case than do time of flight plus characteristic frequency information.

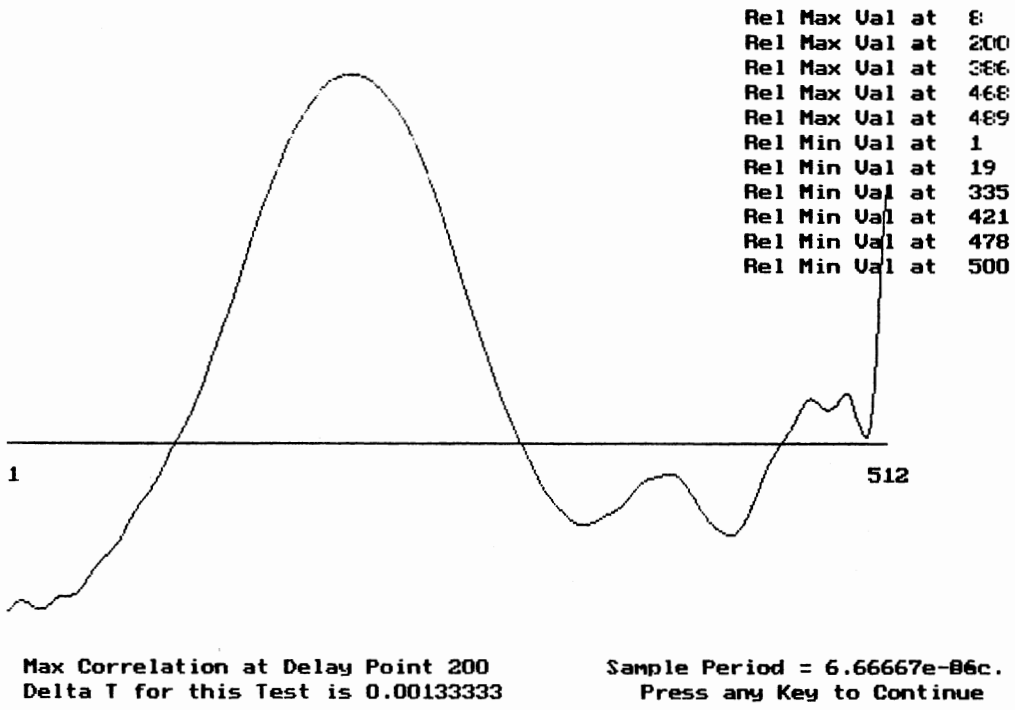


(a) Raw Unfiltered Signal

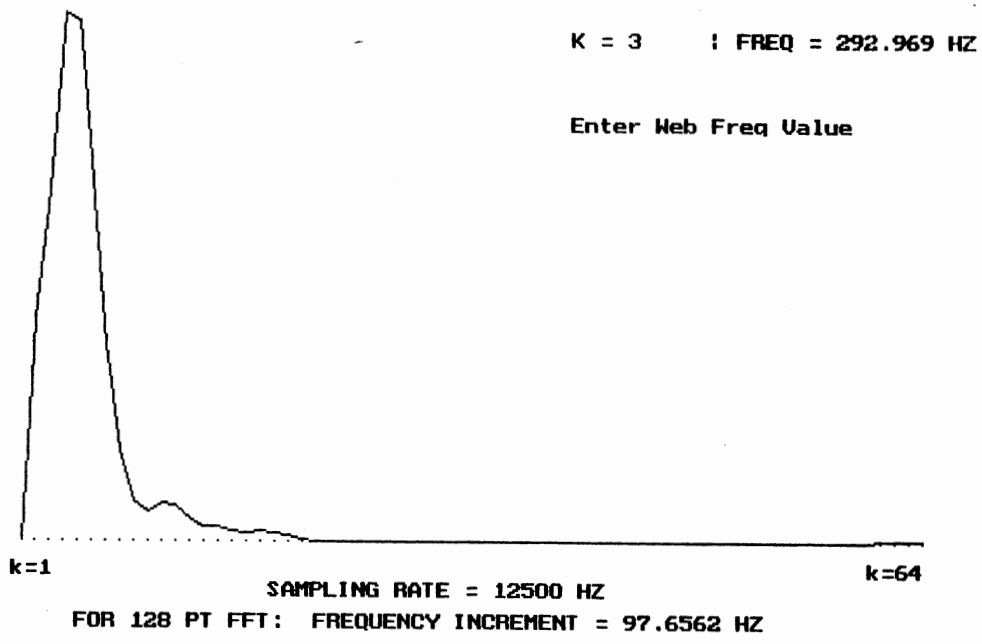


(b) Digital Filtered Signal

Figure 3.6-19. Test J Signals—10 Mil Type 411 Nomex Material

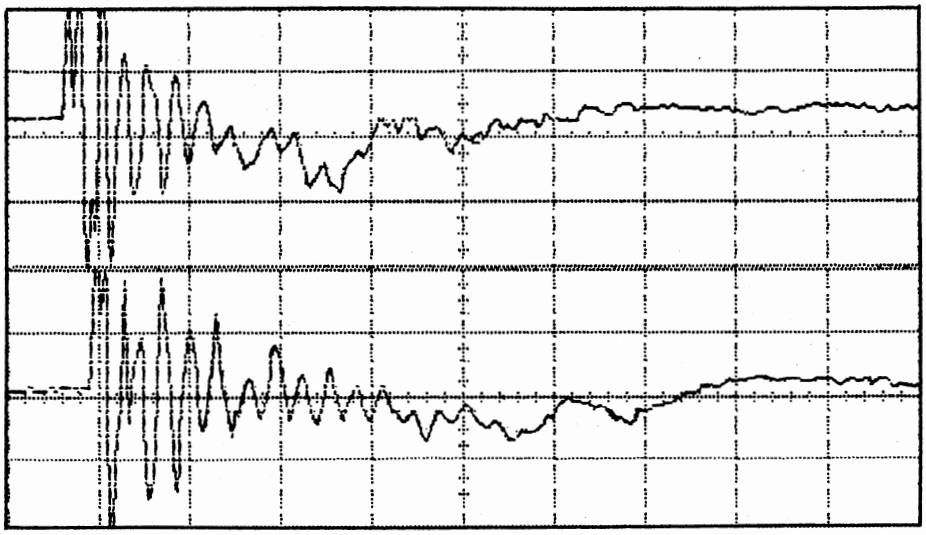


(a) Cross Correlation Function



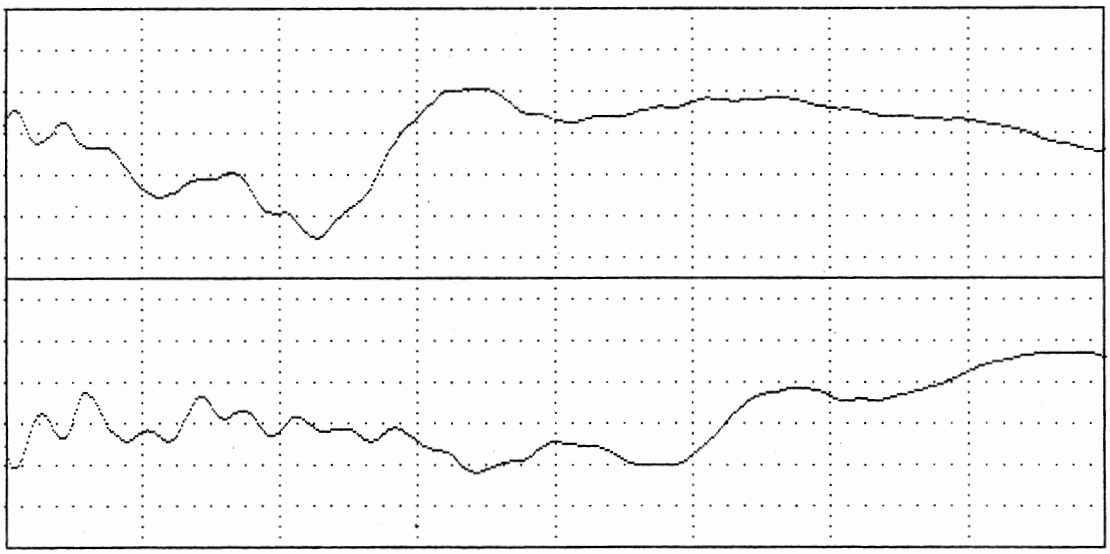
(b) Spectral Density Function

Figure 3.6-20. Test J Signal Processing Results



Time Base 500 microsec/div

(a) Raw Unfiltered Signal



(b) Digital Filtered Signal

Figure 3.6-21. Test K Signals—10 Mil Type 411 Nomex Material

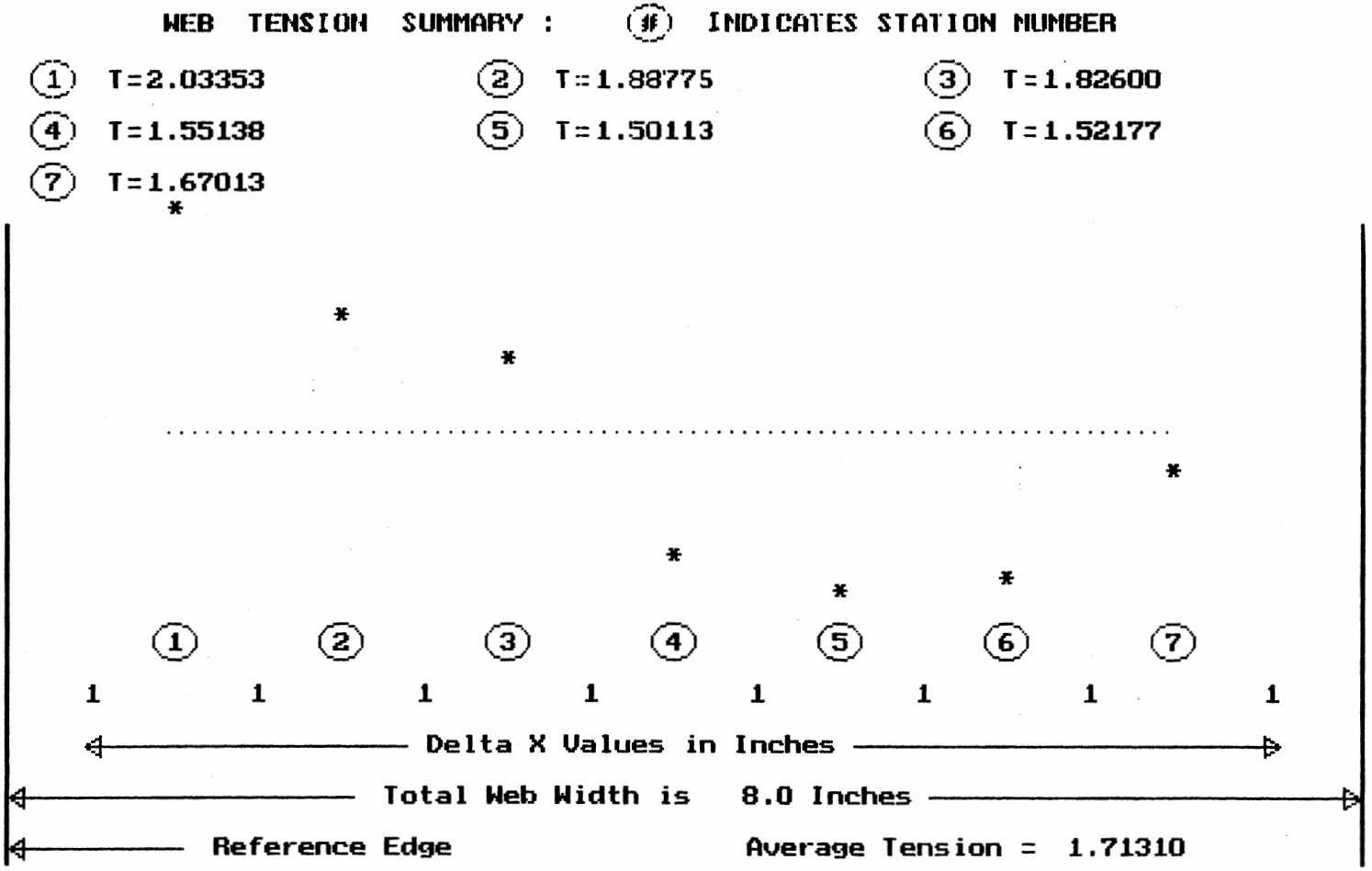


Figure 3.6-22. Test K—10 Mil Nomex Tension Profile Summary

TABLE 5

TIME OF FLIGHT, CHARACTERISTIC FREQUENCY, AND IN-VACUO TENSION FOR TEST K TENSION PROFILE

Station	Δt (msec)	f_{web} (Hz)	$T_{IN\text{-}VACUO}$ (pli)
1	0.965	390.6	1.202
2	0.905	756.8	1.366
3	1.020	366.2	1.076
4	1.080	390.6	0.960
5	1.095	390.6	0.933
6	0.995	756.8	1.130
7	0.995	756.8	1.130

Average in-vacuo tension is 1.114 pli.

Ribbon Equation (3.15) was derived based on the assumption on an air loaded membrane through the membrane dispersion relation (3.12). The effect of this air loading was inversely proportional to frequency of flexural membrane wave travel, which has been referred to as characteristic frequency in this report. Thus, air loading correction of Equation (3.15) is more significant for low characteristic frequency values. For Tests J and K on 10 mil type 411 Nomex, very low values of characteristic frequency were evident. In such cases the effect of the ribbon equation air loading correction term on indicated tension will be much greater than for web material/web tension combinations where characteristic frequency is much higher. Results from Test J could easily be misinterpreted. Test J oscilloscope trace suggests that a soft spot in the material is being examined, which is supported by the rather large time of flight value produced. Tension indications, 0.629 pli for in-vacuo and 1.048 for ribbon models, might lead one to assume that the larger ribbon equation tension value is correct due to close proximity to the applied average 0.97 pli tension. More likely, however, is that the in-vacuo tension value is closer to the actual web tension for this particular case. Thus, for cases where characteristic frequency is quite low, one may desire to examine tension indications based on time of flight only. Again, trial and error experimentation is necessary to formulate such decisions due to no hard and fast rule being available for reference.

Presented above has been a series of laboratory tests in an effort to provide a fairly comprehensive examination of the application and performance of the point source local tension measurement system which is the main topic of this report. A variety of materials over a range of nominal tension levels were tested and results presented. Single-point tension tests along with profile tension tests were performed to examine innerworkings of the measurement process as well as performance over a web span where applied average tension was known. Versatility of variable signal conditioning facility was demonstrated as was performance of different signal processing approaches. Some concluding remarks may be made at this point having examined of tests provided in this chapter.

Performance of the point source pneumatic pulse local tension measurement system appears to be quite good for web materials 3 mils thick or less for all practical nominal tension levels. As has been seen in this section, some thicker materials have been successfully tested, but problems may arise for stiffer, thicker materials experiencing high tension. Materials must be nonporous for proper flexural wave generation. In all cases, experimentation is necessary to formulate optimal system filtering parameters to provide most accurate tension indications.

During these discussions, system performance has been given in terms of adjectives rather than as numerical values. Tension measurement system accuracy may be examined qualitatively at present but not quantitatively. As is, the tension measurement system requires no calibration prior to use. Proper signal conditioning parameter adjustment allows for "optimal" performance. System tension indications may be compared to average applied tension to gauge performance. An error (sensitivity/uncertainty) analysis was performed in section 4 to illustrate the error contribution of each variable to resultant indicated tension error. This error analysis allowed for qualitative assessment of system accuracy, but a quantitative accuracy evaluation may not be performed until an "exact" web tension profile is available for use with the system. An exact web tension profile would have tension known accurately at any point along the web span. If this were available, then experimentally achieved tension profiles could be directly compared to this tension standard and numerical values of error could be presented. Development of a method to calibrate a web tension profile would be quite helpful to further aid in development of the tension measurement system. In a couple of instances, an error statement has been presented with respect

to average tension. Applied average tension was known accurately in the laboratory due to the static test conditions. Thus, profile tests produced a measured average tension indication which could be compared to the applied tension amount. It is believed that presentation of quality average tension comparisons in this chapter have made clear what is meant by "very good," "accurate," "reasonable," etc., when referring to system performance.

This concludes discussion of this research effort's experimental aspects. Web manufacturing and web handling industries will have greatest use for this material. Much information has been provided such that decisions regarding applicability of this system in industrial settings may be formulated. From a research viewpoint, however, questions remain regarding pneumatic shock pulse coupling to web materials with subsequent generation of flexural waveforms that may be sensed, processed, and converted to tension indications. An analytic study will be presented in an attempt to answer some of these questions.

CHAPTER IV

PULSE PROPAGATION THEORY

During development of the experimental tension measurement system, references were made to the coupling between the pneumatic pulse and web, which is an unknown phenomenon. If this coupling were known, one could more easily explain why signals obtained through the system transducer head microphones were shaped as they were. One could also more accurately hypothesize as to what materials and tension levels could be used with the system and still obtain dependable, repeatable tension values. A hypothesis regarding the sensed signal shape was presented in section 3.2, but this hypothesis was based on qualitative information rather than on rigorous theory. The pneumatic pulse-to-web coupling phenomena is due to nonlinear compressible, nonisentropic fluid flow, nonlinear plate or membrane dynamics, and fluid loading effects. Considering all of these factors, an exact coupling mechanism analysis is likely not possible. An approximate analysis will be provided in an attempt to show the effects of the pneumatic pulse on a web interface.

Experimental system pneumatic pulses propagation contained two components. To begin, overlap of the rotary pulser disk hole (Figure 3.1-1) with the pulser tube inlet allowed for a compressed air pulse to enter the pulser tube. One-dimensional pulse propagation through the pulser tube allowed for pressure front steepening. An impedance mismatch exists at the pulser tube exit, which was manifested by a snapping sound as the steepened weak shock wave encountered this mismatch at the pulser tube exit. At this point, a second propagation situation begins. Three-dimensional propagation would occur as the weak shock enters the ambient surrounding, or two-dimensional if considered axisymmetric. Subsequent shock wave contact, interaction with, and reflection from the web interface would generate flexural waves in the web material. An analytical examination of this propagation will serve as the basis for an approximate web deflection analysis to follow.

Variables required to describe a fluid particle state in a compressible flow situation are density ρ , pressure p, temperature T, specific volume τ, internal energy e, specific enthalpy h, and specific entropy S [67] where $\rho\tau = 1$. Two of these variables are independent with thermodynamic relations coupling the remaining variables. Entropy is related to internal energy, temperature, pressure, and density through:

$$de = T\,dS - p\,d\left(\frac{1}{\rho}\right) \tag{4.1}$$

Ideal gas assumption allows for constant specific heats C_v and C_p such that the following relations may be used:

$$\text{(a)}\ e = C_v T \qquad \text{(b)}\ h = C_p T \qquad \text{(c)}\ p = \rho R T$$

$$\text{(d)}\ R = C_p - C_v \qquad \text{(e)}\ \gamma = 1 + \frac{R}{C_v} = \frac{C_p}{C_v} \tag{4.2}$$

Substitution of these relations into (4.1) leads to relations involving internal energy, specific enthalpy, pressure, and density:

$$d\left(e + \frac{p}{\rho}\right) = C_p\,dT \tag{4.3}$$

$$h = e + \frac{p}{\rho} = \frac{\gamma}{\gamma - 1}\frac{p}{\rho} \tag{4.4}$$

$$e = \frac{1}{\gamma - 1}\frac{p}{\rho} \tag{4.5}$$

Differential analysis is used to derive the differential equations of motion for a fluid particle in terms of selected independent variables. The application to be considered in this report will be assumed two-dimensional for a differential area lying in the x-y plane. Velocity components for the x and y directions are u and v, respectively. The partial differential equations of motion are available from many sources such as References [68], [69], and [70], and thus will simply be presented below. Eulerian coordinates have been selected, which is akin to a stationery observer viewing a set of field points. Of interest in this analysis is the behavior and propagation of fluid particles with respect to stationery grid points.

Conservation of Mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{4.6}$$

Conservation of X Momentum:

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2 + p) + \frac{\partial}{\partial y}(\rho u v) = 0 \tag{4.7}$$

Conservation of Y Momentum:

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v^2 + p) = 0 \tag{4.8}$$

Conservation of Energy:

$$\frac{\partial}{\partial t}(\rho E_s) + \frac{\partial}{\partial x}((\rho E_s + p) u) + \frac{\partial}{\partial y}((\rho E_s + p) v) = 0 \tag{4.9}$$

The quantity E_S is the total, stagnation energy given by:

$$E_s = e + \frac{1}{2}(u^2 + v^2) \tag{4.10}$$

Originally, seven variables were presented with the qualifier that only two of the variables were independent. Two additional variables, the u and v velocities, have been introduced. Using ρ and E_S as the remaining independent variables, then Equations (4.1), (4.2a), (4.2b), (4.2c), and (4.4) relate S, h, p, and T to the ρ and E_S variables. An equation of state will consolidate these relations through use of Equations (4.2a), (4.2c), and (4.2e) as follows:

$$p = \rho R T = \rho e\left(\frac{C_p - C_v}{C_v}\right) = \rho e(\gamma - 1) \tag{4.11}$$

The energy variable of interest is the total, stagnation energy given by Equation (4.10). Rearranging Equation (4.10) for substitution into Equation (4.11) results in the equation of state for this analysis:

$$p = \rho\left[E_s - \frac{1}{2}(u^2 + v^2)\right](\gamma - 1) \tag{4.12}$$

The equation set (4.6) through (4.9), when placed in vector form, is known as the "short form" [71] of the inviscid, compressible flow equations:

$$\frac{\partial}{\partial t}\{F\} + \frac{\partial}{\partial x}\{G\} + \frac{\partial}{\partial y}\{H\} = \{0\}$$

$$\frac{\partial}{\partial t}\begin{Bmatrix}\rho \\ \rho u \\ \rho v \\ \rho E_s\end{Bmatrix} + \frac{\partial}{\partial x}\begin{Bmatrix}\rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho E_s + p) u\end{Bmatrix} + \frac{\partial}{\partial y}\begin{Bmatrix}\rho v \\ \rho u v \\ \rho v^2 + p \\ (\rho E_s + p) v\end{Bmatrix} = \begin{Bmatrix}0 \\ 0 \\ 0 \\ 0\end{Bmatrix} \tag{4.13}$$

This equation set will be used in pneumatic shock pulse modeling to be developed in this chapter.

Pneumatic pressure pulse propagation along the pulser tube may be described as a one-dimensional adaptation of the (4.13) equation set. Nonlinear effects are utilized in this process to allow for pressure pulse steepening during propagation along the pulser tube length. If an ideal gas is assumed, pressure front steepening occurs due to the change of phase speed of the gas with respect to the pressure front [72]. As the propagation occurs, air particles near the pulser tube boundary travel faster than particles near the pulser tube centerline. Blackstock [73, 74] has provided methods of simulating this steepening phenomena for tube-piston arrangements through power series analyses. Figure 4.1 from Reference [74] illustrates the result for nonlinear, small signal propagation processes. If sufficient pulser tube length is used, adequate steepening may occur to yield the weak shock wave "snap" that is heard as the shock front exits the pulser tube end. Noted earlier, project associate Ahn [55] experimented with different pulser tube size and lengths of in the development of this portion of the project.

Pneumatic pulse propagation may be examined through study of characteristics of the applicable partial differential equation system. One-dimensional pulse propagation is applicable to the pressure pulse travel and steepening along the pneumatic pulser tube, where independent space variable x and time t are applicable. A one-dimensional analysis is also applicable at the point where the weak shock front first exits the pulser tube end before two-dimensional analysis becomes necessary. A two-dimensional analysis of system characteristics does not provide a great deal of unique information above that of a one-dimensional analysis. Of interest is to become familiar with the propagation process with regard to the fluid properties density, velocity, and stagnation energy.

A one-dimensional analysis of the partial differential equation set (4.13) may be accomplished through the following equation set:

$$\text{a.)} \qquad \frac{\partial \rho}{\partial t} + \frac{\partial (\rho\, u)}{\partial x} = 0$$

$$\text{b.)} \qquad \frac{\partial (\rho\, u)}{\partial t} + \frac{\partial (\rho\, u^2 + p)}{\partial x} = 0$$

$$\text{c.)} \qquad \frac{\partial (\rho\, E_s)}{\partial x} + \frac{\partial ((\rho\, E_s + p)\, u)}{\partial x} = 0 \qquad (4.14)$$

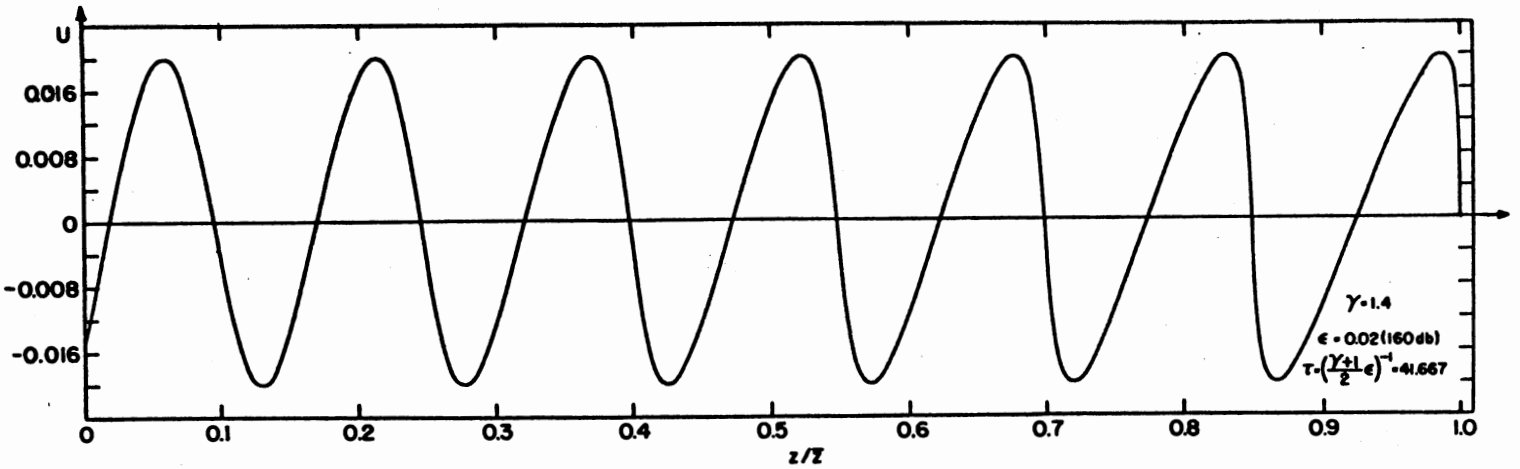


FIG. 10 WAVE OF FINITE AMPLITUDE AT THE INSTANT OF SHOCK FORMATION (FROM (64')).

Figure 4.1. Simulation of Pressure Front Steepening Through Power Series Analysis [75]

System characteristics study provides the speed of fluid particle propagation through solution of system eigenvalues. System eigenvectors reveal relations between the density, velocity, and energy for a given eigenvalue. This is akin to principal coordinates of a linear equation system. Discontinuities of a solution to the above equation system may only occur along characteristics [75] which is applicable to the weak shock wave assumption.

For independent variables x and t, equation set (4.14) may be placed in the form [76]:

$$\frac{\partial}{\partial t}\{U\} + [A]\frac{\partial}{\partial x}\{U\} + [B] = \{0\} \tag{4.15}$$

A characteristic curve for such a system may be described by a function $\phi(x,t)$ = constant. The system characteristic equation may be obtained through determinant formulation below:

$$\left|[I]\frac{\partial \phi}{\partial t} + [A]\frac{\partial \phi}{\partial x}\right| = 0 \;;\; \lambda = -\frac{\partial \phi/\partial t}{\partial \phi/\partial x} = \frac{dx}{dt}$$
$$\left|[A] - \lambda[I]\right| = 0 \tag{4.16}$$

For the nonisentropic, compressible flow equations in conservation form, roots of the above characteristic equation will be real quantities denoting a hyperbolic system. Again, these roots are the system eigenvalues which represent the speed of propagation of the wavefront in question. Right eigenvectors corresponding to the system eigenvalues may be formulated through:

$$[A]\left\{1^{(i)}\right\} = \lambda^{(i)}\left\{1^{(i)}\right\} \tag{4.17}$$

where $\left\{1^{(i)}\right\}$ represents the eigenvector notation for the ith eigenvector. Such an analysis may be extended to examine the partial differential equation system Riemann invariants. Riemann invariants are functions of the fluid property variables density ρ, velocity u, and stagnation energy E_S that are constant along characteristic lines. Thus for the above one-dimensional system there may exist nine such functions due to the possibility of having three eigenvectors with three components in each vector. To gauge the relationship between the above noted fluid properties and the system eigenvectors, one may apply the following equation [76]:

$$\frac{dU_1}{1_1^{(i)}} = \frac{dU_2}{1_2^{(i)}} = \frac{dU_3}{1_3^{(i)}} = \text{constant} \tag{4.18}$$

where i = 1, 2, 3 for the three eigenvectors.

To place equation set (4.14) into the form of Equation (4.15), begin with the continuity equation expanded from Equation (4.14a):

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} = 0 \tag{4.19}$$

Expansion and rearrangement of equations (4.14 b) and (4.14 c) leads to the following results:

$$u\left(\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x}\right) + \rho\frac{\partial u}{\partial t} + \rho\, u\frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} = 0 \tag{4.20}$$

$$E_s\left(\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x}\right) + \rho\frac{\partial E_s}{\partial t} + p\frac{\partial u}{\partial x} + \rho\, u\frac{\partial E_s}{\partial x} + u\frac{\partial p}{\partial x} = 0 \tag{4.21}$$

As can be seen, the continuity Equation (4.19) is enclosed in brackets in Equations (4.20) and (4.21) and is equal to zero. The desired equation set is then easily obtained as follows:

$$\text{(a.)}\quad \frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} = 0$$

$$\text{(b.)}\quad \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0$$

$$\text{(c.)}\quad \frac{\partial E_s}{\partial t} + \frac{p}{\rho}\frac{\partial u}{\partial x} + u\frac{\partial E_s}{\partial x} + \frac{u}{\rho}\frac{\partial p}{\partial x} = 0 \tag{4.22}$$

Equation (4.12) is the equation of state for this analysis, which gives pressure as a function of density ρ, velocity u, and stagnation energy E_S. Partial derivative of pressure with respect to x is then given by the following equations:

$$\begin{aligned} p(\rho,u,E_s) &= \rho\,(\gamma - 1)\left[E_s - \frac{1}{2}u^2\right] \\ \frac{\partial p}{\partial x} &= \frac{\partial p}{\partial \rho}\frac{\partial \rho}{\partial x} + \frac{\partial p}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial p}{\partial E_s}\frac{\partial E_s}{\partial x} \\ \frac{\partial p}{\partial x} &= c^2\frac{\partial \rho}{\partial x} - \rho\,(\gamma-1)\,u\frac{\partial u}{\partial x} + \rho\,(\gamma-1)\frac{\partial E_s}{\partial x} \end{aligned} \tag{4.23}$$

where speed of sound in the fluid c equals $\sqrt{\partial p/\partial \rho}$. Substitution of Equation (4.23) into Equations (4.22b) and (4.22c) with rearrangement leads to the system formulation in the form of Equation (4.15):

$$\frac{\partial}{\partial t}\begin{Bmatrix}\rho\\ u\\ E_s\end{Bmatrix} + \begin{bmatrix} u & \rho & 0 \\ \frac{c^2}{\rho} & u\,(2-\gamma) & (\gamma-1) \\ \frac{u\,c^2}{\rho} & c^2 - u^2(\gamma-1) & \gamma\, u \end{bmatrix} \frac{\partial}{\partial x}\begin{Bmatrix}\rho\\ u\\ E_s\end{Bmatrix} = \begin{Bmatrix}0\\0\\0\end{Bmatrix} \tag{4.24}$$

Application of Equation (4.16) to the above [A] matrix and resultant characteristic equation factoring yields the following eigenvalues:

$$\lambda^{(1)} = u \quad ; \quad \lambda^{(2)} = u + \sqrt{\gamma}\, c \quad ; \quad \lambda^{(3)} = u - \sqrt{\gamma}\, c \tag{4.25}$$

Substitution of these eigenvalues into Equation (4.17) and reduction of the resulting equations reveals the corresponding system eigenvectors:

$$l^{(1)} = \begin{Bmatrix} -\rho\,(\gamma-1) \\ 0 \\ c^2 \end{Bmatrix} = \begin{Bmatrix} -\partial p/\partial E_s \\ 0 \\ \partial p/\partial \rho \end{Bmatrix} \quad ; \quad l^{(2)} = \begin{Bmatrix} \rho \\ \sqrt{\gamma}\, c \\ 0 \end{Bmatrix} \quad ; \quad l^{(3)} = \begin{Bmatrix} \rho \\ -\sqrt{\gamma}\, c \\ 0 \end{Bmatrix} \tag{4.26}$$

To assess the generalized Riemann invariants corresponding to the above eigenvectors, Equation (4.18) may be applied. Eigenvector $l^{(1)}$ from Equation (4.26) provides the following result:

$$-\frac{d\rho}{\partial p/\partial E_s} = \frac{du}{0} = \frac{dE_s}{\partial p/\partial \rho} = \text{constant} \tag{4.27}$$

The relation du = 0 yields the result u = constant. For pressure $p = p(\rho, u, E_S)$, the other two relations provide $\partial p/\partial E_S$ = constant and $\partial p/\partial \rho$ = constant with u = constant already established. Thus, Riemann invariants for this eigenvalue-eigenvector combination are constant velocity and pressure.

Application of Equation (4.18) to eigenvector $I^{(2)}$ of Equation (4.26) yields the following result:

$$\frac{d\rho}{\rho} = \frac{du}{\sqrt{\gamma}\, c} = \frac{dE_s}{0} = \text{constant} \tag{4.28}$$

The relation $dE_S = 0$ provides the result E_S = constant. The remaining relations may be integrated to form the following relation:

$$u - \sqrt{\gamma} \int \frac{c}{\rho}\, d\rho = \text{constant} \tag{4.29}$$

Recall that speed of sound $c^2 = \partial p/\partial \rho = (\gamma-1)\left[E_s - (1/2)\, u^2\right]$ and that E_S = constant has been established. With these substitutions, Equation (4.29) becomes:

$$u - \int \sqrt{\frac{\gamma\,(\gamma-1)\left[E_s - (1/2)\, u^2\right]}{\rho}}\, d\rho = \text{constant} \tag{4.30}$$

Independent variables were assumed to be ρ, u, and E_S with γ a constant such that the integration of Equation (4.30) may be performed with respect to ρ to achieve the final result below:

$$u - 2\sqrt{\rho\,\gamma\,(\gamma-1)\left[E_s - (1/2)\,u^2\right]} = \text{constant} \tag{4.31}$$

Therefore, the Riemann invariants for this eigenvalue-eigenvector combination require constant E_S and $u - 2\sqrt{\rho\,\gamma\,(\gamma-1)\left[E_s - (1/2)\,u^2\right]}$. Visualization of this result is difficult but it does illustrate the innerworkings of the thermodynamic properties of a gas during a propagation process. Application of Equation (4.18) to eigenvector $I^{(3)}$ of Equation (4.26) provides the same information as above with an associated sign change.

The above analysis is an examination of the theory behind shock formation and propagation. Although one-dimensional, the above study of unsteady state, nonisentropic compressible flow equation system characteristics is applicable to the experimental system being analyzed. A two-dimensional study of characteristics can be performed using procedure provided by Jeffrey and Taniuti [76]. Shock pulse interaction with a web would change the system characteristics from that illustrated in the above one- or two-dimensional analysis. Therefore, a general purpose method of integrating the inviscid, nonisentropic, unsteady state, compressible flow equations is necessary for this application. Such an integration process would allow for viewing of the pulse behavior through pressure and velocity evaluation during a complete pulse cycle. A short background discussion of various integration processes will be presented prior to development of the finite difference relations to be used in this study.

The process of inviscid compressional fluid flow equation integration has evolved through the years. Major integration process concerns were with regard to treatment of shock discontinuities and consistent boundary condition formulation in continuous flow problem conversion to discrete problems. Von Neumann and Richtmyer [77] were pioneers in development of the numerical integration methodology. Desired was for a shock interval to be of the same order of width as the discrete grid spacing to avoid aliasing. Additionally desired was to "smear out" the shock discontinuity such that numerical stability might be maintained. This process was to preserve steep gradients in the shock interval while substituting continuous

functions for the discontinuous points. This led to the use of explicit artificial viscosity terms in the governing partial differential equation systems.

Dissipation in compressible, unsteady state flow problems results from two major areas, viscosity effects and heat conduction. Introduction of an explicit artificial viscosity term into the nonviscous, compressible flow Euler equations then allowed for some dissipation, and hence the numerical stability, to occur. Von Neumann and Richtmyer placed the following requirements on the form of the artificial viscosity term [77]:

1. Euler Equations (4.6) through (4.9) must possess solutions without discontinuities.
2. Thickness of the shock layer must be of the order of the grid spacing independent of the shock strength and material in which the shock was propagating.
3. Effect of the artificial viscosity term must be negligible outside of the shock thickness interval.
4. Hugonoit shock relations must hold when dimensions characterizing the flow are large in comparison to the shock thickness interval.

Selected as the artificial viscosity model by Von Neumann and Richtmyer was a term:

$$q = -\frac{(b\,\Delta x)^2}{\tau}\,\frac{\partial u}{\partial x}\left|\frac{\partial u}{\partial x}\right| \tag{4.32}$$

in one dimension where u is the x directional velocity, τ is the gas specific volume, and b is an adjustable constant. For the classic one-dimensional shock tube case, velocity u does not change appreciably with the space variable x outside of the shock region. Hence this artificial viscosity form had the greatest effect in the shock region, where the velocity gradient could become relatively large. Various forms of artificial viscosity terms have been proposed and used in different integration schemes. Pulliam [78] and Roache [71] have provided comprehensive discussions on the merits and effects of the different artificial viscosity terms.

In the shock region, gas velocity, density, pressure, temperature, and entropy are rapidly changing. The Hugonoit equation, given by Equation (4.33) below, was the result of application of conservation of mass, momentum, and energy to the shock interval [72]:

$$(\tau_0 - \tau_1)\frac{(p_1 - p_0)}{2} = e_1 - e_0 \quad ; \quad 0,1\text{: Before,After Shock} \tag{4.33}$$

Maintaining consistent notation, τ is the fluid specific volume and e is the fluid internal energy. Thus, some dissipation was inherent to shock processes by conversion to heat. From Equation (4.33), the level of dissipation was not specified, thus the insertion of the additional artificial viscosity term in the system equations did not adversely affect the Hugonoit relation.

Several methods have been developed for solution of the compressible, inviscid flow equations [71]. The Von Neumann-Richtmyer method used the artificial viscosity term (4.32) in a scheme such that the one-dimensional Equation (4.14) became:

$$\frac{\partial}{\partial t}\begin{Bmatrix}\rho\\ \rho u\\ \rho E_s\end{Bmatrix} + \frac{\partial}{\partial x}\begin{Bmatrix}\rho u\\ p + \rho u^2 - \dfrac{(b\,\Delta x)^2}{\tau}\dfrac{\partial u}{\partial x}\left|\dfrac{\partial u}{\partial x}\right|\\ (\rho E_s + p)u - \dfrac{(b\,\Delta x)^2}{\tau}u\dfrac{\partial u}{\partial x}\left|\dfrac{\partial u}{\partial x}\right|\end{Bmatrix} = \begin{Bmatrix}0\\0\\0\end{Bmatrix} \tag{4.34}$$

Notable in this method was that the dissipation was now proportional to

$$\frac{\partial}{\partial x}\left[\left(\frac{\partial u}{\partial x}\right)^2\right]$$

rather than to

$$\frac{\partial}{\partial x}\left[\frac{\partial u}{\partial x}\right]$$

which allowed for the diffusion to occur over a shorter interval. The b constant required experimentation to select an optimal value. A tradeoff between shock interval thickness and numerical oscillations is dependent on this b value.

An integration method developed by Lax and Wendroff [79] used a Jacobian formulation of the system conservation equations. For the one-dimensional case, Equation (4.14) is cast in the form:

$$\frac{\partial}{\partial t}\begin{Bmatrix}\rho\\ \rho u\\ \rho E_s\end{Bmatrix} + [A]\frac{\partial}{\partial x}\begin{Bmatrix}\rho\\ \rho u\\ \rho E_s\end{Bmatrix} = \begin{Bmatrix}0\\0\\0\end{Bmatrix} \tag{4.35}$$

Elements of the [A] Jacobian matrix are found from the general equation:

$$A_{jk} = \frac{\partial G_j}{\partial F_k} \quad ; \; j,k = 1, 2, 3 \quad ; \; \{F\} = \begin{Bmatrix}\rho\\ \rho u\\ E_s\end{Bmatrix} \quad \{G\} = \begin{Bmatrix}\rho u\\ \rho u^2 + p\\ (E_s + p)u\end{Bmatrix} \tag{4.36}$$

The state Equation (4.12) was used with Equation (4.36) to form the Jacobian matrix for the one-dimensional case:

$$[A] = \begin{bmatrix} 0 & 1 & 0 \\ -\left(\frac{3-\gamma}{2}\right)\rho\, u^2 & (3-\gamma)\, u & (\gamma-1) \\ -\gamma\, u\, E_s - (\gamma-1)\, u^3 & \gamma\, E_s + \frac{3}{2}(\gamma-1)\, u^2 & \gamma\, u \end{bmatrix} \tag{4.37}$$

Finite difference relation for this conservation equation system was then:

$$F_m^{n+1} = F_m^n - \Delta t \frac{\partial G_m}{\partial x} + \frac{1}{2}\Delta t^2 \frac{\partial}{\partial x}\left[A_{m1}\frac{\partial G_1}{\partial x}\right]$$
$$\left[+ A_{m2}\frac{\partial G_2}{\partial x} + A_{m3}\frac{\partial G_3}{\partial x}\right] + O(\Delta t^3)\ ;\ m = 1, 2, 3 \tag{4.38}$$

where the following differencing was used for the $\partial/\partial x\,[A\,\partial G/\partial x]$ terms:

$$\frac{\partial}{\partial x}\left[A\frac{\partial G}{\partial x}\right] = \frac{(A_{i+1} + A_i)(G_{i+1} - G_i) - (A_i + A_{i-1})(G_i - G_{i-1})}{2\ \Delta x^2} + O(\Delta x^2) \tag{4.39}$$

As given, the Lax-Wendroff method does not require an explicit artificial viscosity term. Relatively large numerical oscillations are a consequence of this, such that an artificial viscosity term of the form $(1/2)\, u^2\, \Delta t$ may be used to reduce oscillations.

A more modern and sophisticated method of evaluating the unsteady state compressible Navier Stokes equations was developed by MacCormack [80]. Mehta and Sastri [81] used this method to evaluate the reflection of a shock pulse from a plane interface. Implicit methods were combined with explicit formulations to help increase accuracy and efficiency of the numerical integrations. Predictor-Corrector equations were formulated using the explicit-implicit means such that these equations applied to Equation (4.14) would be:

Predictor Equations:

Explicit

$$\Delta F_{i,j}^n = -\Delta t\left(\frac{G_{i+1,j}^n - G_{i,j}^n}{\Delta x^2} + \frac{H_{i,j+1}^n - H_{i,j}^n}{\Delta y^2}\right) \tag{4.40}$$

Implicit

$$\left([I] - \Delta t\frac{[A]_{i+1,j} - [A]_{i,j}}{\Delta x^2}\right)\left([I] - \Delta t\frac{[B]_{i,j+1} - [B]_{i,j}}{\Delta y^2}\right)\overline{\delta F_{i,j}^{n+1}} = \Delta F_{i,j}^n \tag{4.41}$$

Update

$$\overline{F_{i,j}^{n+1}} = F_{i,j}^{n} + \overline{\delta F_{i,j}^{n+1}} \tag{4.42}$$

Corrector Equations:

Explicit

$$\overline{\Delta F_{i,j}^{n+1}} = -\Delta t \left(\frac{\overline{G_{i,j}^{n+1}} - \overline{G_{i-1,j}^{n+1}}}{\Delta x^2} + \frac{\overline{H_{i,j}^{n+1}} - \overline{H_{i,j-1}^{n+1}}}{\Delta y^2} \right) \tag{4.43}$$

Implicit

$$\left([I] + \Delta t \frac{[A]_{i,j} - [A]_{i-1,j}}{\Delta x^2} \right) \left([I] + \Delta t \frac{[B]_{i,j} - [B]_{i,j-1}}{\Delta y^2} \right) \delta F_{i,j}^{n+1} = \overline{\Delta F_{i,j}^{n}} \tag{4.44}$$

Update

$$F_{i,j}^{n+1} = \frac{1}{2} \left(F_{i,j}^{n} + \overline{F_{i,j}^{n+1}} + \delta F_{i,j}^{n+1} \right) \tag{4.45}$$

where [A] and [B] are Jacobian matrices

$$A_{j,k} = \frac{\partial G_j}{\partial F_k} \quad , \quad B_{j,k} = \frac{\partial H_j}{\partial F_k}$$

and the overbar indicates a predicted value. MacCormack provided the stability criteria for the method and outlined a general evaluation procedure to follow.

Several other methods are available for use in computation of the transient compressible flow equations. Moretti [82] has developed a scheme based on the MacCormack method which was specifically designed for shock wave handling. Godunov developed an integration method which used parts from the Lax and Von Neumann-Richtmyer methods [83]. Rusanov developed a general purpose explicit integration method where two parameters were selectable for stability. This method was designed to work well for nonsymmetric grids $\Delta x \neq \Delta y$ [71]. Lapidus [84] developed a method such that radial flows could be handled in Cartesian coordinates based on the Lax-Wendroff method. Gary [85] provided a comparison of some of these different integration schemes in a 1964 publication. Other methods are available which have been designed to meet certain problem situations.

These different integration methods have tradeoffs with regard to accuracy or ease of implementation. Computational fluid dynamics is such a broad and intricate topic area that this

study cannot do proper justice to the subject. Desired here was to briefly outline the subject to those unfamiliar with these methods so that use of the associated bibliography could be of use in some simple computational fluid dynamics problems.

Pneumatic pulser modeling was not considered to be a severe test with respect to the assumed shock and boundary conditions. A weak shock wave would be assumed which would propagate and then reflect from an assumed rigid boundary. It was believed that a method involving explicit artificial viscosity would be adequate to evaluate this problem scenario. After examination of the work of Tyler [86], Walker [87], and Leutloff and Roesner [88], the Rusanov method was selected for pneumatic pulse analysis. Rusanov's method has been regarded as general purpose in nature and adaptable to a large number of problems with good results being attainable.

The Rusanov method, which utilizes an explicit artificial viscosity term, was applied to the two-dimensional pneumatic shock flow situation as it exits the pulser tube. The artificial viscosity term was of the form [71]:

$$\frac{\partial}{\partial x}\left[A^x \frac{\partial}{\partial x}\{F\}\right] \; ; \quad \frac{\partial}{\partial y}\left[A^y \frac{\partial}{\partial y}\{F\}\right] \tag{4.46}$$

for the x and y coordinates, respectively. With these explicit dissipation terms, the two-dimensional system of equations becomes:

$$\frac{\partial}{\partial t}\{F\} + \frac{\partial}{\partial x}\{G\} + \frac{\partial}{\partial y}\{H\} = \frac{\partial}{\partial x}\left[A^x \frac{\partial}{\partial x}\{F\}\right] + \frac{\partial}{\partial y}\left[A^y \frac{\partial}{\partial y}\{F\}\right] \tag{4.47}$$

where $\{F\}$, $\{G\}$, and $\{H\}$ are equal to the vectors given in the short form system Equation (4.13). Rearranging, the system may be expressed as:

$$\frac{\partial}{\partial t}\{F\} + \frac{\partial}{\partial x}\left[\{G\} - A^x \frac{\partial}{\partial x}\{F\}\right] + \frac{\partial}{\partial y}\left[\{H\} - A^y \frac{\partial}{\partial y}\{F\}\right] = \{0\} \tag{4.48}$$

Thus for dimensional quantities ρ, u, v, E_S, and p, the units of the vectors $\{G\}$, $\{H\}$,

$$\left\{A^x \frac{\partial}{\partial x}\{F\}\right\}$$

and

$$\left\{A^y \frac{\partial}{\partial y}\{F\}\right\}$$

must be consistent. Use will be made of this later in the analysis.

Forward time differencing and center space differencing was used in the Rusanov scheme. Utilizing these forms for the partial derivatives of Equation (4.48) yields the following finite difference approximations:

$$\left(\frac{\partial}{\partial t}\{F\}\right)_{i,j} \cong \frac{1}{\Delta t}\left[\{F\}_{i,j}^{n+1} - \{F\}_{i,j}^{n}\right] + O(\Delta t^2) \tag{4.49}$$

$$\left(\frac{\partial}{\partial x}\{G\}\right)_{i,j} \cong \frac{1}{2\,\Delta x}\left[\{G\}_{i+1,j}^{n} - \{G\}_{i-1,j}^{n}\right] + O(\Delta x^2) \tag{4.50}$$

$$\left(\frac{\partial}{\partial y}\{H\}\right)_{i,j} \cong \frac{1}{2\,\Delta y}\left[\{H\}_{i,j+1}^{n} - \{H\}_{i,j-1}^{n}\right] + O(\Delta y^2) \tag{4.51}$$

$$\left(\frac{\partial}{\partial x}\left[A^x \frac{\partial}{\partial x}\{F\}\right]\right)_{i,j}^{n} \cong \frac{1}{2\,\Delta x^2}\left[\left(A_{i+1,j}^{x} + A_{i,j}^{x}\right)^n \left(\{F\}_{i+1,j} - \{F\}_{i,j}\right)^n - \left(A_{i,j}^{x} + A_{i-1,j}^{x}\right)^n \left(\{F\}_{i,j} - \{F\}_{i-1,j}\right)^n\right] + O(\Delta x^2) \tag{4.52}$$

$$\left(\frac{\partial}{\partial y}\left[A^y \frac{\partial}{\partial y}\{F\}\right]\right)_{i,j}^{n} \cong \frac{1}{2\,\Delta y^2}\left[\left(A_{i,j+1}^{y} + A_{i,j}^{y}\right)^n \left(\{F\}_{i,j+1} - \{F\}_{i,j}\right)^n - \left(A_{i,j}^{y} + A_{i,j-1}^{y}\right)^n \left(\{F\}_{i,j} - \{F\}_{i,j-1}\right)^n\right] + O(\Delta y^2) \tag{4.53}$$

Putting these components together yields the following if $\Delta x = \Delta y = h$ and $\Delta t = \tau$:

$$\begin{aligned}\{F\}_{i,j}^{n+1} = \{F\}_{i,j}^{n} &- \frac{\tau}{2\,h}\left[\{G\}_{i+1,j} - \{G\}_{i-1,j} + \{H\}_{i,j+1} - \{H\}_{i,j-1}\right]^n \\ &+ \frac{\tau}{2\,h^2}\Big[A_{i+1,j}^{x}\{F\}_{i+1,j} + A_{i-1,j}^{x}\{F\}_{i-1,j} + A_{i,j+1}^{y}\{F\}_{i,j+1} + A_{i,j-1}^{y}\{F\}_{i,j-1} \\ &\quad + A_{i,j}^{x}\left(\{F\}_{i+1,j} + \{F\}_{i-1,j}\right) + A_{i,j}^{y}\left(\{F\}_{i,j+1} + \{F\}_{i,j-1}\right) \\ &\quad - \{F\}_{i,j}\left(A_{i+1,j}^{x} + A_{i-1,j}^{x} + A_{i,j+1}^{y} + A_{i,j-1}^{y} + 2\left(A_{i,j}^{x} + A_{i,j}^{y}\right)\right)\Big]^n\end{aligned} \tag{4.54}$$

Definition of the Courant number, used to assess stability of the numerical procedure, is a dimensionless quantity that relates advection speed to computational information speed. The Courant number should be held to less than unity to insure numerical stability. These requirements are summarized in Equation (4.55) below. Advection speed indicates how quickly a particle may travel through the maximum grid interval, as is illustrated in Figure 4.2.

$$C_{2D} = \frac{\text{Advection Speed}}{\text{Computational Information Speed}} = \frac{(|V| + c)\ \Delta t}{\sqrt{\Delta x^2 + \Delta y^2}} \le 1 \tag{4.55}$$

The Rusanov method uses a slightly different set of parameters and definitions to insure stability with respect to other explicit artificial viscosity methods. The above Courant number definition is utilized in the definition of dimensionless dissipation coefficients α^x and α^y such that:

$$\alpha^x = \frac{\omega}{(\Delta x/\Delta y)} \frac{(|V| + c)\ \Delta t}{\sqrt{\Delta x^2 + \Delta y^2}} = \frac{\omega}{(\Delta x/\Delta y)} C_{2D} \tag{4.56}$$

$$\alpha^y = \omega \left(\frac{\Delta x}{\Delta y}\right) \frac{(|V| + c)\ \Delta t}{\sqrt{\Delta x^2 + \Delta y^2}} = \omega \left(\frac{\Delta x}{\Delta y}\right) C_{2D} \tag{4.57}$$

where ω is an adjustable constant.

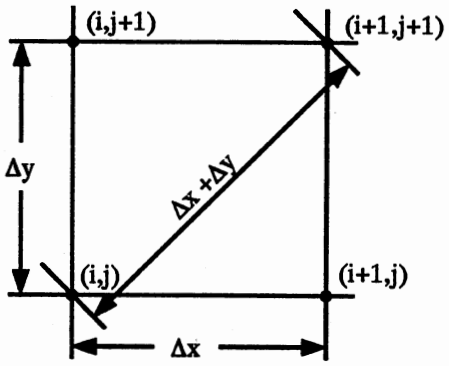


Figure 4.2. Maximum Grid Interval for Advection Speed

Equation (4.54) may be placed in its final form by relating the dimensionless diffusion coefficients α^x and α^y to the dimensional diffusion coefficients A^x and A^y, respectively.

As was indicated earlier, units between the system vectors $\{G\}$, $\{H\}$,

$$\left\{A^x \frac{\partial}{\partial x} \{F\}\right\}$$

and

$$\left\{A^y \frac{\partial}{\partial y}\{\mathbf{F}\}\right\}$$

must be consistent. Examining the first components of these vectors $\{\mathbf{G}\}$ and $\{\mathbf{H}\}$ gives $G_1 = \rho u$ and $H_1 = \rho v$, respectively, that have units lb-sec/in.^{3}. Thus, the units of

$$A^x \frac{\partial}{\partial x}(F_1) = A^x \frac{\partial \rho}{\partial x}$$

and

$$A^y \frac{\partial}{\partial y}(F_1) = A^y \frac{\partial \rho}{\partial y}$$

must give lb-sec/in.^{3}. Units of $\partial\rho/\partial x$ or $\partial\rho/\partial y$ are lb-sec^2/in.^{5} such that:

$$A^x (*) \frac{\partial \rho}{\partial x}\left(\frac{\text{lb-sec}^2}{\text{in}^5}\right) = A^x \frac{\partial \rho}{\partial x}\left(\frac{\text{lb-sec}}{\text{in}^3}\right) \quad \text{such that} \quad (*) = \left(\frac{\text{in}^2}{\text{sec}}\right) \tag{4.58}$$

Examining other components of the $\{\mathbf{G}\}$, $\{\mathbf{H}\}$,

$$\left\{A^x \frac{\partial}{\partial x}\{\mathbf{F}\}\right\}$$

and

$$\left\{A^y \frac{\partial}{\partial y}\{\mathbf{F}\}\right\}$$

vectors yield the same result, units of in.^{2}/sec for A^x and A^y. Making use of this along with the dimensionless diffusion parameter led to the following definitions:

$$A^x = \frac{\Delta x^2}{\Delta t}\alpha^x = \frac{h^2}{\tau}\alpha^x \; ; \qquad A^y = \frac{\Delta y^2}{\Delta t}\alpha y = \frac{h^2}{\tau}\alpha y \tag{4.59}$$

Using these definitions with Equation (4.54) leads to the simplified system relation with $\{\mathbf{F}\}$, $\{\mathbf{G}\}$, and $\{\mathbf{H}\}$ in nondimensional form:

$$\begin{aligned}\{\mathbf{F}\}_{i,j}^{n+1} = \{\mathbf{F}\}_{i,j}^{n} - \frac{\tau}{2h}\left[\{\mathbf{G}\}_{i+1,j} - \{\mathbf{G}\}_{i-1,j} + \{\mathbf{H}\}_{i,j+1} - \{\mathbf{H}\}_{i,j-1}\right]^n \\ + \frac{1}{2}\Big[\alpha^x_{i+1,j}\{\mathbf{F}\}_{i+1,j} + \alpha^x_{i-1,j}\{\mathbf{F}\}_{i-1,j} + \alpha^y_{i,j+1}\{\mathbf{F}\}_{i,j+1} + \alpha^y_{i,j-1}\{\mathbf{F}\}_{i,j-1} \\ + \alpha^x_{i,j}\left(\{\mathbf{F}\}_{i+1,j} + \{\mathbf{F}\}_{i-1,j}\right) + \alpha^y_{i,j}\left(\{\mathbf{F}\}_{i,j+1} + \{\mathbf{F}\}_{i,j-1}\right) \\ - \{\mathbf{F}\}_{i,j}\left(\alpha^x_{i+1,j} + \alpha^x_{i-1,j} + \alpha^y_{i,j+1} + \alpha^y_{i,j-1} + 2\left(\alpha^x_{i,j} + \alpha^y_{i,j}\right)\right)\Big]^n\end{aligned} \tag{4.60}$$

Courant number definition, given in Equation (4.55), is inversely proportional to the maximum speed of particle propagation through the finite difference mesh. This maximum speed is denoted $(|V| + c)_{MAX}$ and is found by searching each grid location for the maximum resultant particle velocity plus the local speed of sound. This maximum velocity is set proportional to the geometrical configuration of the mesh as shown in Figure 4.2 and given in the equation below:

$$(|V| + c)_{MAX} = \left(\sqrt{u^2 + v^2} + c\right)_{max \text{ for all } i,j} = \frac{\sqrt{\Delta x^2 + \Delta y^2}}{\Delta t} \tag{4.61}$$

With this definition, the two-dimensional Courant number becomes:

$$(C_{2D})_{i,j} = \frac{(|V| + c)_{i,j}\,\Delta t}{\sqrt{\Delta x^2 + \Delta y^2}} = \frac{(|V| + c)_{i,j}}{(|V| + c)_{MAX}} \tag{4.62}$$

Since this quantity equals unity at at least one location on the problem grid, a multiplier σ_o was incorporated to insure stability:

$$(C_{2D})_{i,j} = \frac{\sigma_0 (|V| + c)_{i,j}}{(|V| + c)_{MAX}} \tag{4.63}$$

Due to the unique Rusanov α^x and α^y definitions, an additional stability inequality was needed for the parameter ω:

$$C_{2D} \leq \omega \leq \frac{1}{C_{2D}}; \Delta x = \Delta y = h \tag{4.64}$$

Using the above definition for C_{2D} and recognizing that $|(|V| + c)_{ij}|$ will equal $|(|V| + c)_{MAX}|$ at points in the problem grid, the final stability criteria may be written as:

$$\sigma_0^2 \leq \omega\, \sigma_0 \leq 1 \tag{4.65}$$

so that appropriate selection of ω and σ_c will insure numerical stability. Roache [71] indicated that optimal ω and σ_c values were problem-dependent and that experimentation would be required in their determination.

The final system equation incorporates the quantities ω and σ_c described above:

$$\{F\}_{i,j}^{n+1} = \{F\}_{i,j}^{n}\left[1.0 - \frac{\omega\,\sigma_0}{2\,(|V|+c)_{MAX}}((|V|+c)_{i+1,j} + (|V|+c)_{i-1,j} + (|V|+c)_{i,j+1}) \right.$$
$$+ (|V|+c)_{i,j-1} + 4\,(|V|+c)_{i,j}]^n + \frac{\omega\,\sigma_0}{2\,(|V|+c)_{MAX}}[(|V|+c)_{i+1,j}\,\{F\}_{i+1,j} + (|V|+c)_{i-1,j} \quad (4.66)$$
$$\{F\}_{i-1,j} + (|V|+c)_{i,j+1}\,\{F\}_{i,j+1} + (|V|+c)_{i,j-1}\,\{F\}_{i,j-1} + (|V|+c)_{i,j}\,(\{F\}_{i+1,j} + \{F\}_{i-1,j})$$
$$\left.(+\,\{F\}_{i,j+1} + \{F\}_{i,j-1})\right]^n + \frac{\tau}{2\,h}[\{G\}_{i+1,j} - \{G\}_{i-1,j} + \{H\}_{i,j+1} - \{H\}_{i,j-1}]^n$$

Thus, four equations total are to be evaluated for fluid quantities density ρ, u, and v velocity, and stagnation energy, E_S, each following the above form. Evaluation is sequential in nature at each grid location for each time frame.

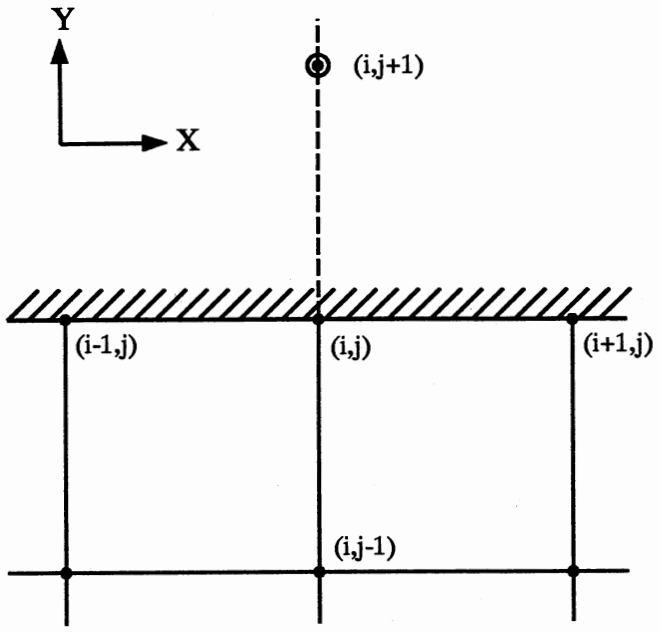
Boundary conditions must be realistically applied for proper performance of the various compressible flow integration schemes. Inviscid assumption allows for the use of slip wall boundaries, where velocity perpendicular to a wall is zero. Reflection principle is then used to reflect the density, energy, and tangential velocity quantities. Walker [87] provided the general reflection principle as applied to an arbitrary boundary. These relations are provided in Equation (4.67):

$$U_n = 0\ ;\ \frac{\partial U_t}{\partial n} = 0\ ;\ \frac{\partial p}{\partial n} = 0\ ;\ \frac{\partial \rho}{\partial n} = 0\ ;\ n = \text{normal}$$
$$\frac{\partial^2 U_t}{\partial n^2} = 0\ ;\ \frac{\partial^2 p}{\partial n^2} = 0\ ;\ \frac{\partial^2 \rho}{\partial n^2} = 0\ ;\ t = \text{tangential} \quad (4.67)$$

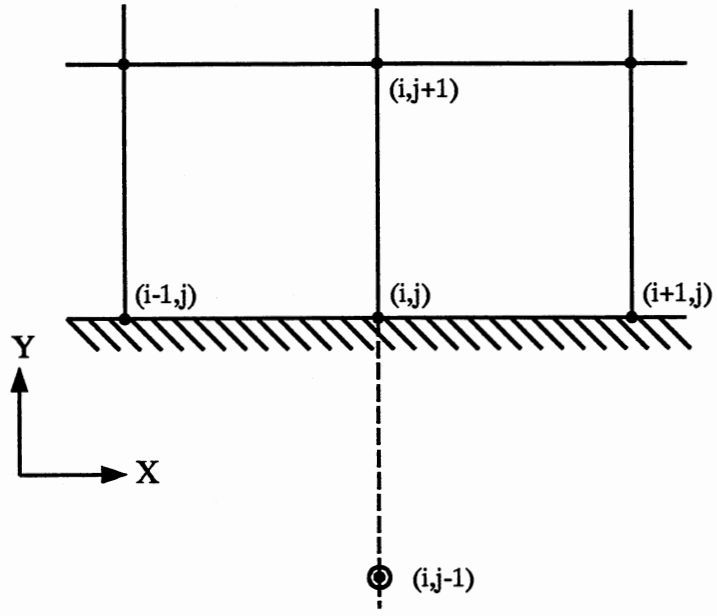
Tyler [86] provided the treatment of the diffusion terms of Equation (4.60) for the various boundary configurations. Figure 4.3a is an illustration of a horizontal boundary by which flow may occur below this boundary and y directional flow at the boundary points is zero. For this case, the amended version of the general system Equation (4.60) becomes:

$$\{F\}_{i,j}^{n+1} = \{F\}_{i,j}^{n} - \frac{\tau}{2\,h}[\{G\}_{i+1,j} - \{G\}_{i-1,j} - 2\,\{H\}_{i,j-1}]^n$$
$$+ \frac{\tau}{2\,h}[A_{i+1,j}^{x}\,\{F\}_{i+1,j} + A_{i-1,j}^{x}\,\{F\}_{i-1,j} + + A_{i,j}^{x}\,(\{F\}_{i+1,j} + \{F\}_{i-1,j}) \quad (4.68)$$
$$- \{F\}_{i,j}\,(A_{i+1,j}^{x} + A_{i-1,j}^{x} + 2\,A_{i,j}^{x})]^n$$

Essentially, the diffusion terms corresponding to differencing with respect to y were zeroed. Figure 4.3b illustrates the similar case of a horizontal boundary with flow occurring above the indicated boundary. For this situation, y directional flow at the boundary is zeroed, resulting in:



(a) Flow Below Boundary



(b) Flow Above Boundary

Figure 4.3. Finite Difference Grid Horizontal Boundary Condition

$$\{F\}_{i,j}^{n+1} = \{F\}_{i,j}^{n} - \frac{\tau}{2h}\left[\{G\}_{i+1,j} - \{G\}_{i-1,j} + 2\{H\}_{i,j+1}\right]^n$$
$$+ \frac{\tau}{2h}\left[A_{i+1,j}^{x}\{F\}_{i+1,j} + A_{i-1,j}^{x}\{F\}_{i-1,j} + + A_{i,j}^{x}\left(\{F\}_{i+1,j} + \{F\}_{i-1,j}\right)\right. \quad (4.69)$$
$$\left. - \{F\}_{i,j}\left(A_{i+1,j}^{x} + A_{i-1,j}^{x} + 2A_{i,j}^{x}\right)\right]^n$$

Figure 4.4a illustrates the case of flow to the left of a vertical wall, such that x velocity is zero at the boundary. For this case Equation (4.60) becomes:

$$\{F\}_{i,j}^{n+1} = \{F\}_{i,j}^{n} - \frac{\tau}{2h}\left[-2\{G\}_{i-1,j} + \{H\}_{i,j+1} - \{H\}_{i,j-1}\right]^n$$
$$+ \frac{\tau}{2h}\left[A_{i,j+1}^{y}\{F\}_{i,j+1} + A_{i,j-1}^{y}\{F\}_{i,j-1} + A_{i,j}^{y}\left(\{F\}_{i,j+1} + \{F\}_{i,j-1}\right)\right. \quad (4.70)$$
$$\left. - \{F\}_{i,j}\left(A_{i,j+1}^{y} + A_{i,j-1}^{y} + 2A_{i,j}^{y}\right)\right]^n$$

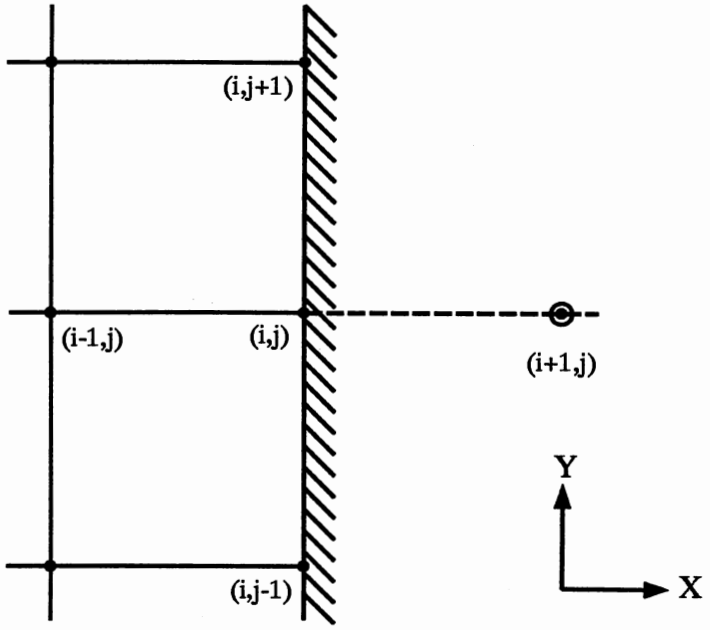
Finally, Figure 4.4b shows the vertical boundary with flow to the right. Again, x directional velocity is considered zero at the boundary. For this case Equation (4.60) becomes:

$$\{F\}_{i,j}^{n+1} = \{F\}_{i,j}^{n} - \frac{\tau}{2h}\left[2\{G\}_{i+1,j} + \{H\}_{i,j+1} - \{H\}_{i,j-1}\right]^n$$
$$+ \frac{\tau}{2h}\left[A_{i,j+1}^{y}\{F\}_{i,j+1} + A_{i,j-1}^{y}\{F\}_{i,j-1} + A_{i,j}^{y}\left(\{F\}_{i,j+1} + \{F\}_{i,j-1}\right)\right. \quad (4.71)$$
$$\left. - \{F\}_{i,j}\left(A_{i,j+1}^{y} + A_{i,j-1}^{y} + 2A_{i,j}^{y}\right)\right]^n$$

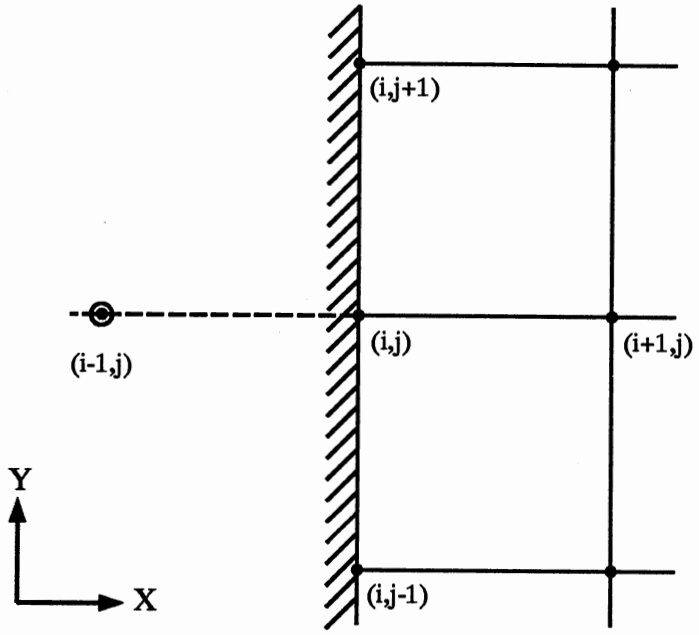
In the last two cases, the diffusion terms corresponding to differencing with respect to x were zeroed.

Roach [71] indicated a modification to the reflection principle when a boundary was not a line of symmetry. This modification was due to the reflection principle adding a redundant boundary condition of zero slope in the case of ρ, e, and the tangential velocity component of Equations (4.7) and (4.8). This modification entailed the zeroing of the $\partial/\partial v$ (ρ,u,v) term of the x momentum Equation (4.7), and the $\partial/\partial x$ (ρ,u,v) term of the y momentum Equation (4.8).

Presented above has been the basic theory to be used in modeling of the experimental tension measurement system pneumatic pulser. This discussion was intended to provide a look at the equations, variables, and analysis techniques that are applicable to this problem. Application of these techniques will be presented in Chapter V. The finite difference problem grid will be presented along with the applicable boundary conditions and initial field conditions for the inviscid fluid density, velocity, and energy quantities.



(a) Flow Left of Boundary



(b) Flow Right of Boundary

Figure 4.4. Finite Difference Grid Vertical Boundary Condition

CHAPTER V

MEMBRANE AND PLATE MODEL DEVELOPMENT

Wave propagation analysis in web materials viewed with respect to an input pressure pulse is quite difficult for a number of reasons. First, assumptions made concerning web structural behavior affects the form of the associated model equations. Next, assumptions made with respect to web boundary conditions affects model complexity and integrity. Finally, assumptions made concerning propagating wave magnitudes affect the realism and applicability of model equations. These concerns will be addressed in a short presentation of the various models examined in this study that may be applied to the web situation.

The general scenario for analytic wave generation and propagation examination in web materials is shown in Figure 5.1. Simple supports may be used to approximate the web rollers and tension is assumed constant across the web width and active in the x direction only. The impulsive web input will be assumed to act at the axis origin, which is the geometrical web center. Data obtained from pneumatic pulse analysis will serve as input stimulus to web models to be developed. Some compromises will be made with respect to web model selection and web configuration of Figure 5.1 for ease of calculation. One such compromise is associated with the free boundaries located at coordinates $-a \leq x \leq a$, $y = \pm b$. One tradeoff deals with linear versus nonlinear model usage. Desired is to sufficiently justify any assumptions such that the subsequent analysis will remain respectable.

Linear membrane and linear plate models will be utilized in this analysis. Clearly, with respect to the experimental system, large web deflections are present which would indicate a need to use large deflection nonlinear membrane and plate models. The reason for this numerical analysis, however, is to gain understanding of the pulse-to-web coupling mechanism rather than to exactly duplicate the web response. The added nonlinear model evaluation complexity was believed to outweigh possible benefits from their use in this study. Thus, basic theory behind

static and dynamic linear membrane and plate models will be presented whereupon results of model use will be presented in Chapter VI.

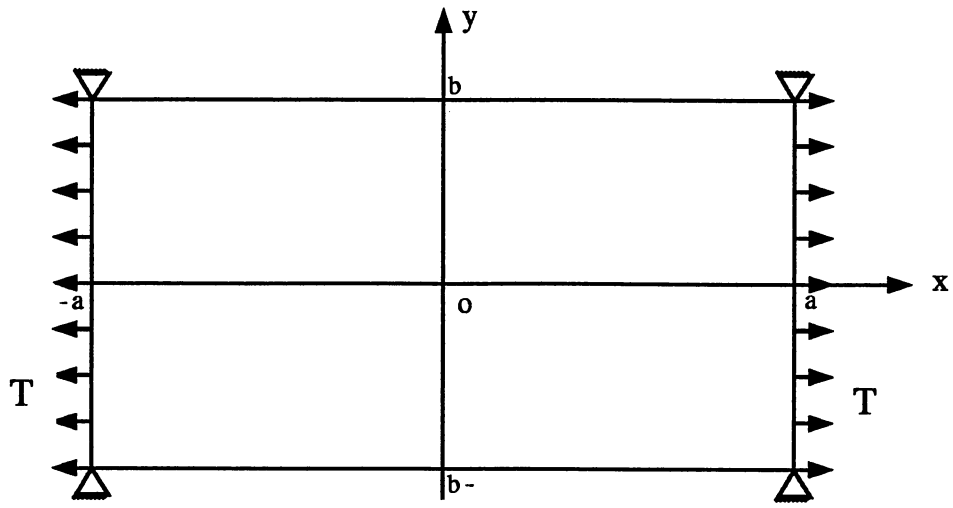


Figure 5.1. General Configuration for Web Analysis

The linear membrane partial differential equation was discussed briefly in section 3.4 with regard to wave propagation speed. Desired in this section is to closely examine the membrane loading to derive the static membrane response equation, then extend this analysis to the dynamic case. Figures 5.2 and 5.3 may be used to aid derivation of the static linear membrane model [89]. Figure 5.2a illustrates a differential membrane element lying in the x-y plane with constant tension T and a transverse loading P_Z (x,y) applied. Figure 5.2b shows a typical transverse membrane displacement w (x,y) as a result of this transverse loading. A differential membrane element is illustrated in Figure 5.3 where Figure 5.3a shows the x directional membrane forces and Figure 5.3b shows the y directional forces. Membrane curvature is given by second partial derivatives of w with respect to the x and y space variables. From Figure 5.3a, summation of forces in the z direction provides:

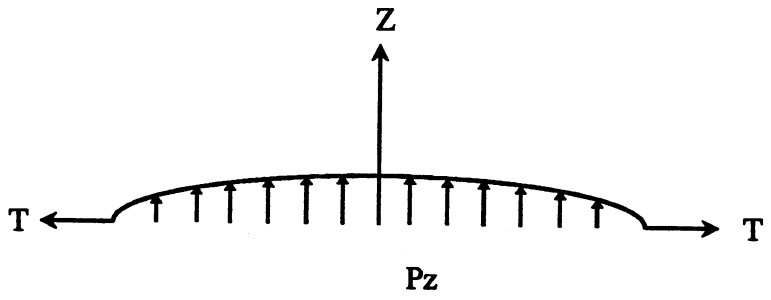
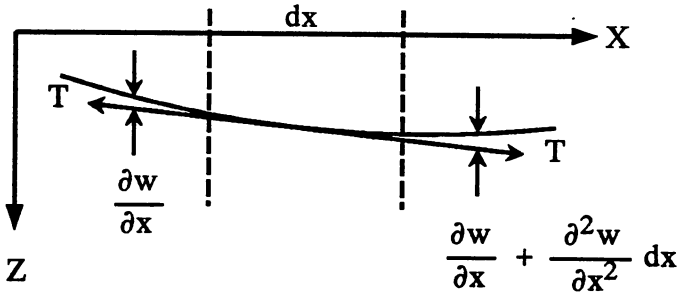
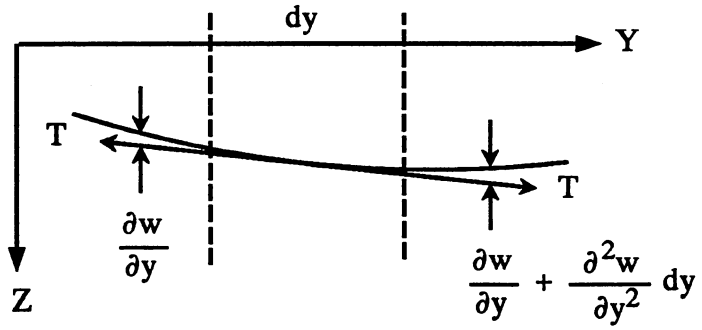


Figure 5.2. Membrane Under Tension and Transverse Loads



(a) X-Directional Membrane Forces



(b) Y-Directional Membrane Forces

Figure 5.3. X- and Y-Directional Membrane Forces

$$\sum F_z = -T\frac{\partial w}{\partial x}dy + T\left(\frac{\partial w}{\partial x} + \frac{\partial^2 w}{\partial x^2}\right)dy + p_z\,dx\,dy = 0 \tag{5.1}$$

From Figure 5.3b, summation of forces in the z direction provides:

$$\sum F_z = -T\frac{\partial w}{\partial y}dx + T\left(\frac{\partial w}{\partial y} + \frac{\partial^2 w}{\partial y^2}\right)dx + p_z\,dx\,dy = 0 \tag{5.2}$$

Adding these equations results in the overall static linear membrane equation for transverse loading:

$$\frac{\partial^2 w(x,y)}{\partial x^2} + \frac{\partial^2 w(x,y)}{\partial y^2} = -\frac{p_z(x,y)}{T} \tag{5.3}$$

Solutions for complex loading functions may be attained through representation of the loading function by Fourier series such that superposition of the membrane response to various frequency components may be performed. Such a Fourier series representation is of the form [90]:

$$p_z(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} P_{mn}\sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right) \tag{5.4}$$

If a sinusoidal response is assumed, then the membrane response to such a pressure input may be described by:

$$w(x,y) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty} W_{mn}\sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right) \tag{5.5}$$

where the amplitude coefficient W_{mn} is given by:

$$W_{mn} = \frac{P_{mn}}{\pi^2\left[\left(m^2/a^2\right) + \left(n^2/b^2\right)\right]T} \tag{5.6}$$

Iterative techniques may be utilized in this static analysis to compute the membrane deflection w(x,y) for a given pressure field $p_z(x,y)$.

To convert the static membrane relation to a dynamic relation, membrane inertia is considered. Figure 5.4 shows a differential membrane element with inertia forces opposing membrane motion. An analysis similar to that given above yields the z directional force summation for this situation:

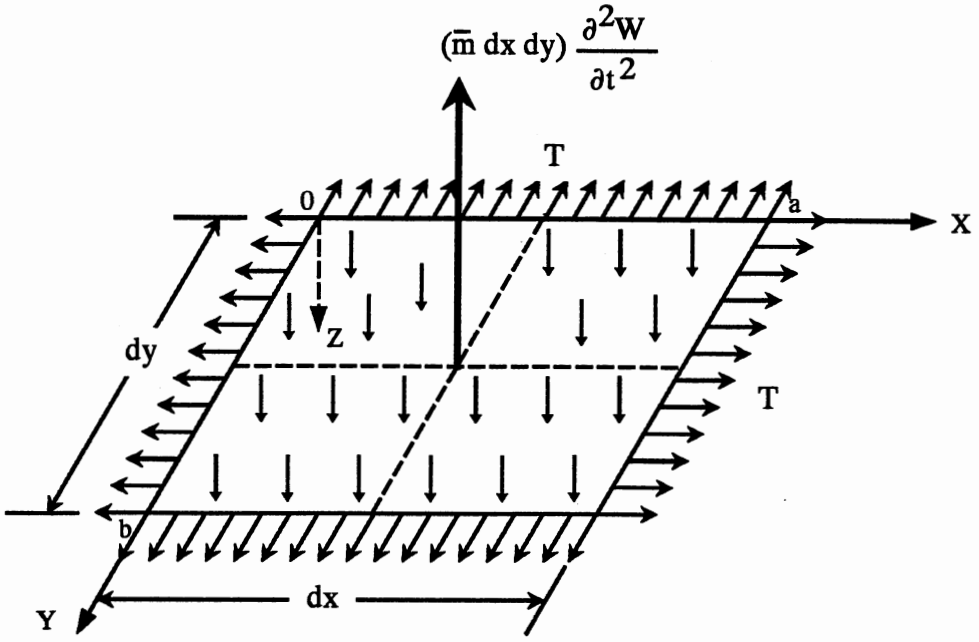


Figure 5.4. Membrane Forces Including Inertial Force

$$\sum F_z = -T\frac{\partial w}{\partial y}dx + T\left(\frac{\partial w}{\partial y} + \frac{\partial^2 w}{\partial y^2}\right)dx - T\frac{\partial w}{\partial x}dy$$
$$+ T\left(\frac{\partial w}{\partial x} + \frac{\partial^2 w}{\partial x^2}\right)dy + p_z\,dx\,dy - \bar{m}\frac{\partial^2 w}{\partial t^2}dx\,dy = 0 \qquad (5.7)$$

where $\bar{m}$ is the membrane mass per unit area. Canceling appropriate terms leads to the dynamic linear membrane equation:

$$\frac{\partial^2 w(x,y,t)}{\partial x^2} + \frac{\partial^2 w(x,y,t)}{\partial y^2} = \frac{\bar{m}}{T}\frac{\partial^2 w(x,y,t)}{\partial t^2} - \frac{p_z(x,y,t)}{T} \qquad (5.8)$$

Separation of variables along with assumed sinusoidal response and superposition techniques are used in formulation of exact solutions to this partial differential equation.

Classical development of plate mechanics is given notably by Timoshenko [35, 91]. Web materials possess some flexural rigidity so that a plate analysis is appropriate with respect to a web in tension. Consider a plate as shown in Figure 5.5, which is equivalent to one-fourth of the plate

area of Figure 5.1. A transverse load p_z (x,y) is applied to some area of the plate perpendicular to the x and y axes. Such a force creates bending moments M_x and M_y and twisting moments M_{xy} as well as shear forces Q_x and Q_y. Both the moment and shear forces are considered to be defined per unit length of the plate. Figure 5.6a and 5.6b shows the differential plate element with moment and shear forces applied, respectively.

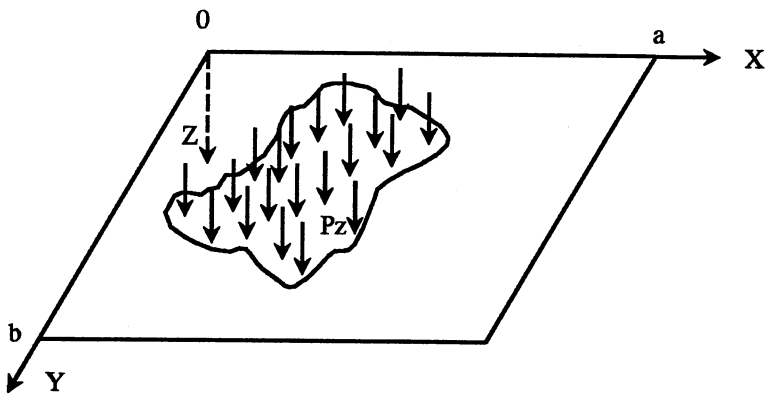


Figure 5.5. Plate Experiencing a Transverse Loading

Considering Figure 5.6, equilibrium conditions may be formulated as follows. Referring to Figure 5.6a, summation of forces in the z direction provides:

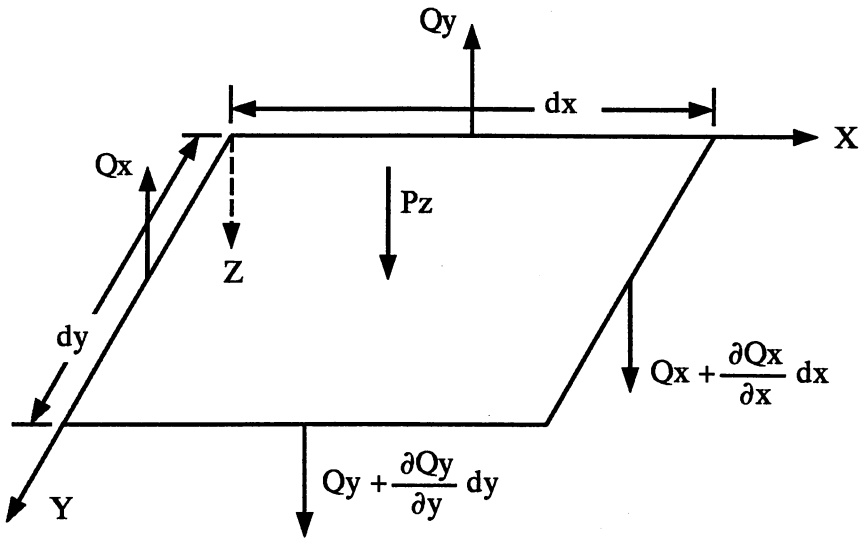
$$\sum F_z = \frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y} + p_z = 0 \tag{5.9}$$

Referring to Figure 5.6a and 5.6b, summation of moments with respect to the x axis provides:

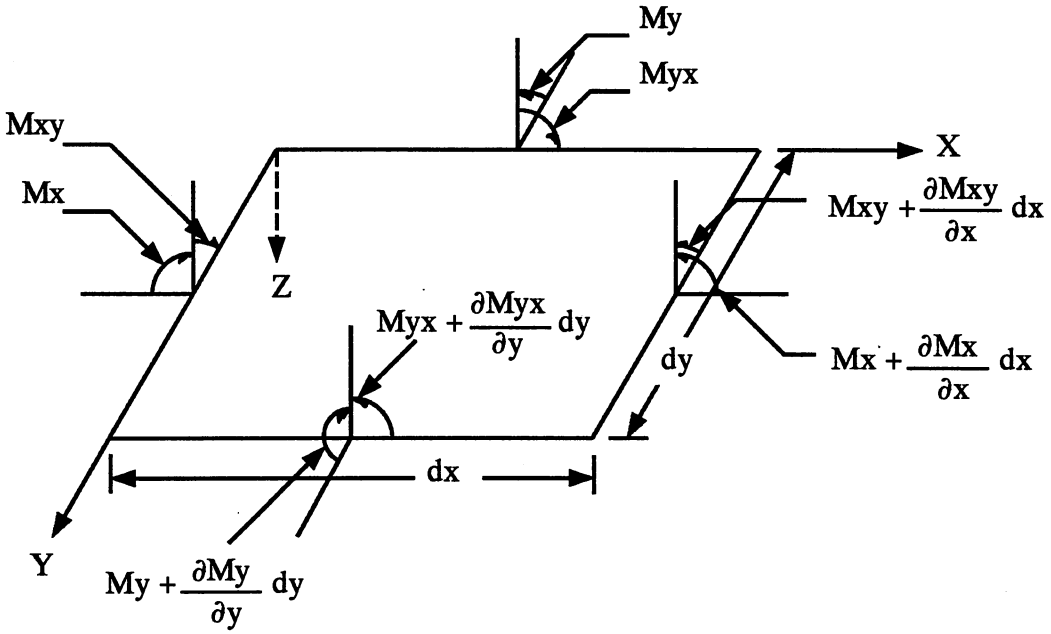
$$\sum M_x = \frac{\partial M_{xy}}{\partial x} - \frac{\partial M_y}{\partial y} + Q_y = 0 \tag{5.10}$$

Similarly, summation of moments with respect to the y axis provides:

$$\sum M_y = \frac{\partial M_x}{\partial x} + \frac{\partial M_{yx}}{\partial y} + Q_x = 0 \tag{5.11}$$



(a) Shearing Forces



(b) Bending Moments

Figure 5.6. Shear and Bending Forces Acting on a Differential Plate Element [35]

If shearing forces are determined from Equations (5.10) and (5.11), then substitution of these quantities into Equation (5.9) yields:

$$\frac{\partial^2 M_x}{\partial x^2} + \frac{\partial^2 M_{yx}}{\partial x\,\partial y} - \frac{\partial^2 M_{xy}}{\partial x\,\partial y} + \frac{\partial^2 M_y}{\partial y^2} = -p_z \tag{5.12}$$

Noting that $M_{xy} = -M_{yx}$, this equation becomes:

$$\frac{\partial^2 M_x}{\partial x^2} - 2\frac{\partial^2 M_{xy}}{\partial x\,\partial y} + \frac{\partial^2 M_y}{\partial y^2} = -p_z \tag{5.13}$$

Moments are expressed in terms of deflection w using the plate flexural rigidity D, Poisson's Ratio v, and the plate curvature relations:

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right) \tag{5.14}$$

$$M_y = -D\left(\frac{\partial^2 w}{\partial y^2} + \nu\frac{\partial^2 w}{\partial x^2}\right) \tag{5.15}$$

$$M_{xy} = -M_{yx} = D(1-\nu)\frac{\partial^2 w}{\partial x\,\partial y} \tag{5.16}$$

Substitution of Equations (5.14), (5.15), and (5.16) into Equation (5.13) leads to the Kirchoff governing equation for small transverse plate static deflections:

$$\frac{\partial^4 w(x,y)}{\partial x^4} + 2\frac{\partial^4 w(x,y)}{\partial x^2\,\partial y^2} + \frac{\partial^4 w(x,y)}{\partial y^4} = \frac{p_z(x,y)}{D} \tag{5.17}$$

For a general plate problem, integration of Equation (5.17), using plate boundary conditions, can yield plate deflections for the given loading p_z (x,y). In many cases, however, plate in-plane loads are also applied to stiffen the plate with respect to transverse loading. This situation, illustrated in Figure 5.7, must be analyzed with respect to tension applied to web materials.

Differential analysis of the Figure 5.7 situation may be illustrated by Figure 5.8, where in-plane forces N_x, N_y, and N_{xy} are acting on the area element in addition to bending and shear forces which were illustrated in Figure 5.6. Equilibrium conditions for the in-plane forces are obtained through force summations with respect to x and y axes, respectively, as referenced to Figure 5.8a [35]:

$$\sum F_x = \frac{\partial N_x}{\partial x} + \frac{\partial N_{yx}}{\partial y} = 0 \tag{5.18}$$

$$\sum F_y = \frac{\partial N_{xy}}{\partial x} + \frac{\partial N_y}{\partial y} = 0 \tag{5.19}$$

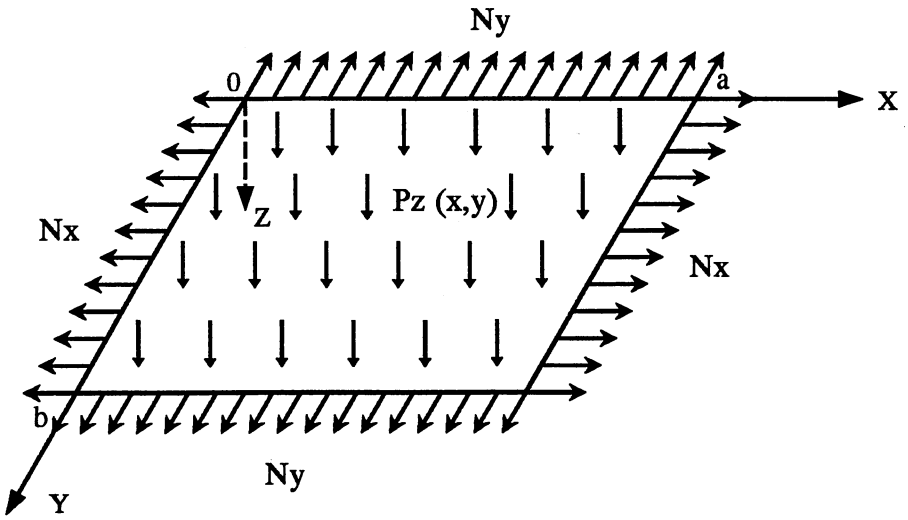
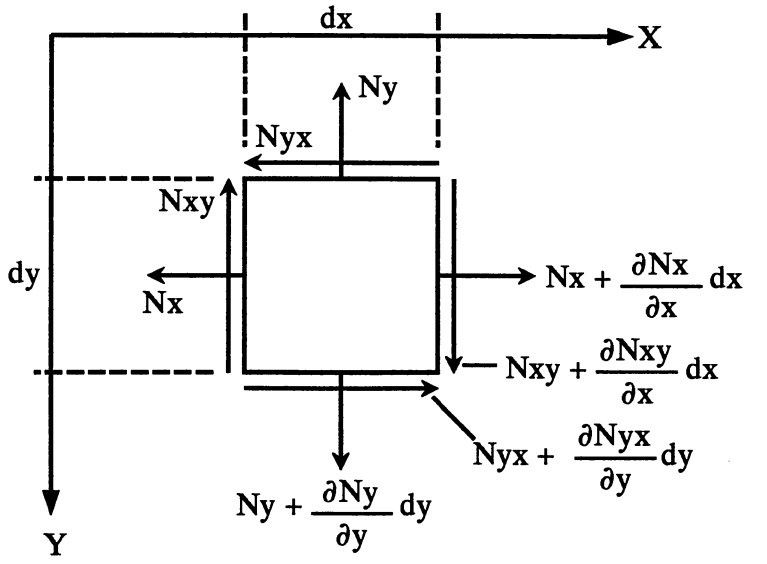


Figure 5.7. Plate Experiencing Transverse Loads and In-Plane Tensile Loads

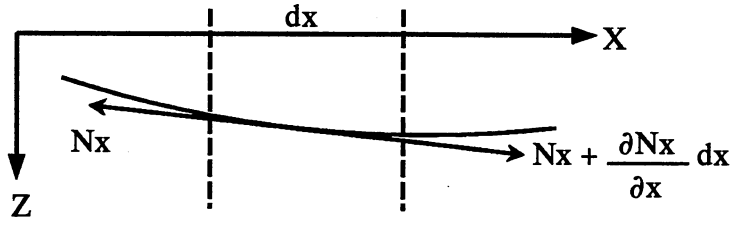
Reference to Figure 5.8b, projection of the in-plane x directional force N_x onto the z axis relates the force to the deflection w. Again, second partial derivatives of w with respect to the x and y space variables represent the plate curvature:

$$\sum F_z = \left(N_x + \frac{\partial N_x}{\partial x}dx\right)dy\left(\frac{\partial w}{\partial x} + \frac{\partial^2 w}{\partial x^2}dx\right) - N_x\frac{\partial w}{\partial x}dy$$

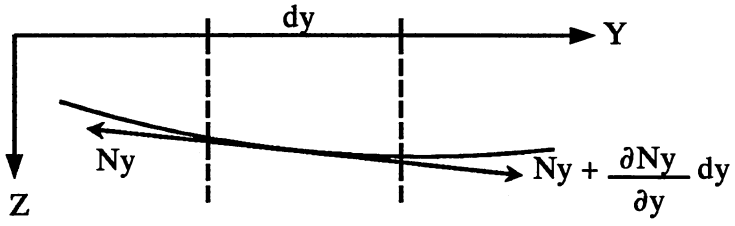
$$\approx N_x\frac{\partial^2 w}{\partial x^2}dx\,dy + \frac{\partial N_x}{\partial x}\frac{\partial w}{\partial x}dx\,dy \tag{5.20}$$



(a)



(b)



(c)

Figure 5.8. Differential Plate Element Experiencing In-Plane Loads [91]

Similarly, using Figure 5.8c, y directional force N_y projection onto the z axis relates the force to the deflection w:

$$\sum F_z = \left(N_y + \frac{\partial N_y}{\partial y}\right) dx \left(\frac{\partial w}{\partial y} + \frac{\partial^2 w}{\partial y^2} dy\right) - N_y \frac{\partial w}{\partial y} dx$$
$$\approx N_y \frac{\partial^2 w}{\partial y^2} dx\, dy + \frac{\partial N_y}{\partial y} \frac{\partial w}{\partial y} dx\, dy \tag{5.21}$$

Shearing forces in the x direction act on displacements with respect to y and vice versa. Projection of the in-plane shearing forces N_{xy} and N_{yx} onto the z axis gives the relation to the deflection w:

$$\sum F_z = \left(N_{yx} + \frac{\partial N_{yx}}{\partial y} dy\right) dx \left(\frac{\partial w}{\partial x} + \frac{\partial^2 w}{\partial x\, \partial y} dx\right) dy - N_{yx} \frac{\partial w}{\partial x} dy$$
$$\approx N_{xy} \frac{\partial^2 w}{\partial x\, \partial y} dx\, dy + \frac{\partial N_{xy}}{\partial x} \frac{\partial w}{\partial y} dx\, dy \tag{5.22}$$

$$\sum F_z = \left(N_{xy} + \frac{\partial N_{xy}}{\partial x} dx\right) dy \left(\frac{\partial w}{\partial x} + \frac{\partial^2 w}{\partial x\, \partial y} dy\right) dx - N_{yx} \frac{\partial w}{\partial x} dx\, dy$$
$$\approx N_{yx} \frac{\partial^2 w}{\partial x\, \partial y} dx\, dy + \frac{\partial N_{yx}}{\partial y} \frac{\partial w}{\partial x} dx\, dy \tag{5.23}$$

The magnitude of N_{xy} equals N_{yx}. The resultant from these projections, obtained through summation of the results of Equations (5.20) through (5.23) with equilibrium conditions (5.18) and (5.19) imposed, result in fictitious lateral loads which are additive with the transverse load p_z (x,y). Thus for a differential element area dA = dx·dy, the moment relation (5.13) becomes:

$$\frac{\partial^2 M_x}{\partial x^2} - 2 \frac{\partial^2 M_{xy}}{\partial x\, \partial y} + \frac{\partial^2 M_y}{\partial y^2} = - \left(p_z + N_x \frac{\partial^2 w}{\partial x^2} + N_y \frac{\partial^2 w}{\partial y^2} + 2\, N_{xy} \frac{\partial^2 w}{\partial x\, \partial y}\right) \tag{5.24}$$

Again, substitution of curvature relations (5.14) through (5.16) results in the final linear plate equation to be used in this analysis:

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2\, \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{1}{D} \left(p_z + N_x \frac{\partial^2 w}{\partial x^2} + N_y \frac{\partial^2 w}{\partial y^2} + 2\, N_{xy} \frac{\partial^2 w}{\partial x\, \partial y}\right) \tag{5.25}$$

Classical solution of the small deflection plate Equations (5.17) and (5.25), known as the Kirchoff plate equations, may be achieved through use of plate boundary conditions and Fourier series representations of the input loading function p_z (x,y). Complex loading functions may be

described by Fourier series as was shown in the membrane analysis. Equation (5.26) is the Fourier series representation of a complex pressure input function, which is equivalent to the membrane case of Equation (5.4). Equation (5.27) provides the corresponding plate deflection. This relation differs from the membrane equivalent, Equation (5.5), due to the effect of flexural rigidity on the plate stiffness:

$$p_z(x,y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} P_{mn} \sin\left(\frac{m \pi x}{a}\right) \sin\left(\frac{n \pi y}{b}\right) \tag{5.26}$$

$$w(x,y) = \frac{1}{\pi^4 D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{P_{mn}}{\left[\left(\frac{m^2}{a^2}\right) + \left(\frac{n^2}{b^2}\right)\right]^2} \sin\left(\frac{m \pi x}{a}\right) \sin\left(\frac{n \pi y}{b}\right) \tag{5.27}$$

Szilard [90] has tabulated the load coefficient relations P_{mn} for several different complex loading functions.

For the situation where in-plane loads N_x, N_y, and N_{xy} are applicable, using Equation (5.26) in Equation (5.25) yields:

$$\nabla^4 w - \frac{1}{D}\left(N_x \frac{\partial^2 w}{\partial x^2} + N_y \frac{\partial^2 w}{\partial y^2} + 2 N_{xy} \frac{\partial^2 w}{\partial x \, \partial y}\right)$$
$$= \frac{1}{D} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} P_{mn} \sin\left(\frac{m \pi x}{a}\right) \sin\left(\frac{n \pi y}{b}\right) \tag{5.28}$$

Using the assumption of simply supported edges, the equation solution may be expressed in the form:

$$w(x,y) = \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} W_{mn} \sin\left(\frac{m \pi x}{a}\right) \sin\left(\frac{n \pi y}{b}\right) \tag{5.29}$$

where W_{mn} are unknown series coefficients. Differentiation of Equation (5.29) and substitution into Equation (5.28) would allow for solution of the W_{mn} coefficients. The in-plane shear component, N_{xy}, complicates the analysis in that cosine series terms become mixed with sine series terms, not allowing for complete cancellation of the series terms. The web model described in Figure 5.1, however, allows for the elimination of the shear terms so that the W_{mn} coefficients may be obtained quite easily.

Due to web configuration symmetry, only half of the plate shown in Figure 5.1 need be analyzed. Selected for analysis will be the upper half of the plate corresponding to $y > 0$. Noted

earlier, simple supports are assumed along boundaries $x = \pm a$, $0 \leq y \leq b$ to simulate the web rollers. Additionally, simple supports will be assumed along the free boundary $y = b$, $-a \leq x \leq a$ to simplify the analysis. A web under tension tends to channel pressure pulse response in the direction of applied tension. Thus, with x dimensional tension applied, web material along a free boundary a lateral distance b away from the pressure pulse is only slightly influenced by said pressure pulse, making an assumption of simple supports at this free boundary tolerable. Figure 5.9 is then the approximate plate model to be used in this analysis. Simply supported boundary conditions are summarized below by Equation (5.30).

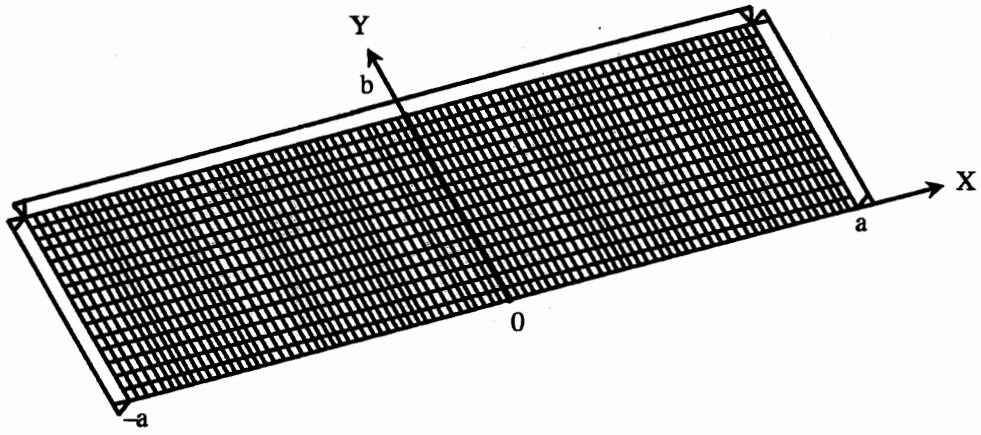


Figure 5.9. Simplified Plate Model for Web Simulation

$$w = 0 \;;\; M_x = \frac{\partial^2 w}{\partial x^2} = 0 \quad \text{along} \quad x = +a, -a$$
$$w = 0 \;;\; M_y = \frac{\partial^2 w}{\partial y^2} = 0 \quad \text{along} \quad y = +b \qquad (5.30)$$

Figure 5.1 indicates web tension parallel to the X axis such that $N_x \neq 0$, $N_y = N_{xy} = 0$. With this assumption, Equation (5.28) becomes:

$$\nabla^4 w - \frac{N_x}{D}\frac{\partial^2 w}{\partial x^2} = \frac{1}{D}\sum_{m=1}^{\infty}\sum_{n=1}^{\infty} P_{mn}\sin\left(\frac{m\pi x}{a}\right)\sin\left(\frac{n\pi y}{b}\right) \tag{5.31}$$

whereupon differentiation of Equation (5.29) and substitution yields the form of W_{mn} [90]:

$$W_{mn} = \frac{P_{mn}}{D\left\{\left[(m\pi/a)^2 + (n\pi/b)^2\right] + \frac{N_x}{D}(m\pi/a)^2\right\}} \tag{5.32}$$

Thus, addition of in-plane tension N_X has served to stiffen the web, reducing the displacement response. This was the expected result.

The above linear plate equations are valid for static pressure input conditions. Thus, iterative methods may be used to calculate the plate deflection w (x,y) for a given pressure field p_z (x,y). As with the membrane, however, a dynamic response is desired to most realistically model the pulser-to-web interaction.

The plate model given by Equation (5.31) is of interest in this analysis due to its inclusion of in-plane, x directional tension. This equation may be converted to a dynamic equation through incorporation of plate inertial effects [90]. Figure 5.10 shows the inertial force along with the previously detailed external forces.

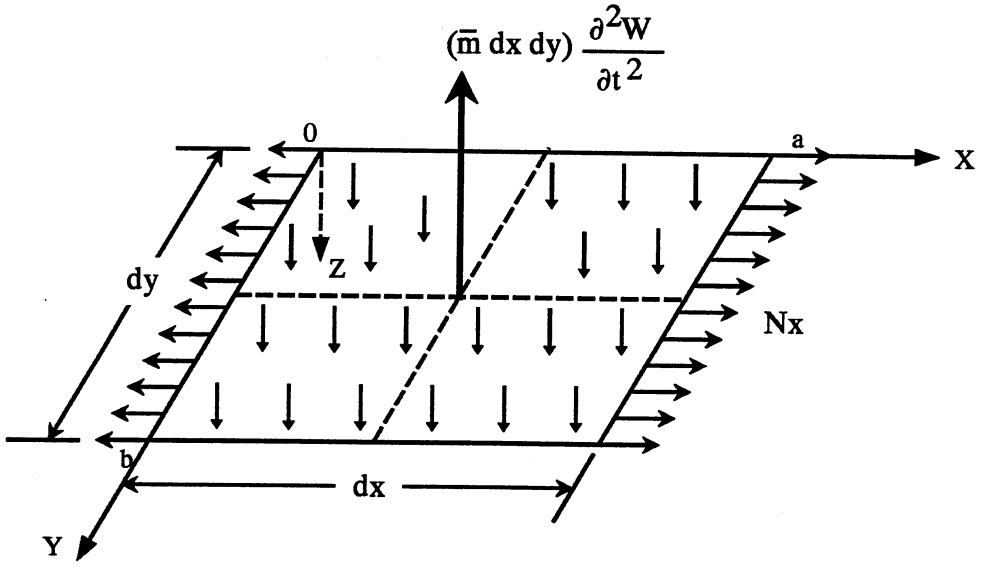


Figure 5.10. Plate Forces Including Inertial Force

The small deflection plate equations were derived through study of shear and moment forces on a differential plate element. If the inertial load, shown opposing plate motion in Figure 5.10, is considered in the analysis, then this force would be additive to the equilibrium equations used in the derivations. This inertial force is $\overline{m}\left(\partial^2 w/\partial t^2\right) dx\, dy$ where $\overline{m}$ is the plate mass per unit area. Thus, Equations (5.9) through (5.11) would incorporate such a term. Equation (5.17) for the case of no in-plane loads is then:

$$\frac{\partial^4 w(x,y,t)}{\partial x^4} + 2\frac{\partial^4 w(x,y,t)}{\partial x^2 \partial y^2} + \frac{\partial^4 w(x,y,t)}{\partial y^4} = \frac{p_z(x,y,t)}{D} - \frac{\overline{m}}{D}\frac{\partial^2 w(x,y,t)}{\partial t^2} \tag{5.31}$$

Equation (5.25) for the case of transverse pressure loading plus in-plane tensile loading is then:

$$\frac{\partial^4 w(x,y,t)}{\partial x^4} + 2\frac{\partial^4 w(x,y,t)}{\partial x^2 \partial y^2} + \frac{\partial^4 w(x,y,t)}{\partial y^4} = -\frac{\overline{m}}{D}\frac{\partial^2 w(x,y,t)}{\partial t^2}$$
$$+ \frac{p_z(x,y,t)}{D} + \frac{N_x}{D}\frac{\partial^2 w(x,y,t)}{\partial x^2} + \frac{N_y}{D}\frac{\partial^2 w(x,y,t)}{\partial y^2} + 2\frac{N_{xy}}{D}\frac{\partial^2 w(x,y,t)}{\partial x\, \partial y} \tag{5.32}$$

Exact solutions to the above equations again are usually found through use of harmonic excitation and response assumptions and separation of variables techniques.

CHAPTER VI

MODELING RESULTS

Pneumatic pulse modeling and membrane/plate modeling developed in Chapters IV and V, respectively, will be applied at this time. Detailed in the respective developments were the system governing equations along with any assumptions that were used in the derivations. Of interest is exploration of the pulse-to-web coupling mechanism. This coupling phenomenon determines the performance of the experimental system in that poor coupling results in a poor signal to analyze which increases uncertainty and chance of error in the indicated tension result. As was indicated in section 3.6, soft spots in a web tend to attenuate the web response whereas tight spots aid the coupling process resulting in larger amplitude signals. The modeling results to be presented in this chapter will provide some insights regarding the characteristic shape of the experimental waveforms and the pulser-to-web coupling process.

Chapter IV detailed the development of finite difference equations to be used in the pneumatic shock modeling procedure. Provided also were equations to be used with rigid boundaries which utilized the reflection principal. The finite grid for this modeling problem will be presented at this time along with applicable boundary conditions. This will be followed by a discussion of equation evaluation over the problem grid, fluid property initialization, and modeling results.

Figure 6.1 is the finite grid used to simulate the pulser tube near a web surface. The pulser tube is represented by the vertical boundaries on Region 6 whereas the web surface is simulated by the horizontal upper boundary along Regions 1, 2, 3, 7, and 8. As shown, one pulser tube diameter spacing is used between the pulser tube exit and upper web boundary. This is typical of spacing used with the experimental system. Remaining boundaries between regions and along outside grid borders are used to couple fluid properties between the various regions. These

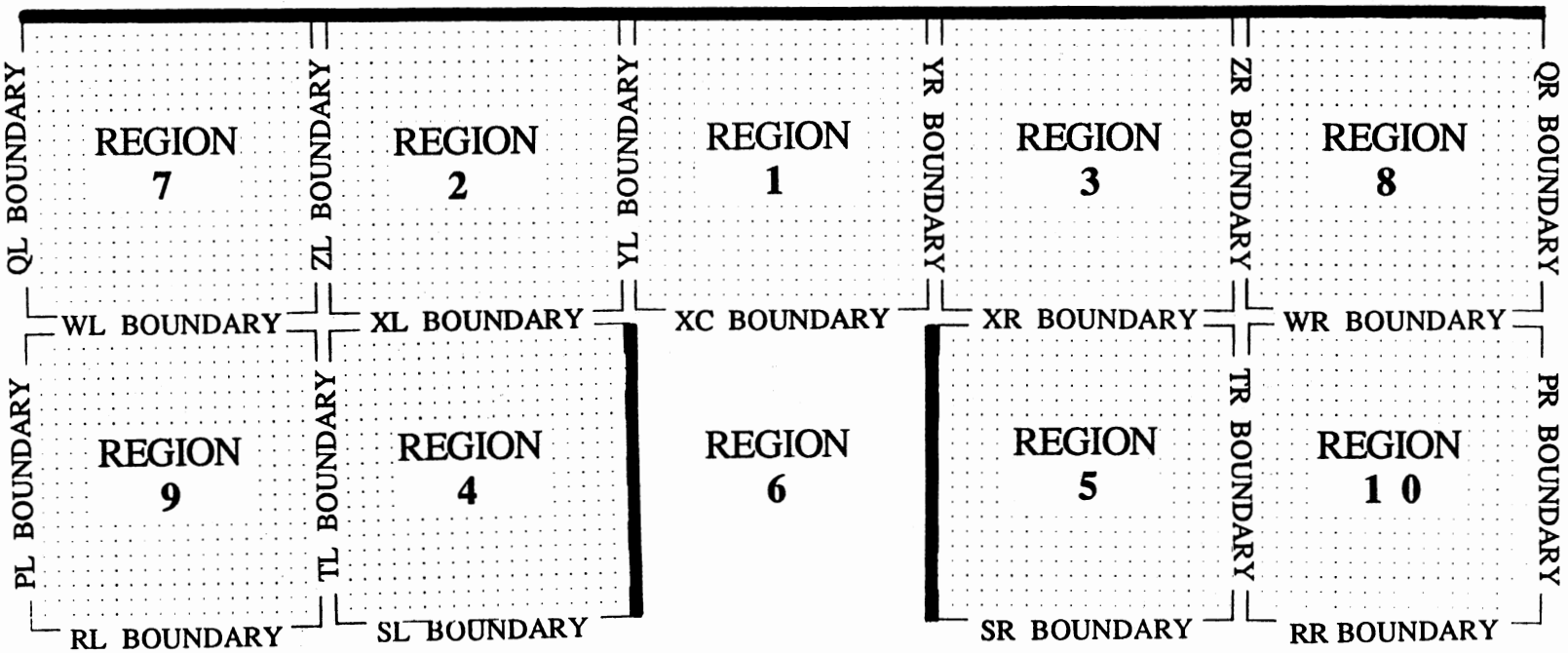


Figure 6.1. Grid used in Finite Difference Pneumatic Pulse Modeling

boundaries are appropriately updated between time frames as the evaluation process marches through time.

The Rusanov method used for inviscid compressible flow equation integration required a fixed grid for the finite difference representation. Thus, a fixed grid implies rigid rather than flexible boundaries. A web, of course, is not a rigid surface but deflects under pressure loading. This flexibility will affect generated interface pressure levels and thus pulse-to-web coupling. For this approximate analysis, pressure levels at this rigid "web boundary" are of prime interest such that these pressure levels may be input to membrane or plate models. Differences in pressure levels and pressure distribution between the actual and the modeled systems are thus inherent. If one considers the web response with respect to the pneumatic pulse duration, however, this approximation does not appear to be dreadful. The pneumatic pulse is quite quick, having a duration on the order of 50 microseconds. Web response to these fast pulses has been much slower, on the order of 250 microseconds or more. Thus, instantaneous web response is not possible, implying that perhaps the web has not deflected a significant amount during the active pulse period. If so, this would help to offset flaws associated with using a stationary web boundary in the pulse modeling procedure.

Finite difference Equation (4.66) is used to evaluate fluid density, u and v directional fluid velocities, and fluid stagnation energy as was developed in Chapter IV. Figure 6.2 is an expanded view of the problem grid Region 1 with boundary values and notation given. Equation (4.66) is contained in a subroutine within the computer program used in this process, which is provided in Appendix G. For each region, corner locations (x,y) = (1,1), (1,N), (N,1), and (N,N) are evaluated first while incorporating any applicable boundary values. For example, evaluation of density, u and v velocity, and stagnation energy at point (1,1), is accomplished through use of said fluid properties at Region 1 points (1,1), (2,1), and (1,2), boundary location YL(1), and boundary location XC(1). Next, remainder of the region edge locations are evaluated using the governing equation in loops. For the case of rigid boundaries along region edges, the governing equation is appropriately modified, resulting in boundary Equations (4.68) through (4.71). For example, referring to Figure 6-2 for Region 1, left-hand side and right-hand side edges of the region are evaluated in standard fashion using $x = 1, 2 \leq y \leq N-1$ and then $x = N, 2 \leq$

$y \leq N-1$, respectively. Similarly, bottom edge values are evaluated using $y = 1, 2 \leq x \leq N-1$. The top edge lies along the rigid web boundary. This condition is illustrated in Figure 4.3a and presented in equation form by Equation (4.68). An indicator is used to indicate the applicable boundary condition such that appropriate terms of the governing Equation (4.60) or (4.66) are zeroed, resulting in the appropriately modified equation of the form of Equations (4.68) through (4.71). After Equation (4.66) has been applied to all edge locations of a region, interior locations are then evaluated. Applicable coupling boundary conditions are updated before proceeding to the next region where the above procedure is repeated.

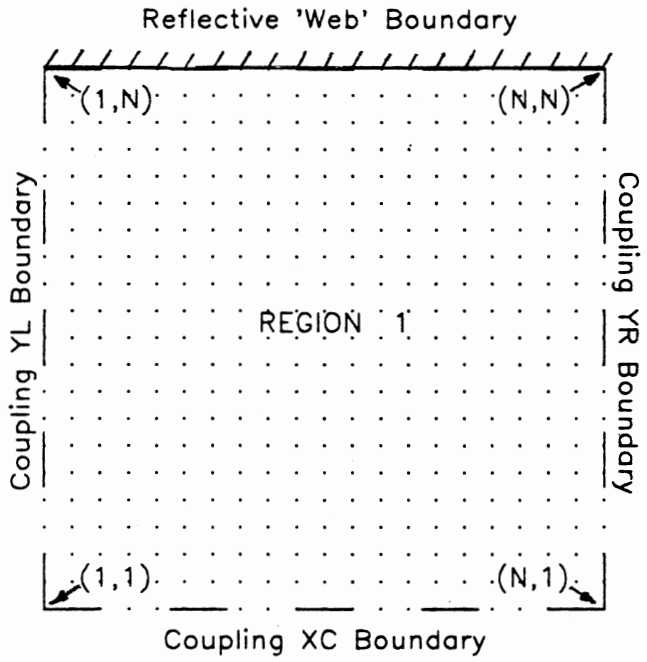


Figure 6.2. Expanded View of Region 1
From Figure 6.1

Stability of the Rusanov scheme, detailed in Chapter IV, required knowledge of the advection speed of the fluid. Courant number was then defined as the ratio of this advection speed to the computational information speed (Equation (4.55)). Maximum fluid velocity is used

in this definition to calculate the time step for the next time frame. Thus, after evaluation of fluid density, u and v velocity, and stagnation energy for each region, the region is searched for maximum velocity by the equation below:

$$(|V| + c)_{\text{all i,j}}^{\text{Region k}} = (\sqrt{u^2 + v^2} + c)_{\text{all i,j}}^{\text{Region k}} \tag{6.1}$$

After evaluation of all ten regions, an absolute maximum velocity is determined:

$$(|V| + c)_{MAX} = MAX\left[(|V| + c)_{\text{all i,j}}^{\text{Region k}}\right] \; ; \; k = 1, 2, ..., 10 \tag{6.2}$$

This value is then used to determine the elapsed time and time step for the next time frame:

$$\tau^n = \left[\frac{\sigma_o \sqrt{\Delta x^2 + \Delta y^2}}{(|V| + c)_{MAX}}\right]^n \; ; \; \tau = \text{Nondimensional Time Step} \tag{6.3}$$

$$t = \sum_n \tau^n \; ; \; t' = \frac{L' \sqrt{\gamma}}{\sqrt{2}\, N\, c_o'} \sum_n \frac{\sigma_o}{(|V| + c)_{MAX}} \qquad \begin{matrix} t = \text{Nondimensional Time Sum} \\ t' = \text{Dimensional Time Sum} \end{matrix} \tag{6.4}$$

where c_0' is dimensional speed of sound, L'/N is dimensional space increment where L' is a characteristic dimension of the problem grid. This time step value in dimensional form is used in the membrane/plate finite difference equation modeling procedure to be outlined later in this chapter.

Nondimensional quantities were used in the Rusanov formulation of the inviscid compressible flow equations. Fluid properties are nondimensionalized with respect to the undisturbed regions of the problem. Thus, from Figure 6.1, undisturbed regions are initially Regions 1, 2, 3, 4, 5, 7, 8, 9, and 10 where fluid properties in undisturbed regions ahead of a shock front will be denoted by an "F" (Field) subscript. Nondimensional variables and initial nondimensional values for field locations are:

$$p = \frac{p_F}{p_F} = 1.0 \; ; \; \rho = \frac{\rho_F}{\rho_F} = 1.0 \; ; \; u = \frac{u_F}{\sqrt{\frac{\gamma\, p_F}{\rho_F}}} = 0.0$$

$$v = \frac{v_F}{\sqrt{\frac{\gamma\, p_F}{\rho_F}}} = 1.0 \; ; \; E_s = \frac{1}{2}(u^2 + v^2) + \frac{p}{\gamma - 1} = \frac{1.0}{0.4} = 2.5 \tag{6.5}$$

Selection of initial conditions within the shock region, Region 6 of the problem grid, was based primarily on pressure levels used with the rotary pulser (Figure 3.1.1). Supply pressure was regulated at approximately 30 psig, which would result in a nondimensional pressure level of 3.0.

Losses occur during pressure pulse propagation along the pulser tube length such that use of a nondimensional pressure of 3.0 in Region 6 would be excessive. Nondimensional pressure of 2.0 was believed to be more realistic. Ratio of shock region to field region pressures is an indication of the shock strength being considered.

Fluid properties on either side of a shock gradient may be used to fit initial fluid properties at the shock gradient median. Recall that the Rusanov integration method called for spreading of a shock gradient over two to four grid intervals. Figure 6.3 is a schematic showing the shock gradient relation to problem grid Region 6. Region 6 row j = N and the coupling boundary values XC were defined as initial shock gradient locations. Averaged fluid property values at the XC coupler locations insure sound coupling of the shock region to the field regions. Fluid properties behind the shock gradient are denoted with an "S" (full Shock), fluid properties on the shock gradient are denoted with a "G" (shock Gradient), with fluid properties ahead of the shock being denoted with an "F" subscript. Initial condition $\rho_S/\rho_F = 2.0$ has been established. At this pressure ratio, shock tables [70] may be used to obtain the density ratio $\rho_S/\rho_F = 1.633$ such that $\rho_S = 1.633$. Initial velocity used in the shock region was assumed parallel to the pulser tube such that $u_S = 0.0$, $v_S = 1.0$; that is, flow out of Region 6 is assumed initially perpendicular to the region outlet. This allows for initial condition calculations to be based on a one-dimensional analysis.

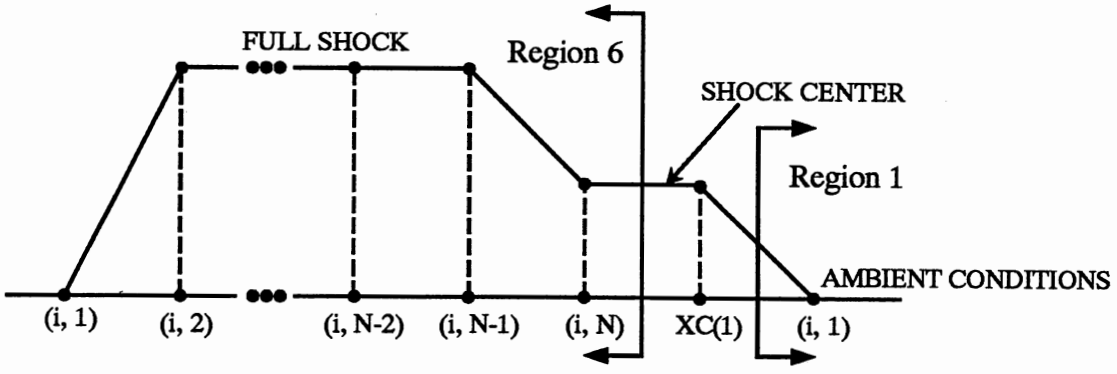


Figure 6.3. Treatment of Shock Interval in Finite Difference Method

The shock fitting method used [88] entailed examination of fluid density, speed of sound, and velocity at the shock gradient with respect to conditions ahead of and behind of the shock front. Pressure at the shock gradient was assumed to be equal to the average pressure $p_G = (p_S + p_F)/2 = 1.5$. With respect to full shock conditions, use of shock tables provided the information shown in Figure 6.4a. From the figure, values of ρ_G^S and v_G^S represent density and velocity of the shock gradient as viewed from the full shock condition. With respect to the undisturbed conditions, similar use of shock tables allowed for information given in Figure 6.4b. Density and velocity of the shock gradient as viewed from the undisturbed region then are ρ_G^F and v_G^F. These values of v_G^S and v_G^F indicate velocity with respect to speed of sound in the fluid rather than velocity of actual fluid particles.

Fluid particle velocities are found through a second set of shock fitting procedures. An assumed particle velocity in the full shock region is used to calculate relative velocities on either side of the shock front. Figure 6-5a provides an examination of the shock gradient referred to the full shock condition, resulting in particle velocity estimate u_G^S. Figure 6.5b is the counterpart with respect to the undisturbed condition, which provides the second particle velocity estimate u_G^F. Finally, average values of the above mentioned quantities are used as initial fluid conditions a given below.

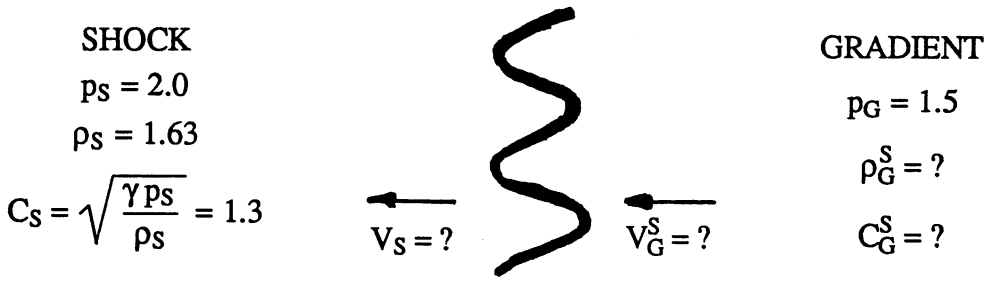
$$\rho_G = (\rho_G^S + \rho_G^F) / 2.0 = 1.33 \tag{6.6}$$

$$u_G = (u_G^S + u_G^F) / 2.0 = 0.558 \tag{6.7}$$

$$p_G = (p_S + p_F) / 2.0 = 1.5 \tag{6.8}$$

With respect to the pulser problem grid, Figure 6.1, velocity value u_G is actually the y directional velocity v where the x directional velocity component will be initially zero.

Computer program pulse f, a listing of which is included in Appendix G, was used to integrate the inviscid compressible flow equations from initial states until the pulse had sufficiently died out. At predetermined intervals during program execution, pressure and velocity contour data files were generated through use of user specified contour increments ΔP and ΔV. Contour bracketing proceeded in both x and y directions, where linear interpolation was used once a pressure/velocity contour was bracketed by two adjacent grid points. Contour data files were



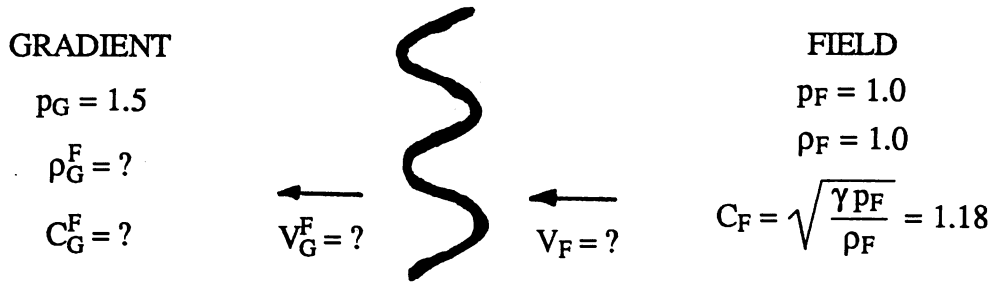
FROM SHOCK TABLES AT $P_S/P_G = 2.0/1.5 = 1.33$:

$M_S = 0.89$, $M_G = 1.13$, $\rho_S/\rho_G = 1.22$

$$\therefore \quad \rho_G^S = \rho_S/1.22 = 1.33 \quad ; \quad V_S = M_S\, C_S = 1.17$$

$$C_G^S = \sqrt{\frac{\gamma P_G}{\rho_G^S}} = \sqrt{\frac{(1.4)(1.5)}{1.33}} = 1.26 \quad ; \quad V_G^S = M_G\, C_G^S = 1.42$$

(a) Referred to Full Shock Condition



FROM SHOCK TABLES AT $P_G/P_F = 1.5/1.0 = 1.5$:

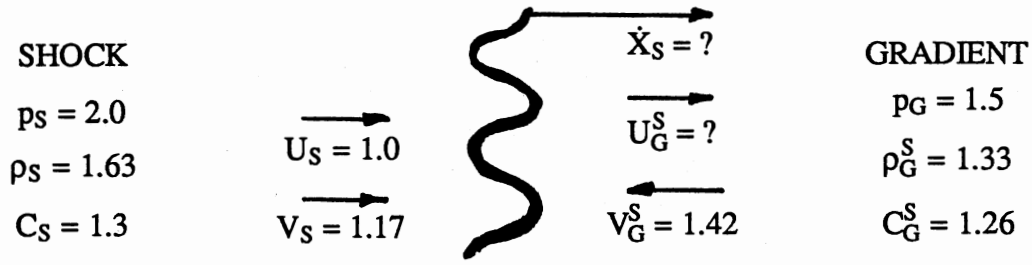
$M_G = 0.84$, $M_F = 1.2$, $\rho_G^F/\rho_F = 1.33$

$$\therefore \quad \rho_G^F = 1.33\, \rho_F = 1.33 \quad ; \quad V_F = M_F\, C_F = 1.42$$

$$C_G^F = \sqrt{\frac{\gamma P_G}{\rho_G^F}} = \sqrt{\frac{(1.4)(1.5)}{1.33}} = 1.26 \quad ; \quad V_G^F = M_G\, C_G^F = 1.06$$

(b) Referred to Undisturbed Condition

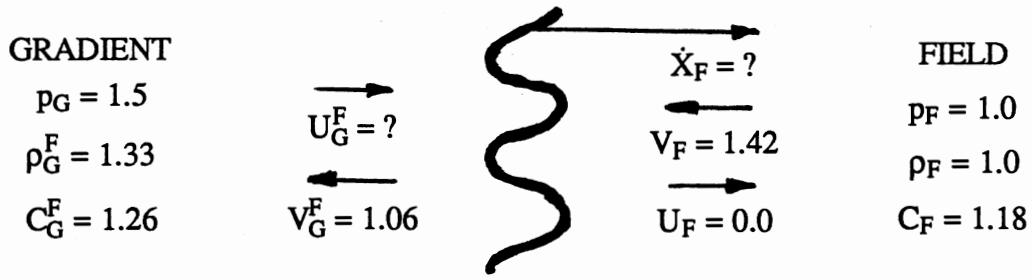
Figure 6.4. Shock Gradient Speed of Sound Fitting



$$\therefore \quad \dot{X}_S = V_S + U_S = 2.17$$

$$U_G^S = \dot{X}_S - V_G^S = 0.75$$

(a) Referred to Full Shock Condition



$$\therefore \quad \dot{X}_F = V_F = 1.42$$

$$U_G^F = \dot{X}_F - V_G^F = 0.36$$

(b) Referred to Undisturbed Condition

Figure 6.5. Shock Gradient Particle Velocity Fitting

generated every twenty time frames during periods of relatively slow pressure variation and every ten time frames during periods of rapidly changing conditions. At these times, the entire problem grid field is scanned to obtain the maximum pressure and velocity values. These values are displayed to the user such that intelligent selection of ΔP and ΔV values may be accomplished. Pressure contours are presented in Appendix A and corresponding velocity contours are presented in Appendix B. An examination of these figures is in order to gain insight into pneumatic shock pulse behavior.

From the initial state, the shock pulse propagated hemispherically with respect to the pulser tube outlet, as shown in time frames 20, 40, and 60, corresponding to Figures A.1, A.2, and A.3 for pressure and B1, B2, and B3 for velocity. Note the contour increments $\Delta P = 0.2$ and $\Delta V = 0.1$. Recall that initial x directional velocity was zero such that the constant response of pressure and velocity along the width of Region 1 would be expected, with small radial propagation at the edges.

Contact with the rigid web boundary begins at 70 time frames, as is shown in Figures A.4 and B.4. The next five contours, for time frames 80, 90, 100, 110, and 120, reveal a very rapid pressure buildup at the web boundary. Pressure Figures A.5 through A.9 dramatically illustrate this effect where the contour increment ΔP was doubled from 0.2 to 0.4 in going from Figure A.5 to A.6. Maximum pressure of 5.684 occurred at time frame 90. In comparison to the initial pressure ratio, 2.0, one can see that the general rule of thumb of weak shock pressure doubling upon contact with a rigid boundary has been exceeded in this analysis. Courant and Friedrichs [67] derived the relation for pressure build up at a rigid boundary as a function of gas constant γ. For the cases of a strong incident shock, p_S/p_F is large. If p_R is denoted as the reflected over-pressure at the boundary, then the following approximation was formulated:

$$\frac{p_R - p_F}{p_S - p_F} \approx 2 + \frac{\gamma + 1}{\gamma - 1} \tag{6.9}$$

Note that if $\gamma = 1.4$ and $p_S - p_F = 1.0$, then $p_R - p_F = 8.0$, which is substantially greater than 2.0. The numerical pressure levels used in this analysis appear to be somewhat stronger than "weak" and thus the maximum overpressure result p_R/p_S was greater than 2.0.

Rapid pressure build-up is of particular interest with respect to the web response hypothesis formulated in section 3.4. To appreciate this pressure build-up, the pressure gradient in moving from 80 to 90 time frames is:

$$\frac{\Delta P}{\Delta t} = \frac{(5.684 - 3.423)\,\text{atm}}{(12.47 - 11.02)\,\text{microsec}} \frac{14.7\ \text{psi}}{1\ \text{atm}} = 22.92\ \text{Mpsi/sec} \qquad (6.10)$$

Figures B.5 through B.9 reveal corresponding velocity contours during this interval. Minimal velocity is evident in Region 1. Velocity vectors begin to point outward as propagation occurs parallel to the web boundary. This behavior is expected and lends credence to the overall results.

Beginning with 140 time frames, gradual dissipation of pressure begins to occur with propagation proceeding laterally outward from the pressure source. Figures A.10 through A.16 for pressure and Figures B.10 through B.16 for velocity provide these graphic results. Contour increments are back to initial values, $\Delta P = 0.2$ and $\Delta V = 0.1$. For comparative purposes, the pressure gradient for a pair of time increments may be examined. Proceeding from 160 to 180 time frames provides a pressure gradient:

$$\frac{\Delta P}{\Delta t} = \frac{(2.347 - 2.438)\,\text{atm}}{(25.97 - 23.08)\,\text{microsec}} \frac{14.7\ \text{psi}}{1\ \text{atm}} = -0.463\ \text{Mpsi/sec} \qquad (6.11)$$

A similar analysis from 220 to 240 time frames provides:

$$\frac{\Delta P}{\Delta t} = \frac{(1.535 - 1.770)\,\text{atm}}{(35.63 - 32.17)\,\text{microsec}} \frac{14.7\ \text{psi}}{1\ \text{atm}} = -0.998\ \text{Mpsi/sec} \qquad (6.12)$$

Thus, pressure dissipation is occurring at a much slower rate than the initial build-up. Velocity contours are quite interesting during this dissipation sequence. One may view evidence of swirl, velocity pockets, and other interesting effects as time progresses. Generally, propagation is outward and downward, as one would expect, Initial ambient pressure returns to Regions 2 and 3 beginning with time frame 200 and later to Region 1 at time frame 260. Region 1 contours tend to be shifted with respect to Region 2 and 3 contours. From this it may be stated that coupling between Region 1 and Regions 2 and 3 was the most difficult to accomplish, likely due to the input discontinuity at the region borders. As can be seen, coupling between other regions was quite successfully accomplished.

Contours presented above were achieved through use of an approximate analysis of the inviscid compressible flow equations coupled with a rigid boundary to represent the web material. The assumption of a rigid interface is obviously flawed but was necessary in order to perform the analysis. Pressure and velocity contours exhibit behavior that one would expect from this type of problem in that pressure at the web boundary increased directly above the simulated pulser tube outlet, velocities were minimal in areas of highest pressure, and directions of propagation were as expected. Time period examined for pulse duration was 260 time frames, corresponding to 39.29 microseconds dimensional time. This is comparable to pulse durations found from experimental work. One may further examine the above results in a qualitative manner to gauge the quality of hypotheses formulated with regard to this pulse-to-web coupling problem.

The objective of this analytic study is to explain why experimental traces acquired through development of the tension measurement system were shaped as they were. Indicated was the general signal behavior of a sharp downward trend followed by a relatively slower rebound. Outlined above through examination of pressure and velocity contours was the same sort of trend in that a sharp pressure surge was followed by a more gradual dissipation. Some interesting qualitative results may be achieved by examining these trends with respect to analytical and experimental results.

Figure 6.6 is a plot of web boundary pressure versus time for three grid boundary locations. Figure 6.6a corresponds to Region 1 point x = 11, y = 21, the central web boundary point of Region 1. Similarly, Figures 6.6b and 6.6c correspond to Regions 2 and Region 7 points x = 11, y = 21, respectively. Each figure contains a small problem grid with this indicated boundary location marked for easy interpretation. These plots clearly reveal the result anticipated from the above observations. A dramatic result of this analysis may be viewed if the result of Figure 6.6a is turned upside down and then compared to an actual trace from the experimental system. This comparison is shown in Figure 6.7. Keep in mind that time scales corresponding to these traces are radically different, due to relatively slow web impulse response. Of interest, however, is the basic shape of the two traces, which for the most part agree. Dynamic effects present in the experimental trace are not applicable to the pressure trace. Qualitatively, Figure 6.7 provides some understanding of the pneumatic pulse-to-web coupling that is desired. Results of membrane

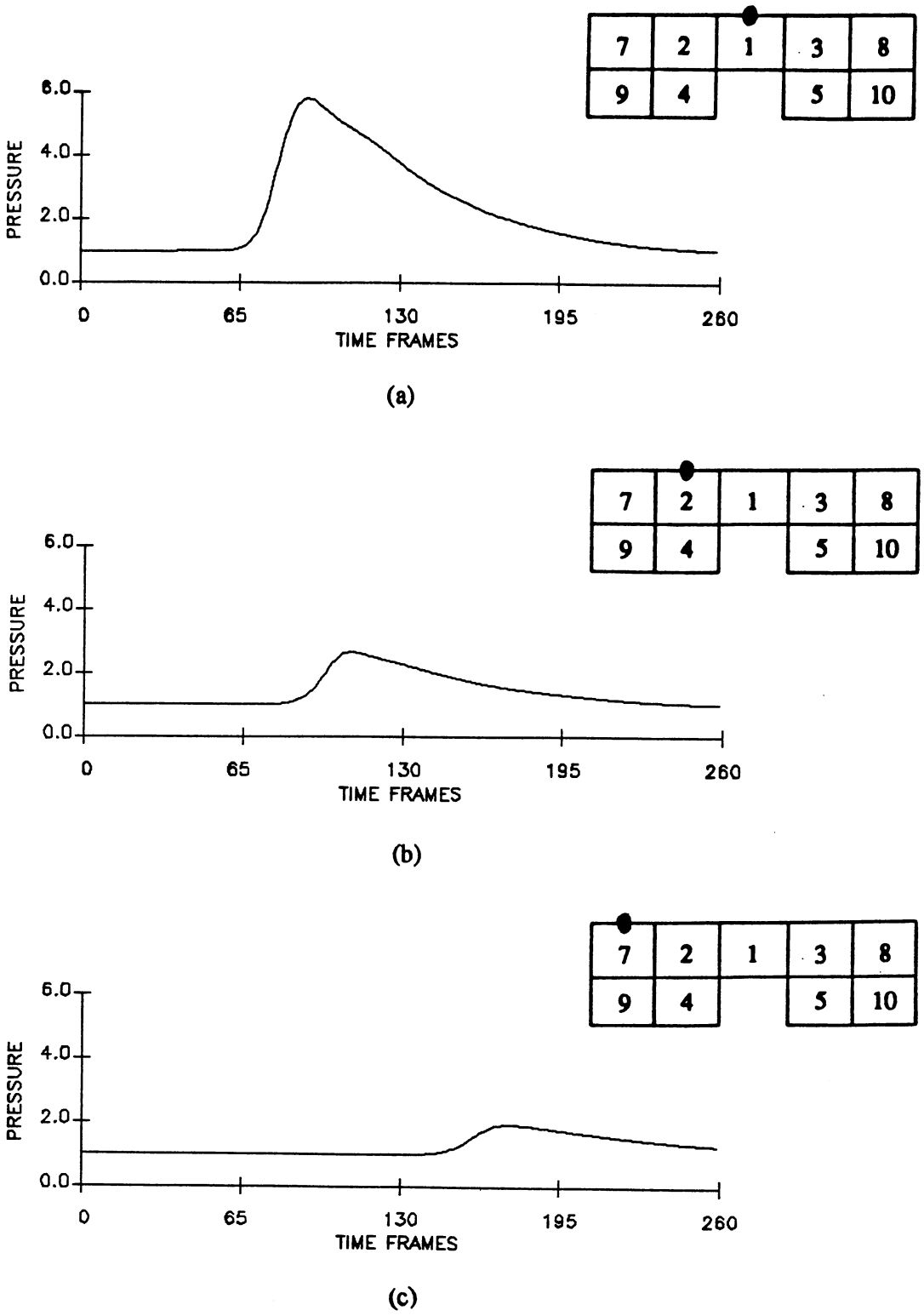
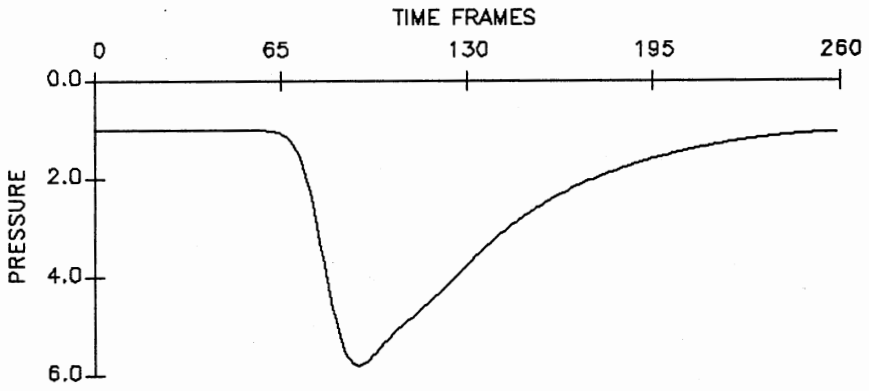
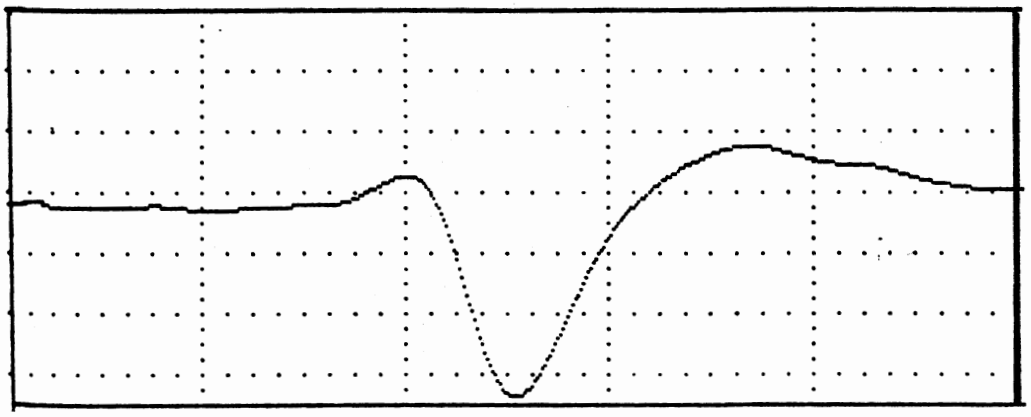


Figure 6.6. Nondimensional Pressure vs. Time for Discrete Locations on the Rigid Web Boundary



(a) From Figure 6.6



(b) From Figure 3.5-3

Figure 6.7. Comparison of Numerical and Experimental Waveform Shape

and plate response to these numerical pressure levels will be presented next to further enhance this understanding.

Chapter V provided an examination of membrane and plate models to be used in this analysis. Indicated was the desire to perform a dynamic analysis of this problem if possible. Linear membrane and linear plate dynamic response equations were presented for use in such a simulation. These equations, in finite difference form, were used in the pulse modeling computer program for simultaneous computation of the membrane/plate response. A presentation of the problem grid, finite difference equation formulation, and pressure level distribution will be provided prior to the simulation results.

Figure 6.8 is a schematic illustrating the relation of the pulse modeling problem grid to the membrane/plate problem grid. The membrane/plate grid represents half of the actual web model in that symmetry is assumed about the grid centerline, consistent with that given in Figure 5.9. Pressure values from the pulse modeling are applied along this membrane/plate centerline with a distribution function assumed to spread the one-dimensional pressure values over the two dimensional membrane/plate grid.

From use and study of pulse propagation with respect to the experimental system, observed has been nearly total propagation in the direction of applied web tension. The pneumatic shock pulse is in effect "channeled" in the direction of web tension such that influence in the cross direction is minimal. A distribution function was used to simulate this channeling effect with respect to the numerically derived web pressure values. A Gaussian function was used for the cross directional distribution, as illustrated in Figure 6.9. This Gaussian function:

$$F(y) = \exp\left[- y^2/2\,\sigma_y^2\right] = \exp\left[- (\text{float}\,(j))^2/2\,\sigma_y^2\right] \qquad (6.13)$$

has a value of unity at y = 0, given by the grid centerline (j = 1), a value of 0.606 at one standard deviation (j = 20), and a value of 0.023 at the grid edge y = b (j = Ny = 55). Using a distribution function such as this is taking liberties with regard to the actual situation due to the problem grid, Figure 6.8, being only 2-1/2 pulser tube diameters deep. Certainly, in the actual system, the pneumatic shock pulse would have a definite effect on a point located a lateral distance of 2-1/2 pulser tube diameters away. A distribution function such as this became necessary due to the

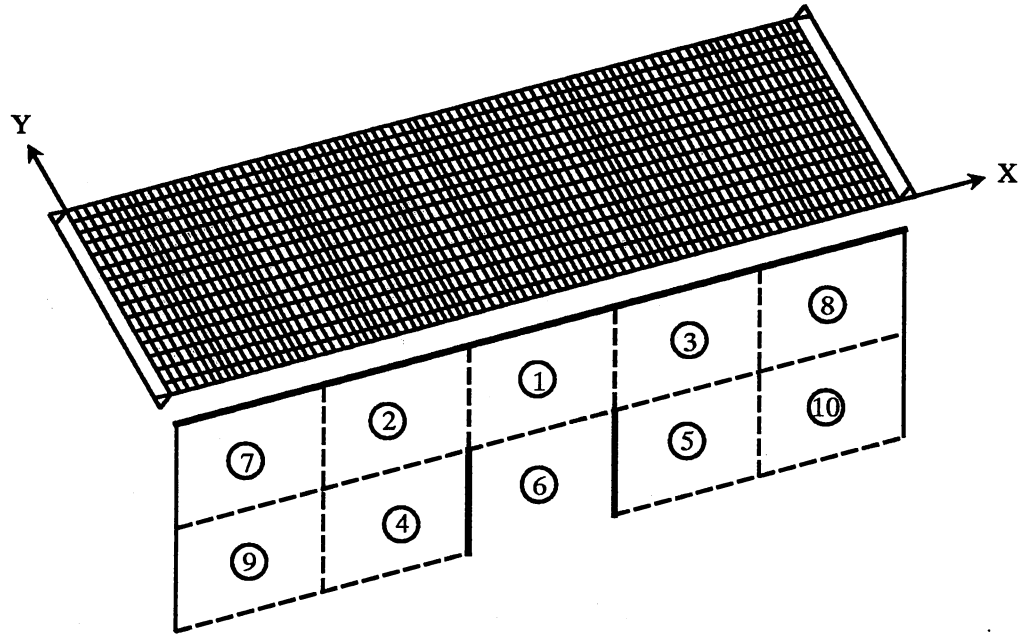


Figure 6.8. Grid Used in Membrane/Plate Finite Difference Pulse Response Modeling

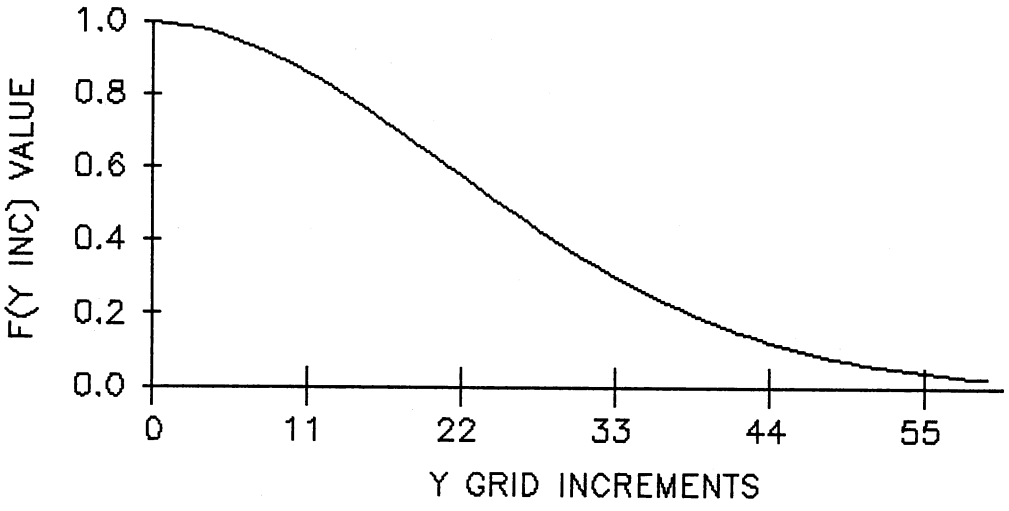


Figure 6.9. Gaussian Pressure Distribution Function

assumed simple support along the web lateral edge. Pressure along this lateral edge is attenuated by the distribution function; thus, less distortion and more quality views of the membrane/plate centerline deflection would be available.

The dynamic linear membrane equation was presented in Chapter V and is repeated in Equation 6-13 below:

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} = \frac{\overline{m}}{T}\frac{\partial^2 w}{\partial t^2} - \frac{p_z}{T} \tag{6.14}$$

As was indicated in Chapter V, T is the membrane tension (pli), $\overline{m}$ is the membrane material area density (lbm/in.^{2}), w is the transverse membrane deflection (in.), and p_z is the transverse pressure loading (psi). Conversion of this equation to finite difference equivalent was performed using second order central difference approximations:

$$\left(\frac{\partial^2 w}{\partial x^2}\right)_{i,j}^{n} \cong \frac{1}{\Delta x^2}\left[w_{i+1,j}^{n} - 2\,w_{i,j}^{n} + w_{i-1,j}^{n}\right] + O(\Delta x^2) \tag{6.15}$$

$$\left(\frac{\partial^2 w}{\partial y^2}\right)_{i,j}^{n} \cong \frac{1}{\Delta y^2}\left[w_{i,j+1}^{n} - 2\,w_{i,j}^{n} + w_{i,j-1}^{n}\right] + O(\Delta y^2) \tag{6.16}$$

$$\left(\frac{\partial^2 w}{\partial t^2}\right)_{i,j}^{n} \cong \frac{1}{\Delta t^2}\left[w_{i,j}^{n+1} - 2\,w_{i,j}^{n} + w_{i,j}^{n-1}\right] + O(\Delta t^2) \tag{6.17}$$

Mentioned earlier, the time step indicated in Equation (6.17) is available from the pulse modeling process where conversion to dimensional form is performed. Substitution of Equations (6.15) through (6.17) into Equation (6.14), with sufficient grouping of terms and rearrangement, provides the dynamic linear membrane finite difference relation used in this analysis ($\Delta x = \Delta y = h$):

$$w_{i,j}^{n+1} = -\,w_{i,j}^{n-1} + 2\,w_{i,j}^{n} + \left(\frac{T\,g_c}{\rho_m}\right)\left(\frac{\Delta t^2}{\Delta x^2}\right)\left[w_{i+1,j}^{n} + w_{i-1,j}^{n} + w_{i,j+1}^{n} + w_{i,j-1}^{n} - 4\,w_{i,j}^{n}\right] + \left(\frac{g_c\,\Delta t^2}{\rho_m}\right)\left[14.7\,(p_z(i) - 1.0) - \rho_m\right] \quad ;\ 2 \le i \le N_x - 1,\ j > 1 \tag{6.18}$$

The 14.7 multiplier converts nondimensional pressure levels to units of psi. Evaluation of this equation with respect to the problem grid of Figure 6.8 was straightforward. A modified version of Equation (6.18) was required at the web centerline (j = 1) to incorporate symmetry conditions. This modified expression is provided below:

$$w_{i,1}^{n+1} = - w_{i,1}^{n-1} + 2\, w_{i,1}^{n} + \left(\frac{T\, g_c}{\rho_m}\right)\left(\frac{\Delta t^2}{\Delta x^2}\right)\left[w_{i+1,1}^{n} + w_{i-1,1}^{n}\right.$$

$$\left. + 2\, w_{i,2}^{n} - 4\, w_{i,1}^{n}\right] + \left(\frac{g_c\, \Delta t^2}{\rho_m}\right)\left[14.7\,(p_z(i) - 1.0) - \rho_m\right] \quad ;\ 2 \le i \le N_x - 1,\ j = 1 \qquad (6.19)$$

The dynamic linear plate equation was developed in Chapter V, resulting in Equation (5.25) which was modified for x directional in-plane loads only and repeated as Equation (6.20) below.

$$\frac{\partial^4 w}{\partial x^4} + 2\,\frac{\partial^4 w}{\partial x^2\, \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{p_z}{D} - \frac{\overline{m}}{D}\frac{\partial^2 w}{\partial t^2} + \frac{N_x}{D}\frac{\partial^2 w}{\partial x^2} \qquad (6.20)$$

Quantities of interest are flexural rigidity D (lb-in.), x directional tension N_x (pli), with quantities $\overline{m}$, w, and p_z remaining as was given above in the membrane discussion. One may take a couple of different paths in the evaluation of this equation. When simple supports are used, the above biharmonic expression may be split into two second order partial differential equations. One partial differential equation is in terms of moments and the other is in terms of transverse displacements. This procedure was noted in the literature survey by References [34], [35], and [91]. This moment sum formulation was utilized in some preliminary work on the static linear plate equation. In-plane forces, given by the N_x term of Equation (6.20), served to couple the moments to the displacements such that the moment sum formulation no longer provided independent partial differential equations. Longer iteration times were necessary for sufficient solution convergence due to this displacement/ moment coupling.

Direct finite difference formulation of Equation (6.20) was performed in this analysis. For equal grid spacing, $\Delta x = \Delta y = h$, the biharmonic operator in finite difference form is:

$$\left(\nabla^4 w\right)_{i,j}^{n} \cong \left(\frac{1}{\Delta^4}\right)\left[20\, w_{i,j}^{n} - 8\left(w_{i+1,j}^{n} + w_{i-1,j}^{n} + w_{i,j+1}^{n} + w_{i,j-1}^{n}\right)\right.$$

$$+ 2\left(w_{i+1,j+1}^{n} + w_{i+1,j-1}^{n} + w_{i-1,j+1}^{n} + w_{i-1,j-1}^{n}\right)$$

$$\left. + w_{i+2,j}^{n} + w_{i-2,j}^{n} + w_{i,j+2}^{n} + w_{i,j-2}^{n}\right] + O\left(\Delta^2\right) \qquad (6.21)$$

Remaining partial derivatives of equation (6.20) are provided in Equations (6.15) and (6.17). Substitution of these finite difference approximations into Equation (6.20) yields the linear finite difference plate equation to be used in this study:

$$w_{i,j}^{n+1} = - w_{i,j}^{n-1} - \left(\frac{g_c \Delta t^2}{\rho_p}\right)\left[w_{i,j}^{n}\left(\frac{20\,D}{\Delta^4} + \frac{2\,T}{\Delta^2} - \frac{2\,\rho_p}{g_c \Delta t^2}\right) - \left(w_{i+1,j}^{n} + w_{i-1,j}^{n}\right)\left(\frac{8\,D}{\Delta^4} + \frac{T}{\Delta^2}\right) - \left(w_{i,j+1}^{n} + w_{i,j-1}^{n}\right)\left(\frac{8\,D}{\Delta^4}\right) + \left(w_{i+1,j+1}^{n} + w_{i+1,j-1}^{n} + w_{i-1,j+1}^{n} + w_{i-1,j-1}^{n}\right)\left(\frac{2\,D}{\Delta^4}\right) + \left(w_{i+2,j}^{n} + w_{i-2,j}^{n} + w_{i,j+2}^{n} + w_{i,j-2}^{n}\right)\left(\frac{D}{\Delta^4}\right) - 14.7\left(p_z(i) - 1.0\right) F(j) + \rho_p\right] \tag{6.22}$$

Again, evaluation of this equation with respect to the problem grid, Figure 6.8, was straightforward. Due to the fourth-order nature of the equation, two expressions were required for reflection of deflection conditions at the web centerline. Equations (6.23) and (6.24) below give the necessary reflective modifications for j = 1 and j = 2, respectively.

$$w_{i,1}^{n+1} = - w_{i,1}^{n-1} - \left(\frac{g_c \Delta t^2}{\rho_p}\right)\left[w_{i,1}^{n}\left(\frac{20\,D}{\Delta^4} + \frac{2\,T}{\Delta^2} - \frac{2\,\rho_p}{g_c \Delta t^2}\right) - \left(w_{i+1,1}^{n} + w_{i-1,1}^{n}\right)\left(\frac{8\,D}{\Delta^4} + \frac{T}{\Delta^2}\right) - \left(w_{i,2}^{n}\right)\left(\frac{16\,D}{\Delta^4}\right) + \left(w_{i+1,2}^{n} + w_{i-1,2}^{n}\right)\left(\frac{2\,D}{\Delta^4}\right) + \left(w_{i+2,j}^{n} + w_{i-2,j}^{n} + 2\,w_{i,3}^{n}\right)\left(\frac{D}{\Delta^4}\right) - 14.7\left(p_z(i) - 1.0\right) F(j) + \rho_p\right] \qquad 3 \le i \le N_x = 2;\ j = 1 \tag{6.23}$$

$$w_{i,2}^{n+1} = - w_{i,2}^{n-1} - \left(\frac{g_c \Delta t^2}{\rho_p}\right)\left[w_{i,2}^{n}\left(\frac{20\,D}{\Delta^4} + \frac{2\,T}{\Delta^2} - \frac{2\,\rho_p}{g_c \Delta t^2}\right) - \left(w_{i+1,2}^{n} + w_{i-1,2}^{n}\right)\left(\frac{8\,D}{\Delta^4} + \frac{T}{\Delta^2}\right) - \left(w_{i,3}^{n} + w_{i,1}^{n}\right)\left(\frac{8\,D}{\Delta^4}\right) + \left(w_{i+1,3}^{n} + w_{i+1,1}^{n} + w_{i-1,3}^{n} + w_{i-1,1}^{n}\right)\left(\frac{2\,D}{\Delta^4}\right) + \left(w_{i+2,2}^{n} + w_{i-2,2}^{n} + w_{i,4}^{n} + w_{i,2}^{n}\right)\left(\frac{D}{\Delta^4}\right) - 14.7\left(p_z(i) - 1.0\right) F(j) + \rho_p\right] \qquad 3 \le i \le N_x = 2;\ j = 2 \tag{6.24}$$

Response of plate or membrane models is dependent on various parameters of the problem, that is, structural parameters of the membrane/plate and the orientation and magnitudes of applied in-plane loads. Membrane tension T and plate in-plane x directional load N_x were given an assumed value of 1.0 pli. Area density was based on plastic web material specific density 0.9 for a resultant 3.25 (10^{-2}) lbm/in.^{2} density times the membrane/plate thickness. Test case examined in this study assumed web thickness h = 0.001 inch. Linear plate flexural rigidity is affected by thickness through:

$$D = \frac{E\,h^3}{12\,(1 - \nu^2)} \tag{6.25}$$

In this analysis, E (Young's Modulus) was assumed to be (10^6) psi such that flexural rigidity for 0.001 in. thick web material equaled (10^{-4}) lb-in.

Computer programs combining pulse modeling and membrane/plate features were used to simulate membrane/plate pulse response. Program listings are available in Appendices H and I for membrane and plate response, respectively. Coding associated with acquisition of pressure/ velocity contours was replaced by the membrane/plate finite difference equations given above along with coding to write deflection information to output files. Time step calculated in the pulse portion of the programs was converted to a dimensional quantity and used to step the membrane/plate simulations through time. Stability problems arose due to the time step size, whereupon the time step was divided by ten and applied to the membrane/plate equations. Thus, membrane/plate equations were evaluated ten times per pulse pressure evaluation. This analysis involved a good deal of computation which helped to verify some of the peculiarities associated with the experimental web-pulse response.

Effective duration of the modeled pressure pulse was approximately 300 time frames. Simulation using membrane/plate dynamic models was carried out for 3200 time frames, which corresponds to 700.91 microseconds in dimensional time. Relation between time frames and dimensional time, given by equation (6.4), is illustrated in Figure 6.10. The figure shows essentially two linear regions. The bend in the curve occurs at approximately 300 time frames, which indicates that maximum fluid velocity for time frames greater than 300 is nearly constant at the fluid speed of sound. For time steps prior to 300 time frames, relatively less dimensional time per time frame indicates that fluid particle velocity is nonzero resulting in maximum fluid velocity $(|V| + c)_{MAX}$ greater than the fluid speed of sound.

Referring to the problem grid of Figure 6.8, no mention has been made as to the dimensions of the pulser tube diameter or to area dimensions of the web model. Dimensional quantities selected are consistent with dimensions of the experimental apparatus. First, square grids were selected such that $\Delta x = \Delta y = \Delta$. The pulser tube was assigned a diameter 0.2 inches wide whereupon 20 equal increments, each 0.01 inch wide, would be used with square regions to

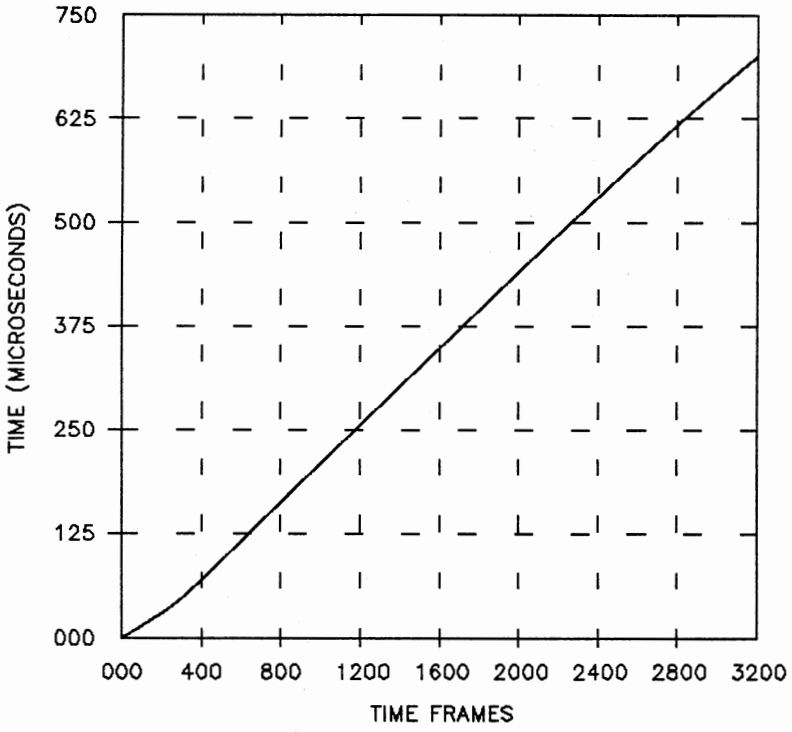


Figure 6.10. Graphical Relation of Dimensional Time and Time Frames

model the pneumatic pulser. From the problem grid, 0.2 inch is also the spacing between pulser tube and web, which is typical of such spacing in the laboratory. Thus, the web model is five regions wide with a grid increment included between regions, resulting in 1.04 inches. Web depth was selected somewhat arbitrarily. In this case, depth was selected as half of the web width.

Graphical display of 0.001 inch thick linear membrane and linear plate pulse modeled responses are provided in Appendices C and D, respectively. A qualitative examination of the graphical series is appropriate as a preface to more specific examination.

Membrane and plate response to the modeled shock front begins in earnest at time frame 300, as shown in Figures C.3 and D.3. The sharp crest on the web structures is quite severe. Effect of the Gaussian distribution function is observable due to reduced displacement away from the center axis of symmetry. Displacement levels here proceeded from 0.004 to roughly 0.2 inch in marching from 100 to 300 time frames. Thickness of the model was 0.001 inch, so clearly, extremely large deflections are present which likely pose problems with respect to the linear nature of the membrane/plate governing equations. Similarities exist between the experimental system and displacement levels indicated by the membrane response during this initial positive displacement sequence. Air loading and damping effects would serve to attenuate actual web deflections somewhat.

Continuing through the graphical displays, displacement amplitude builds as radial propagation effects begin to occur. This phenomenon may be observed by viewing membrane Figures C.4 through C.7 and plate Figures D.4 through D.9. Recall that the pneumatic pulse had an effective duration of 300 time frames. Thus, beginning with approximately 300 time frames, input transverse pressure is effectively zero and the Gaussian distribution function no longer affects the problem. Maximum displacement is evident in Figure D.9 at 0.3762 inch. All of the graphical displacement results presented in Appendices C and D are scaled to this maximum deflection such that relative displacements between membrane and plate may be appreciated.

Downward membrane and plate movement begins at 800 and 1000 time frames, respectively. This progression back to zero displacement is given by Figures C.8 through C.12 and D.10 through D.14. At this point, negative values of displacement begin. Note that the "Maximum

Displacement" values given on each figure is an absolute value rather than a signed value of displacement. Membrane and plate travel down to maximum negative displacements is illustrated in Figures C.13 through C.16 and D.15 through D.22, respectively. Shape of the deflected plate at positive maximum (Figure D.9) and negative maximum (Figure D.22) appear to be nearly mirror images of each other. The same statement may not be made with respect to the corresponding membrane responses, given by Figures C.7 and C.16. In fact, the membrane has traveled 63% further in the downward direction. One would not think that gravitational body forces alone would account for such a difference.

Remainder of the traces, Figures C.17 through C.32 and Figures D.23 through D.32, show the membrane and plate rebounding back to zero and then beginning a second oscillation sequence. Of interest is the time period of one complete oscillation. Figure C.21 shows the linear membrane effectively zeroed after a complete oscillation, with a resulting time period of 462.04 microseconds corresponding to 2100 time frames. An estimate of characteristic frequency from this period is then:

$$f_{membrane} \approx \frac{1}{T^{membrane}_{cycle}} = \frac{1}{462.04\ (10^{-6})\ \text{sec}} = 2164.3\ \text{Hz} \tag{6.26}$$

Similarly for the plate model, rezeroing occurs at approximately 2900 time frames, shown in Figure D.29. Dimensional time at this point is 638.84 microseconds such that characteristic frequency is:

$$f_{plate} \approx \frac{1}{T^{plate}_{cycle}} = \frac{1}{638.84\ (10^{-6})\ \text{sec}} = 1565.3\ \text{Hz} \tag{6.27}$$

These numbers are quite consistent with respect to experimental results. One must be careful, however, to consider the geometry of the membrane/plate grid in any discussions. Characteristic frequency values of Equations (6.26) and (6.27) above may be related to the fundamental frequency of a membrane or plate of a fixed length subjected to an arbitrary transverse load. One conclusion that one may make, however, is that flexural rigidity of the thin (0.001 in.) material is so small that it has little effect on the model. Deflections were smaller and characteristic frequency was higher for the membrane model compared to the plate model due to the membrane having

assumed equal tension T in both x and y directions while the plate had only x directional tension N_x.

An additional examination of the membrane/plate deflection data was performed to gauge the propagation effect at different points along the model centerline. A similar examination was performed with respect to web boundary pressures, culminating in Figure 6.6. The same three grid locations will be used in this analysis which will illustrate deflection with respect to space and time.

Figures 6.11 and 6.12 provide deflection versus time response for plate and membrane respectively. Figures 6.11a and 6.12a correspond to a point centered directly above the pulser tube outlet. Here, the plate exhibits an almost sine-like displacement history. Hypothesized has been a sharp initial deflection followed by a slower rebound period. Plate response here does not follow such a sequence, but the membrane response shows signs consistent with the hypothesis. Troubling here is the larger negative displacement than initial positive displacement. As was suggested earlier, air loading would likely dampen out a portion of this rebound magnitude. Many more harmonics are indicated by the unsymmetric, peculiar membrane response compared to that of the plate.

Figures 6.11b and 6.12b provide displacement versus time histories corresponding to points on the model symmetry line located above the center of pulse grid Region 2. Figures 6.11c and 6.12c provide displacement versus time histories for similar locations above pulser grid Region 7. These figures do not reveal any additional information that is especially enlightening. Obviously, simple supports at the model borders causes the displacement attenuation that is evident with respect to displacement indicated in Figures 6.11a and 6.12a.

This modeling project presented here was a fair sized undertaking. Much study into computational fluid dynamics and structural mechanics was necessary to become familiar with the terminology, usage, and limitations of the various available analysis procedures. Literature included in the literature review with respect to these topics is just a small portion of the material that was searched out and reviewed. From the beginning, graphical plots as have been presented here were the desired analytic output information. It was believed that graphical pictures would be very effective in explaining the nature of the pneumatic pulse-to-web coupling phenomenon.

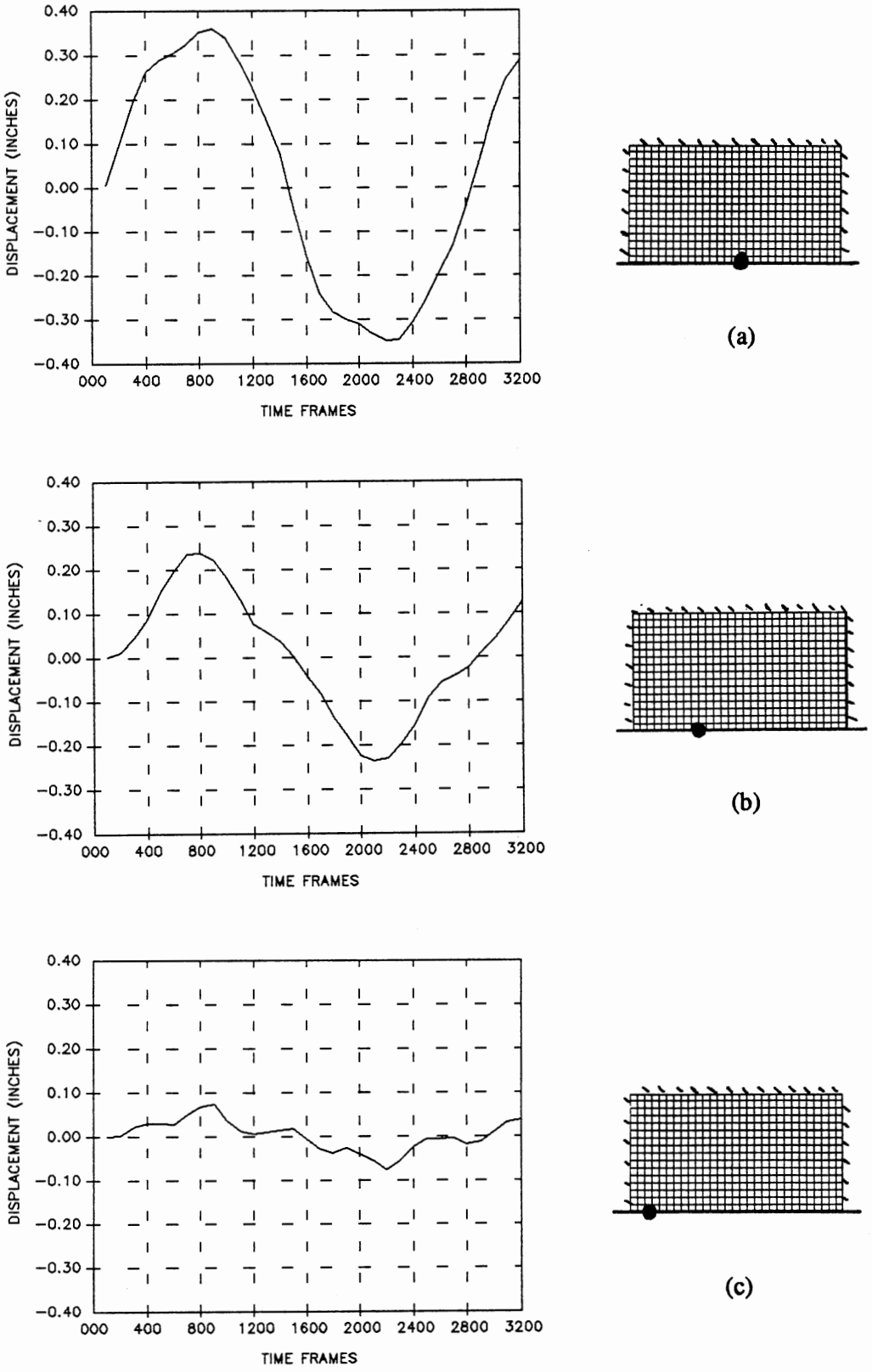


Figure 6.11. Transverse Displacement vs. Time for Discrete Locations on Plate Center Axis of Symmetry

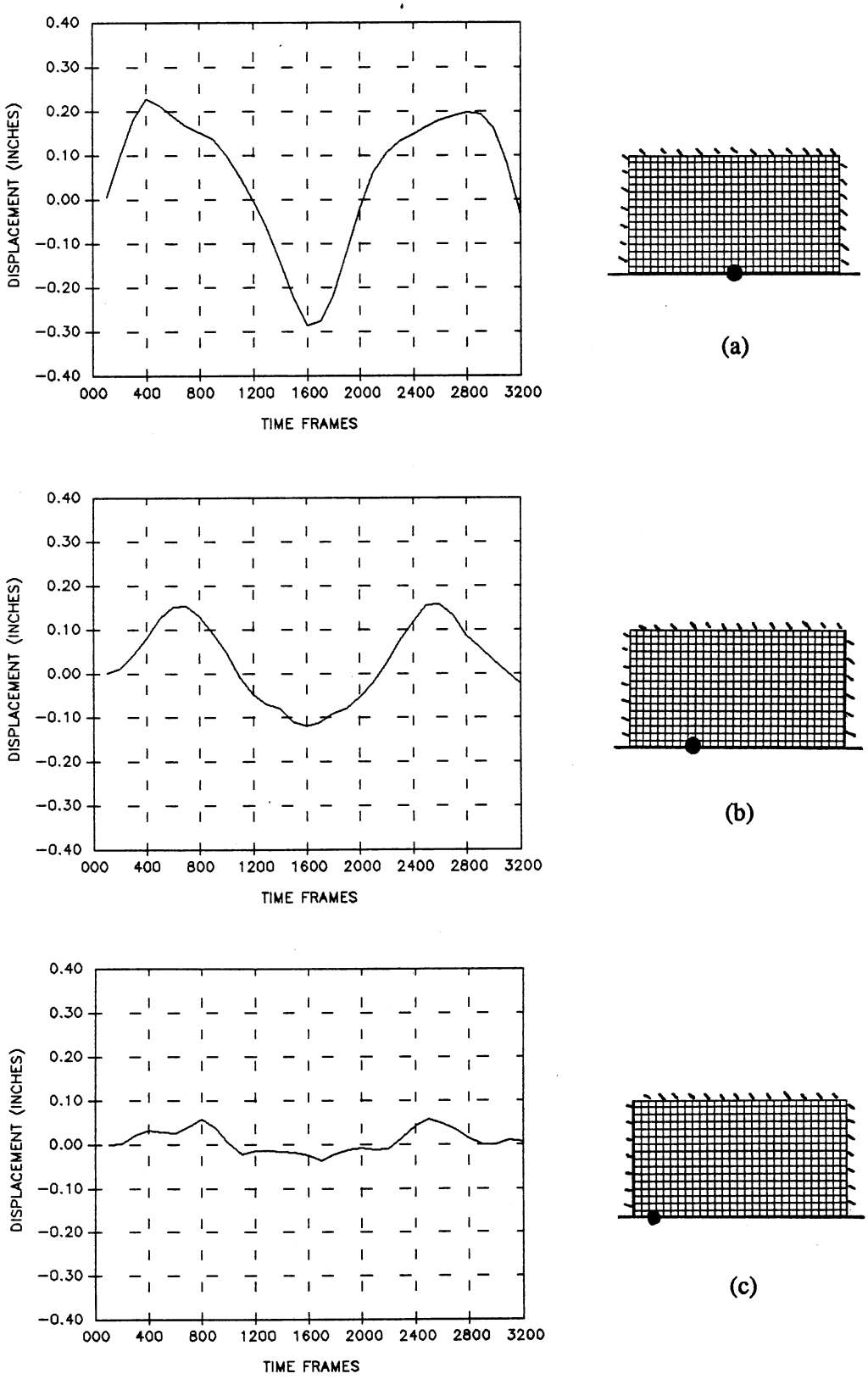


Figure 6.12. Transverse Displacement vs. Time for Discrete Locations on Membrane Center Axis of Symmetry

Pressure and velocity contours achieved through modeling of the pneumatic pulser alone helped illustrate what was actually happening during a pulse cycle. One could examine the contours as the simulated web boundary was encountered. Subsequent pressure and velocity variations were viewed as the impingement, reflection, and radial propagation sequences unfolded. Noted was the fact that experimental and numerical pulse sequences had comparable time periods.

Web boundary pressure was examined to see variation with time as propagation through space occurred. Figure 6.7 was a dramatic result of this investigation, which went far to explain the shape of experimentally derived signals.

Plate and membrane response to the modeled pneumatic pulser was obtained and graphical results displayed. One could examine relative displacement through time to aid in comparisons of the two models. Encouraging was the fact that the plate and membrane behaved differently. Characteristic oscillation frequency for these models was comparable to that obtained in the laboratory. Numerical displacement amplitude results were generally higher than that experienced in the laboratory, but noted also was absence of air loading and air damping effects on the numerical model.

It is believed that some very useful information was derived as a result of the analytic modeling efforts presented in this report. Results were not perfect by any means and areas of possible improvement were noted. The bottom line, however, was better understanding of the pulse-to-web coupling mechanism. Through analytic examination of the pulser and typical web structures, it is believed that much improved understanding has been achieved.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

Research objectives outlined in this report were the development of an inexpensive, compact, multi-purpose, noncontacting and accurate method of local tension measurement. Supported by the Web Handling Research Center, this project has evolved from a few simple ideas to a workable system. Several individuals have contributed to the effort, most notably Darren Nutter, Il Young Ahn, and Marla Bradley under supervision of Dr. Richard L. Lowery, to lay the groundwork for the efforts of this researcher. Many positive statements may be made regarding the system performance to date.

Tension measurement at small, discrete points along a web span was desired by the web handling industry. This would allow for tension distribution measurement across a web span. Tension measurement systems already established were either contacting in nature or were more of an area tension rather than point tension measurement devices. Desired was a point source impulsive input stimulus to the web material to provide a broadband, high frequency excitation to negate air loading effects. Hence the pneumatic pulser was developed which provided a weak pneumatic shock pulse. Noncontacting in nature, this pulse was very sharp and crisp in nature. Strength of the pulse, however, was such that flexural bending waves dominated the web response. These bending waves were of lower characteristic frequency than originally desired. Other adverse effects such as trailing air flow and surrounding fluid loading led to use of the ribbon equation rather than the simpler in-vacuo membrane relation for conversion of experimental data to tension indications.

Detailed has been the transducer head development to provide web damage protection while not adversely affecting induced web flexural waveforms. Concurrent with these activities was the development of data acquisition and signal processing hardware and software. Noted in the text were performance tradeoffs that were encountered. Provided was reasoning used in dealing with

such tradeoffs and any compromises that were formed. Desired for use were the best possible procedures compatible with equipment and computer memory constraints.

Tension indications from the ribbon equation were affected by signal conditioning used in the measurement process. Detailed were the various approaches to raw filtering signals. Low frequency flutter and biases were rejected as was high frequency noise from inherent sources and industrial environmental sources. Analog and digital methods were explored and relative merits and faults were noted. Sensitivity of indicated tension values to filtering was found to be a significant factor in the system accuracy.

System testing has been performed in both laboratory and industrial environments. Laboratory environment was used in development of various hardware, signal conditioning, data acquisition, and data processing components. Industrial tests allowed for real world testing of these components and procedures where an on-line, traveling, fluttering web was present for study. Difficulties to overcome in these field tests included variable air gap spacing due to bowed rollers, eccentricities creating tension fluctuations, and raw signal corruption due to static electricity. The system performed well in this environment if air gap between the transducer head and the web was not overly excessive. Quality information was obtained from weak signals as well as strong signals, indicating that the system would perform well in cases of absolute web noncontact as well as for random web contact. Again, proper filtering was necessary for optimal system performance.

Various web materials were laboratory tested to assess relative system performance. It was found that anisotropic behavior of web materials, that is, change from "tight" spots to "soft" spots, had a great effect on performance of the system. In general, system performance was dictated by the flexural waveform amplitude that could be generated. Flexural waveforms in thicker web materials tended to be of lower characteristic frequency due to increased flexural rigidity of the material. Variable parameter selection in the measurement system signal conditioning facility was of great service in tuning of the system to a particular web material and nominal tension level. Many of the thicker, anisotropic materials would require individual study before concrete statements could be made regarding tension measurement system adaptation to the web material. Shown, however, was good system performance for any web material of thickness

0.003 inch or less over a wide tension range. This adaptability was a quality initially desired at the project beginning.

Out of a sincere desire to attain knowledge regarding the physics of the pneumatic pulse to web interaction process, a multifaceted theoretical and numerical analysis was performed. It was believed that such an analysis, if successful, would greatly expand the knowledge base of the Web Handling Research Center. Upon successful completion, this numerical modeling procedure could provide an estimate of system performance for linear membrane and plate web structures. Computer modeling would allow viewing of system response for varied web density, thickness, and flexural rigidity parameters, and also varied applied tension levels.

Modeling results were presented illustrating behavior of the pneumatic shock pulse upon contact with a rigid boundary that was to simulate the web surface. Pressure and velocity contours revealed a sharp pressure build up followed by a more gradual decay as propagation away from the pressure source occurred. Pressure levels at the rigid boundary were used as input pressure to linear membrane and plate web models. Graphical results of the web model response to this pressure input were presented and compared to the experimentally derived web response. Comparisons of deflection shape, displacement amplitudes, and characteristic period of the numerical results to experimental signal traces were favorable. This computer analysis could be reconfigured to analyze other approaches suggested by Professor Lowery and project associates. Web response modeling for the pulser tube mounted parallel to the web surface, for example, could be achieved in this manner.

The research scope covered in this study is quite large. Included in the problem were aspects of Instrumentation theory, Sampling/Signal Processing theory, Wave Propagation theory, Fluid Dynamics, Plate/Membrane Mechanics, Computer Science, and Numerical Analysis. An effort was made to study literature associated with these project aspects such that sound decisions could be made regarding system development. It is believed that questions that arose from the Bradley experiments have been fully addressed and answered. The noncontacting system developed provided good performance for a wide variety of web materials and tension levels. Transducer and positioning mechanisms arrived at during this research process are compact, simple, and inexpensive. Signal acquisition and processing is compatible to an ordinary personal computer,

also helping to reduce costs. In short, it is felt that requirements suggested by the Web Handling Research Center industrial consortium have been successfully fulfilled. Modeling procedures developed have supported the experimental efforts by providing needed insights into the reasons for the experimental system behavior and performance.

Work at developing the experimental and analytical results that have been presented in this report has led to some understanding for possible improvement areas. Some of the upcoming recommendations would allow for fine tuning and further use of the modeling procedures whereas some recommendations deal exclusively with the experimental system. Many areas of study could be examined to more fully develop tension measurement system performance and provide better understanding of the mechanics involved in the system.

Detailed in section 3.6 was tension measurement system response to arbitrary web materials and tension levels. A benefit in set-up and operation of this system in various industrial settings would be some recommendations as to signal conditioning to be used for general classes of web materials. A good deal of testing would be involved here, but the end result could be recommended starting filter parameters for different web materials based on nominal web tension, web thickness, web flexural rigidity, and web anisotropy. Engineers in charge of these systems would then have some starting point from which to become familiar with the system response characteristics.

A calibration method for laboratory use is needed such that system accuracy may be more properly quantified. Context here refers to some means of precisely identifying a known static laboratory tension profile prior to experimental system use. In this sense, contact with the web sample would be acceptable. In essence this would result in yet another local tension measurement method if new ideas were formulated and developed. This would allow for close evaluation of system performance for any applicable web material/web tension situation.

Measurement of pneumatic pressure input and web displacements would be of value with regard to the modeling procedure. Assumed in the modeling was an initial nondimensional pressure ratio of 2.0 between full shock and field locations. Use of miniature pressure sensors to measure the pulse output pressure would provide perhaps a better initial pressure ratio estimate in

the modeling computer programs. Web displacement measurement to pneumatic pulse inputs has similar use with respect to modeling. One could judge modeling results accuracy for varying web material thickness and stiffness. Optical sensors have been applied to this problem with limited success. Such a measurement scheme is hampered in that web contact in most cases would greatly affect the web response.

The modeling program could be used to examine analytic system response for varying pulser tube-to-web spacing. Spacing of one pulser tube diameter was used in modeling presented in this report. Thus, square regions were used with $\Delta x = \Delta y = h$. The Rusanov method used here allows for $\Delta x \neq \Delta y$ grid spacing. Alternately, grid lines with respect to y could be removed from Regions 1, 2, 3, 7, and 8 to move the pulser tube closer to the web or added to Regions 1, 2, 3, 7, and 8 to move the pulser tube away from the web. One could obtain an optimum spacing criterion from such an examination.

Work on this research project has led to observance of many interesting phenomena. Difficulty arises in determination of a basis for further examination of some of these phenomena. Dispersive effects have been noted in various web materials using optical sensors as waveform pickups. Applicable theory may be applied to this identifiable case. The pneumatic pulse seems to promote an extremely high amplitude nonlinear web response in close vicinity to the pulse center point. Modeling suggested the possibility of such behavior due to the rapid high amplitude pressure build up directly above the simulated pulser tube outlet. Speculated was the presence of compressive web buckling waves as a result of such large amplitude stimulus. An examination of hyperelastic structures and theory could perhaps provide some insights into these areas.

REFERENCES

[1] Ketterer, A. "Web Control—Balancing Guidance and Tension." *Formed Fabrics Industry*, Vol. 8, No. 10 (Oct. 1977), pp. 6-8.

[2] "Equipment for Web Guiding, Web Viewing, and Tension Control." *Converter* (June 1978), pp. 22-29.

[3] "Improved Paper Roll Quality Through Improved Tension Control." *Paper, Film, and Foil Converter*, Vol. 57, No. 9 (Sept. 1983), pp. 122-123.

[4] Deeg, R. "New Web Tension Control System Reduces Oscillation." *PIMA Magazine*, Vol. 61, No. 11 (Nov. 1979), pp. 27-28.

[5] "Three Pole Magnets Control Web Tension." *Machine Design*, Vol. 56, No. 5 (Mar. 8, 1984), p. 62.

[6] Farese, F. "Better Methods for Controlling Tension." *Machine Design*, Vol. 61, No. 8 (Apr. 20, 1989), pp. 128, 130, 133.

[7] Murray, C. J. "Closed Loop Feedback Controls Web Tension." *Design News*, Vol. 44, No. 23 (Dec. 5, 1988), pp. 82-83.

[8] Haggstrom, R. P. "Experiments on Tension Measurement in Manufacturing Processes." *Experimental Mechanics*, Vol. 22, No. 3 (Mar. 1982), pp. 106-110.

[9] Bak, D. J. "Dancer Arm Feedback Regulates Tension Control." *Design News*, Vol. 43, No. 7 (Apr. 6, 1987), pp 132-133.

[10] Hutzenlaub, A. "Static and Dynamic Effects During Winding of Films." *Kunststoffe (German Plastics)*, Vol. 57 (1967), pp. 163-165.

[11] Roisum, D. R. "How to Measure Roll Quality." *Tappi Journal*, Vol. 71, No. 10 (Oct. 1988), pp. 91-103.

[12] Ernst, J. R. "Winder Headaches: From Web Breaks to Web Brakes." *Tappi Proceedings Finishing and Converting Conference*, Albany, NY, 1985. pp. 57-63.

[13] Dandan, I. R. *Mechanics of Paper Failure*. Report No. D86-005. Beloit Corporation, Rockton Research Center, Feb. 4, 1986.

[14] Luukkala, M. , Heikkila, P., and Surakka, J. "Plate Wave Resonance—A Contactless Test Method." *Ultrasonics*, Vol. 9, No. 4 (Oct. 1971), pp. 201-208.

[15] Mann, R. W., Baum, G. A., and Habeger, C. C. "Elastic Wave Propagation in Paper." *Tappi Journal*, Vol. 62, No. 8 (Aug. 1979), pp. 115-118.

[16] Olofsson, K., Molin, N., and Kyosti, A. "A New Method to Detect Anisotropy and Local Variations in Paper." *Tappi Journal*, Vol. 74, No. 3 (Mar. 1991), pp. 195-200.

[17] Baum, G. A., and Habeger, C. C. "On-Line Measurement of Paper Mechanical Properties." *Tappi Journal*, Vol. 63, No. 7 (July 1980), pp. 63-66.

[18] Habeger, C. C., and Baum, G. A. "On-Line Measurement of Paper Mechanical Properties." *Tappi Journal*, Vol. 69, No. 6 (June 1986), pp. 106-111.

[19] Habeger, C. C. "An Ultrasonic Technique for Testing the Orthotropic Symmetry of Polymeric Sheets by Measuring Their Elastic Shear Coupling Coefficients." AMD Symposium Series ASME Applied Mechanics Division. Vol. 99. Mechanics of Cellulosic and Polymeric Materials. *Third Joint ASCE/ASME Mechanics Conference*, San Diego, CA, July 1989, pp. 223-229.

[20] Hansen, A. "A Portable Instrument for Web-Tension Control and Cross-Profile Recording." *Tappi Journal*, Vol. 69, No. 12 (Dec. 1986), pp. 48-51.

[21] Linna, H. and Moilanen, P. "Comparison of Methods for Measuring Web Tension." *Tappi Journal*, Vol. 71, No. 10 (Oct. 1988), pp. 134-138.

[22] Walbaum, H. H., and Lisnyansky, K. "Review of Process Control Instruments for Measuring Paper Quality Variables." *Paper Trade Journal*, Vol. 167, No. 13 (July 15 and 30, 1983), pp. 37-42.

[23] Meinander, S., and Marttinen, T. "Measuring Web Tension Using Contactless Tension Meter." Technical Research Center of Finland. *Graphic Arts Laboratory*, Vol. 11, No. 1 (1983), pp. 34-36.

[24] Marttinen, T., and Luukkala, M. "An Acoustic Noncontacting Instrument to Measure Tension in a Moving Paper Web." *1985 IEEE Ultrasonics Symposium Proceedings*, Vol. 1, Oct. 16-18, 1985, San Fransisco CA, pp. 553-556.

[25] Rye, T. W. "Using Tenscan to Measure the Tension Profile." *1988 Tappi Finishing and Converting Conference Proceedings*, pp. 175-178.

[26] Lamb, H. "On the Propagation of Tremors Over the Surface of an Elastic Solid." *Phylosophical Transactions of the Royal Society, Series A*, Vol. 203 (1904), pp. 1-42.

[27] Lamb, H. "On Waves in an Elastic Solid." *Proceedings of the Royal Society, Series A*, Vol. 93 (1917), pp. 114-128.

[28] Tolstoy, I. *Wave Propagation.* New York: McGraw-Hill Publishing Co., 1973.

[29] Cohen, H., and Berkal, A. B. "Wave Propagation in Elastic Membranes." *Journal of Elasticity*, Vol. 2, No 1 (Mar. 1972), pp. 45-57.

[30] Cohen, H. "Wave Propagation in Elastic Plates." *Journal of Elasticity*, Vol. 6, No. 3 (July 1976), pp. 245-259.

[31] Babaev, A. E., Kubenko, V. D., and Krishtalev, V. V. "Transient Processes in a Cylindrical Shell With Liquid Under the Action of a Shock Load." *Soviet Applied Mechanics*, Vol. 23, No. 1 (Jan. 1987), pp. 25-31.

[32] Kubenko, V. D., and Moseenkov, B. "Action of Weak Shock Waves on a Membrane Separating Two Acoustic Half Spaces." *Soviet Applied Mechanics*, Vol. 18, No. 8 (Feb. 1983), pp. 705-711.

[33] Stippes, M. "Large Deflections of Rectangular Plates." *Proceedings of First U.S. National Congress of Applied Mechanics*, Chicago, IL, June 11-16, 1951, pp. 339-345.

[34] Cadambe, R., and Kaul, R. K. "Deflection of Clamped Plates by Two-Step Membrane Analogue." *Journal of the Royal Aeronautical Society*, Vol. 59 (May 1955), pp. 358-360.

[35] Timoshenko, S., and Woinowsky-Kreiger, S. *Theory of Plates and Shells*. 2nd Ed. New York: McGraw-Hill, 1959.

[36] Shaw, F. S., and Perrone, N. "A Numerical Solution for the Nonlinear Deflection of Membranes." *Transactions of the ASME Journal of Applied Mechanics*, Vol. 21, No. 2 (1954), pp. 117-128.

[37] Kao, R., and Perrone, N., "Large Deflections of Flat Arbitrary Membranes." *Computers and Structures*, Vol. 2 (1972), pp. 535-546.

[38] Al-Khaiat, H., and West, H. H. "Analysis of Plates by the Initial-Value Method." *Computers and Structures*, Vol. 24, No. 3 (1986), pp. 475-483.

[39] Al-Khaiat, H. "Initial-Value Analysis of Continuous Orthotropic Plates." *Computer Methods in Applied Mechanics and Engineering*, Vol. 69 (1988), pp. 153-165.

[40] Jones, R. "A Simplified Approach to the Large Deflection of Membranes." *International Journal of Nonlinear Mechanics*, Vol. 9 (1974), pp. 141-145.

[41] Berger, H. M. "A New Approach to the Analysis of Large Deflections of Plates." *Transactions of the ASME Journal of Applied Mechanics*, Vol. 22 (1955), p. 465.

[42] Nerantzaki, M. S., and Katsikadelis, J. T. "A Green's Function Method for Large Deflection Analysis of Plates." *Acta Mechanica*, Vol. 75, No. 1-4 (1988), pp. 211-225.

[43] Stern, M. "A General Boundary Integral Formulation For the Numerical Solution of Plate Bending Problems." *International Journal of Solids and Structures*, Vol. 15 (1979), pp. 769-782.

[44] Burgess, G., and Mahajerin, E. "A Numerical Method for Laterally Loaded Thin Plates." *Computer Methods in Applied Mechanics and Engineering*, Vol. 49 (1985), pp. 1-15.

[45] Bauer, H. F. "Nonlinear Response of Elastic Plates to Pulse Excitations." *Transactions of the ASME Journal of Applied Mechanics*, Vol. 35 (Mar. 1968), pp. 47-52.

[46] Chandrasekharappa, G., and Srirangarajan, H. R. "Response of Plates to Pulse Excitations." *Mechanics Research Communications*, Vol. 13, No. 2 (1986), pp. 107-117.

[47] Chandrasekharappa, G., and Srirangarajan, H. R. "Nonlinear Response of Elastic Plates to Pulse Excitations." *Computers and Structures*, Vol. 27, No. 3 (1987), pp. 373-378.

[48] Nagaya, K. "Dynamic Response of a Plate With Arbitrary Shape." *Transactions of the ASME Journal of Applied Mechanics*, Vol. 47 (Sept. 1980), pp. 620-626.

[49] Nagaya, K. "Dynamic Response of a Membrane With Both Curved and Straight Line Boundaries." *Transactions of the ASME Journal of Applied Mechanics*, Vol. 46 (Sept. 1979), pp. 667-671.

[50] Assadi-Lamouki, A., and Krauthammer, T. "An Explicit Finite Difference Approach for the Mindlin Plate Analysis." *Computers and Structures*, Vol. 31, No. 4 (1989), pp. 487-494.

[51] Sansalone, M., Carino, N. J., and Hsu, N. N. "A Finite Element Study of Transient Wave Propagation in Plates." *Journal of the National Bureau of Standards*, Vol. 92, No. 4 (July-Aug. 1987), pp. 267-278.

[52] Beskos, D. E., and Leung, K. L. "Dynamic Response of Plate Systems by Combining Finite Differences, Finite Elements, and Laplace Transforms." *Computers and Structures*, Vol. 19, Nos. 5/6 (1984), pp. 763-775.

[53] Niemi, J., and Pramila, A.. "FEM Analysis of Transverse Vibrations of an Axially Moving Membrane Immersed in Ideal Fluid." *International Journal for Numerical Methods in Engineering*, Vol. 24 (1987), pp. 2301-2313.

[54] Nutter, D. W. "Investigation of Experimental Tension Measurement Methods." Unpublished M.S. report, Oklahoma State University, May 1988.

[55] Ahn, I. "Optimization of a Pneumatic Pulser for Tension Measurement in Webs." Unpublished M.S. report, Oklahoma State University, Dec. 1989.

[56] Lee, J. "Wave Propagation Velocity of an Air Loaded Web." Unpublished M.S. report, Oklahoma State University, Dec. 1986.

[57] Bradley, M. E. "Noncontact Tension Measurement In Webs by Acoustical Point Source Excitation." Unpublished M.S. report, Oklahoma State University, Dec. 1989.

[58] Johnson, D. E., Johnson, J. R., and Moore, H. P. *A Handbook of Active Filters*. Englewood Cliffs, NJ: Prentice-Hall Publishing Company, 1980.

[59] Bendat, J. S., and Piersol, A. G. *Random Data: Analysis and Measurement Procedures*. 1st Ed. New York: Wiley Interscience Publishing Company, 1971.

[60] Kinsler, L. E., Frey, A. R., Coppens, A. B., and Sanders, J. V. *Fundamentals of Acoustics*. 3rd Ed. New York: John Wiley and Sons Publishing Company, 1982.

[61] Merilainen, P. "Propagation and Excitation of Membrane Waves Loaded by Air." *Microwaves, Optics, and Acoustics*, Vol. 2, No. 5 (Sept. 1978), pp. 147-152.

[62] Bingulac, S. P., and Djorovic, M. "On the Generation of Sensitivity and Pseudosensitivity Functions for Multivariable Systems." *Proceedings of the Third IFAC Symposium*, Ischia, Italy, June 18-23, 1973, pp. 68-75.

[63] Beckwith, T. G., and Marangoni, R. D. *Mechanical Measurements*. 4th Ed. Reading, MA: Addison-Wesley, 1990.

[64] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. *Numerical Recipes*. Cambridge, NY: Cambridge University Press, 1986.

[65] Oppenheim, A. V., and Schafer, R. W. "The Discrete Fourier Transform." *Digital Signal Processing*. 1st Ed. Englewood Cliffs, NJ: Prentice-Hall Publishing Inc., 1975.

[66] Oppenheim, A. V., and Schafer, R. W. "Digital Filter Design Techniques." *Digital Signal Processing*. 1st Ed. Englewood Cliffs, NJ: Prentice-Hall Publishing, Inc., 1975.

[67] Courant, R., and Friedrichs, K. O. *Supersonic Flow and Shock Waves*. Vol. 1. New York: Interscience Publishers Inc., 1948.

[68] Whitham, G. B. *Linear and Nonlinear Waves*. New York: John Wiley and Sons, Inc., 1974.

[69] Shapiro, A. H. *The Dynamics and Thermodynamics of Compressible Fluid Flow*. Vol. 1. New York: Ronald Press Company, 1953.

[70] Friedlander, F. G. *Sound Pulses*. Cambridge, Great Britain: Cambridge University Press, 1958.

[71] Roache, P. J. *Computational Fluid Dynamics*. Albuquerque: Hermose Publishing Co., 1976.

[72] Emmons, H. W., Ed. *Fundamentals of Gas Dynamics*. Vol. 3. Princeton, NJ: Princeton University Press, 1958.

[73] Blackstock, D. T. "Propagation of Plane Sound Waves of Finite Amplitude in Nondissipative Fluids." *Journal of the Acoustical Society of America*, Vol. 34, No. 1 (Jan. 1962), pp. 9-30.

[74] Blackstock, D. T. "Propagation and Reflection of Plane Sound Waves of Finite Amplitude in Gases." Tech. Mem. 43. Acoustics Research Laboratory, Harvard University, June1960.

[75] Courant, R., and Hilbert, D. *Methods of Mathematical Physics*. Vol. 2. New York: John Wiley and Sons Interscience Publishers, 1962.

[76] Jeffrey, A., and Taniuti, T. *Nonlinear Wave Propagation: Mathematics in Science and Engineering*. Vol. 9. New York: Academic Press, 1964.

[77] Von Neumann, J., and Richtmyer, R. D. "A Method for the Numerical Calculation of Hydrodynamic Shocks." *Journal of Applied Physics*, Vol. 21 (March 1950), pp 232-237.

[78] Pulliam, T. H. "Artificial Dissipation Models for the Euler Equations." *AIAA Journal*, Vol. 24, No. 12 (Dec. 1986), pp. 1931-1940.

[79] Lax, P. D., and Wendroff, B. "Systems of Conservation Laws." *Communications on Pure and Applied Mathematics*, Vol. 13 (1960), pp. 217-237.

[80] MacCormack, R. W. "A Numerical Method for Solving the Equations of Compressible Viscous Flow." *AIAA Journal*, Vol. 20, No. 9 (Sept. 1982), pp. 1275-1281.

[81] Mehta, R. C., and Sastri, V. M. K. "Numerical Investigation of the Impingement of an Oblique Shock on a Flat Plate." *Warme-und Stoffubertragung*, Vol. 23 (1988), pp. 123-125.

[82] Moretti, G. "A Technique for Integrating the Two-Dimensional Euler Equations." *Computers and Fluids*, Vol. 15, No. 1 (1987), pp. 59-75.

[83] Einfeldt, B. "On Godunov-Type Methods for Gas Dynamics." *SIAM Journal of Numerical Analysis*, Vol. 25, No. 2 (Apr. 1988), pp. 294-318.

[84] Lapidus, A. "Computation of Radially Symmetric Shocked Flows." *Journal of Computational Physics*, Vol. 8 (1971), pp. 106-118.

[85] Gary, J. "On Certain Finite Difference Schemes for Hyperbolic Systems." *Mathematics of Computation*, Vol. 18 (Jan. 1964), pp. 1-18.

[86] Tyler, L. D. "Numerical Solutions of the Flow Field Produced by a Plane Shock Wave Emerging Into a Cross Flow." Ph.D. thesis, Oklahoma State University, May 1965.

[87] Walker, W. F. "A Numerical Solution for the Interaction of a Moving Shock Wave With a Turbulent Mixing Region." Ph.D. thesis, Oklahoma State University, May 1966.

[88] Leutloff, D., and Roesner, K. G. "Numerical Solution of Converging Shock Problem." *Computers and Fluids*, Vol. 16, No. 2 (1988), pp. 175-182.

[89] Rivello, R. M. *Theory and Analysis of Flight Structures*. New York: McGraw-Hill Book Company, 1969.

[90] Szilard, R. *Theory and Analysis of Plates—Classical and Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall Publishing Co., Inc., 1974.

[91] Timoshenko, S. P., and Gere, J. M. *Theory of Elastic Stability*. 2nd Ed. New York: McGraw-Hill Publishing Co., 1988.

[92] Bendat, J. S. *Principals and Applications of Random Noise Theory*. Huntington, NY: Robert E. Kreiger Publishing Co., Inc., 1977.

[93] Stearns, S. D., and David, R. A. Signal Processing Algorithms. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1988.

APPENDIX A

PRESSURE CONTOURS FROM PULSE MODELING

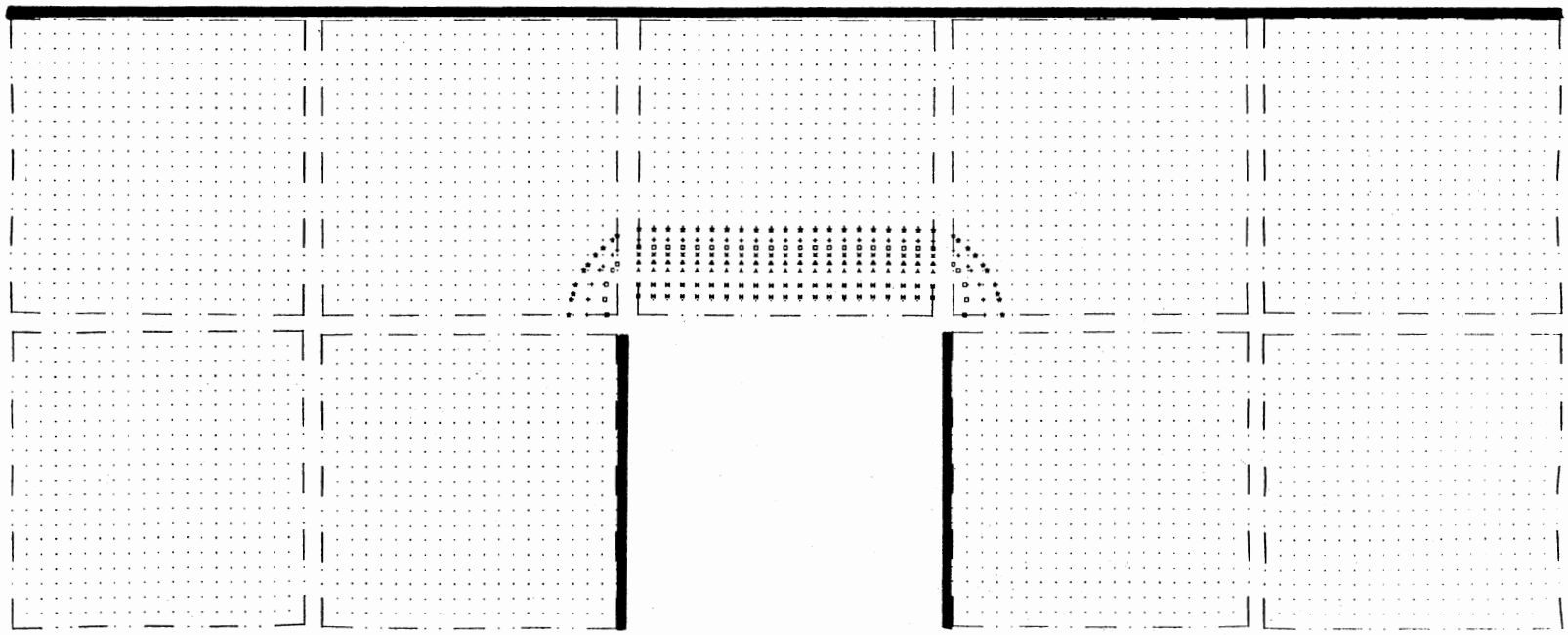


Figure A.1. Pressure Contours at 20 Time Frames; Dimensional Time = 2.68 Microsec

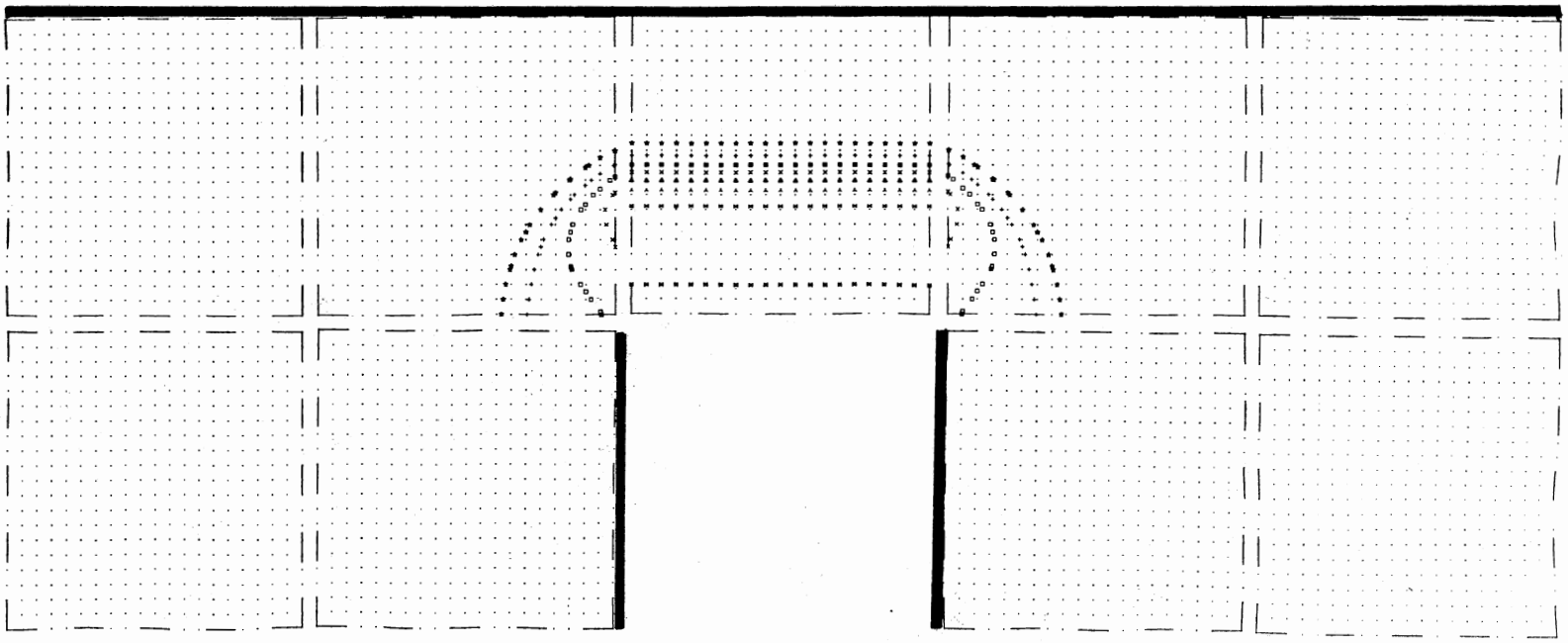


Figure A.2. Pressure Contours at 40 Time Frames; Dimensional Time = 5.38 Microsec

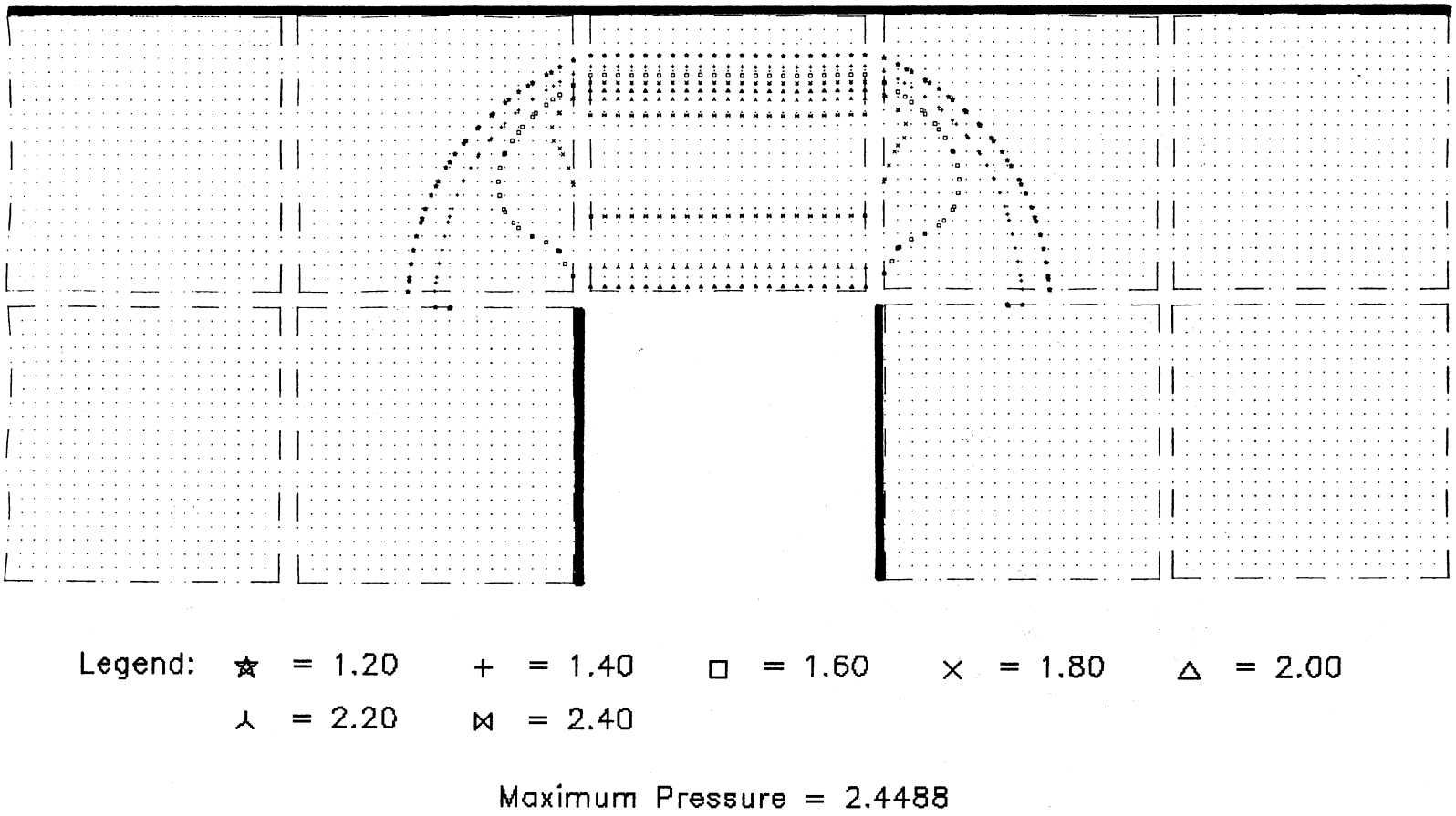


Figure A.3. Pressure Contours at 60 Time Frames; Dimensional Time = 8.18 Microsec

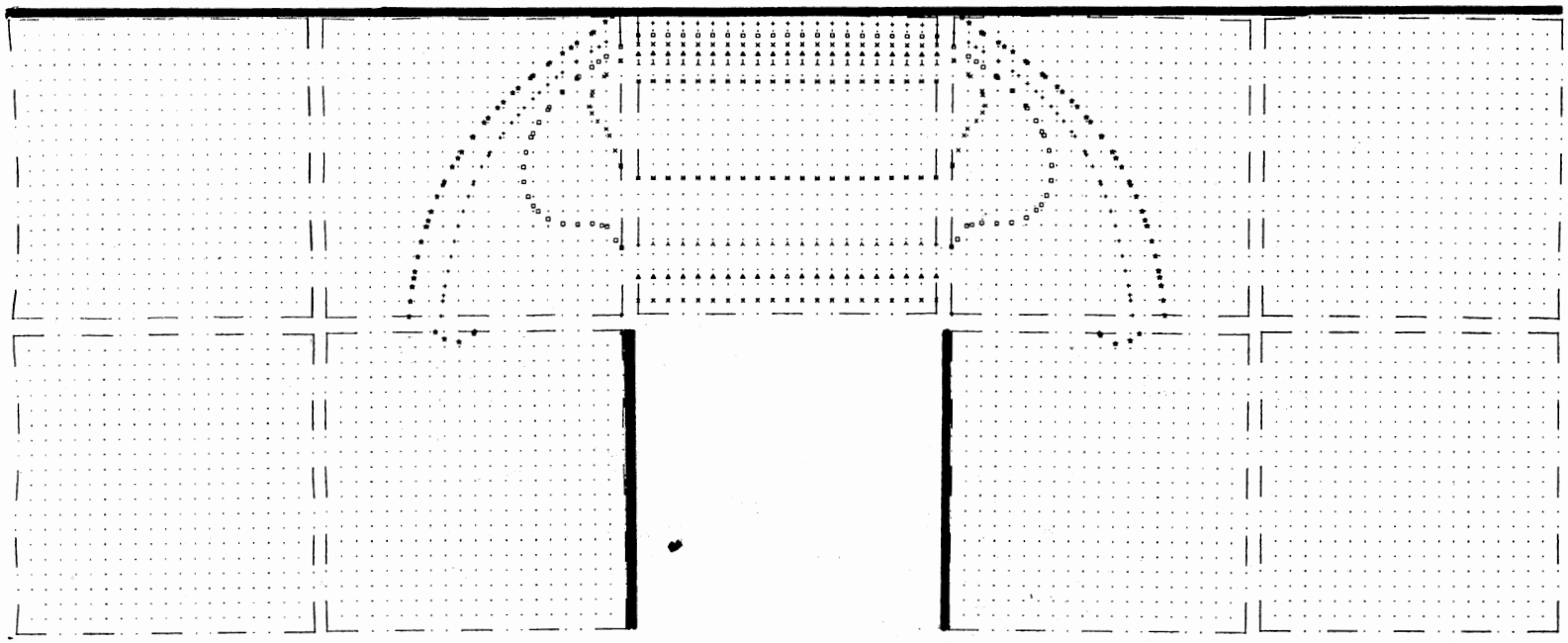


Figure A.4. Pressure Contours at 70 Time Frames; Dimensional Time = 9.60 Microsec

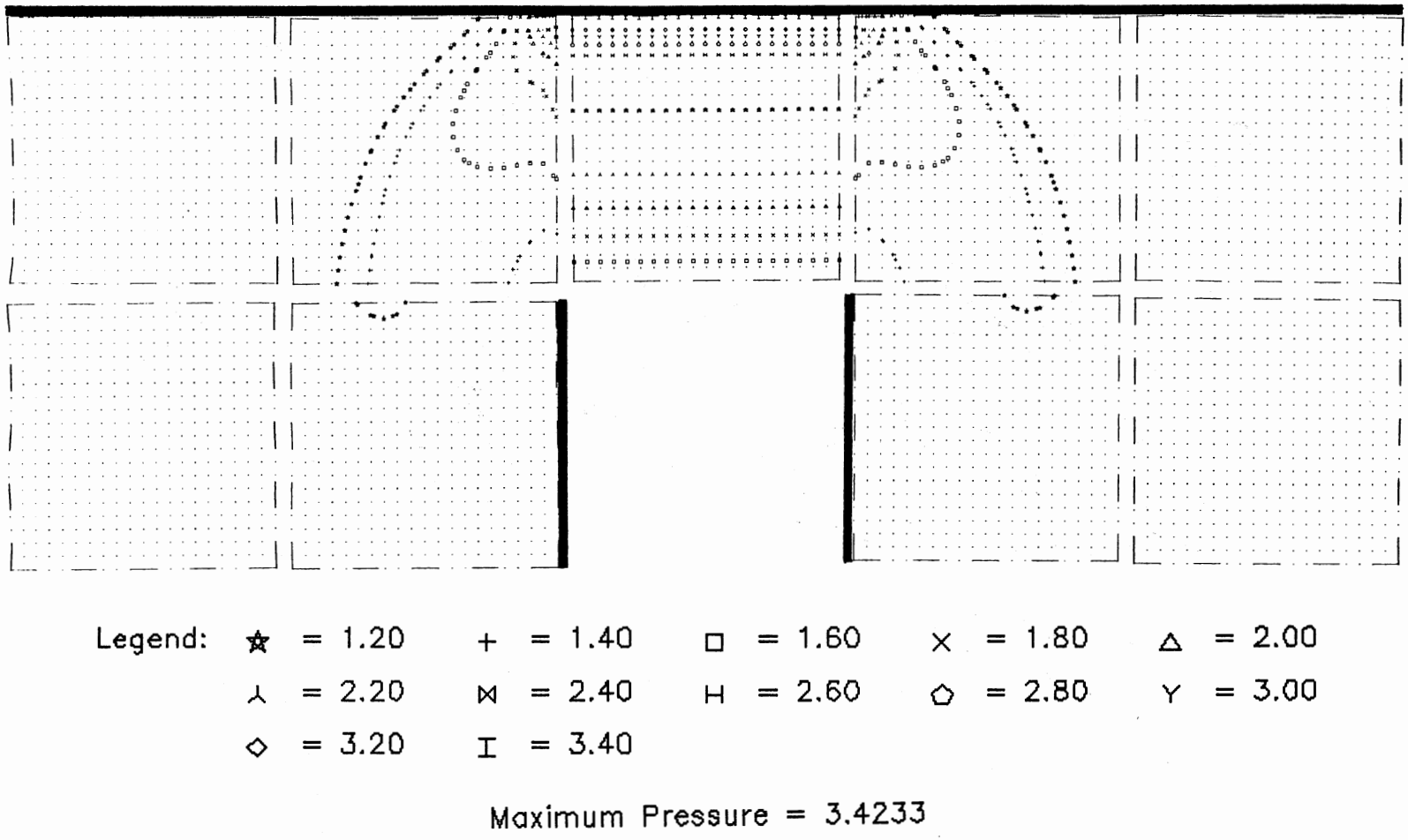


Figure A.5. Pressure Contours at 80 Time Frames; Dimensional Time = 11.02 Microsec

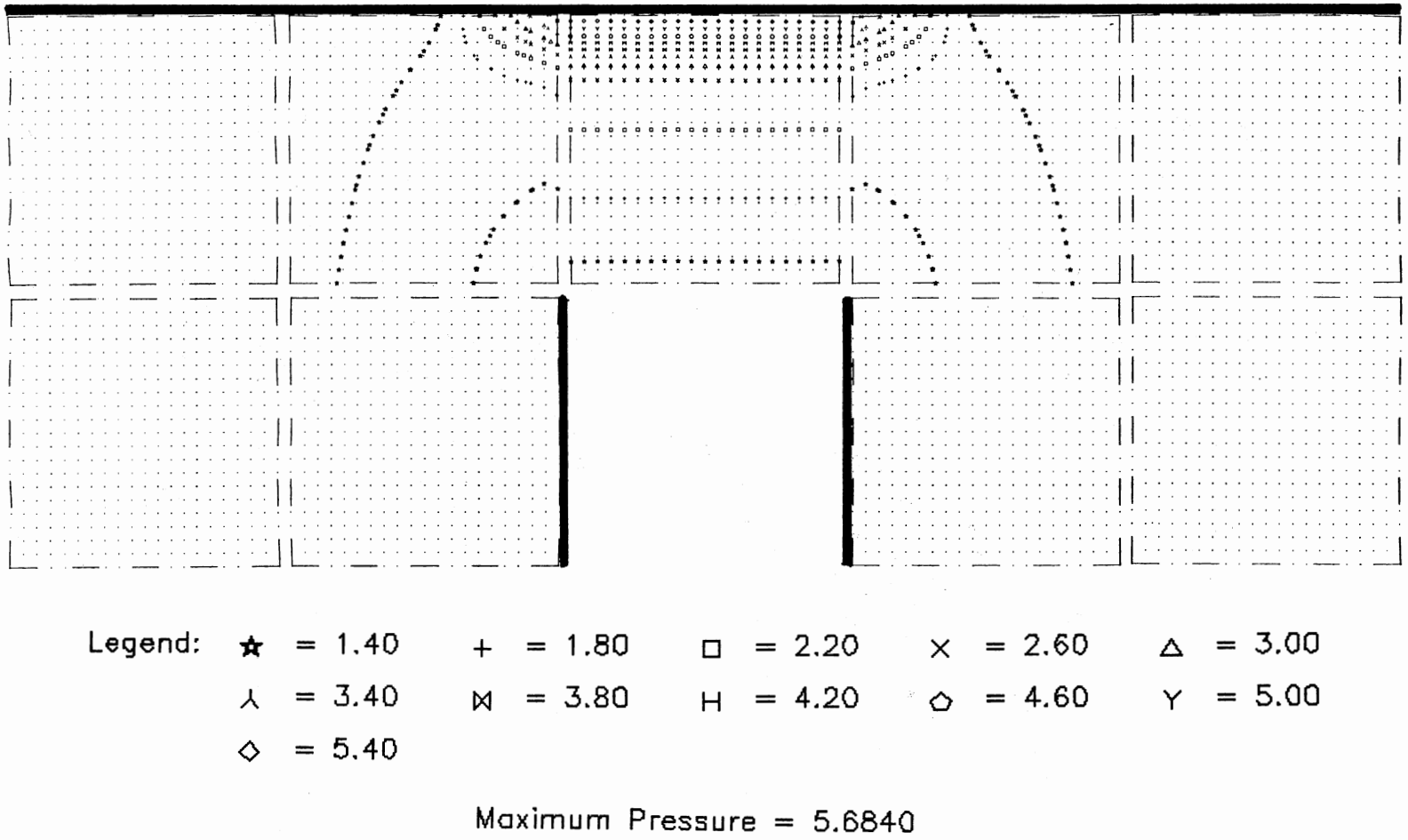


Figure A.6. Pressure Contours at 90 Time Frames; Dimensional Time = 12.47 Microsec

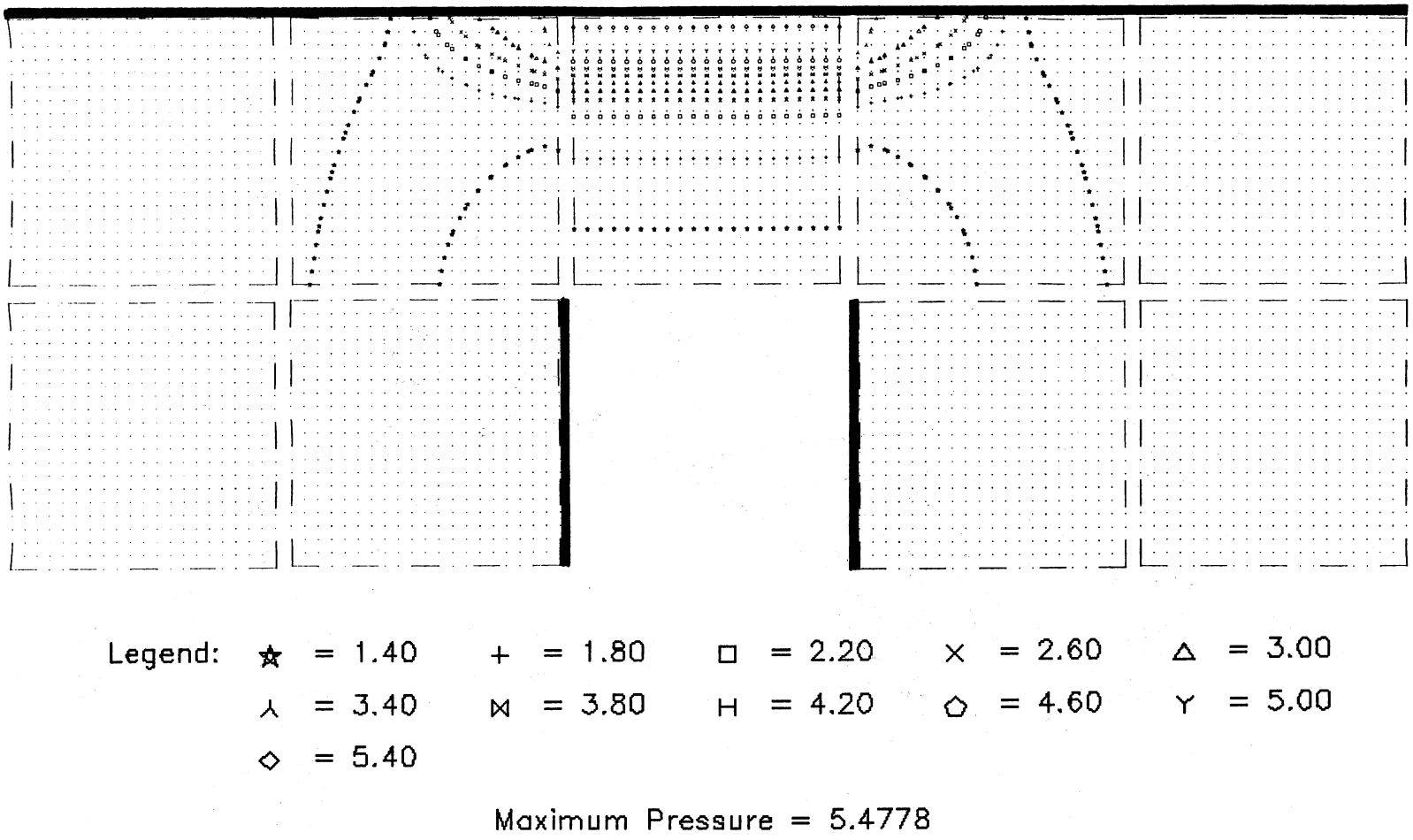


Figure A.7. Pressure Contours at 100 Time Frames; Dimensional Time = 13.99 Microsec

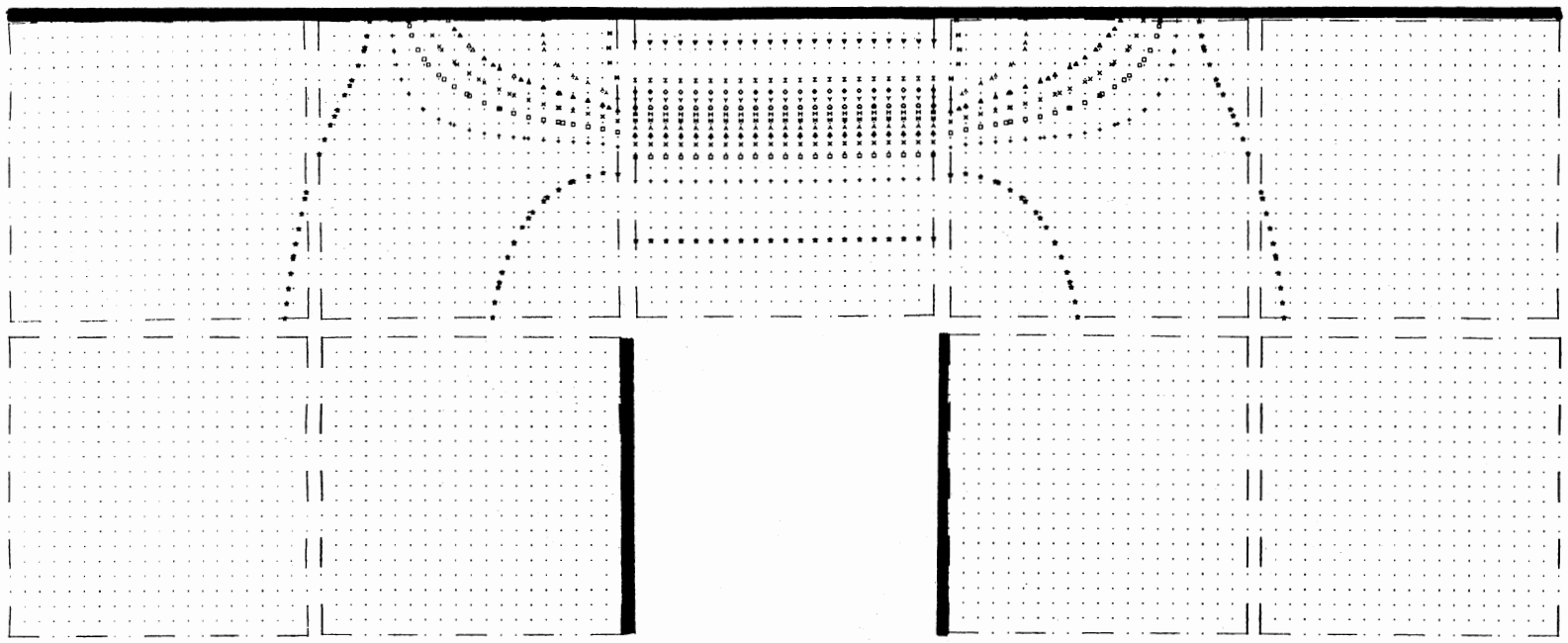


Figure A.8. Pressure Contours at 110 Time Frames; Dimensional Time = 15.52 Microsec

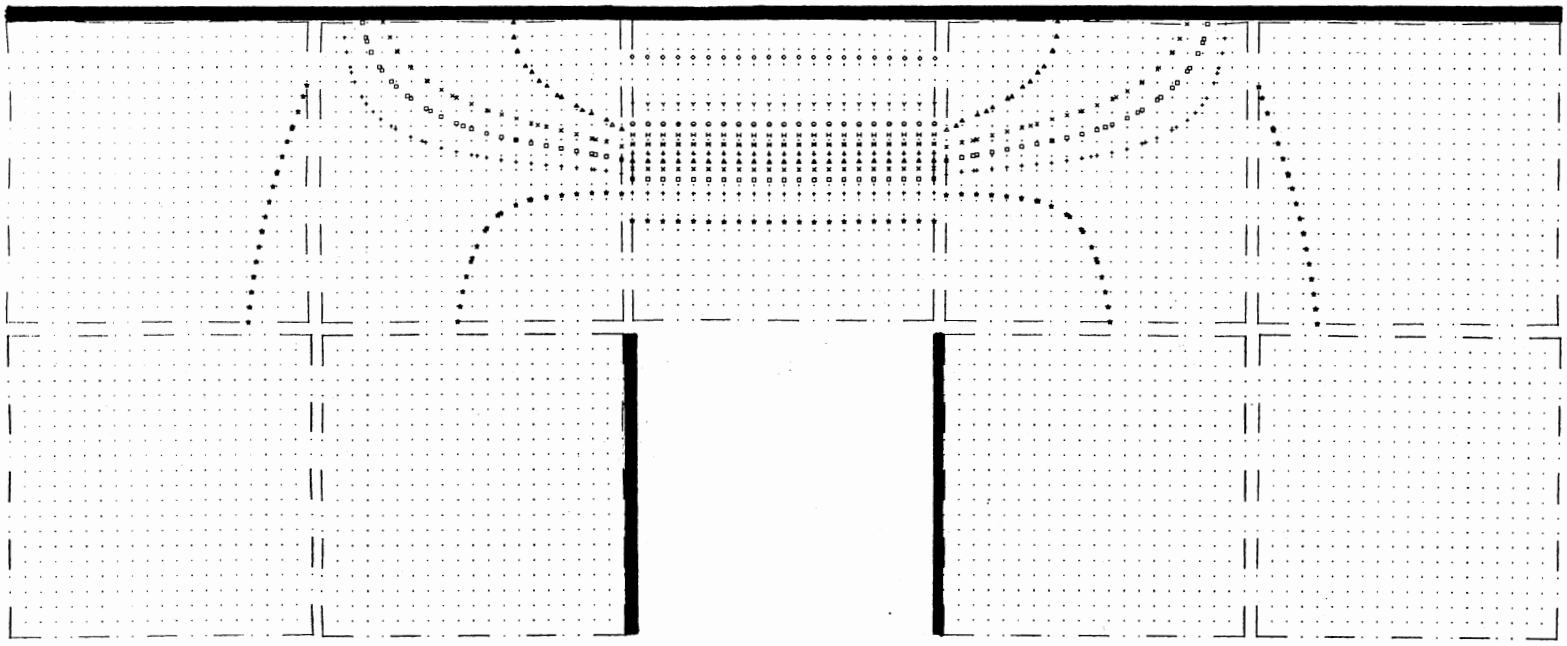


Figure A.9. Pressure Contours at 120 Time Frames; Dimensional Time = 17.05 Microsec

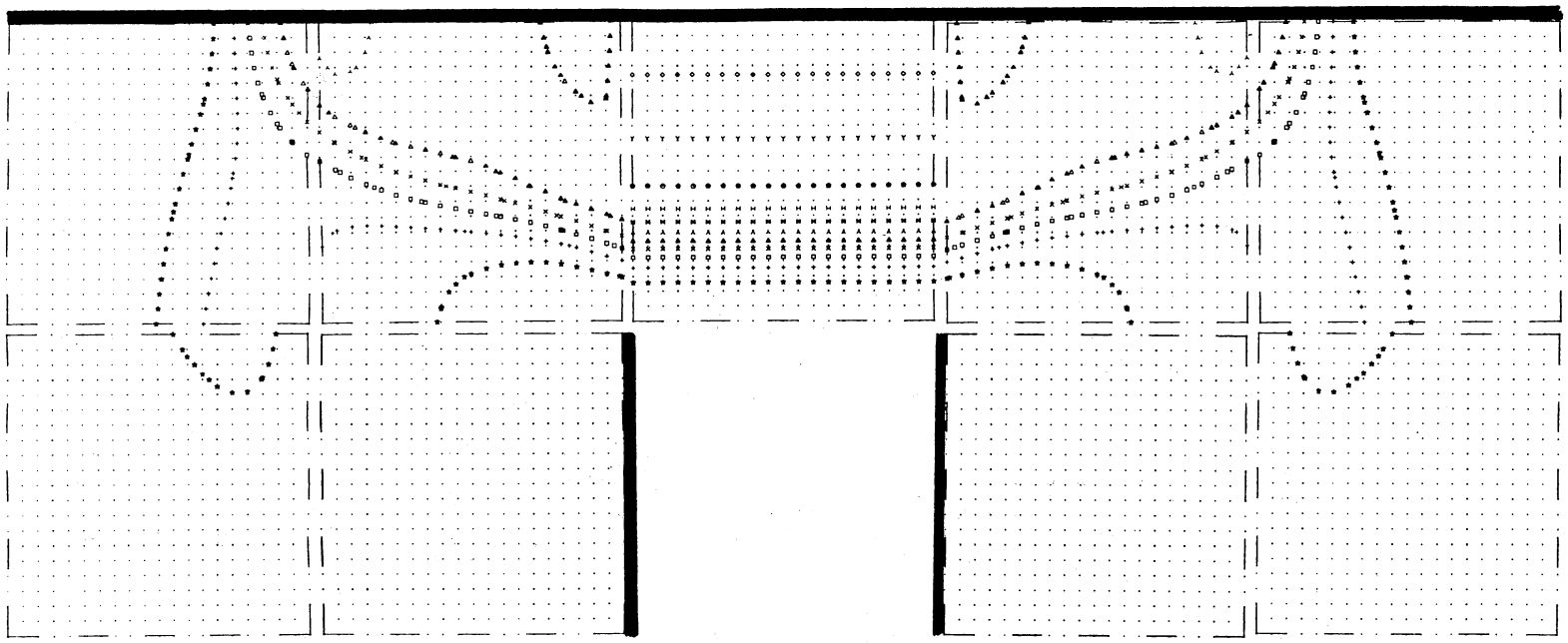
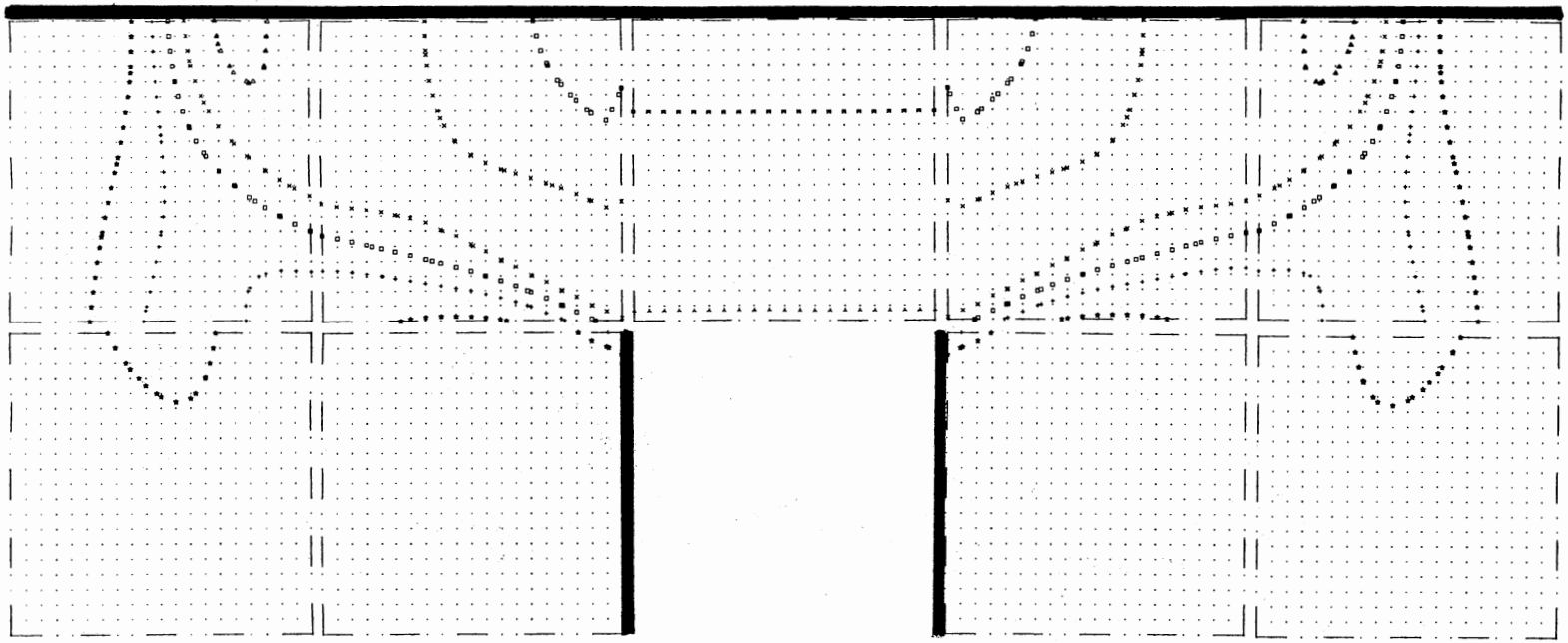


Figure A.10. Pressure Contours at 140 Time Frames; Dimensional Time = 20.13 Microsec



Legend: ☆ = 1.20 + = 1.40 □ = 1.60 × = 2.00 △ = 2.20 ⋈ = 2.40

Maximum Pressure = 2.4381

Figure A.11. Pressure Contours at 160 Time Frames; Dimensional Time = 23.08 Microsec

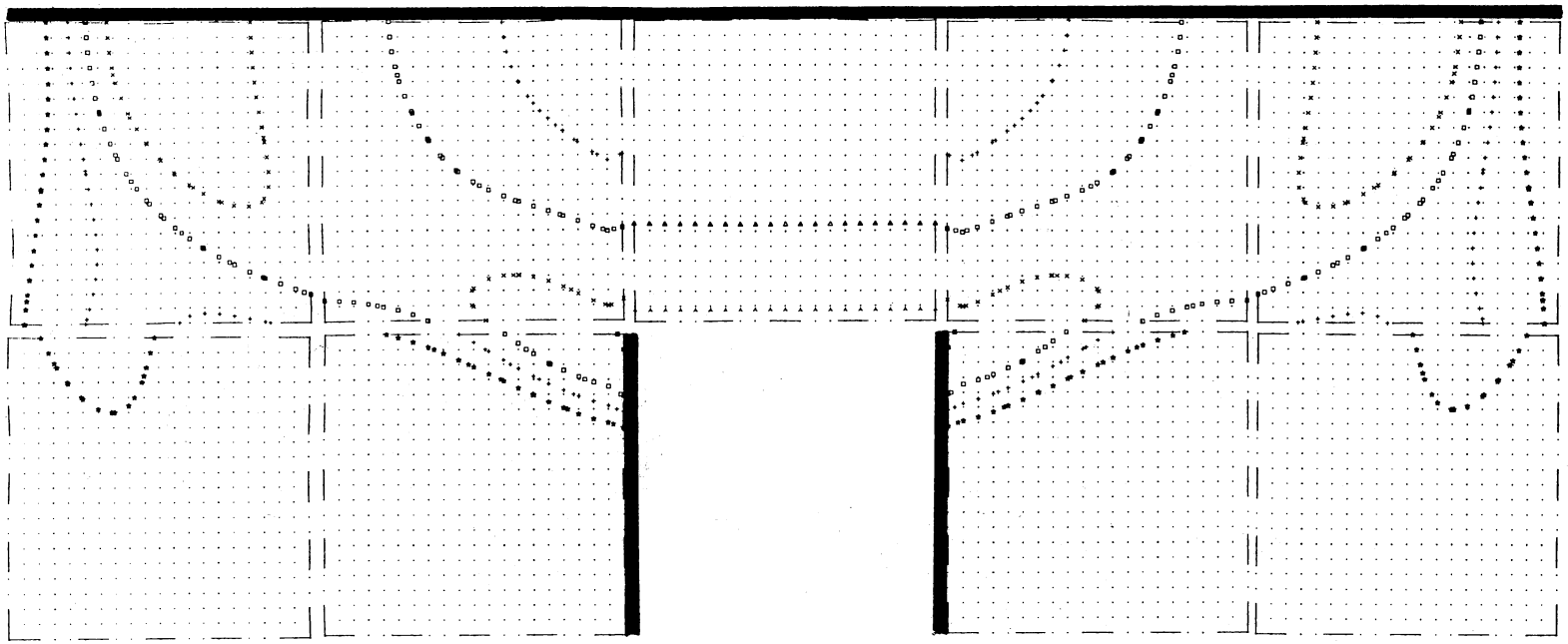


Figure A.12. Pressure Contours at 180 Time Frames; Dimensional Time = 25.97 Microsec

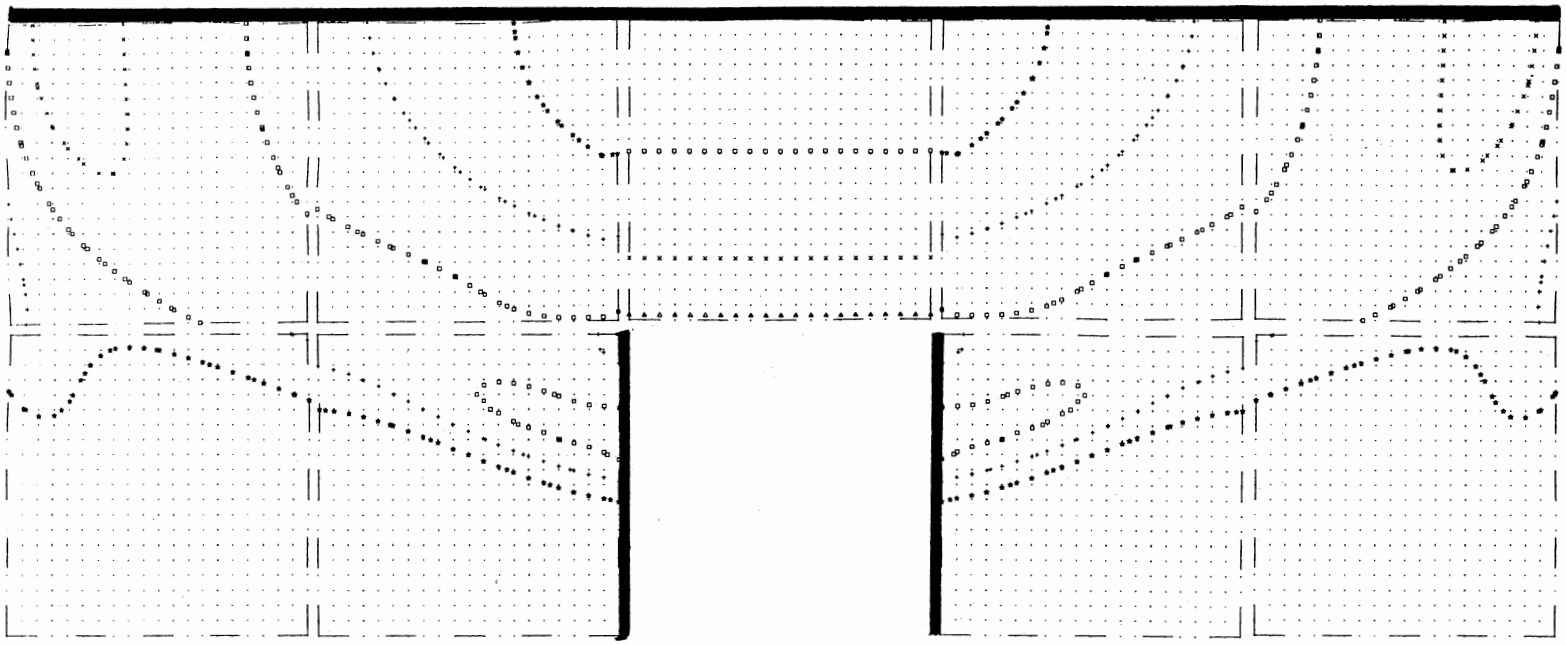


Figure A.13. Pressure Contours at 200 Time Frames; Dimensional Time = 28.98 Microsec

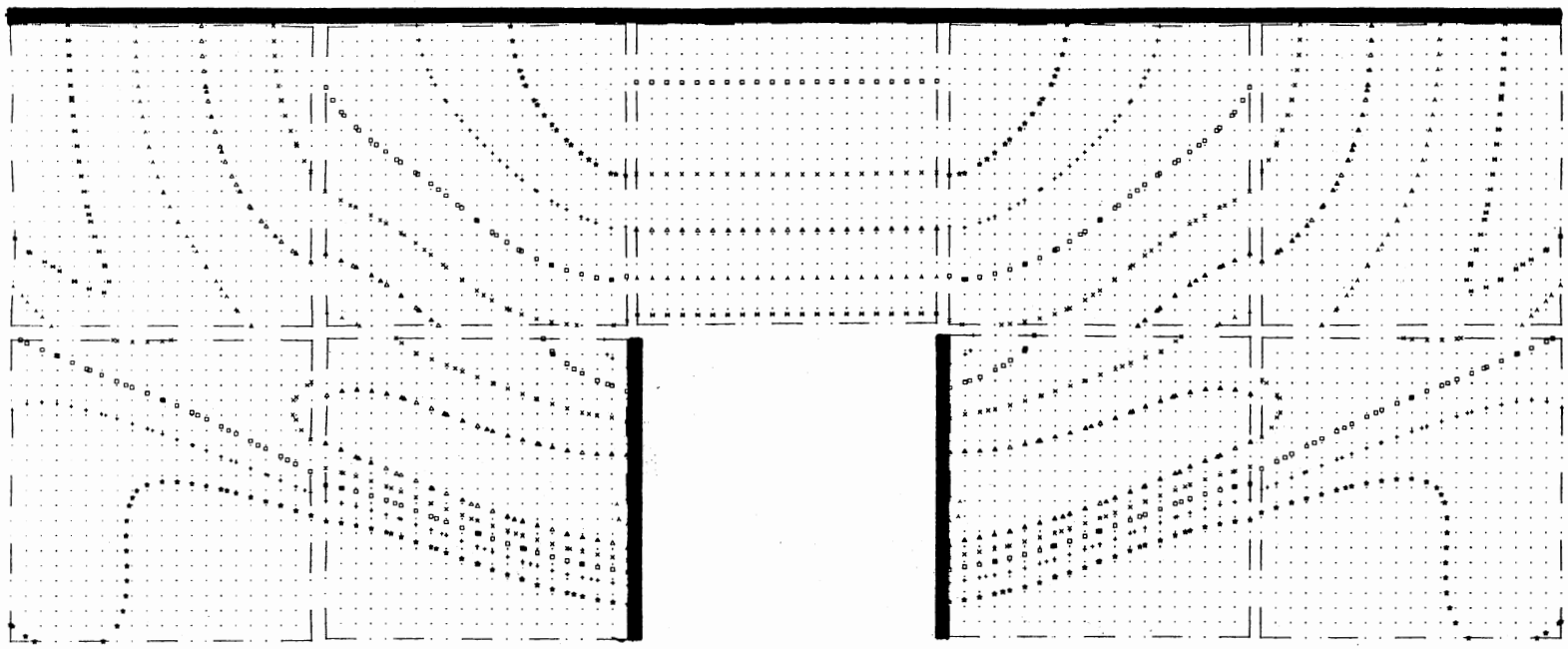


Figure A.14. Pressure Contours at 220 Time Frames; Dimensional Time = 32.18 Microsec

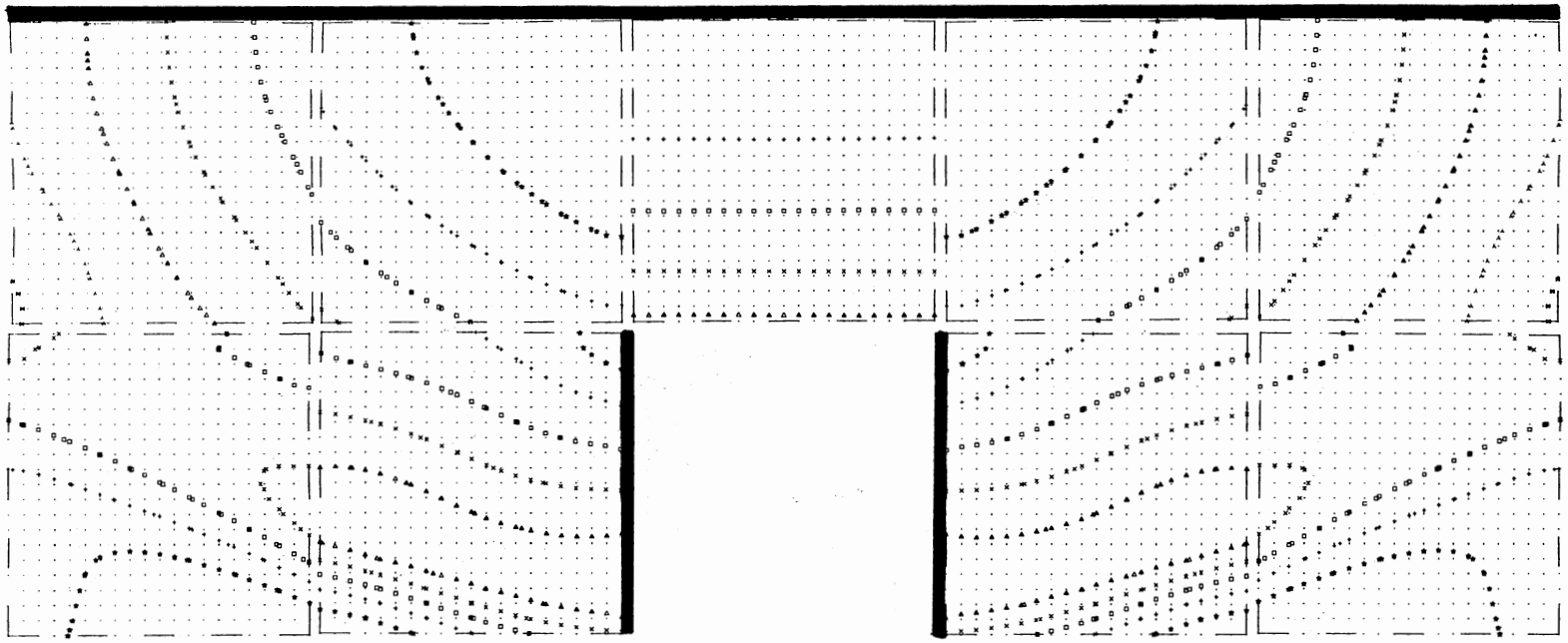


Figure A.15. Pressure Contours at 240 Time Frames; Dimensional Time = 35.63 Microsec

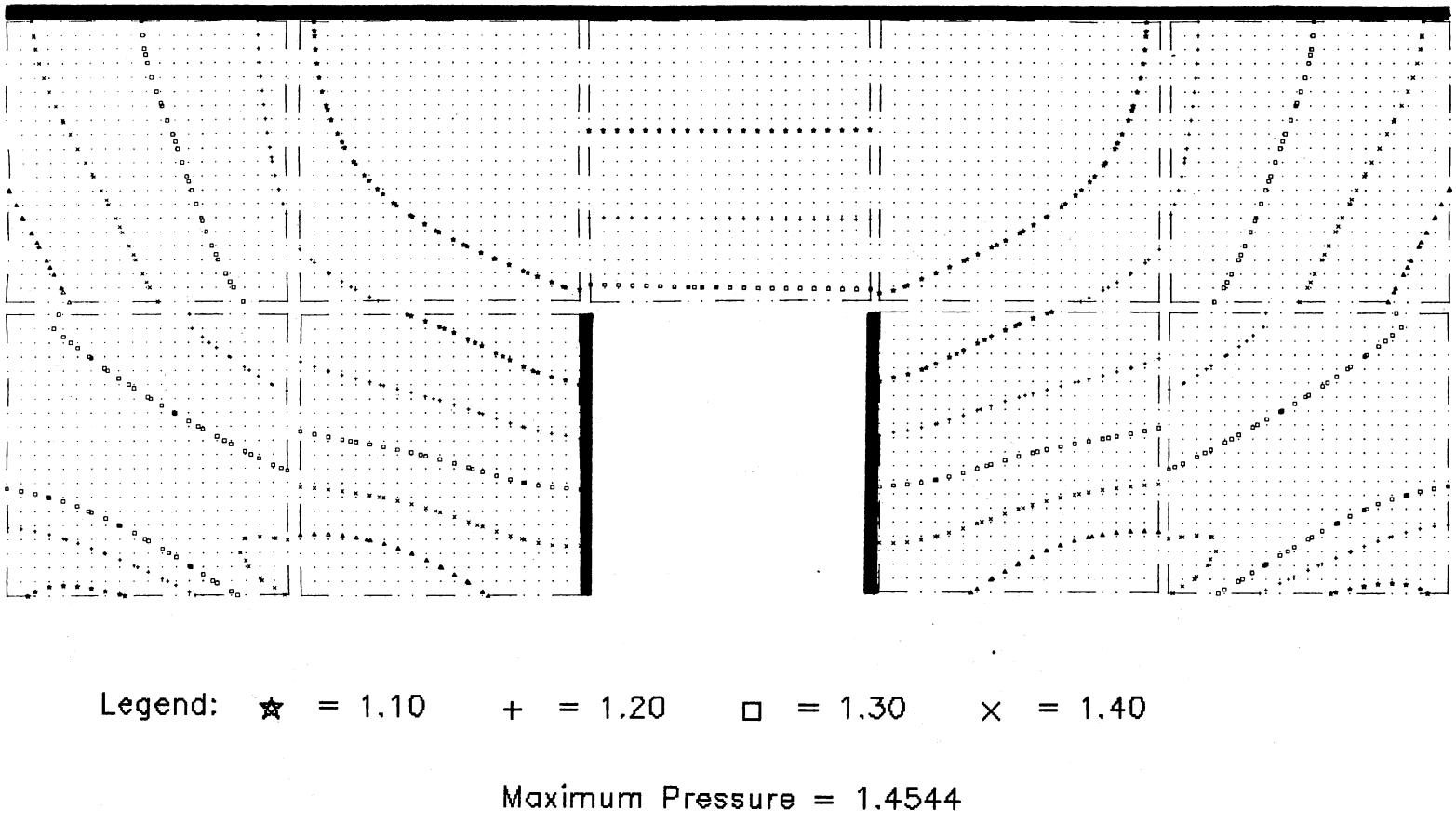


Figure A.16. Pressure Contours at 260 Time Frames; Dimensional Time = 39.29 Microsec

APPENDIX B

VELOCITY CONTOURS FROM PULSE MODELING

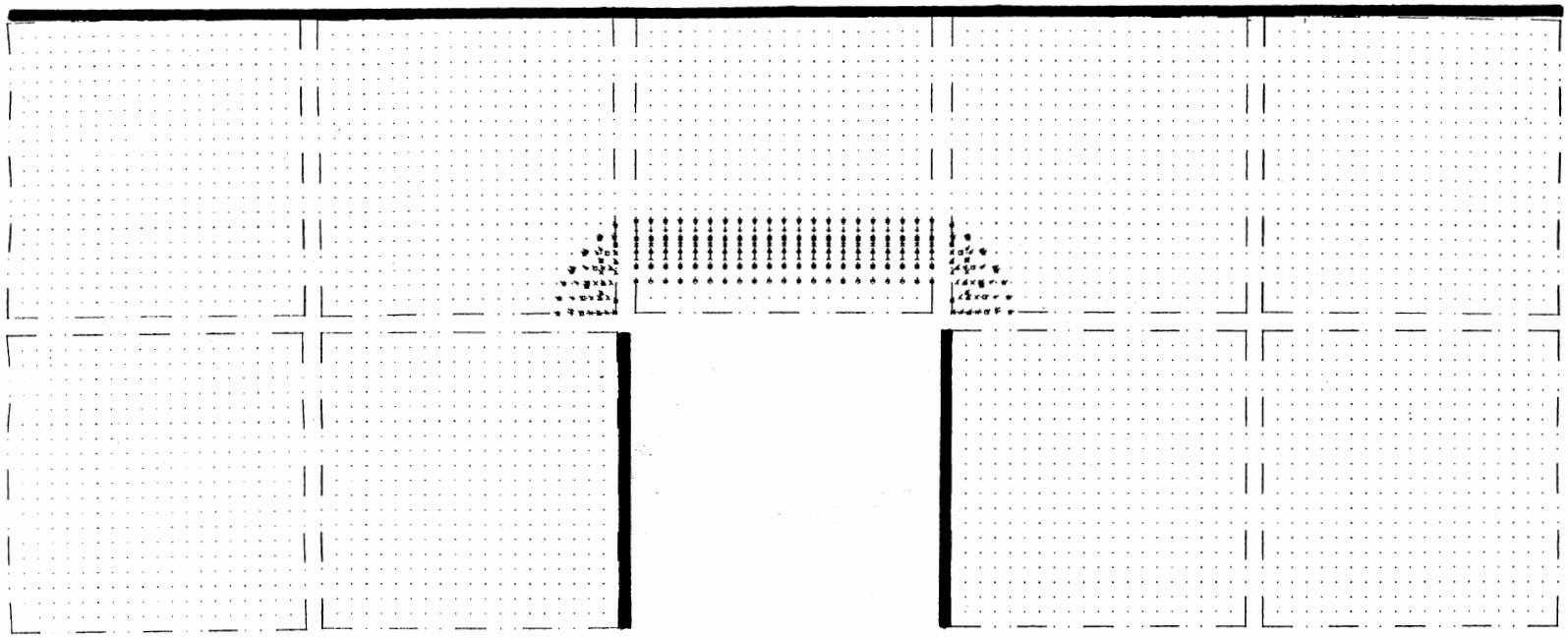


Figure B.1. Velocity Contours at 20 Time Frames; Dimensional Time = 2.68 Microsec

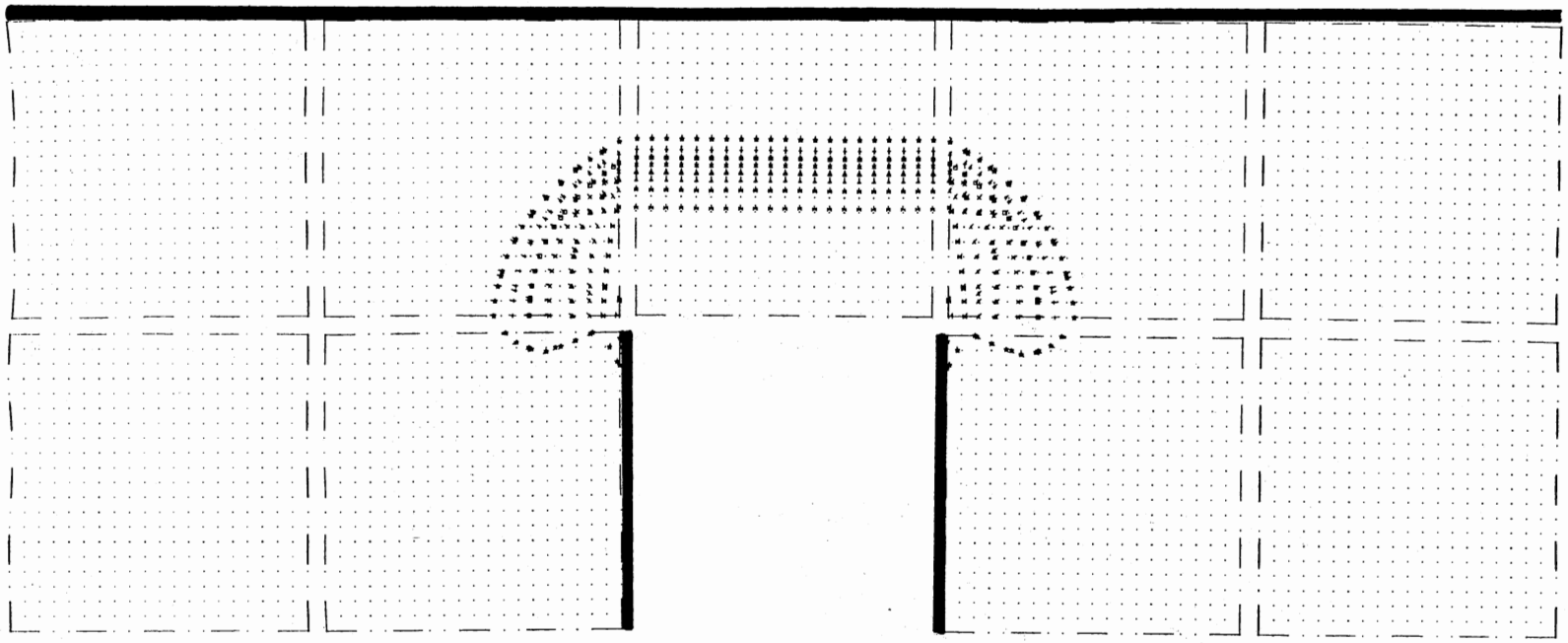


Figure B.2. Velocity Contours at 40 Time Frames; Dimensional Time = 5.38 Microsec

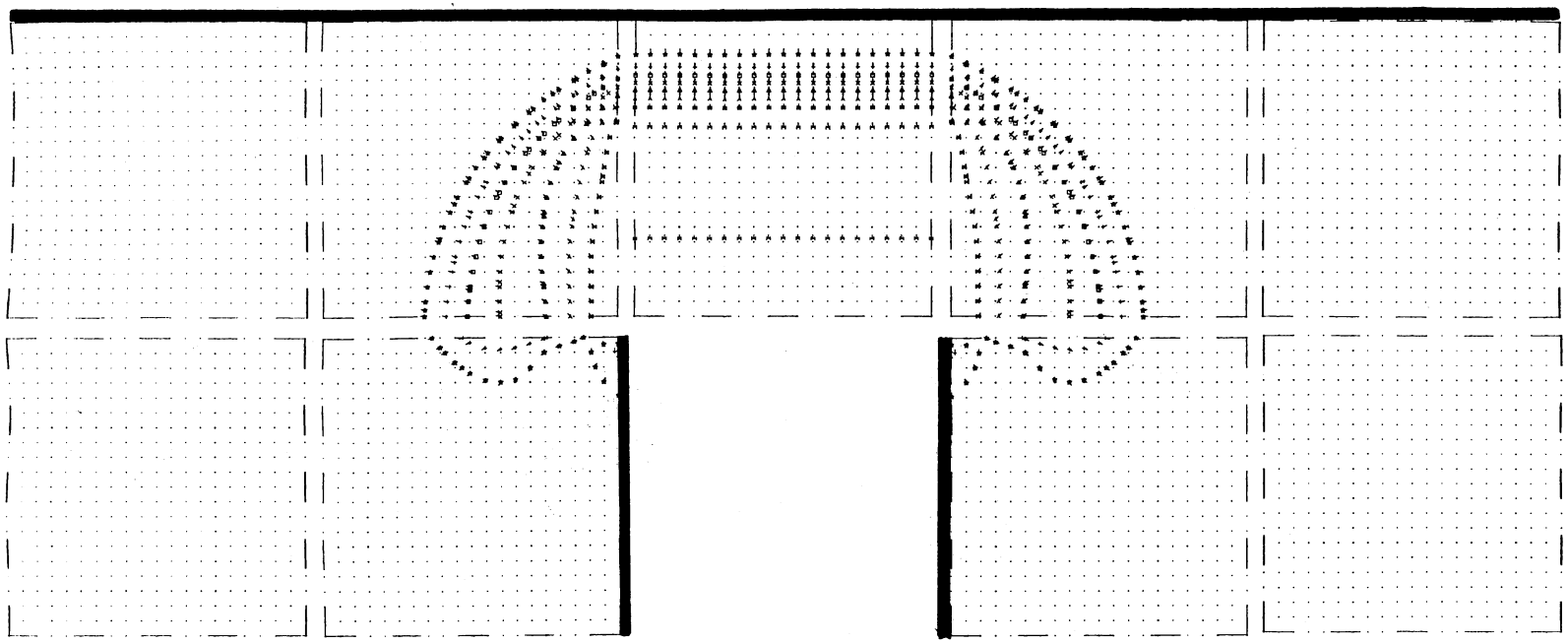


Figure B.3. Velocity Contours at 60 Time Frames; Dimensional Time = 8.18 Microsec

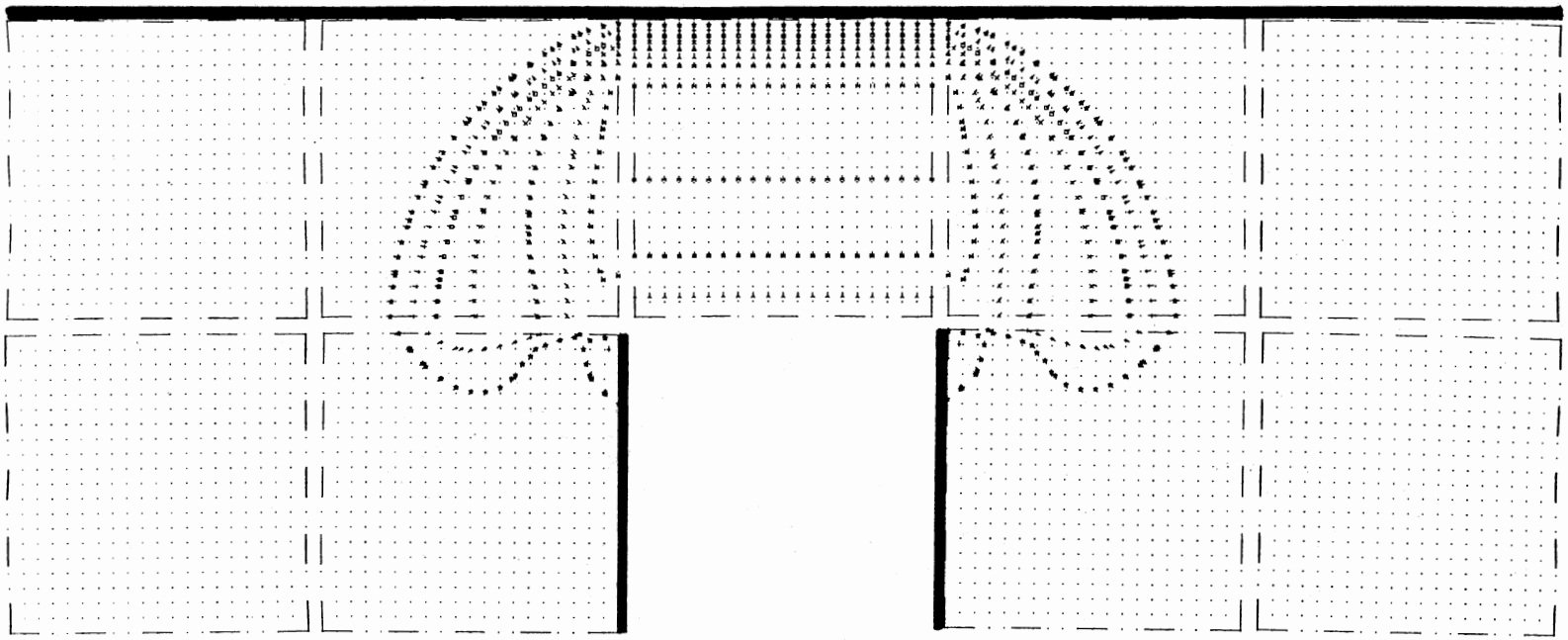


Figure B.4. Velocity Contours at 70 Time Frames; Dimensional Time = 9.60 Microsec

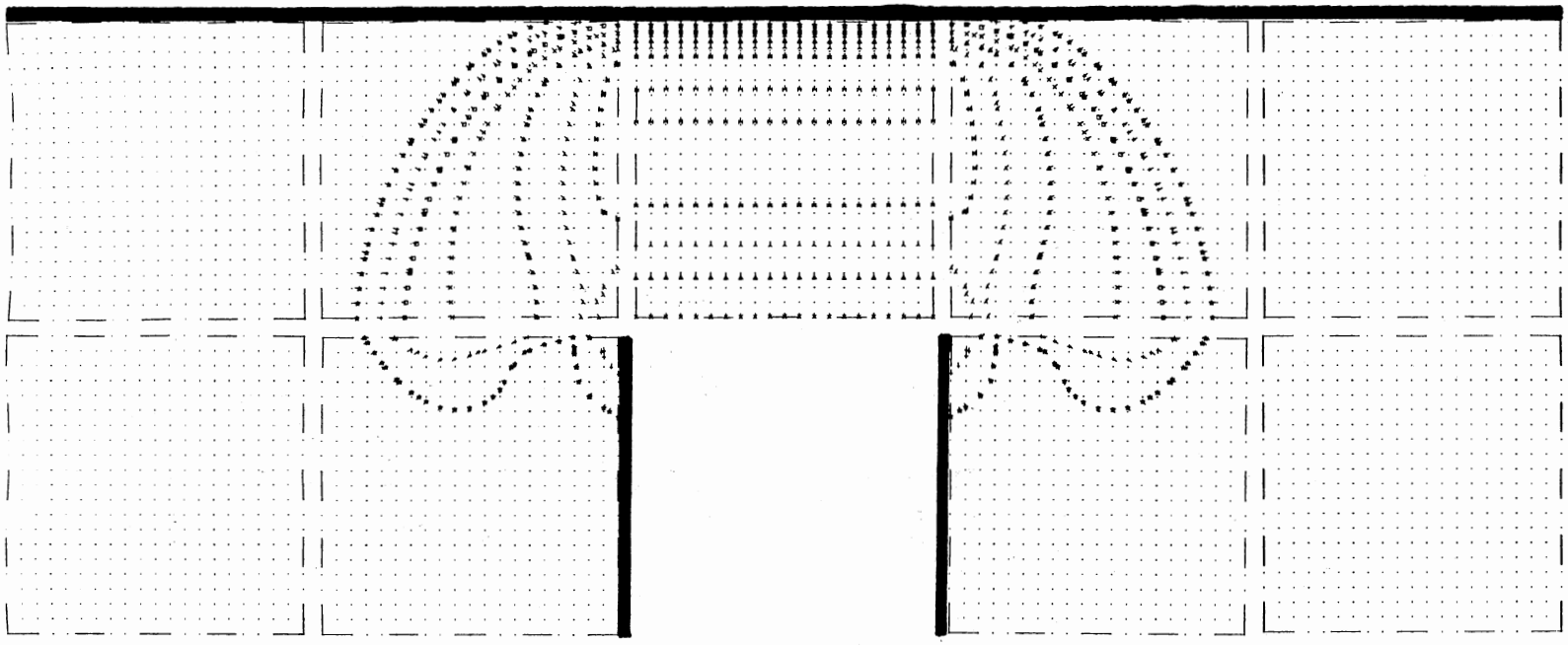


Figure B.5. Velocity Contours at 80 Time Frames; Dimensional Time = 11.02 Microsec

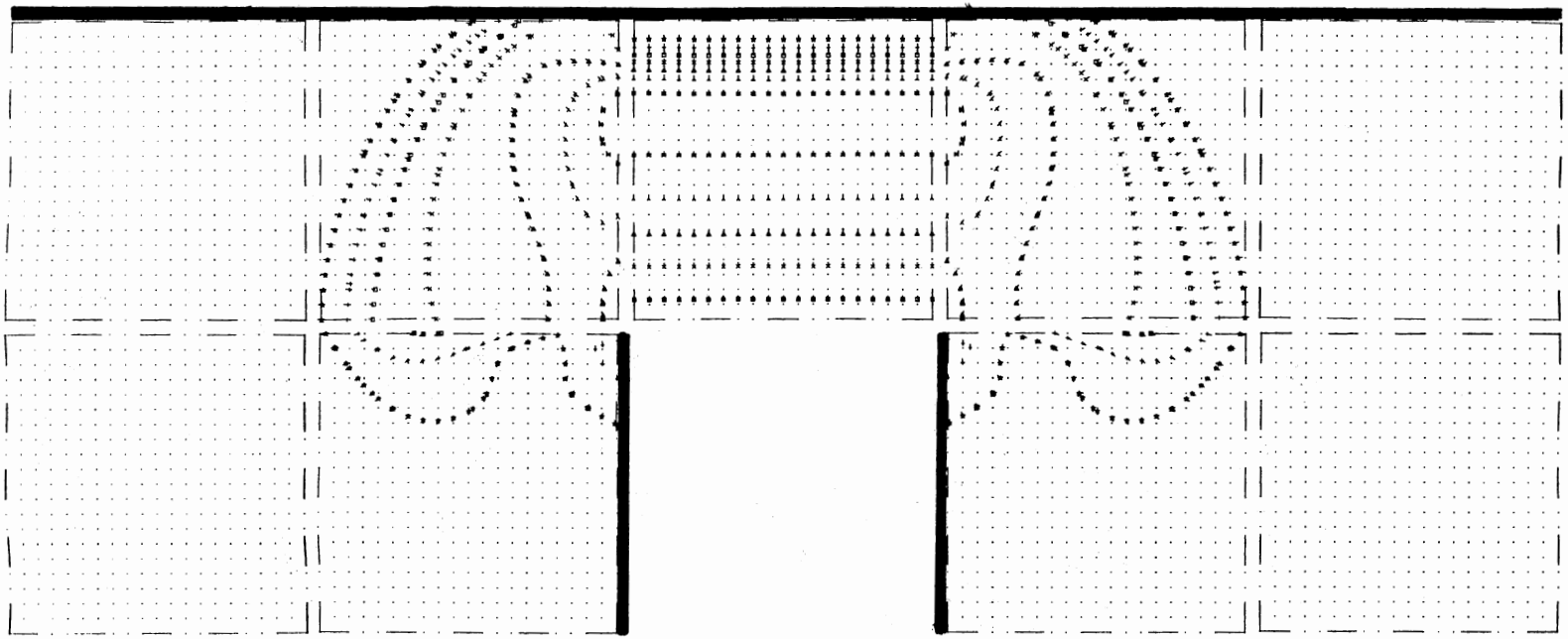
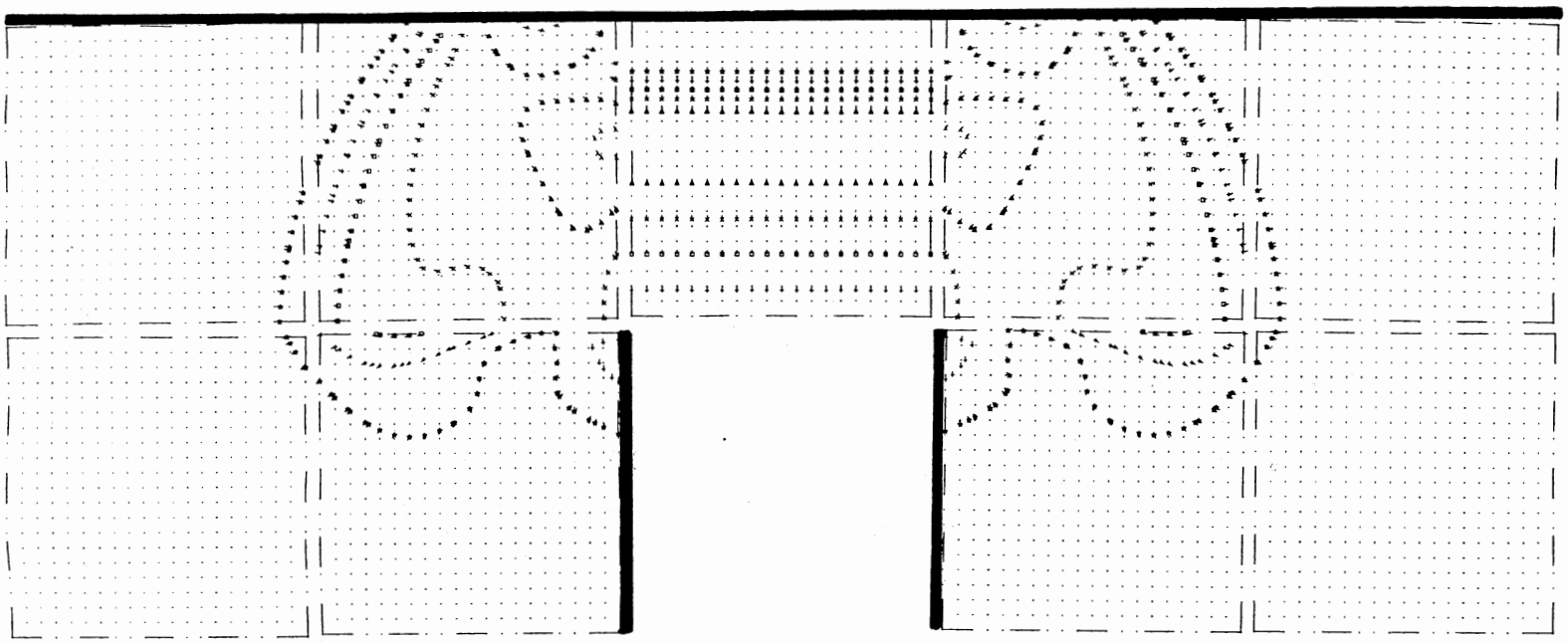


Figure B.6. Velocity Contours at 90 Time Frames; Dimensional Time = 12.47 Microsec



Legend: ☆ = 0.10 + = 0.20 □ = 0.30 × = 0.40 △ = 0.50

Maximum Velocity = 0.5817

Figure B.7. Velocity Contours at 100 Time Frames; Dimensional Time = 13.99 Microsec

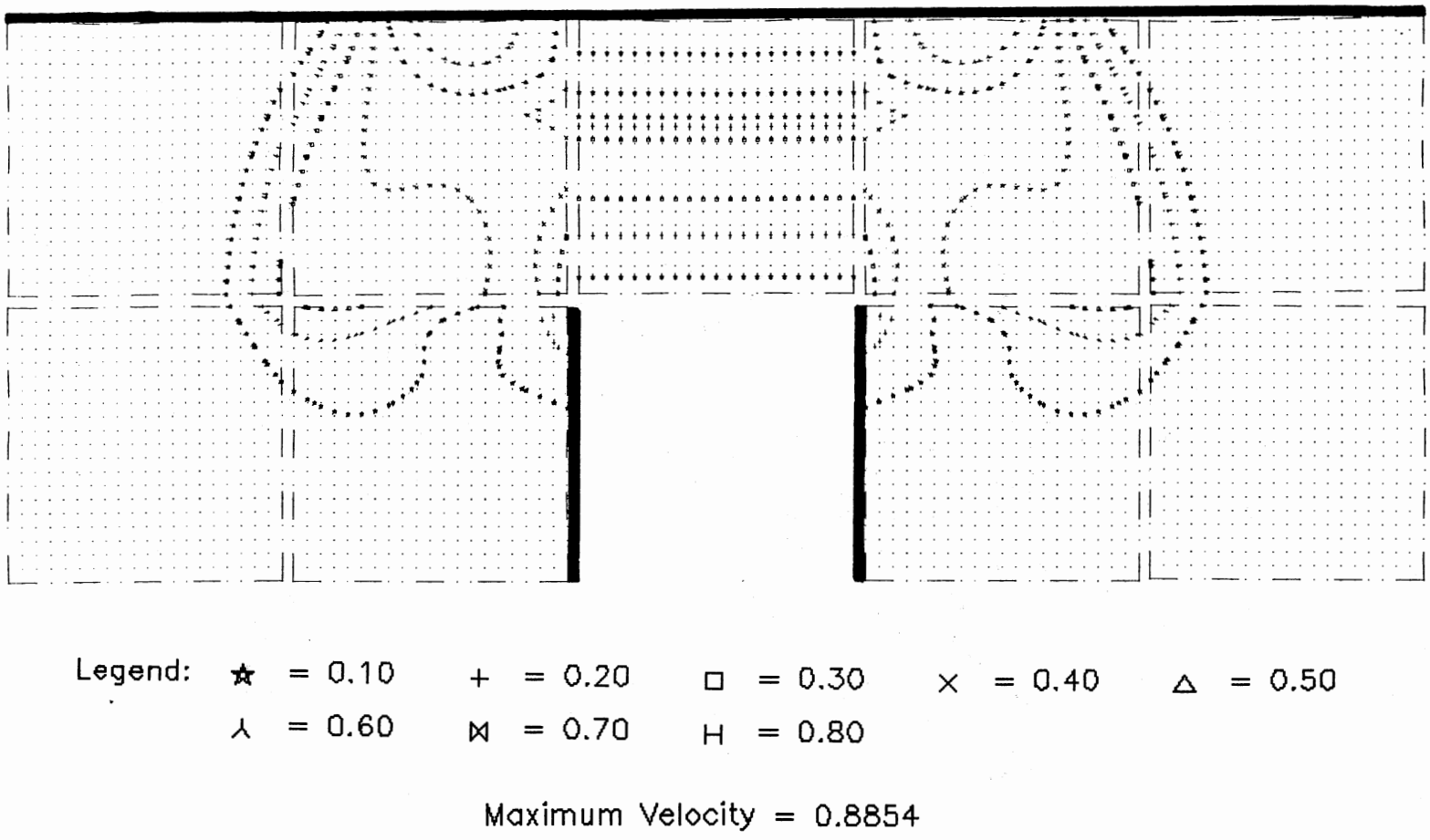


Figure B.8. Velocity Contours at 110 Time Frames; Dimensional Time = 15.52 Microsec

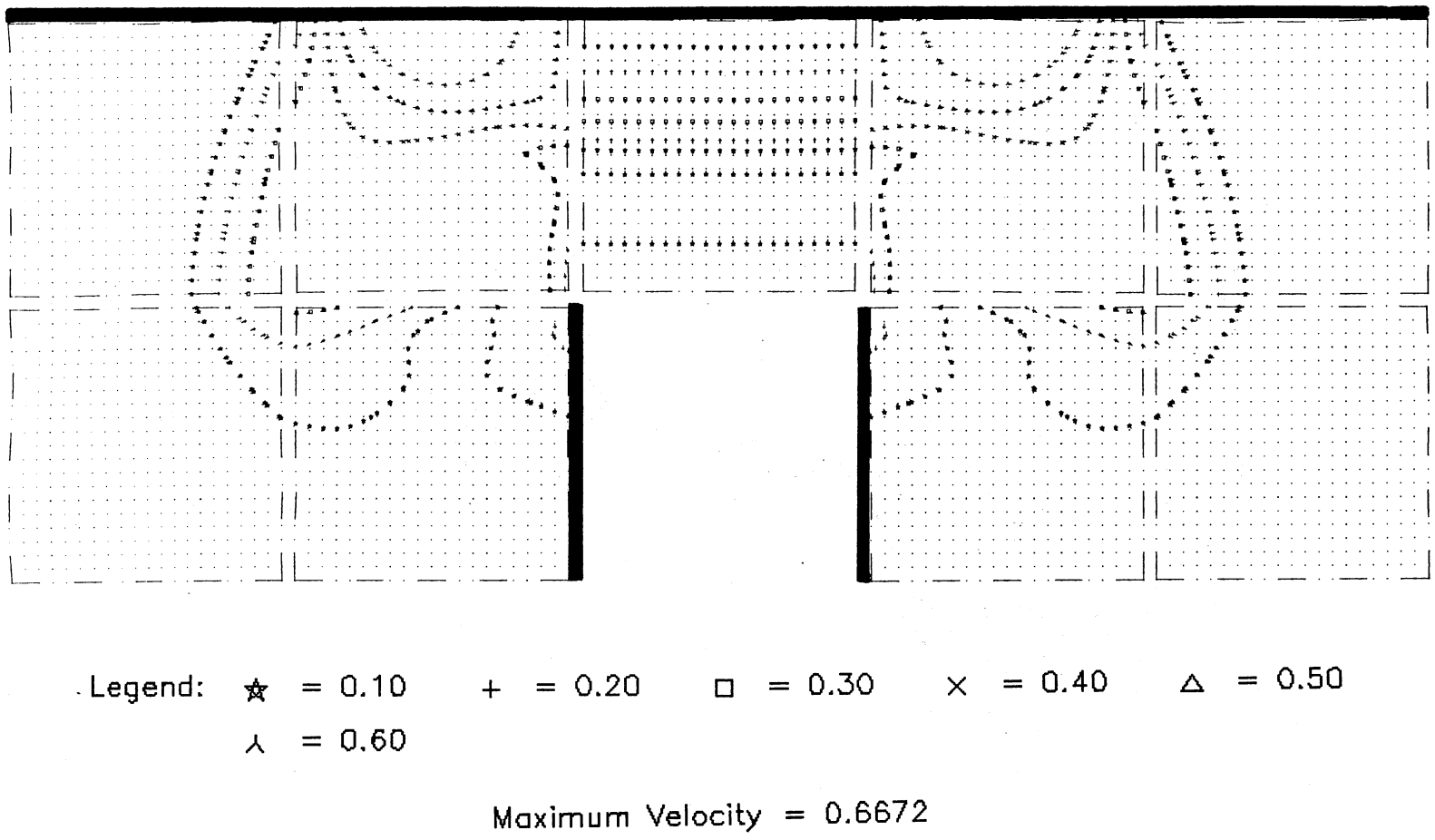


Figure B.9. Velocity Contours at 120 Time Frames; Dimensional Time = 17.05 Microsec

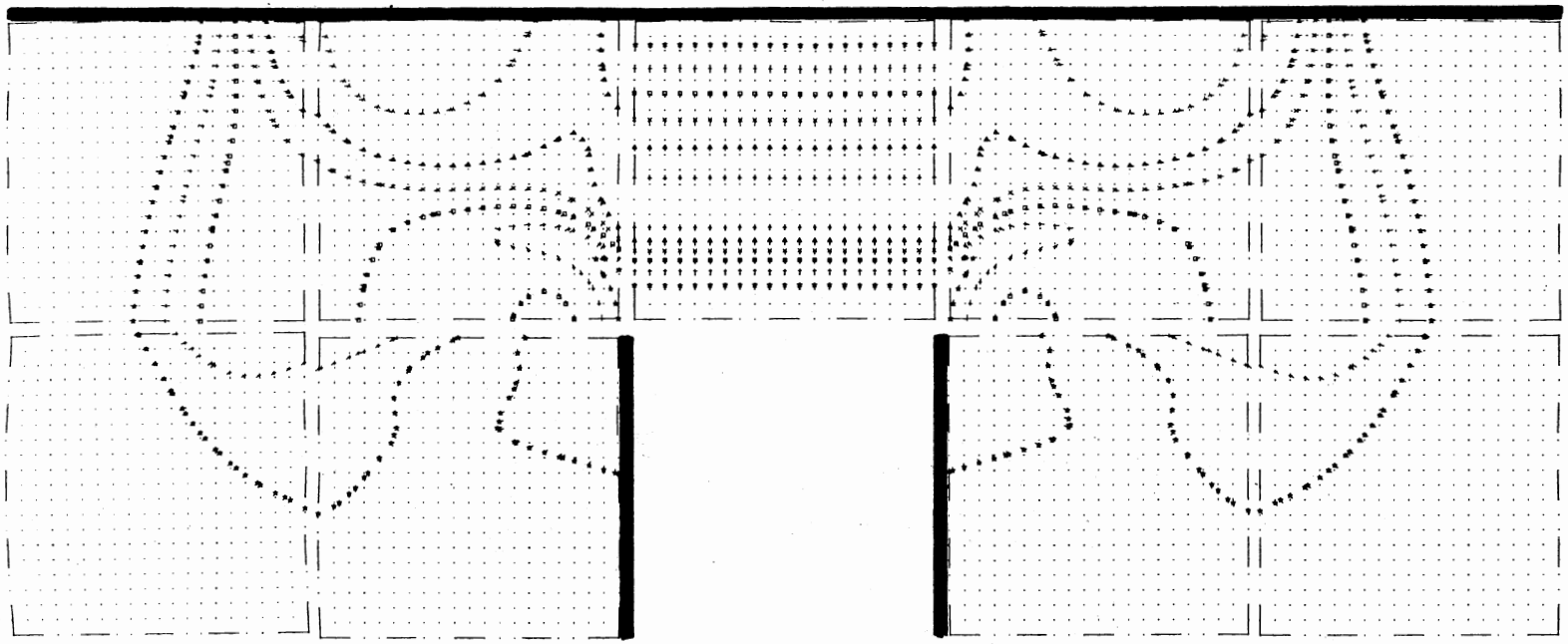


Figure B.10. Velocity Contours at 140 Time Frames; Dimensional Time = 20.13 Microsec

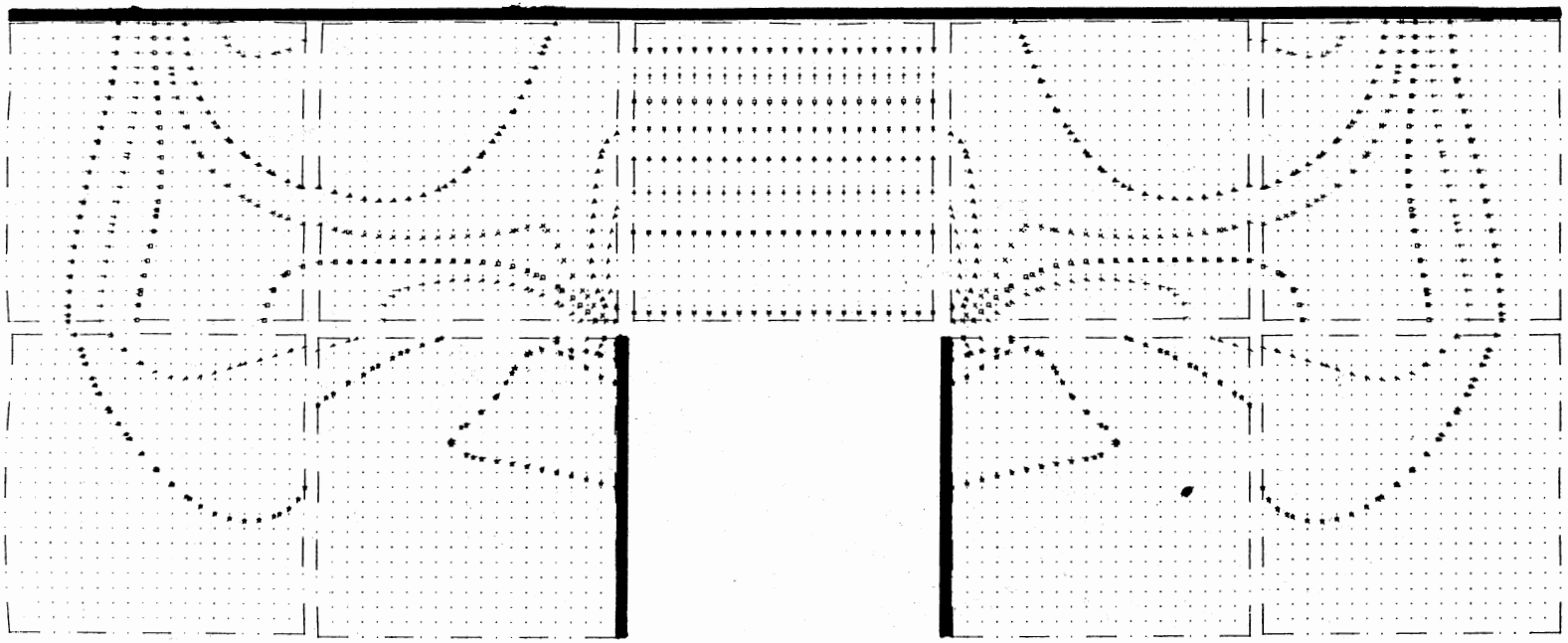


Figure B.11. Velocity Contours at 160 Time Frames; Dimensional Time = 23.08 Microsec

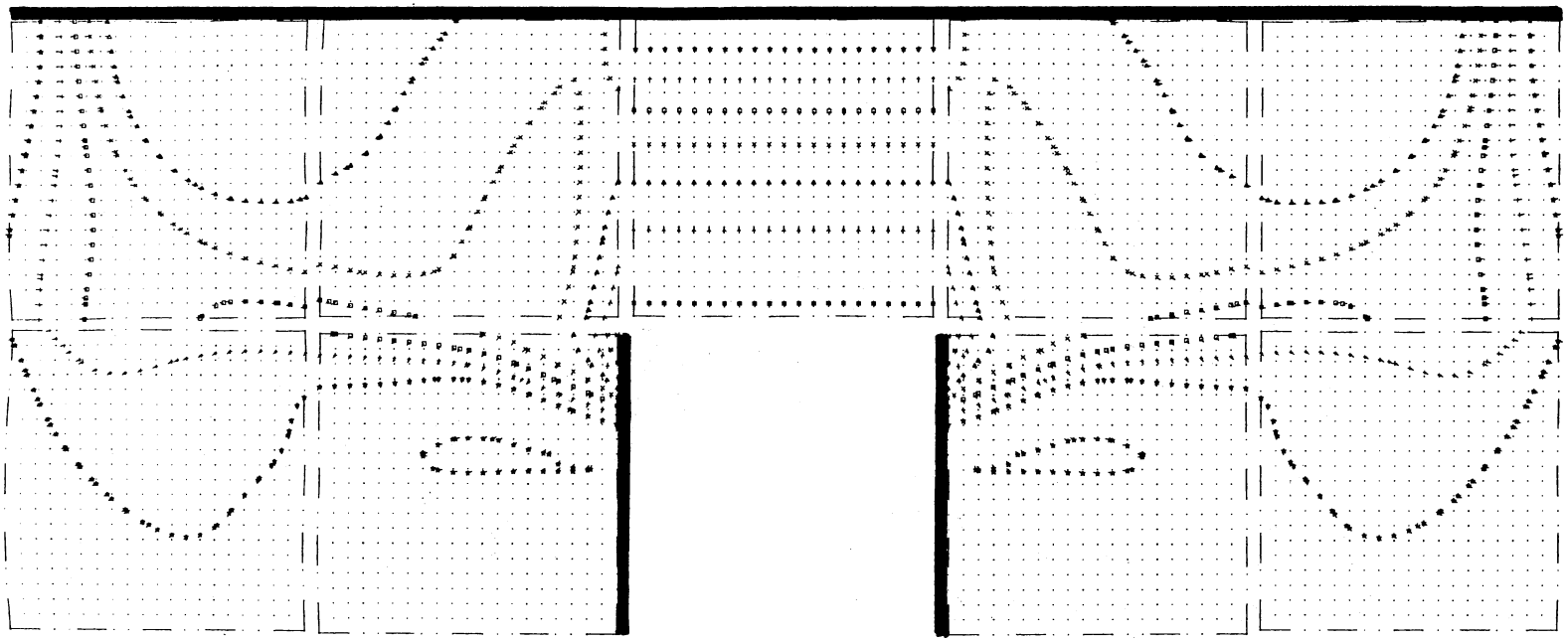


Figure B.12. Velocity Contours at 180 Time Frames; Dimensional Time = 25.97 Microsec

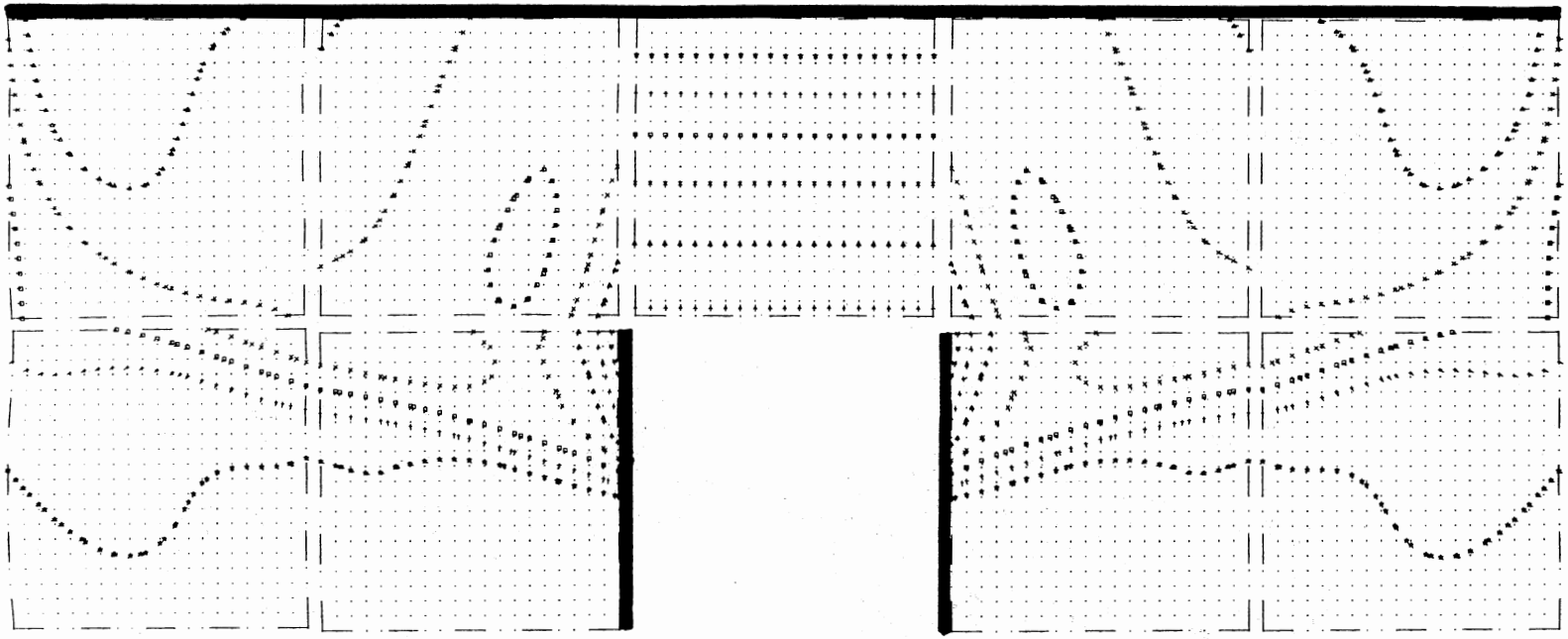


Figure B.13. Velocity Contours at 200 Time Frames; Dimensional Time = 28.98 Microsec

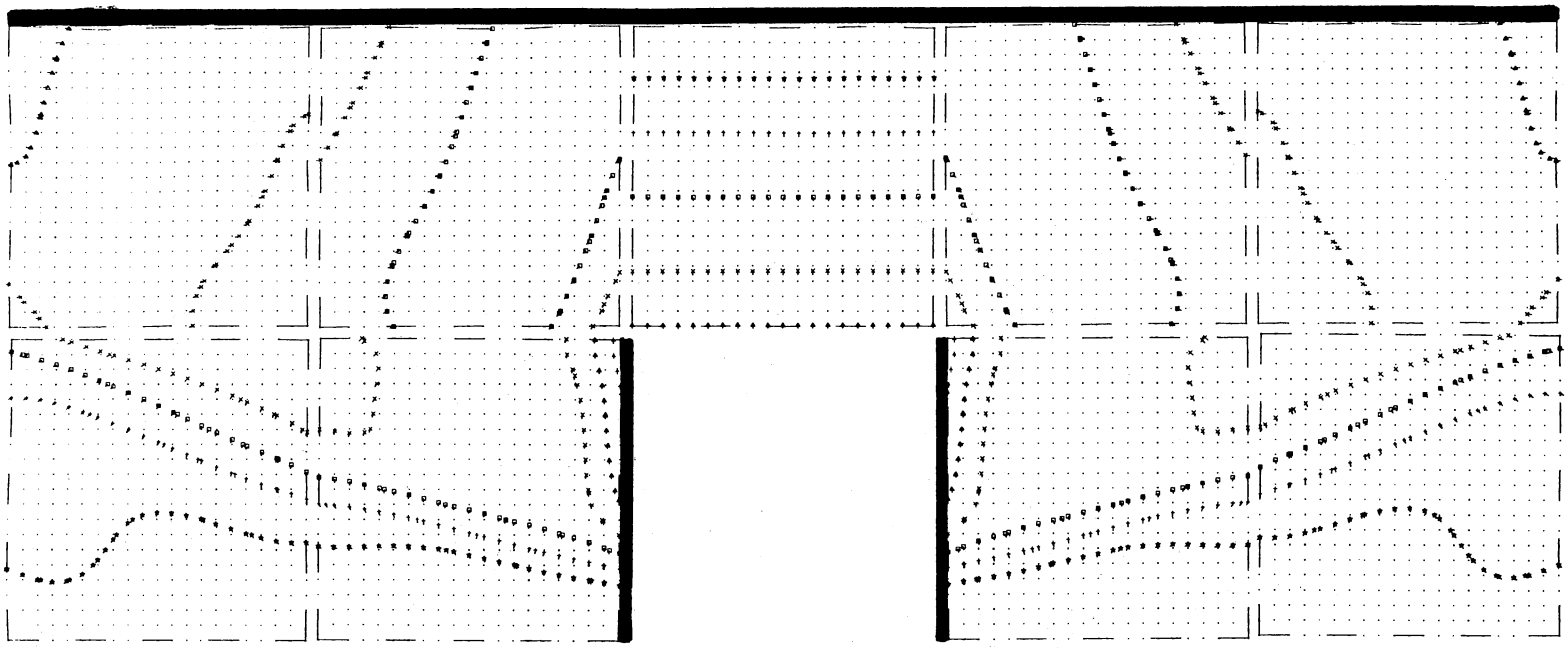


Figure B.14. Velocity Contours at 220 Time Frames; Dimensional Time = 32.18 Microsec

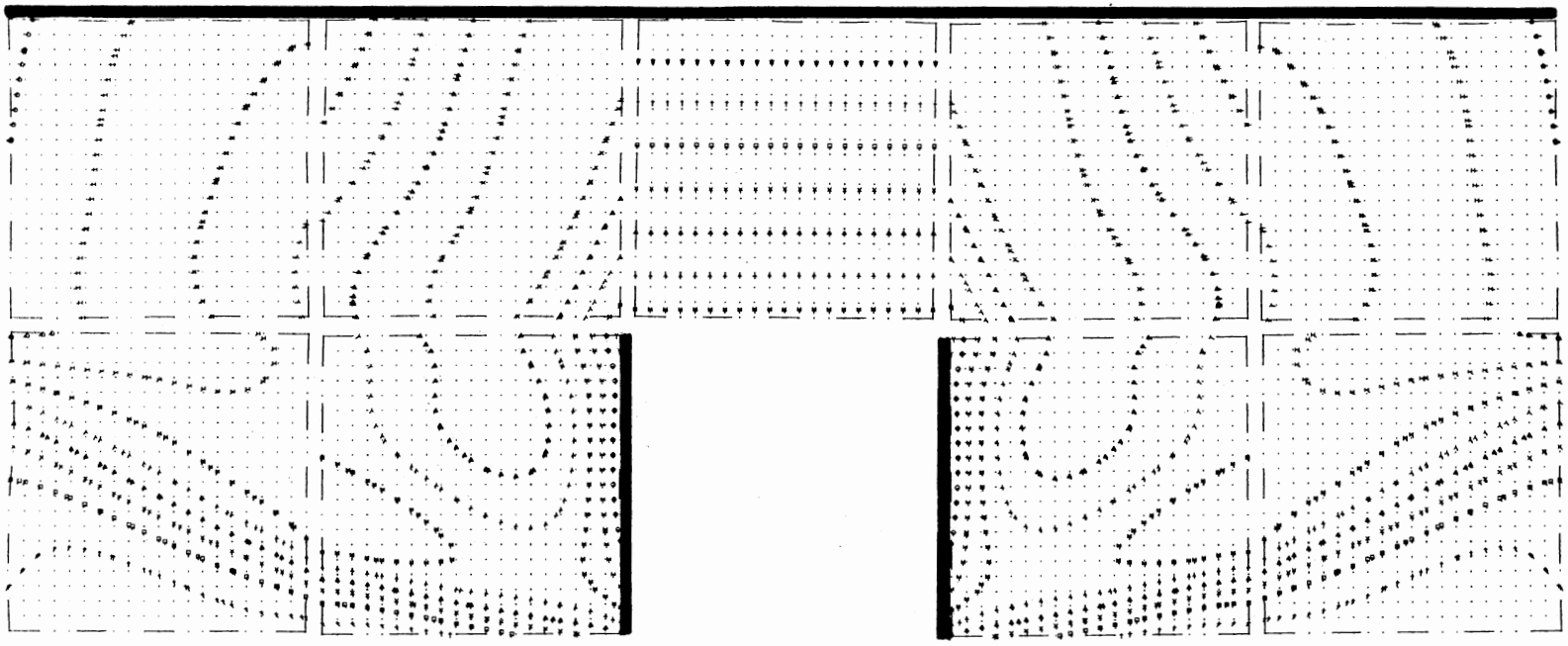


Figure B.15. Velocity Contours at 240 Time Frames; Dimensional Time = 35.63 Microsec

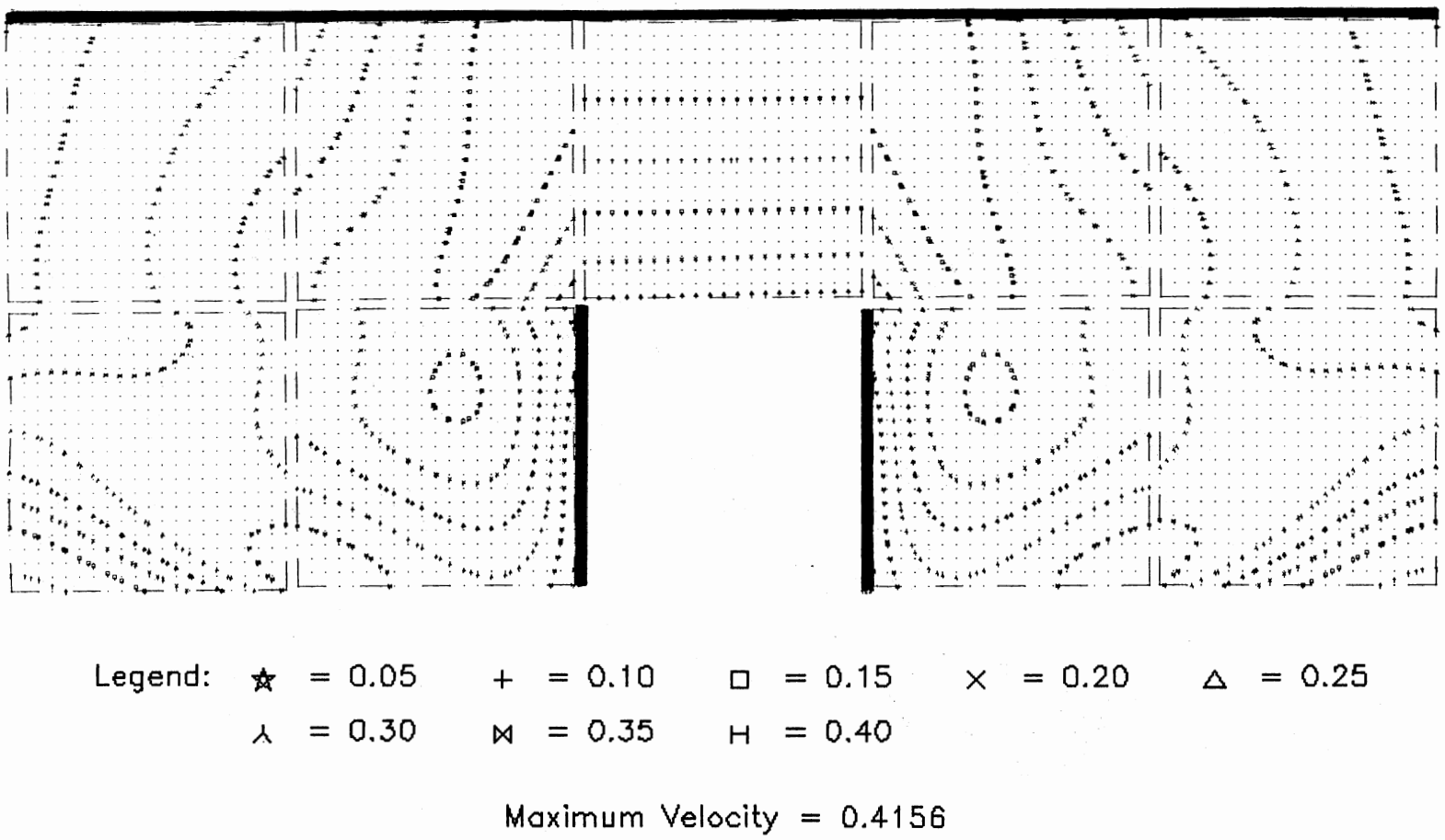


Figure B.16. Velocity Contours at 260 Time Frames; Dimensional Time = 39.29 Microsec

APPENDIX C

LINEAR MEMBRANE DEFLECTION FROM PULSE-TO-MEMBRANE MODELING

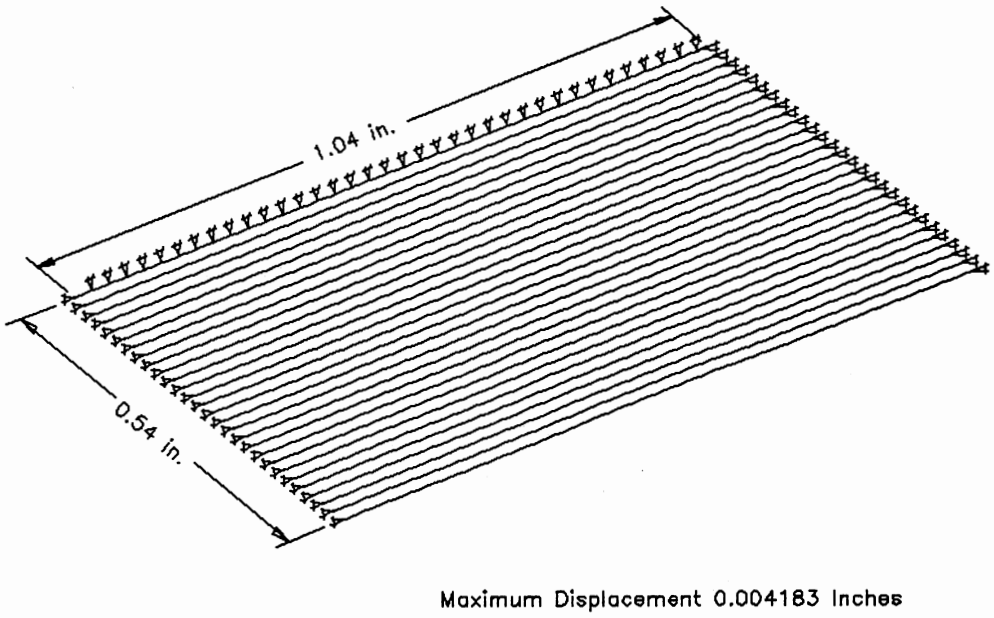


Figure C.1. Linear Membrane Displacement at 100 Time Frames;
Dimensional Time = 13.98 Microsec

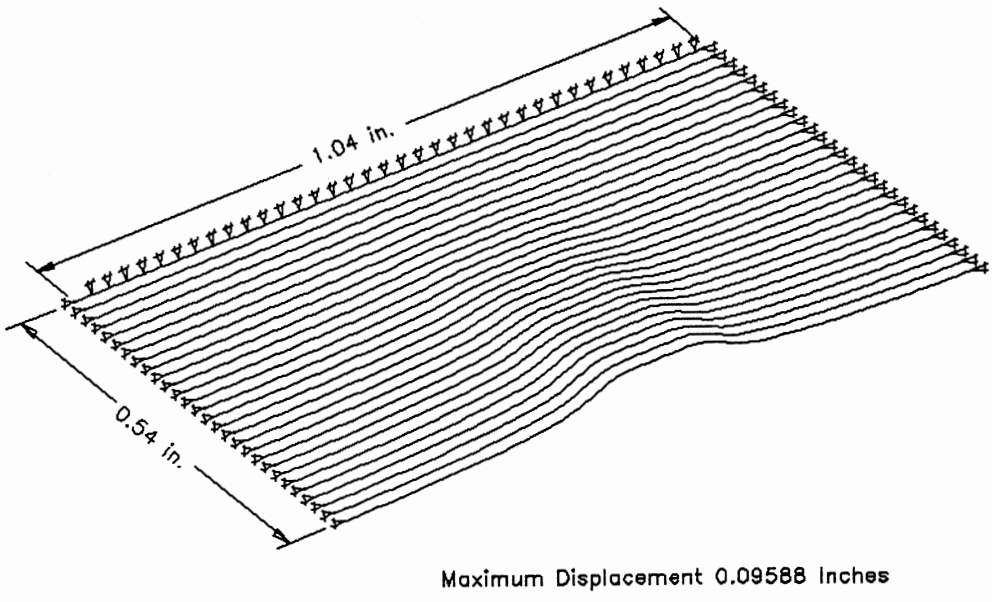


Figure C.2. Linear Membrane Displacement at 200 Time Frames;
Dimensional Time = 28.96 Microsec

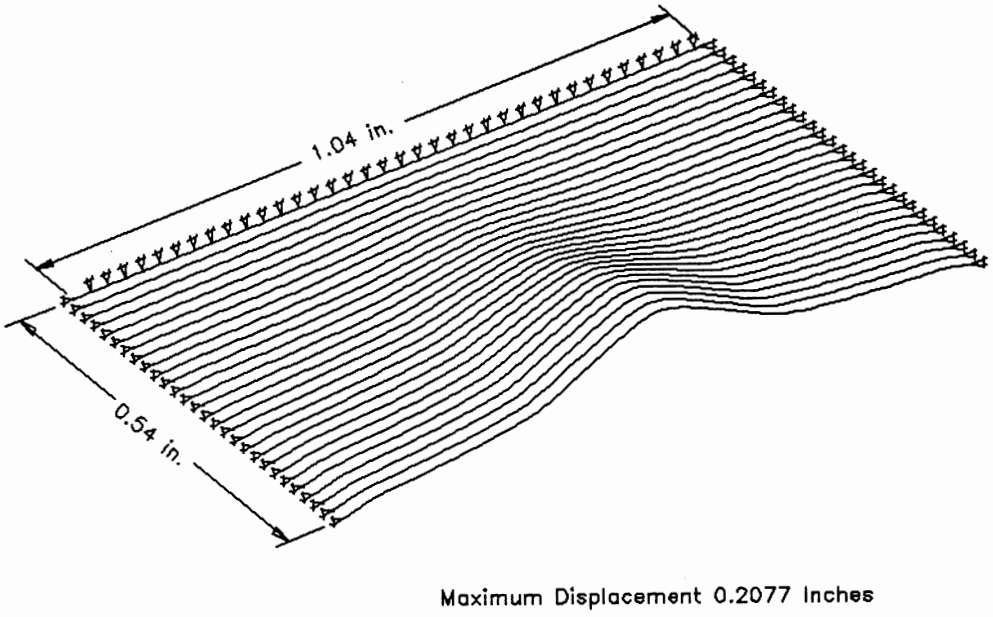


Figure C.3. Linear Membrane Displacement at 300 Time Frames;
Dimensional Time = 47.05 Microsec

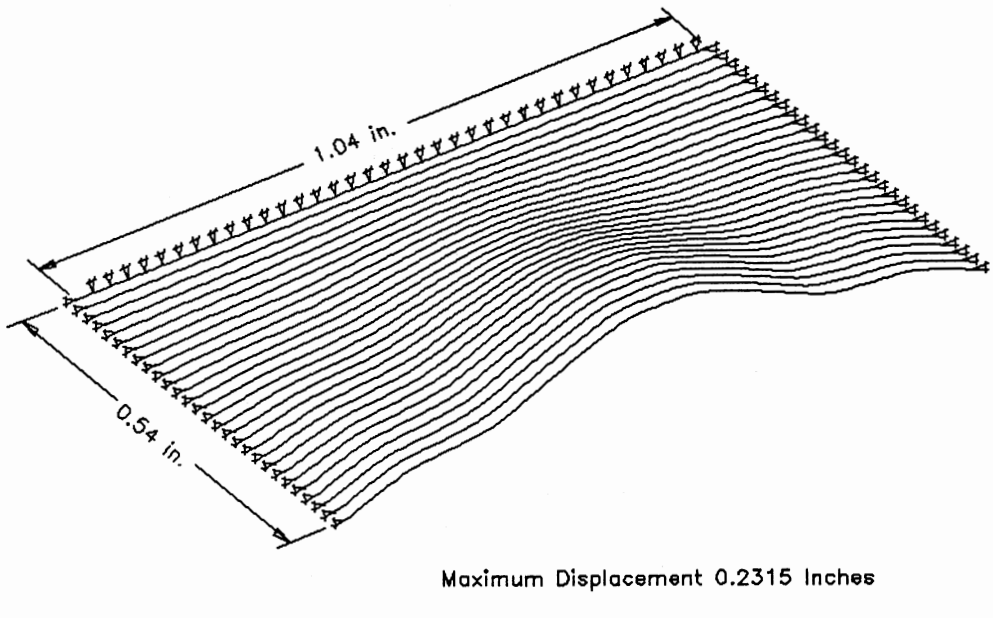


Figure C.4. Linear Membrane Displacement at 400 Time Frames;
Dimensional Time = 69.05 Microsec

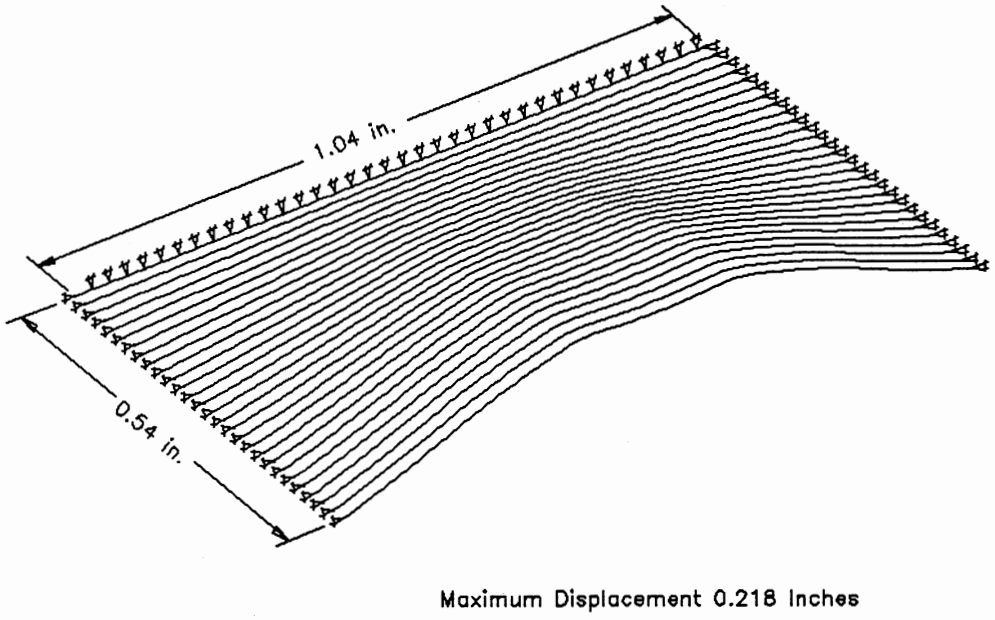


Figure C.5. Linear Membrane Displacement at 500 Time Frames;
Dimensional Time = 92.23 Microsec

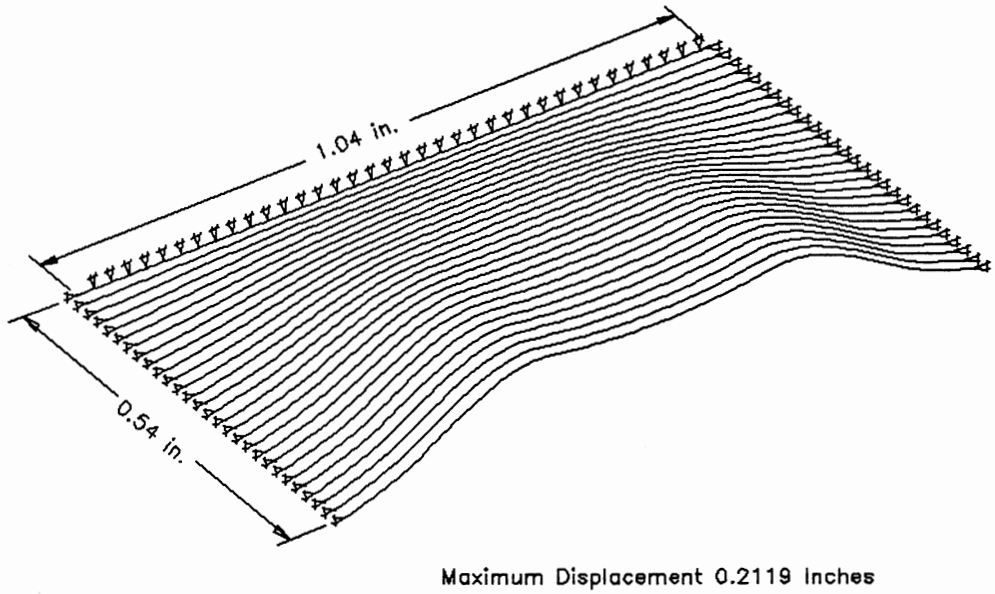


Figure C.6. Linear Membrane Displacement at 600 Time Frames;
Dimensional Time = 115.45 Microsec

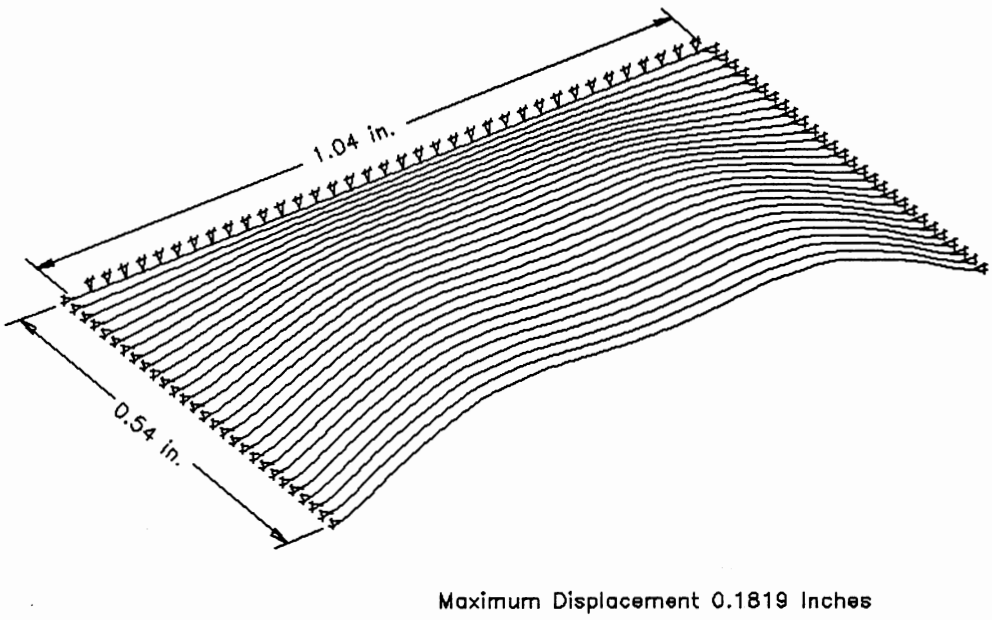


Figure C.7. Linear Membrane Displacement at 700 Time Frames;
Dimensional Time = 138.70 Microsec

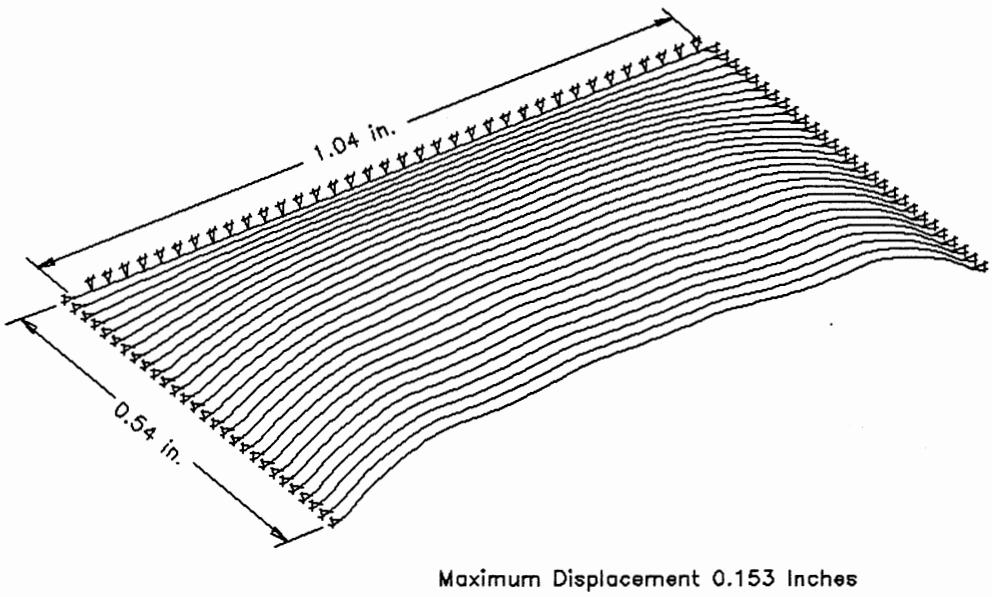


Figure C.8. Linear Membrane Displacement at 800 Time Frames;
Dimensional Time = 161.97 Microsec

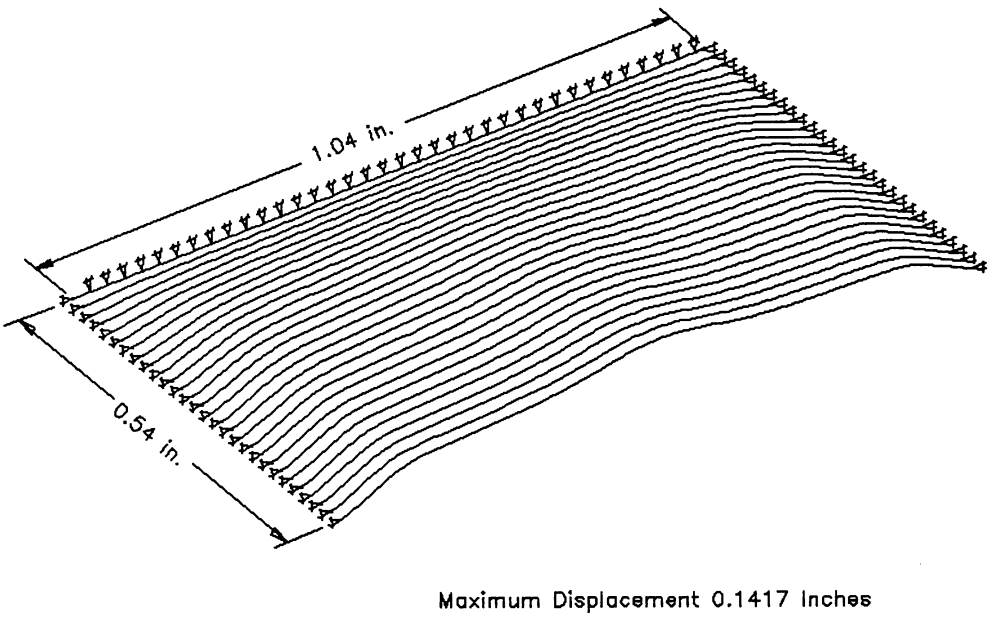


Figure C.9. Linear Membrane Displacement at 900 Time Frames;
Dimensional Time = 185.22 Microsec

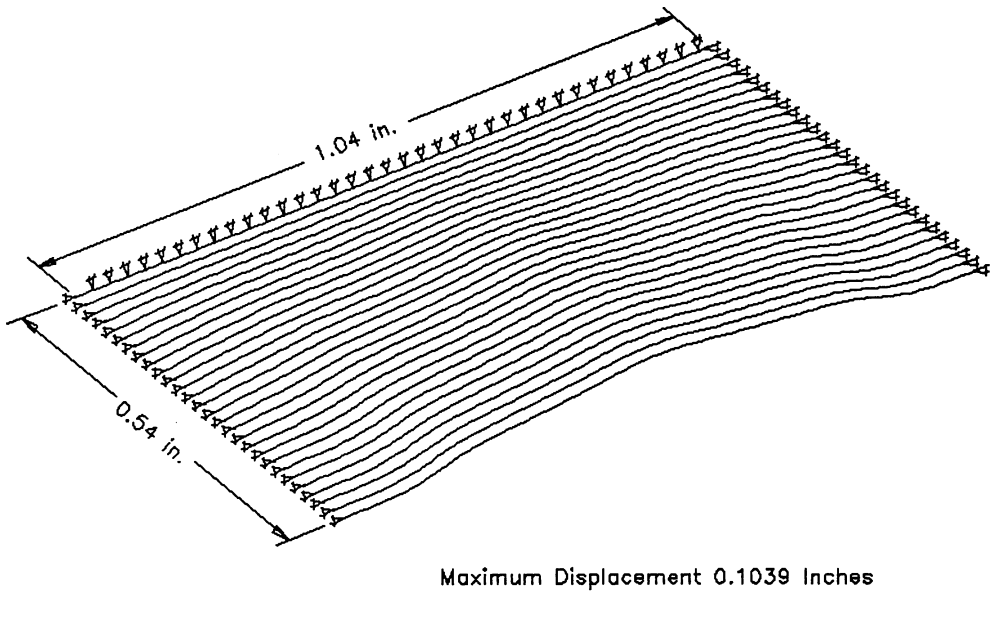


Figure C.10. Linear Membrane Displacement at 1000 Time Frames;
Dimensional Time = 208.44 Microsec

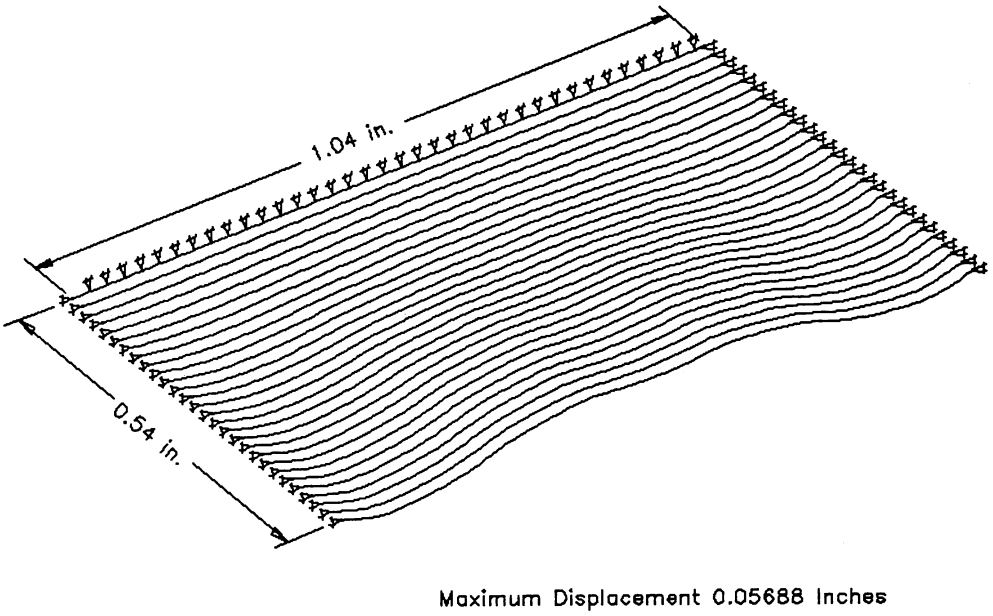


Figure C.11. Linear Membrane Displacement at 1100 Time Frames;
Dimensional Time = 231.60 Microsec

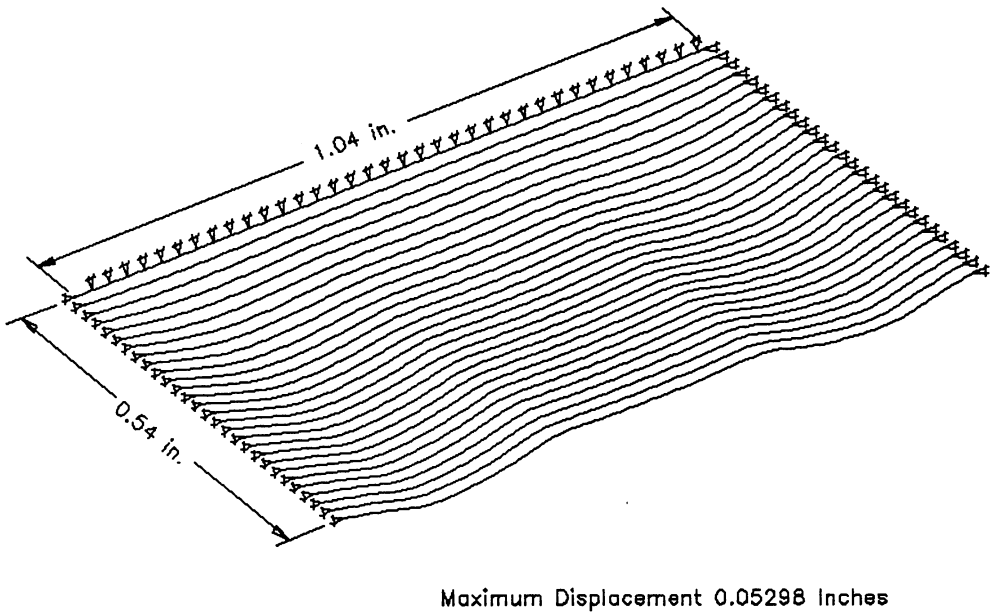


Figure C.12. Linear Membrane Displacement at 1200 Time Frames;
Dimensional Time = 254.73 Microsec

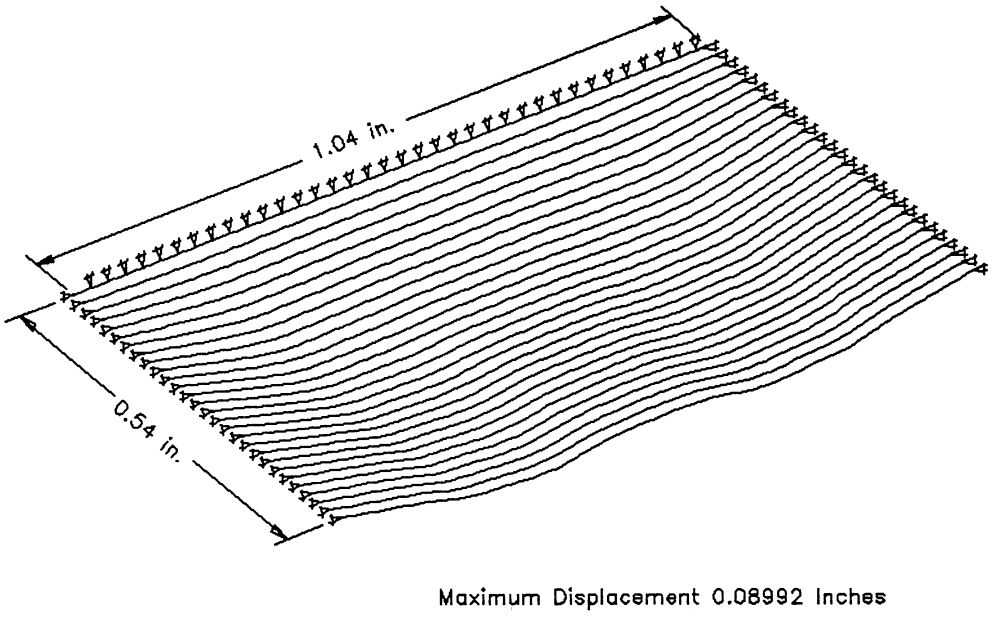


Figure C.13. Linear Membrane Displacement at 1300 Time Frames;
Dimensional Time = 277.84 Microsec

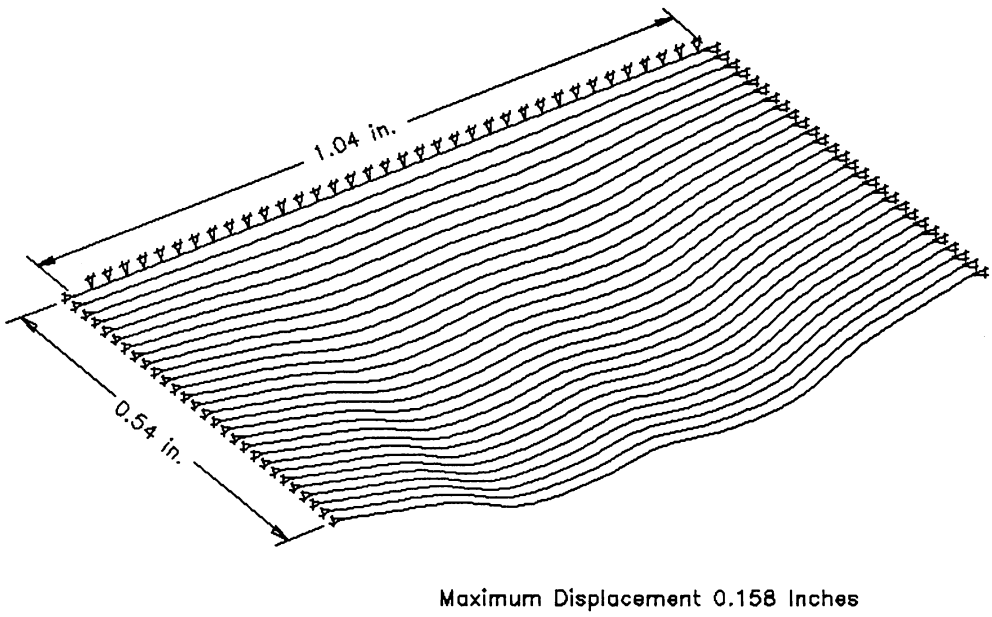


Figure C.14. Linear Membrane Displacement at 1400 Time Frames;
Dimensional Time = 300.93 Microsec

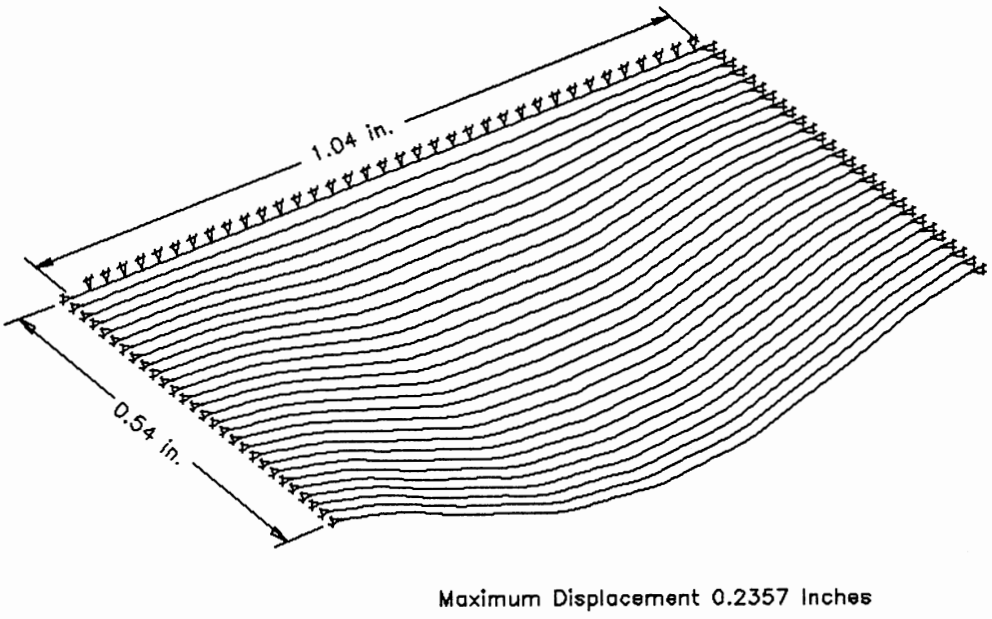


Figure C.15. Linear Membrane Displacement at 1500 Time Frames;
Dimensional Time = 324.02 Microsec

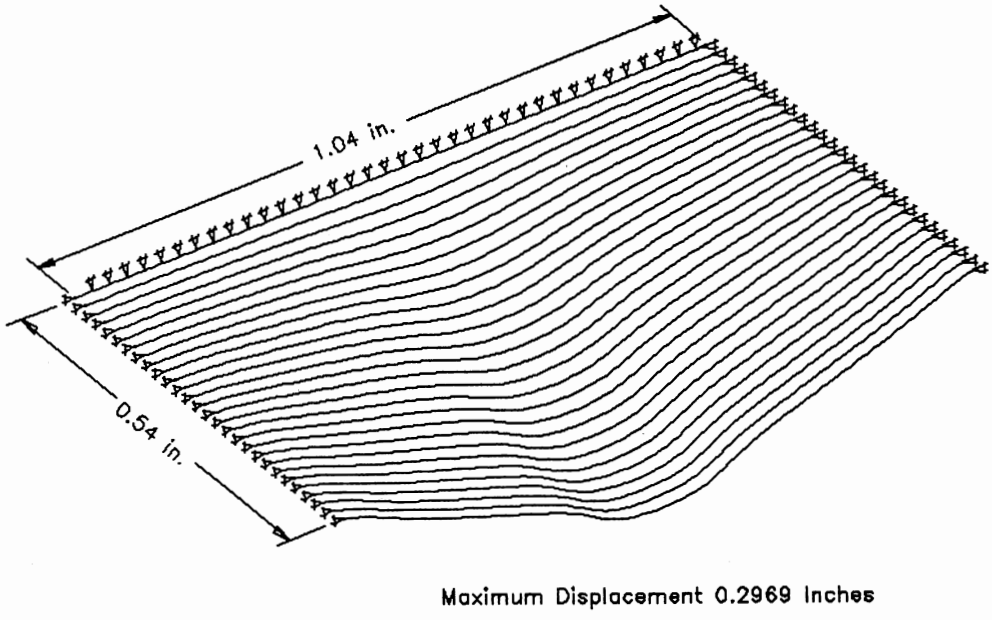


Figure C.16. Linear Membrane Displacement at 1600 Time Frames;
Dimensional Time = 347.10 Microsec

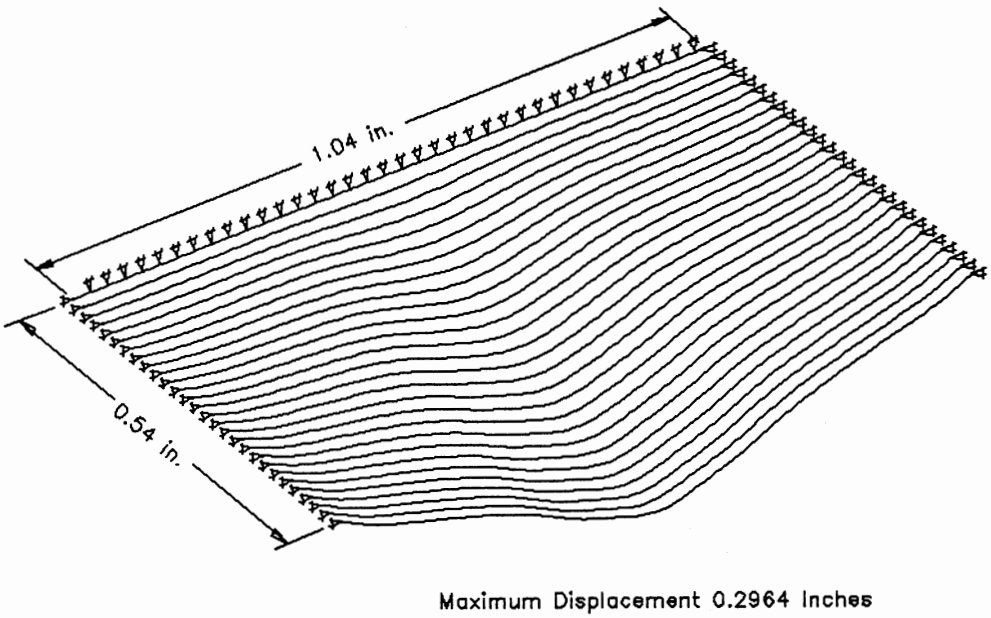


Figure C.17. Linear Membrane Displacement at 1700 Time Frames;
Dimensional Time = 370.17 Microsec

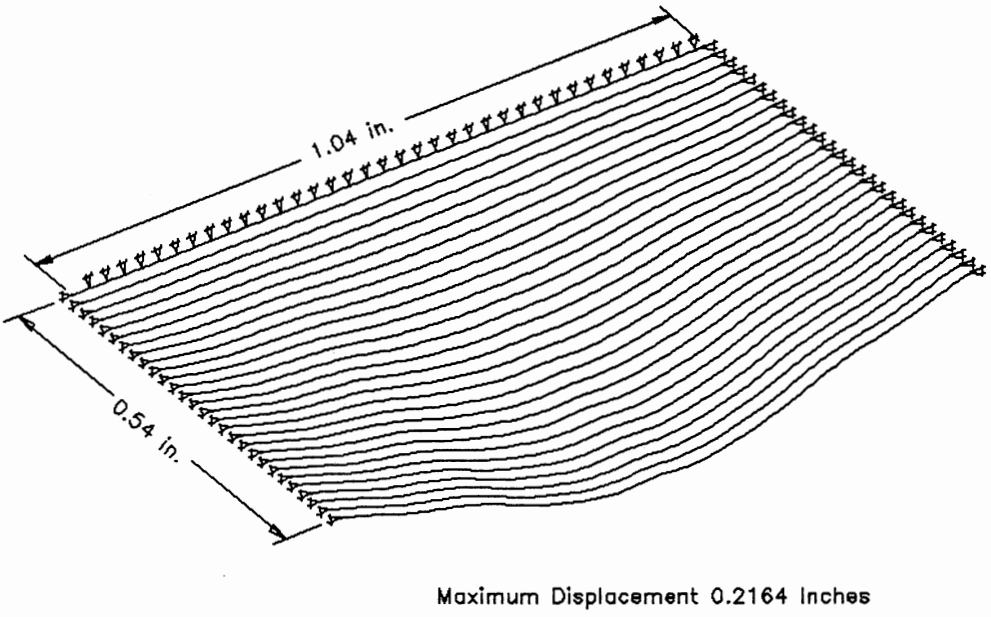


Figure C.18. Linear Membrane Displacement at 1800 Time Frames;
Dimensional Time = 393.23 Microsec

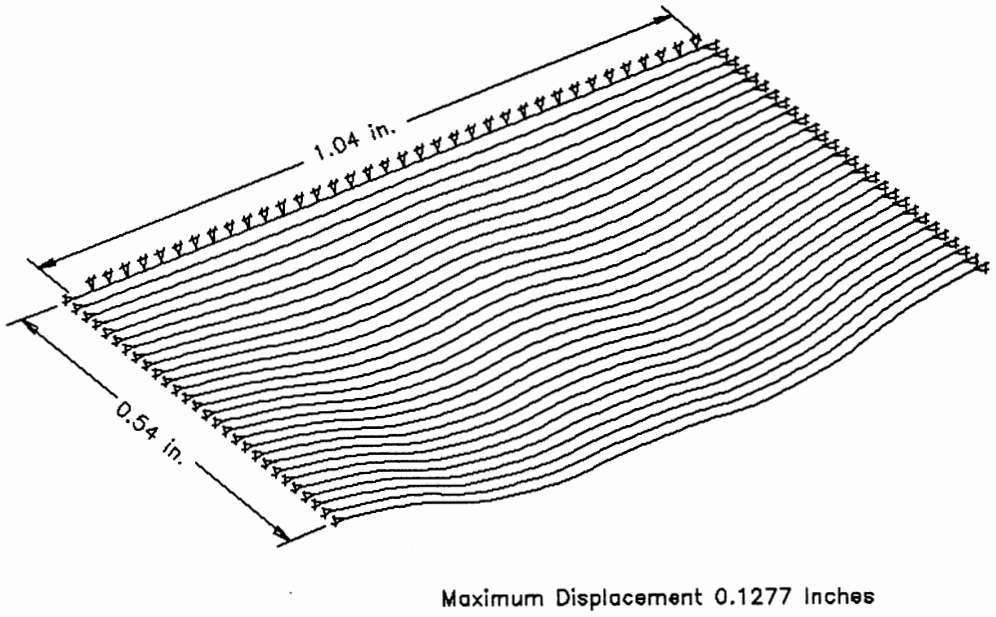


Figure C.19. Linear Membrane Displacement at 1900 Time Frames;
Dimensional Time = 416.25 Microsec

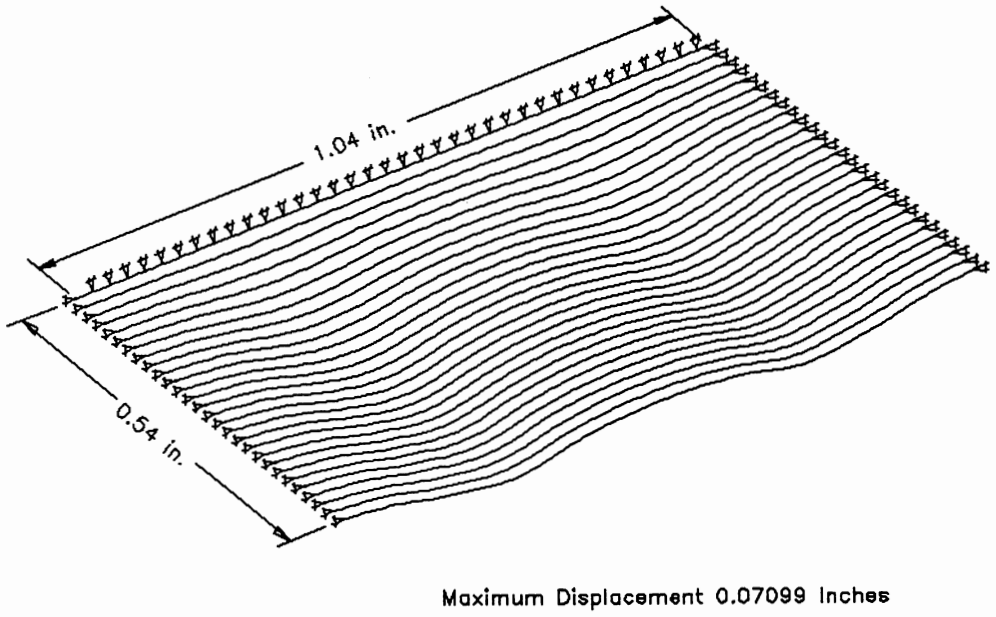


Figure C.20. Linear Membrane Displacement at 2000 Time Frames;
Dimensional Time = 439.19 Microsec

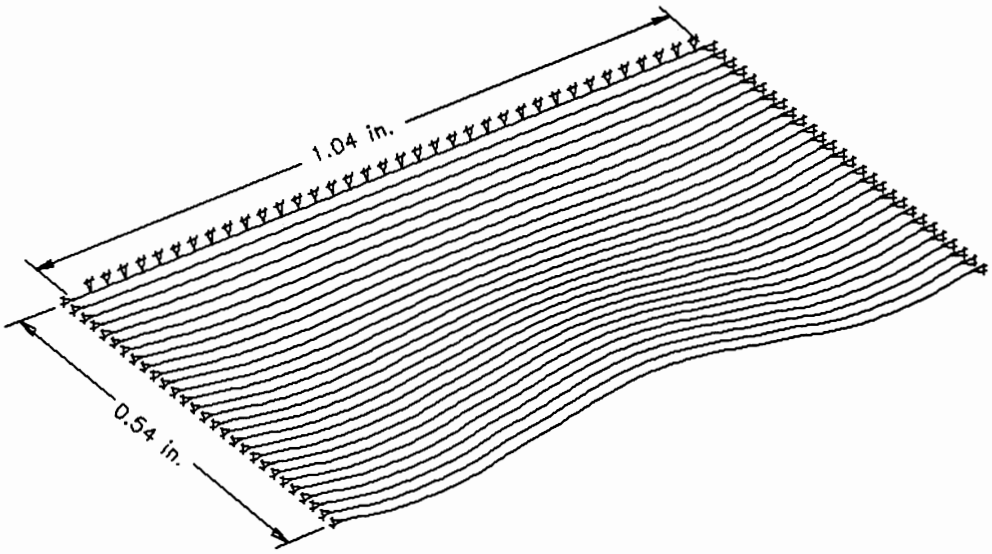


Figure C.21. Linear Membrane Displacement at 2100 Time Frames;
Dimensional Time = 462.04 Microsec

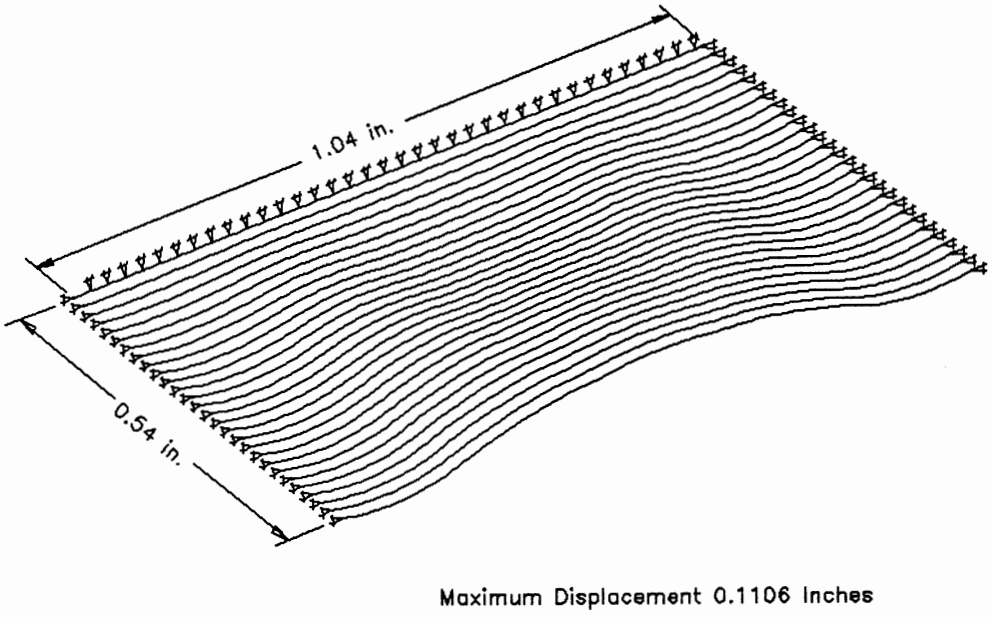


Figure C.22. Linear Membrane Displacement at 2200 Time Frames;
Dimensional Time = 484.78 Microsec

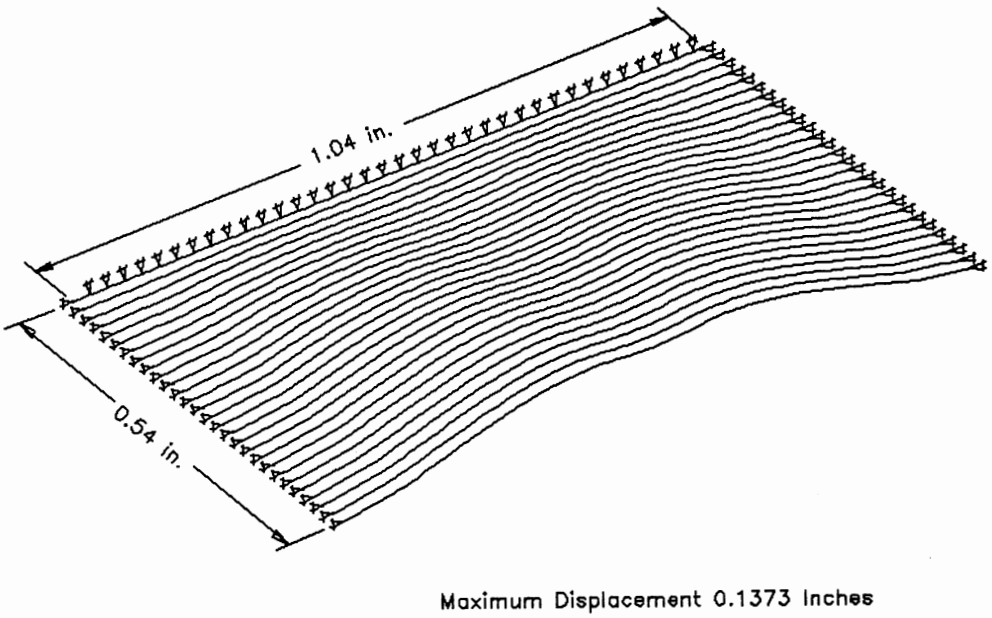


Figure C.23. Linear Membrane Displacement at 2300 Time Frames;
Dimensional Time = 507.38 Microsec

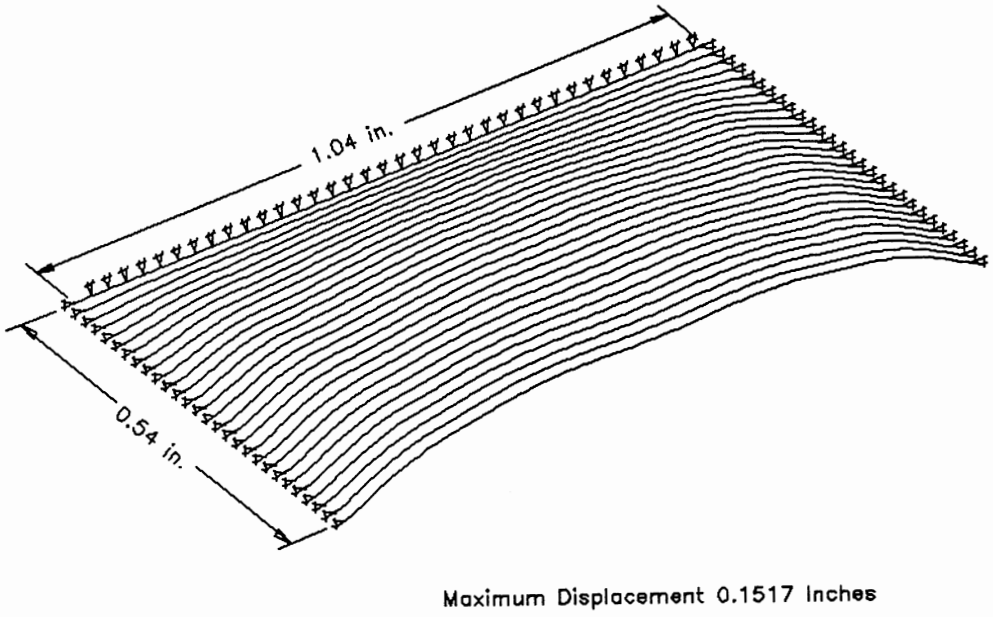


Figure C.24. Linear Membrane Displacement at 2400 Time Frames;
Dimensional Time = 529.82 Microsec

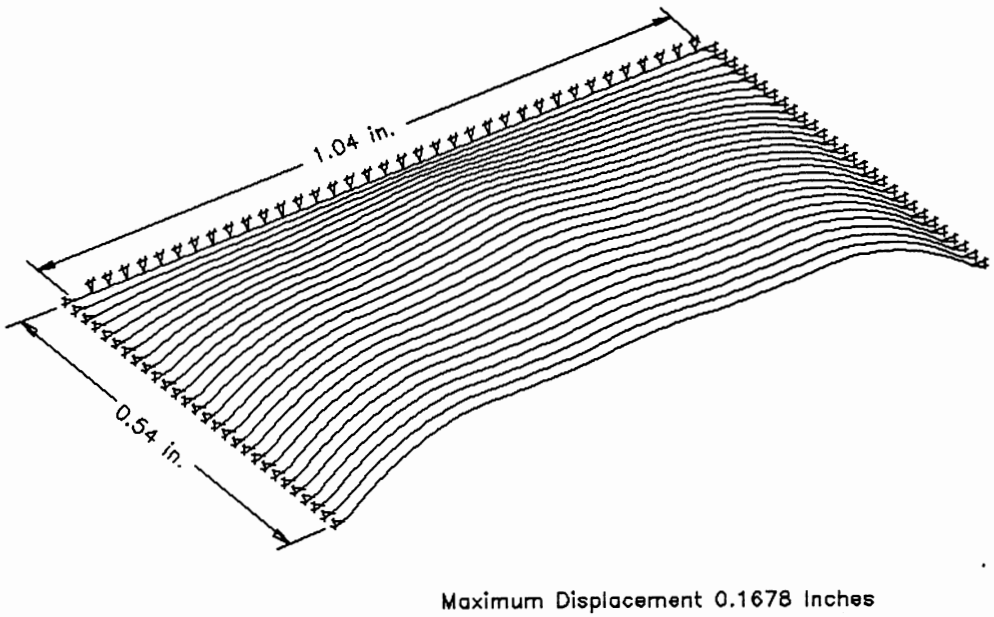


Figure C.25. Linear Membrane Displacement at 2500 Time Frames;
Dimensional Time = 552.08 Microsec

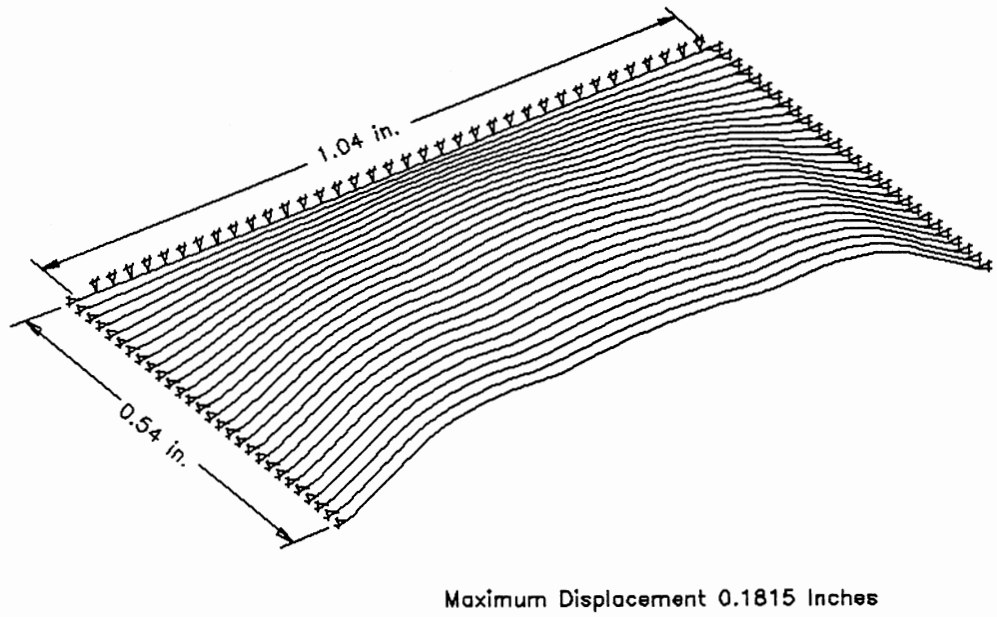


Figure C.26. Linear Membrane Displacement at 2600 Time Frames;
Dimensional Time = 574.14 Microsec

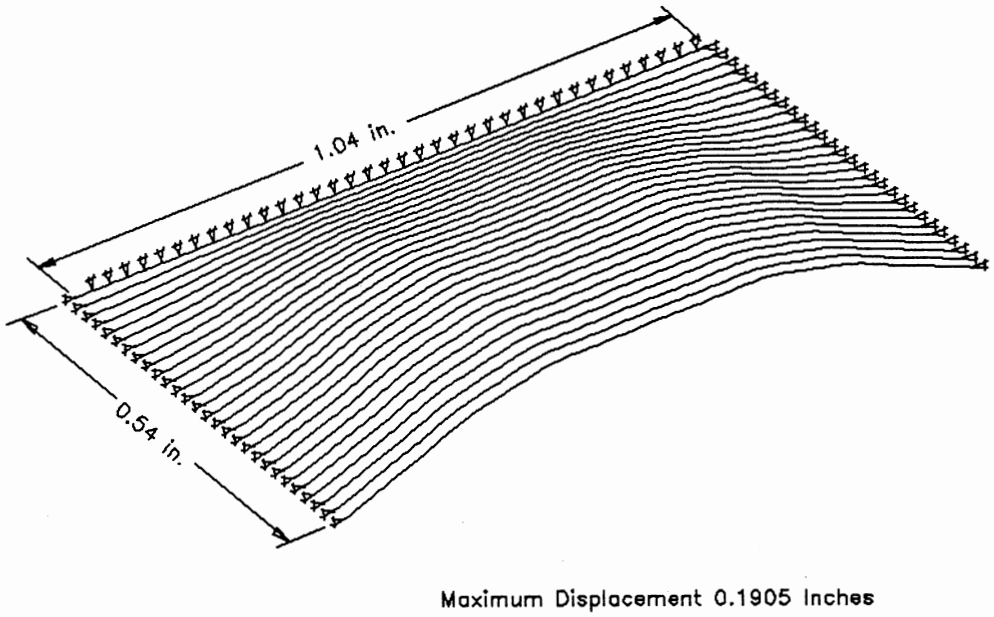


Figure C.27. Linear Membrane Displacement at 2700 Time Frames;
Dimensional Time = 595.96 Microsec

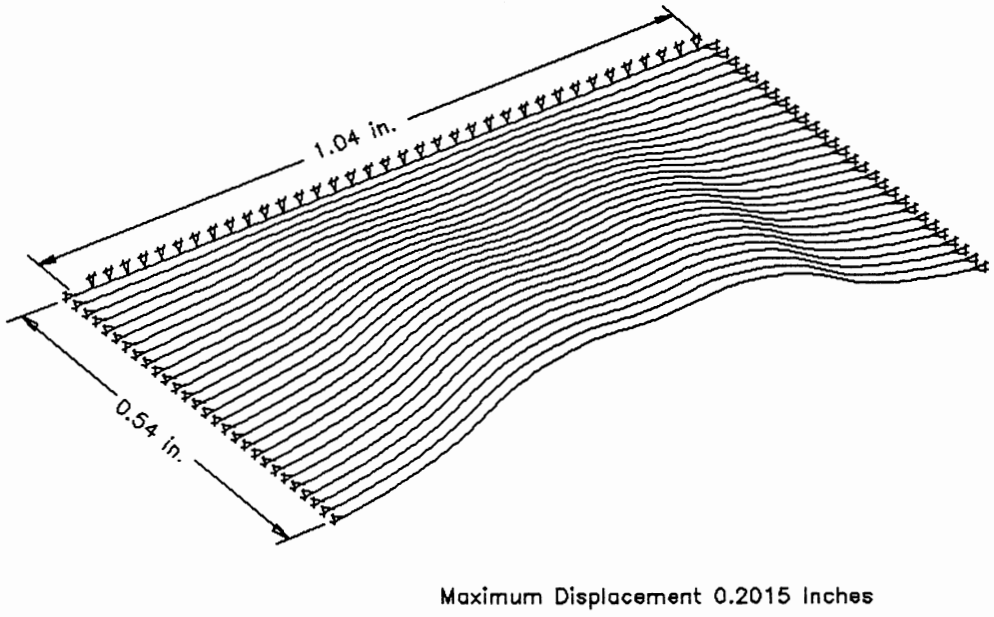


Figure C.28. Linear Membrane Displacement at 2800 Time Frames;
Dimensional Time = 617.54 Microsec

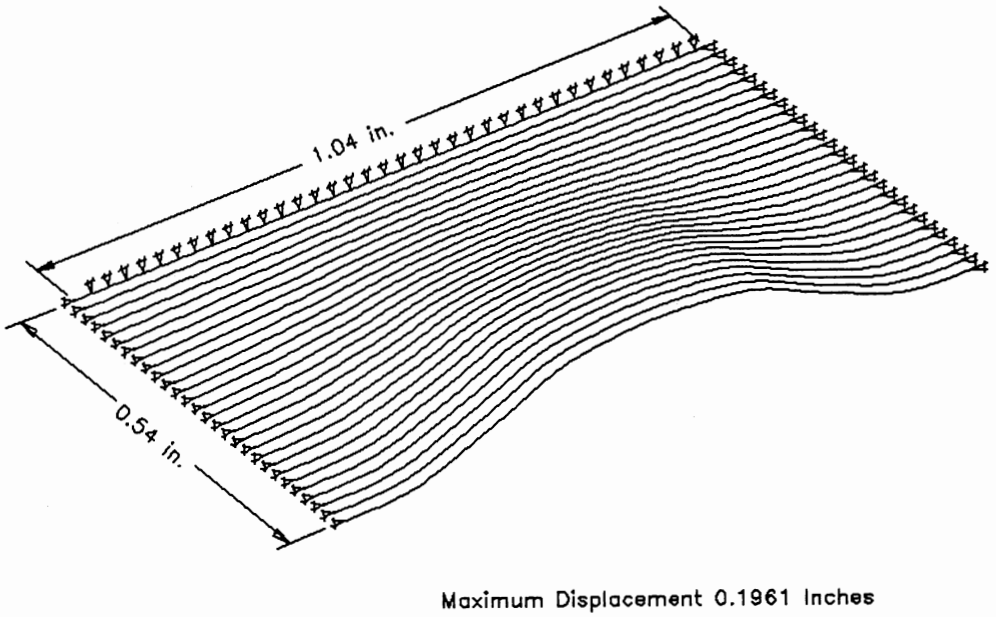


Figure C.29. Linear Membrane Displacement at 2900 Time Frames;
Dimensional Time = 638.84 Microsec

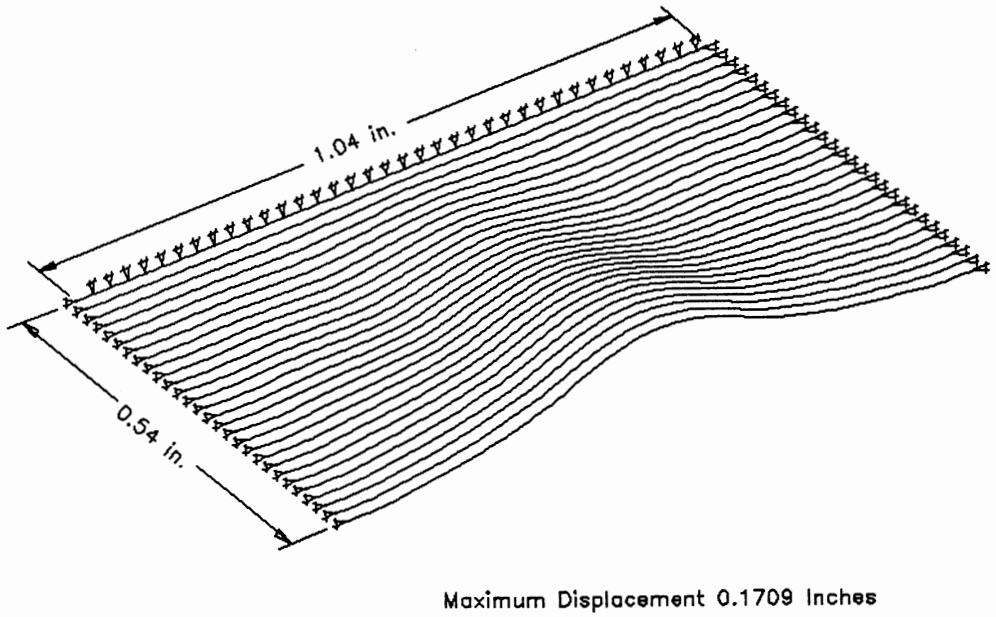


Figure C.30. Linear Membrane Displacement at 3000 Time Frames;
Dimensional Time = 659.85 Microsec

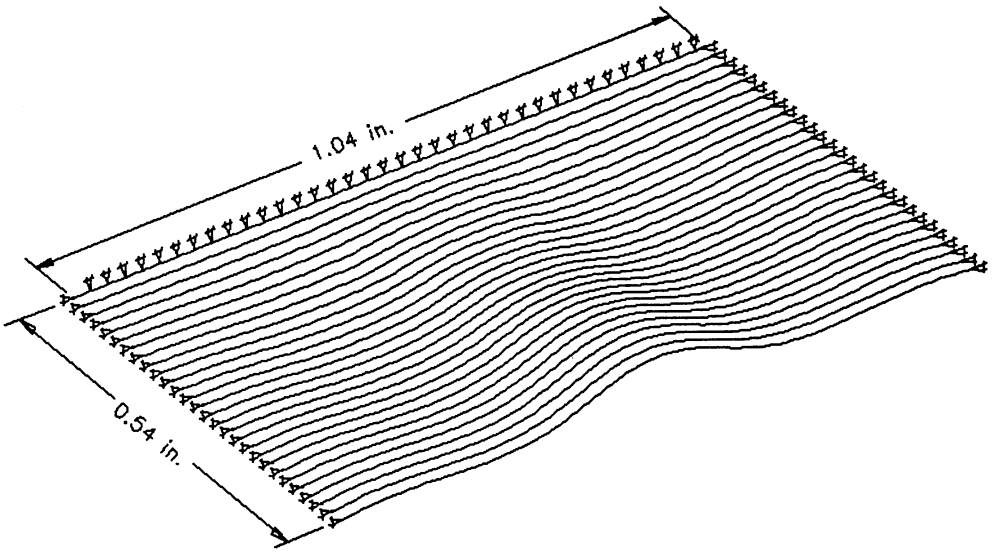


Figure C.31. Linear Membrane Displacement at 3100 Time Frames;
Dimensional Time = 680.54 Microsec

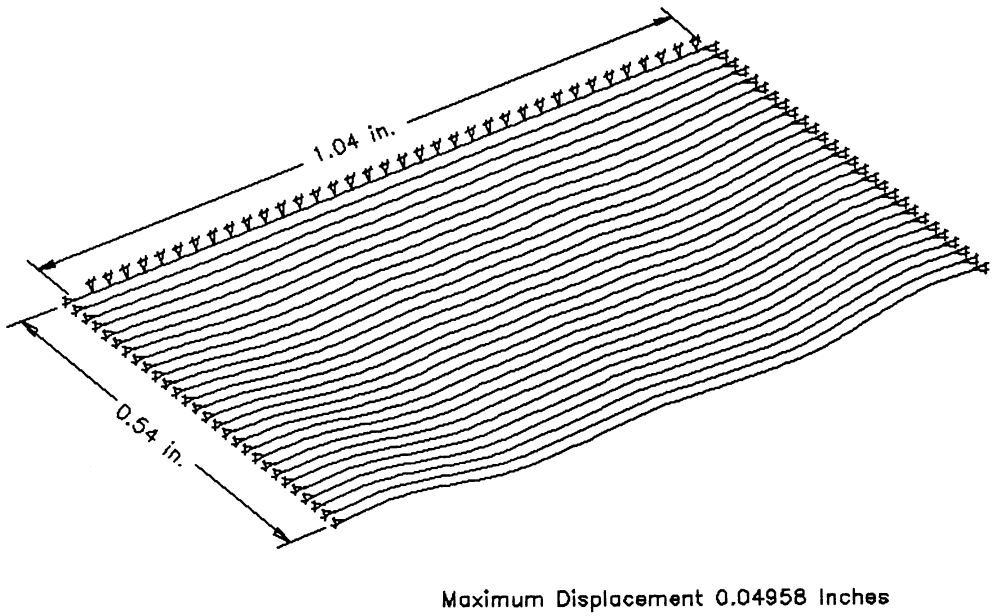


Figure C.32. Linear Membrane Displacement at 3200 Time Frames;
Dimensional Time = 700.91 Microsec

APPENDIX D

LINEAR PLATE DEFLECTION FROM PULSE-TO-PLATE MODELING

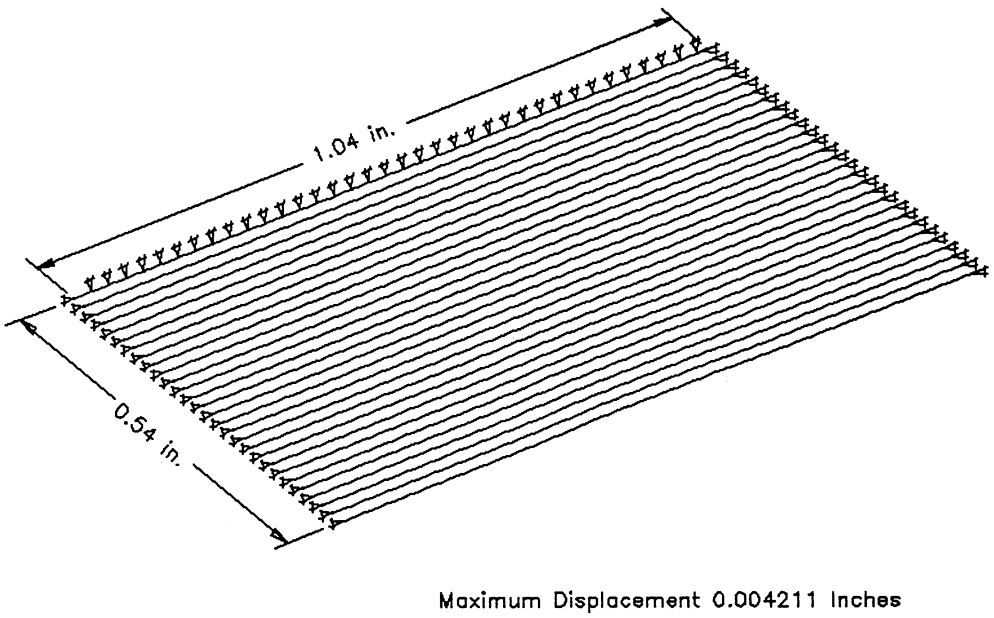


Figure D.1. Linear Plate Displacement at 100 Time Frames;
Dimensional Time = 13.98 Microsec

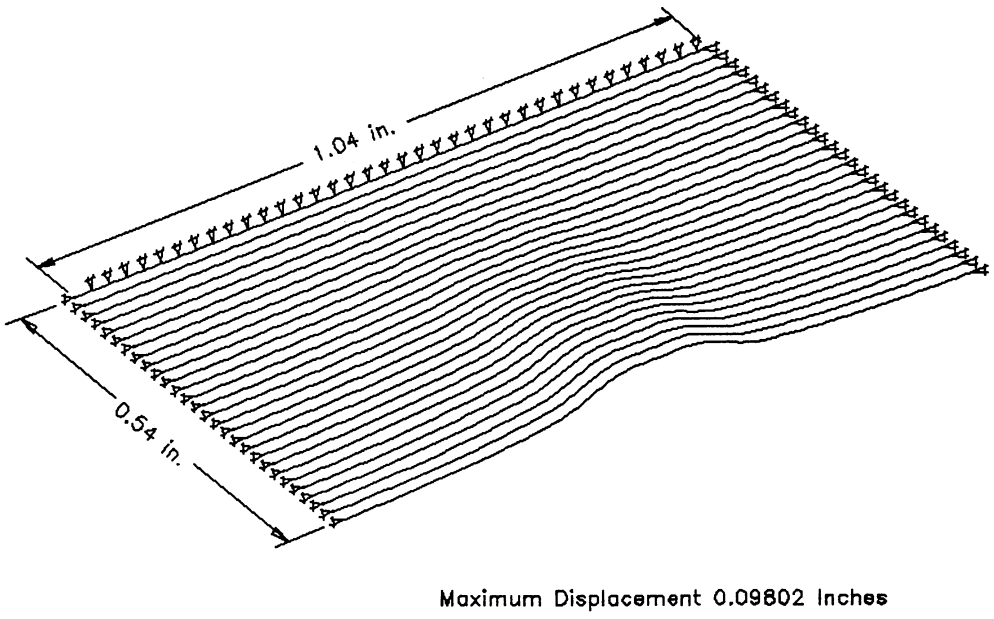


Figure D.2. Linear Plate Displacement at 200 Time Frames;
Dimensional Time = 28.96 Microsec

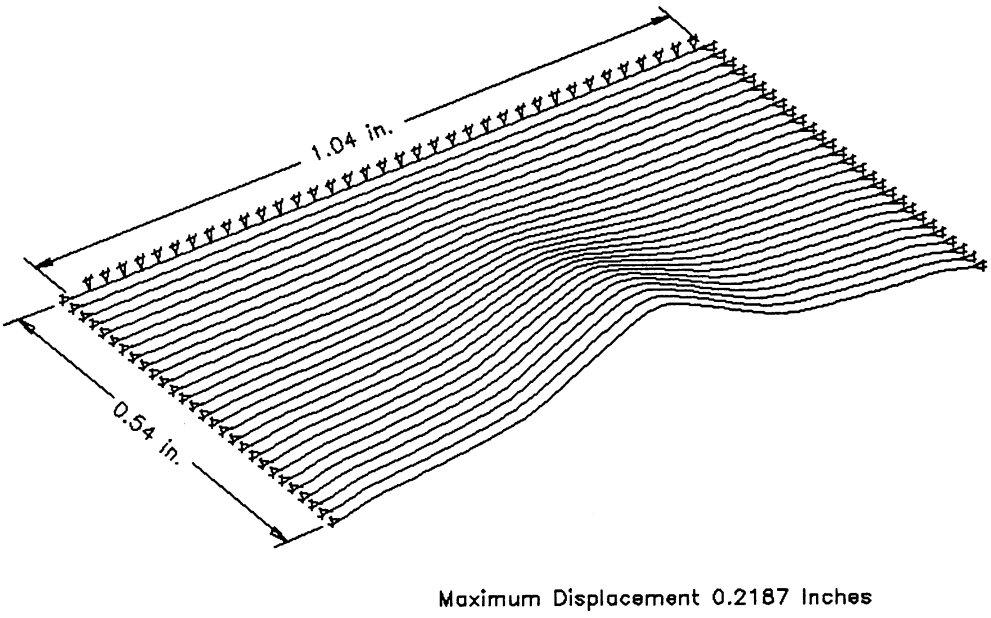


Figure D.3. Linear Plate Displacement at 300 Time Frames;
Dimensional Time = 47.05 Microsec

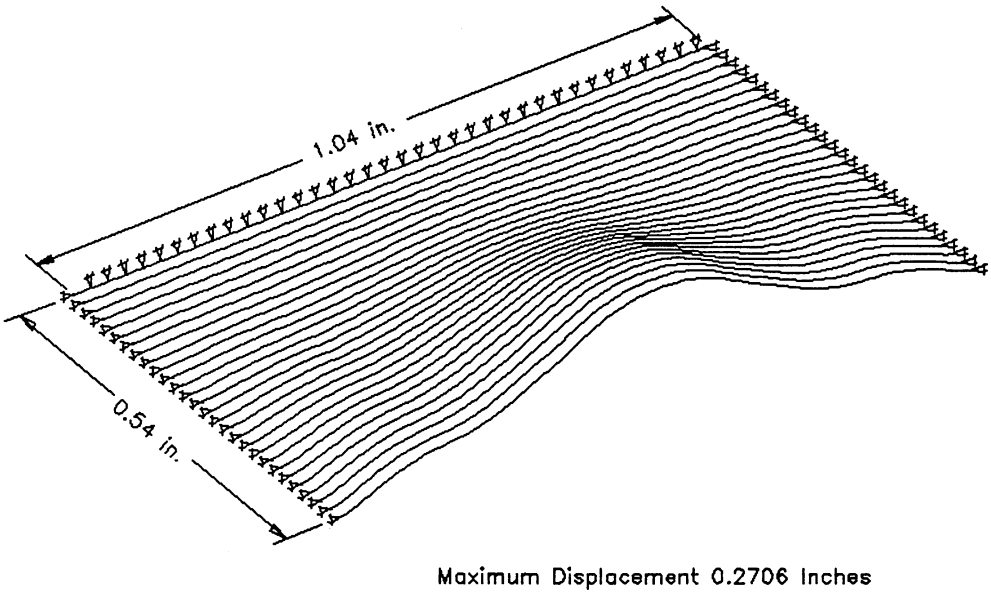


Figure D.4. Linear Plate Displacement at 400 Time Frames;
Dimensional Time = 69.05 Microsec

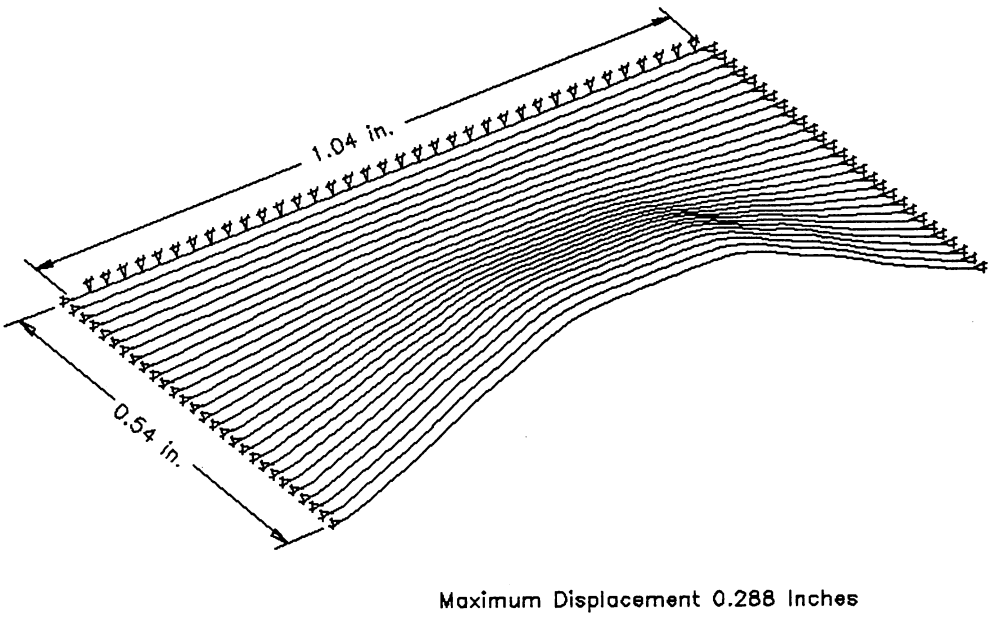


Figure D.5. Linear Plate Displacement at 500 Time Frames;
Dimensional Time = 92.23 Microsec

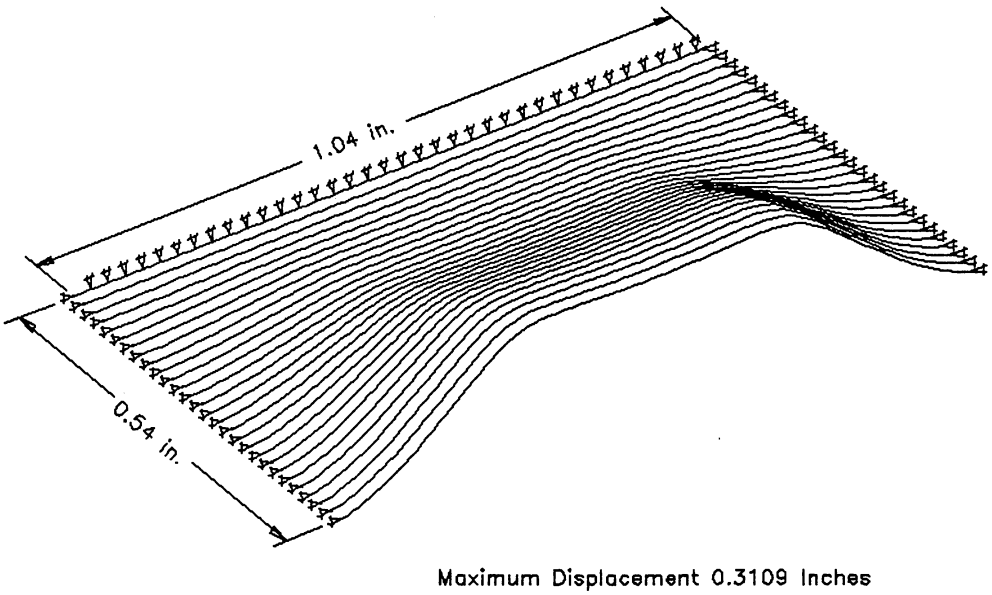


Figure D.6. Linear Plate Displacement at 600 Time Frames;
Dimensional Time = 115.45 Microsec

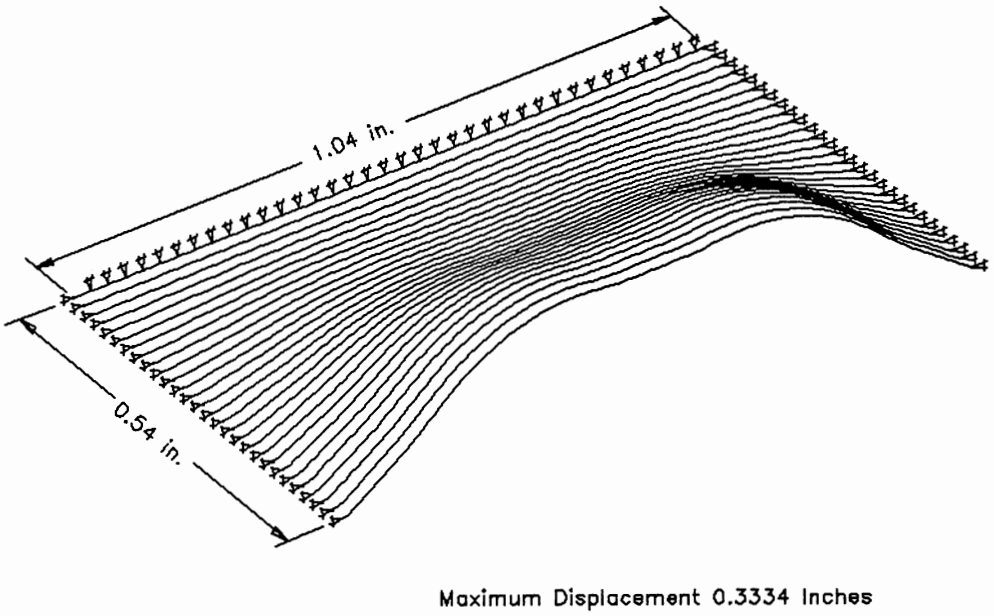


Figure D.7. Linear Plate Displacement at 700 Time Frames;
Dimensional Time = 138.70 Microsec

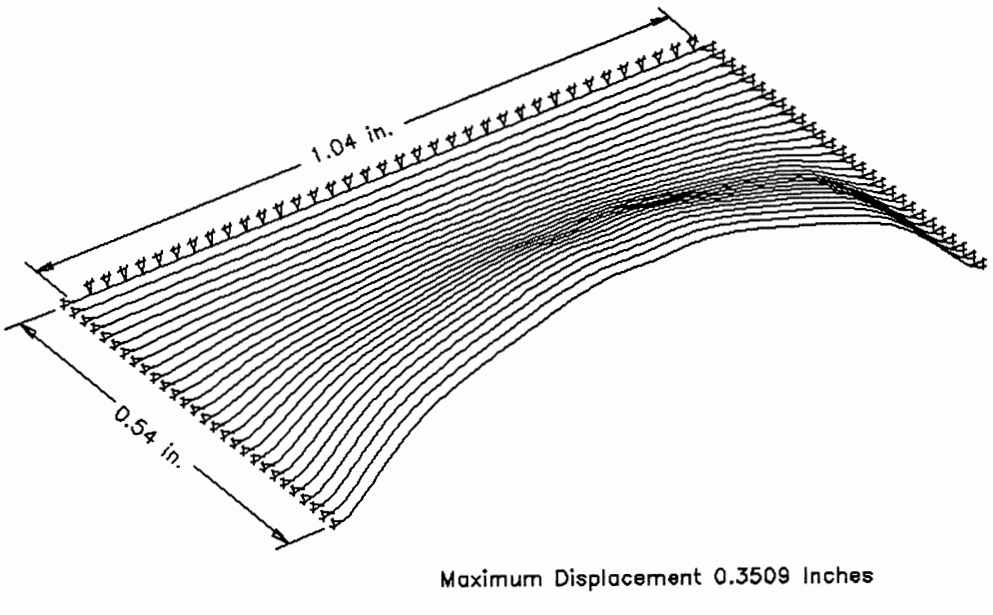


Figure D.8. Linear Plate Displacement at 800 Time Frames;
Dimensional Time = 161.97 Microsec

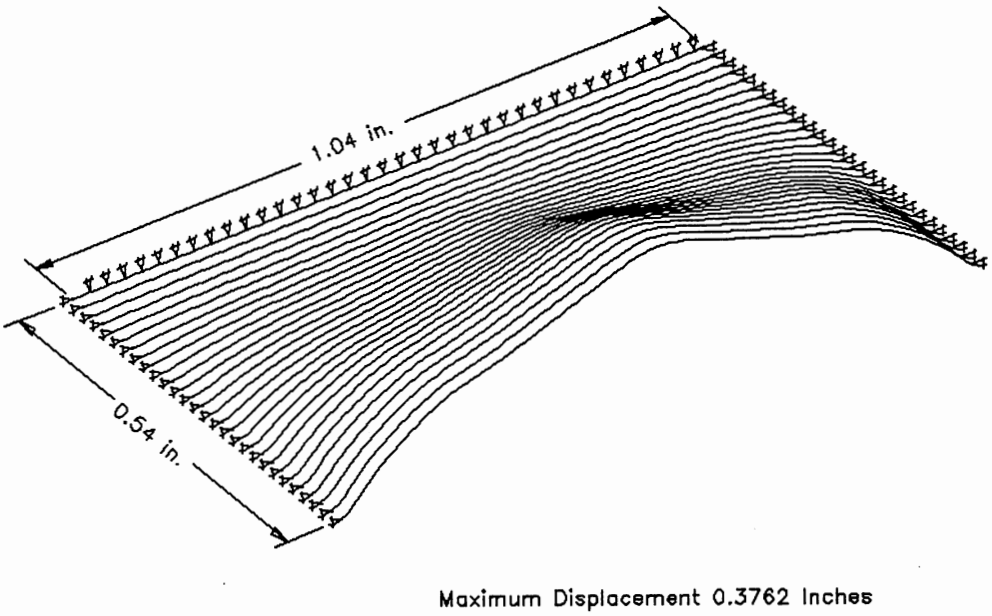


Figure D.9. Linear Plate Displacement at 900 Time Frames;
Dimensional Time = 185.22 Microsec

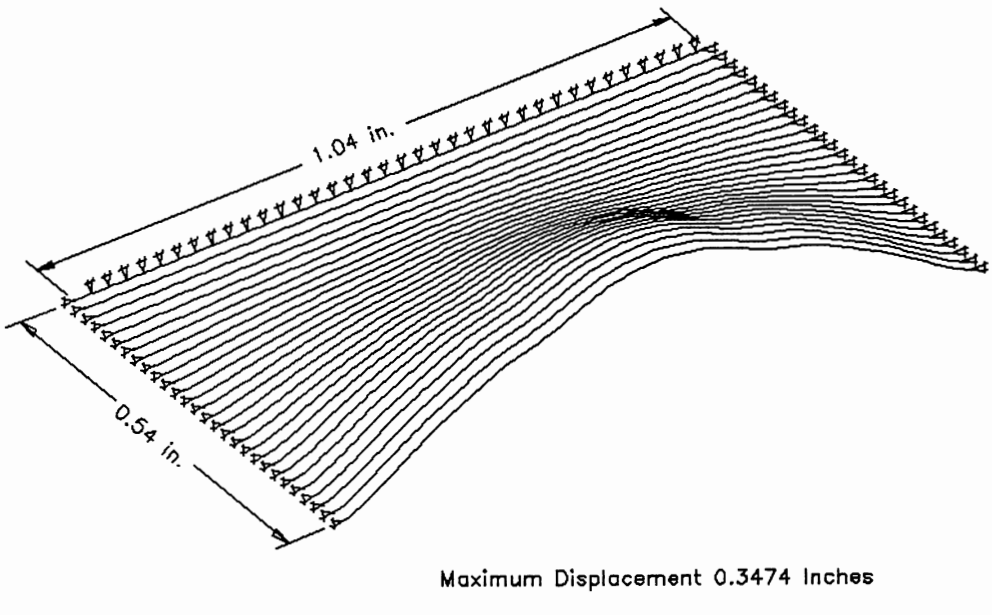


Figure D.10. Linear Plate Displacement at 1000 Time Frames;
Dimensional Time = 208.44 Microsec

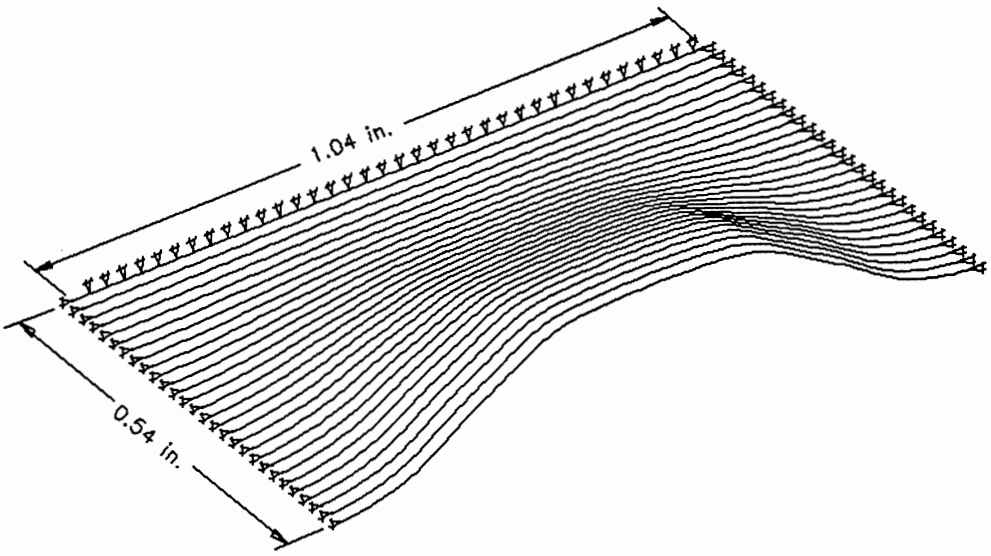


Figure D.11. Linear Plate Displacement at 1100 Time Frames;
Dimensional Time = 231.60 Microsec

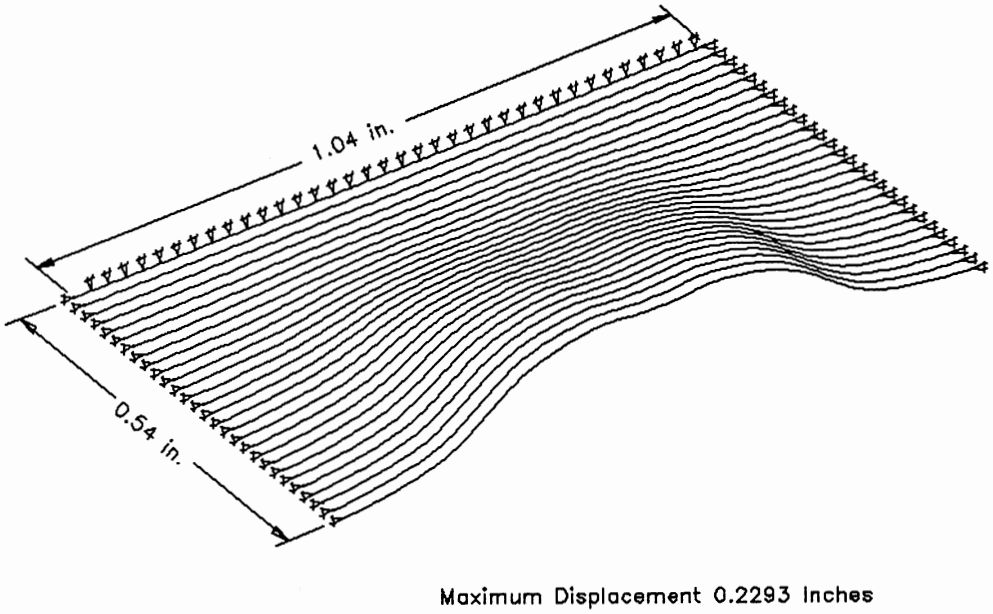


Figure D.12. Linear Plate Displacement at 1200 Time Frames;
Dimensional Time = 254.73 Microsec

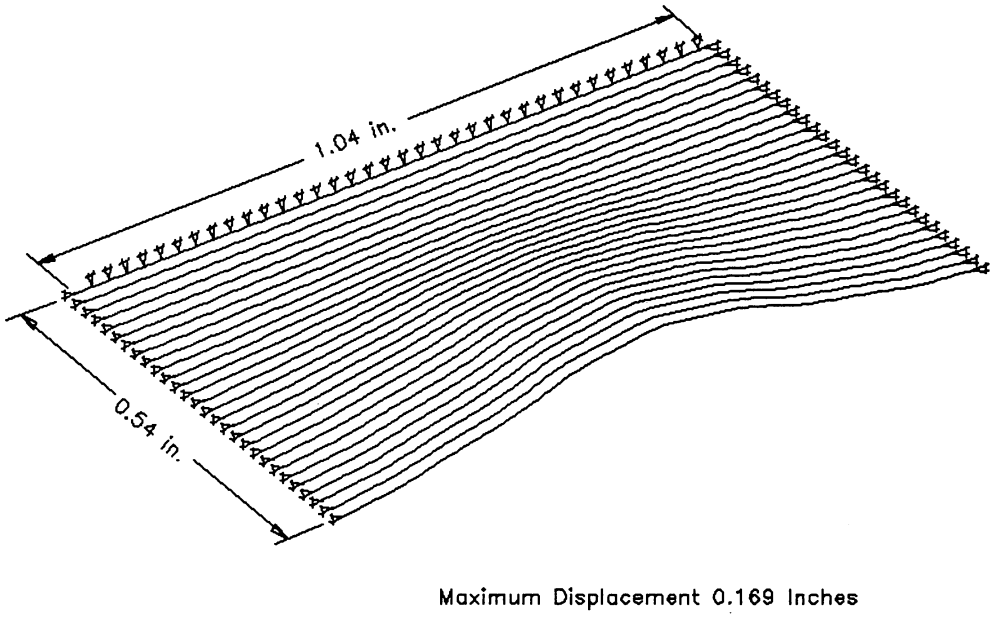


Figure D.13. Linear Plate Displacement at 1300 Time Frames;
Dimensional Time = 277.84 Microsec

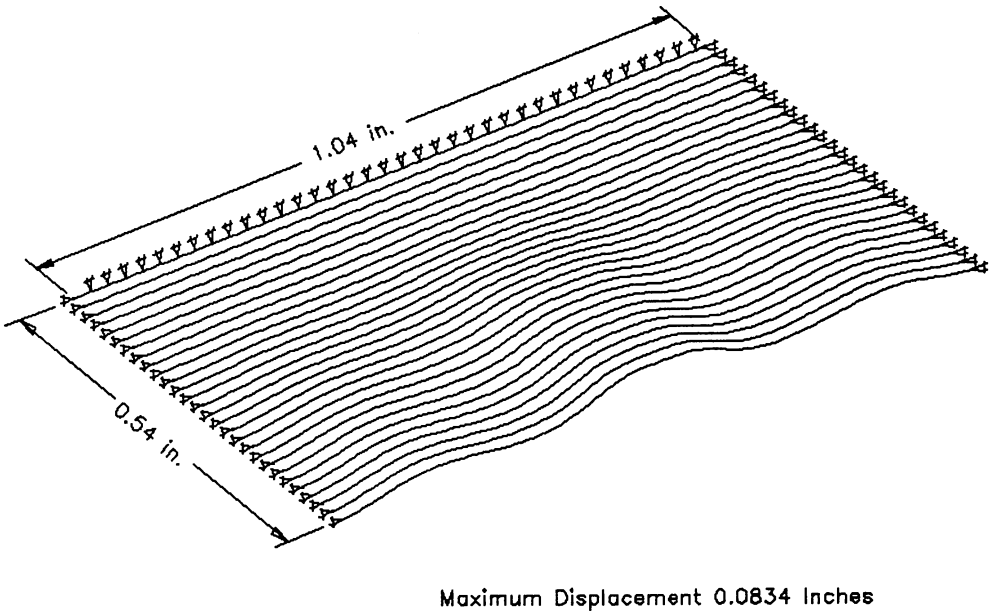


Figure D.14. Linear Plate Displacement at 1400 Time Frames;
Dimensional Time = 300.93 Microsec

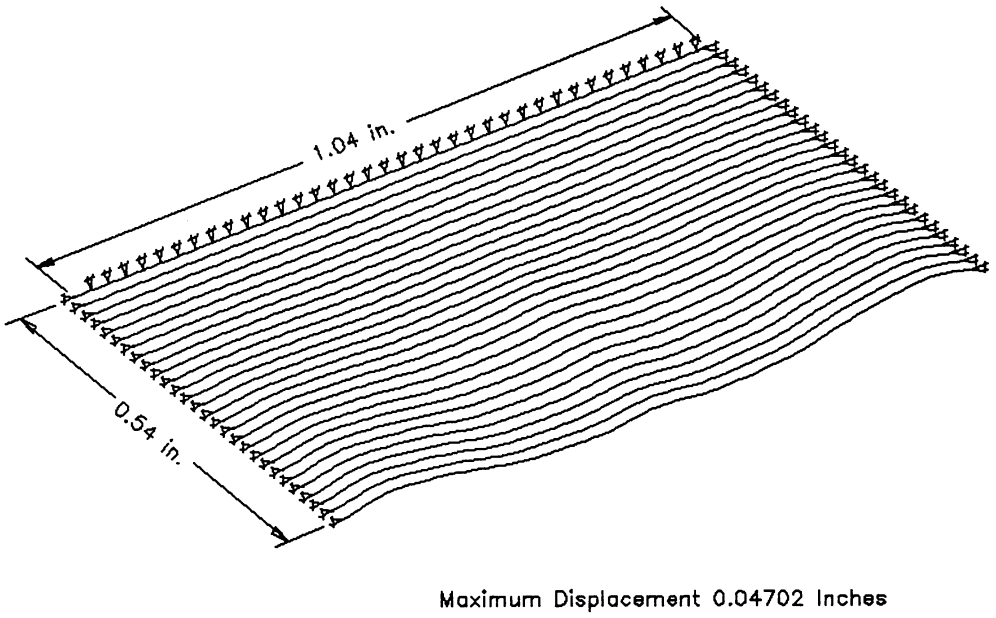


Figure D.15. Linear Plate Displacement at 1500 Time Frames;
Dimensional Time = 324.02 Microsec

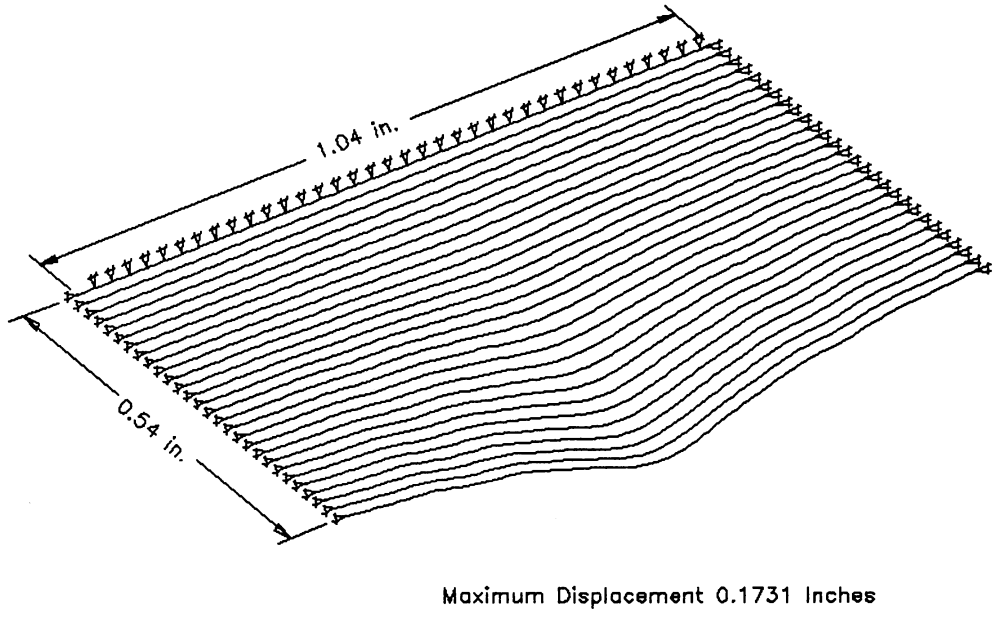


Figure D.16. Linear Plate Displacement at 1600 Time Frames;
Dimensional Time = 347.10 Microsec

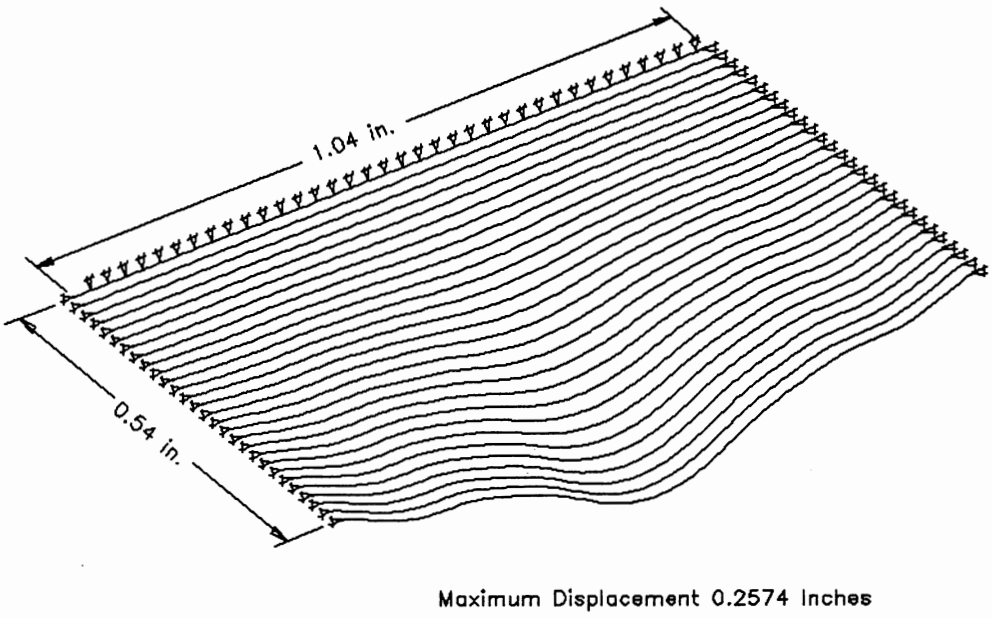


Figure D.17. Linear Plate Displacement at 1700 Time Frames;
Dimensional Time = 370.17 Microsec

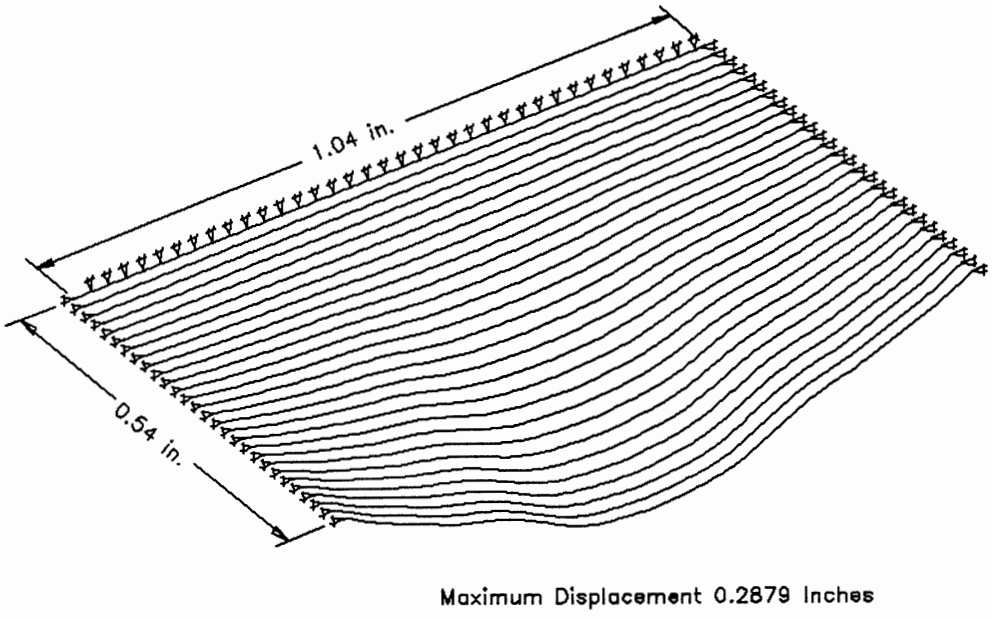


Figure D.18. Linear Plate Displacement at 1800 Time Frames;
Dimensional Time = 393.23 Microsec

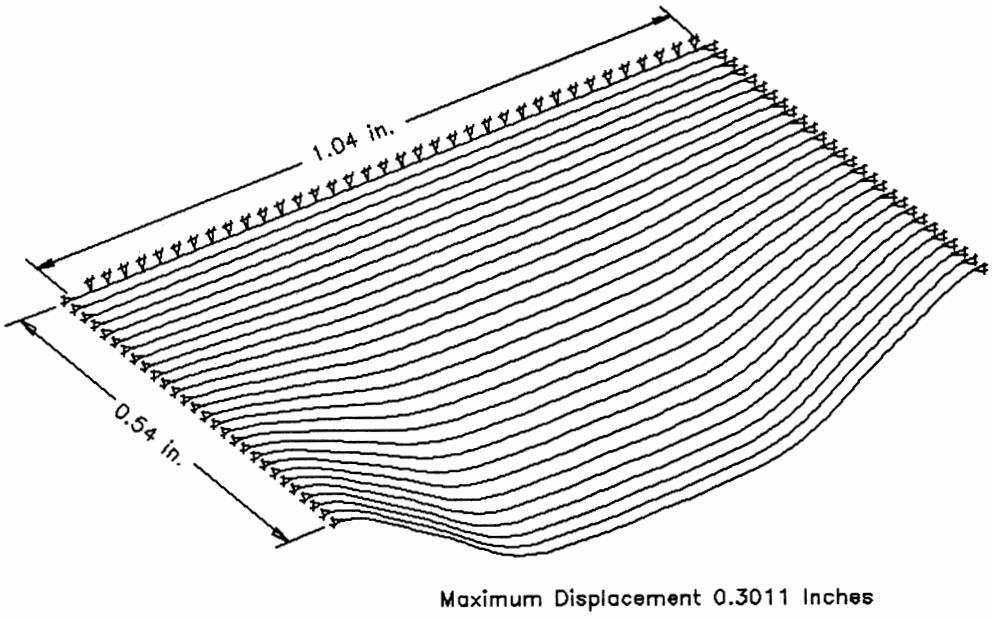


Figure D.19. Linear Plate Displacement at 1900 Time Frames;
Dimensional Time = 416.25 Microsec

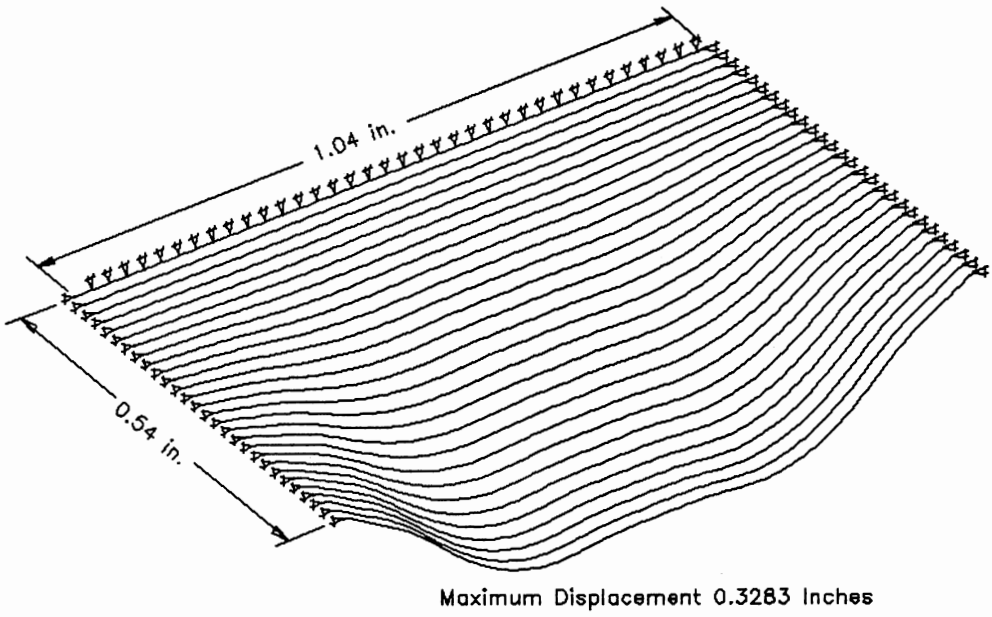


Figure D.20. Linear Plate Displacement at 2000 Time Frames;
Dimensional Time = 439.19 Microsec

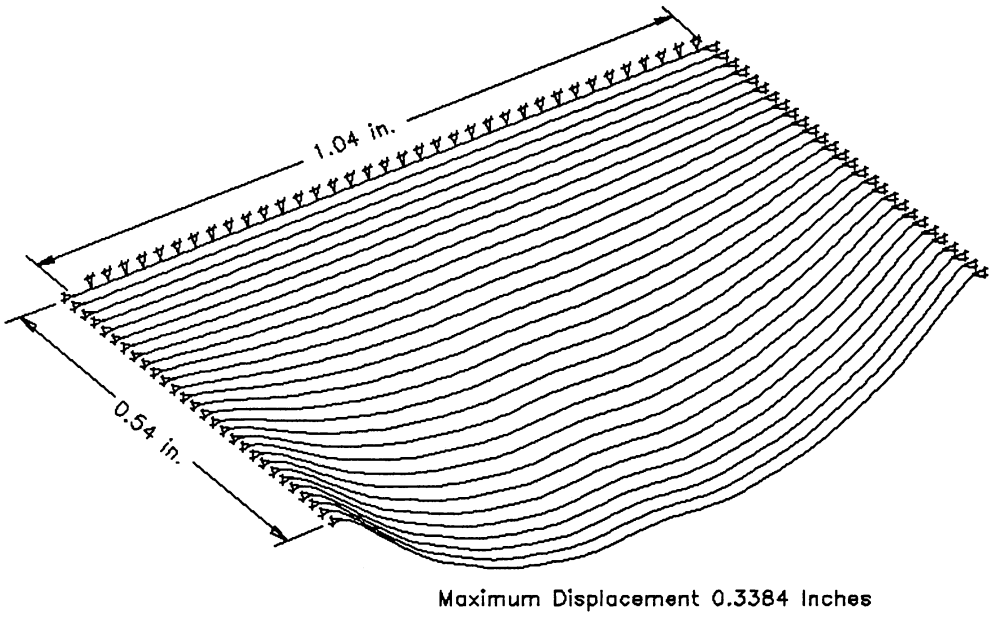


Figure D.21. Linear Plate Displacement at 2100 Time Frames;
Dimensional Time = 462.04 Microsec

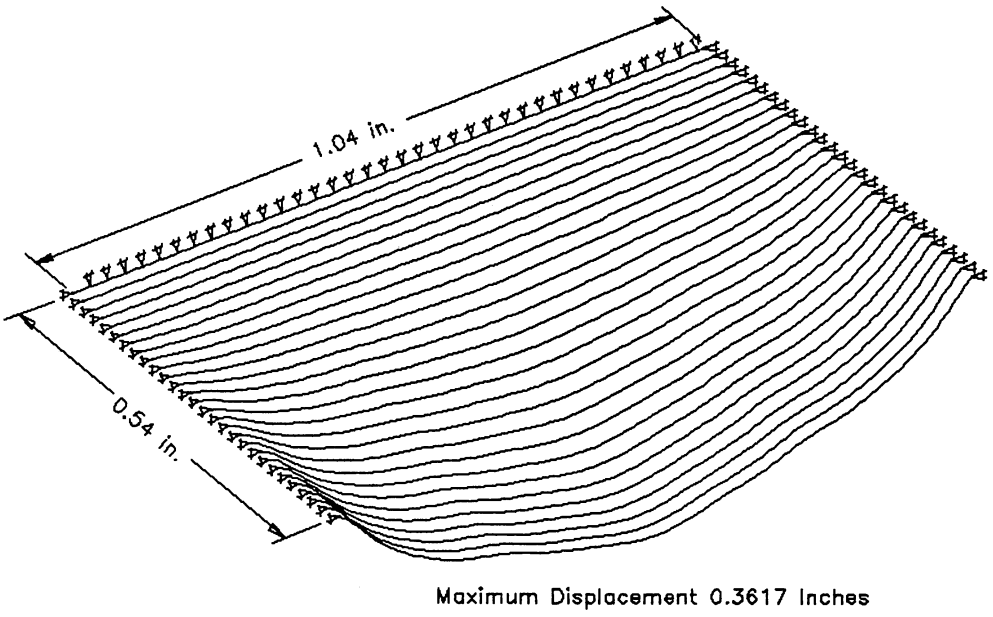


Figure D.22. Linear Plate Displacement at 2200 Time Frames;
Dimensional Time = 484.78 Microsec

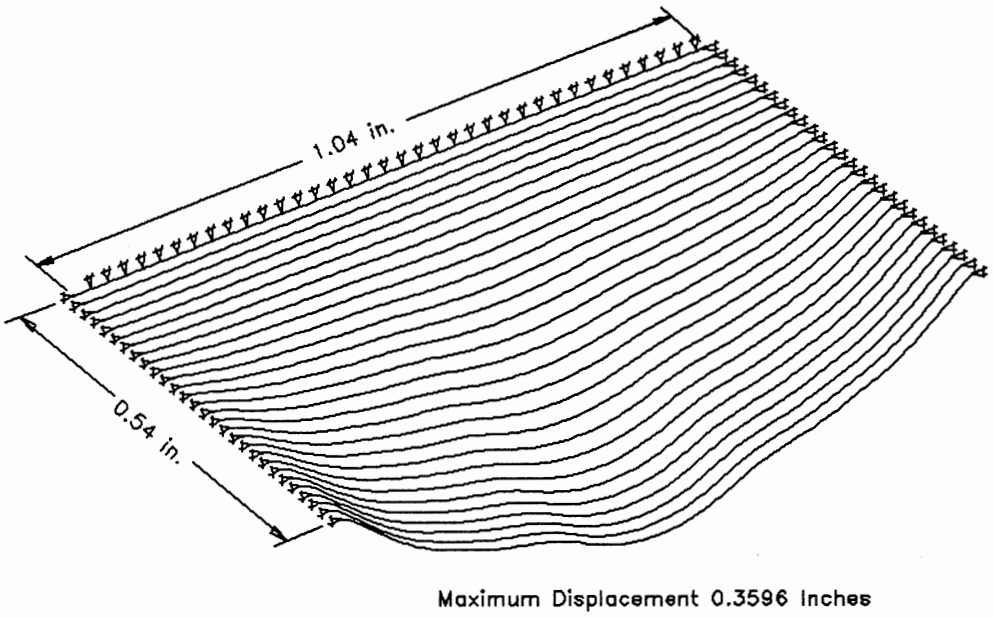


Figure D.23. Linear Plate Displacement at 2300 Time Frames;
Dimensional Time = 507.38 Microsec

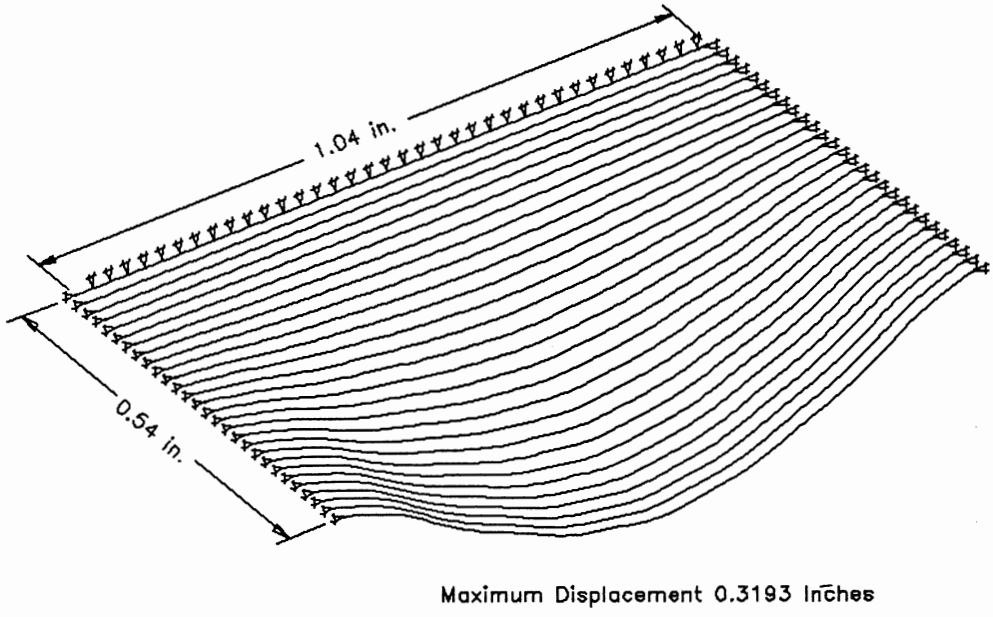


Figure D.24. Linear Plate Displacement at 2400 Time Frames;
Dimensional Time = 529.82 Microsec

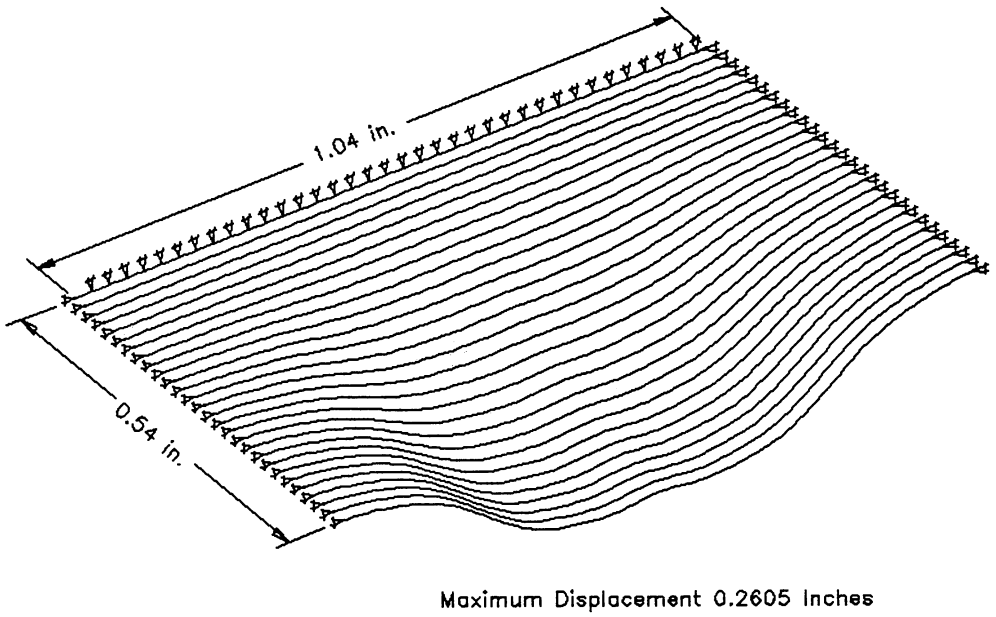


Figure D.25. Linear Plate Displacement at 2500 Time Frames;
Dimensional Time = 552.08 Microsec

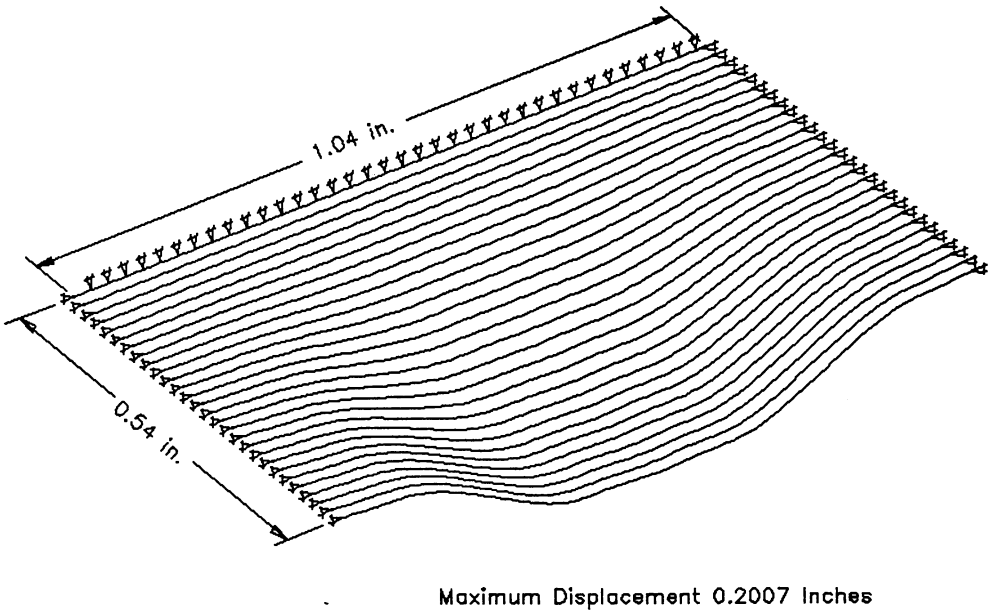


Figure D.26. Linear Plate Displacement at 2600 Time Frames;
Dimensional Time = 574.14 Microsec

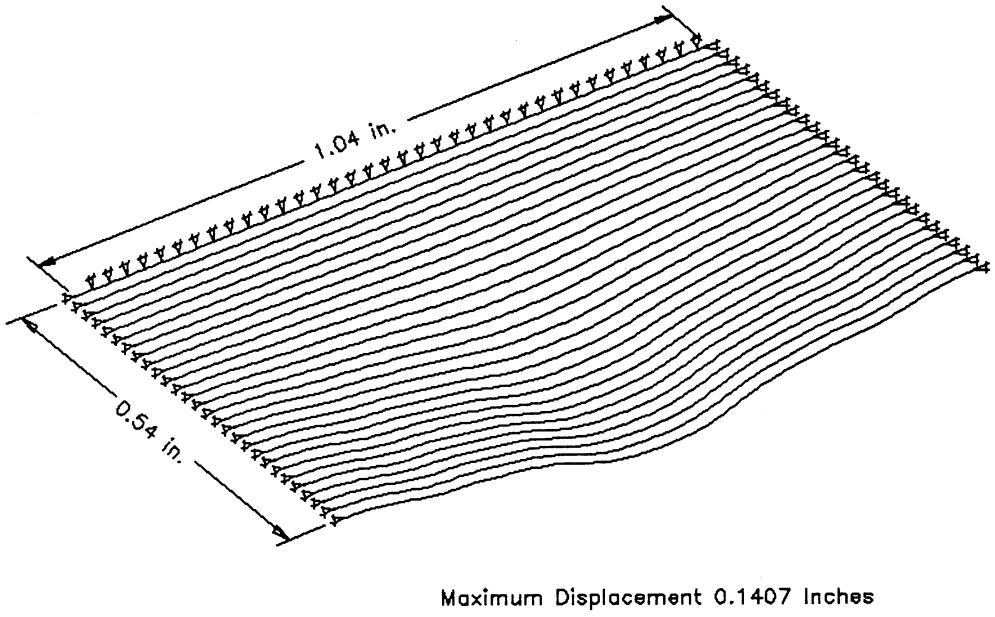


Figure D.27. Linear Plate Displacement at 2700 Time Frames;
Dimensional Time = 595.96 Microsec

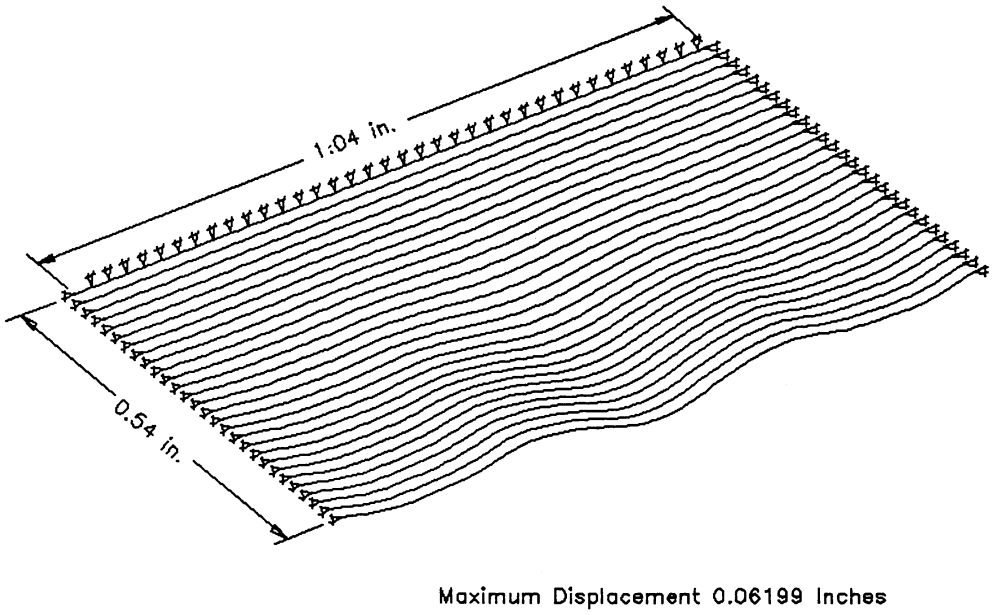


Figure D.28. Linear Plate Displacement at 2800 Time Frames;
Dimensional Time = 617.54 Microsec

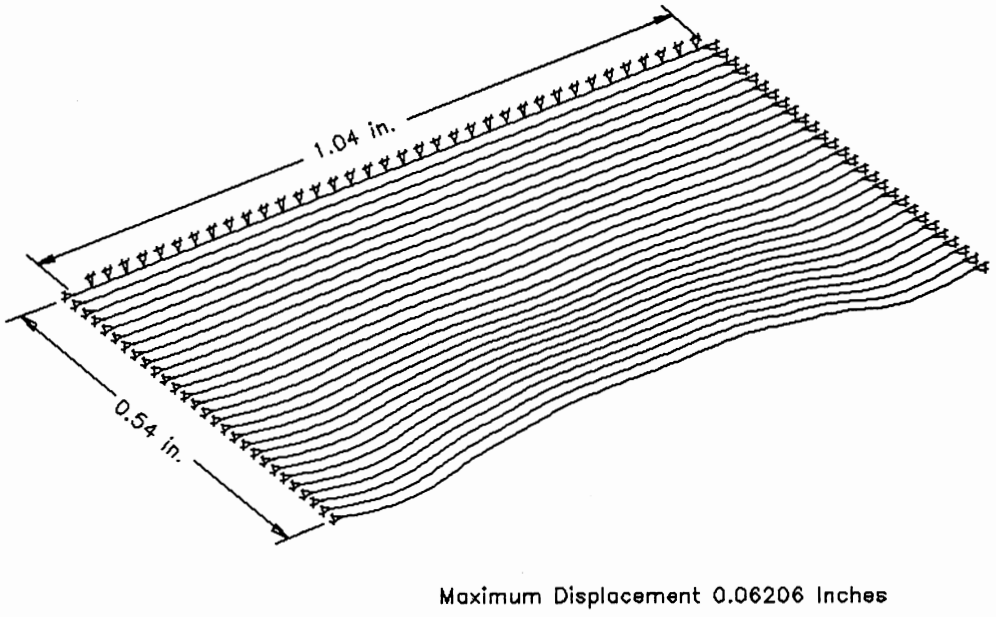


Figure D.29. Linear Plate Displacement at 2900 Time Frames;
Dimensional Time = 638.84 Microsec

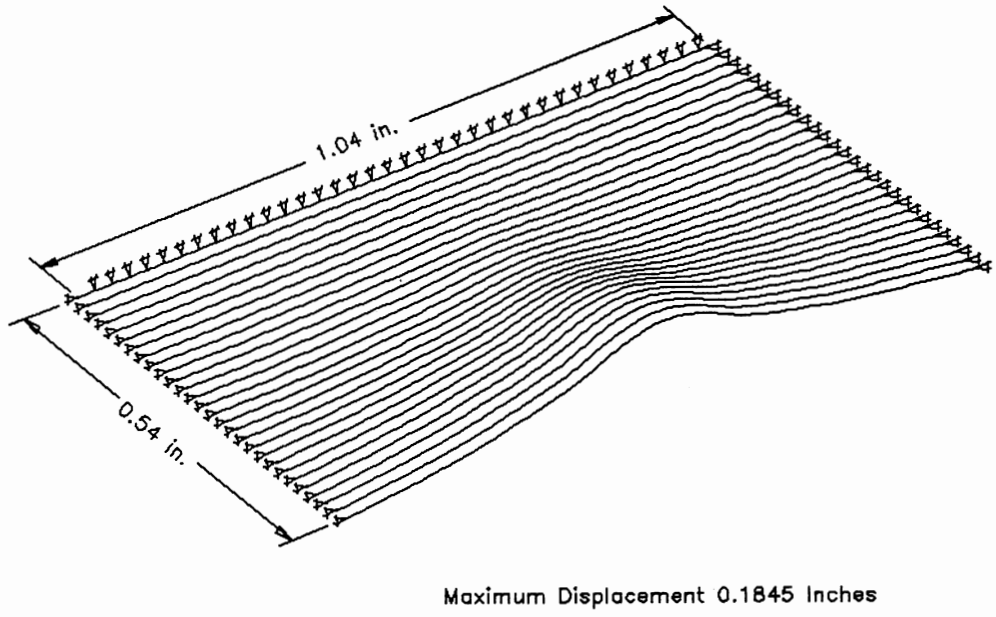
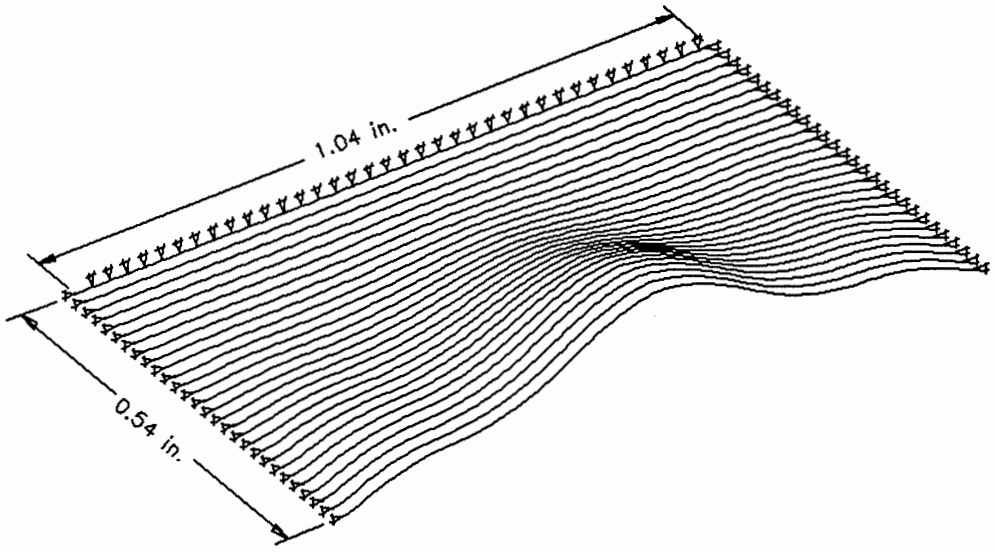
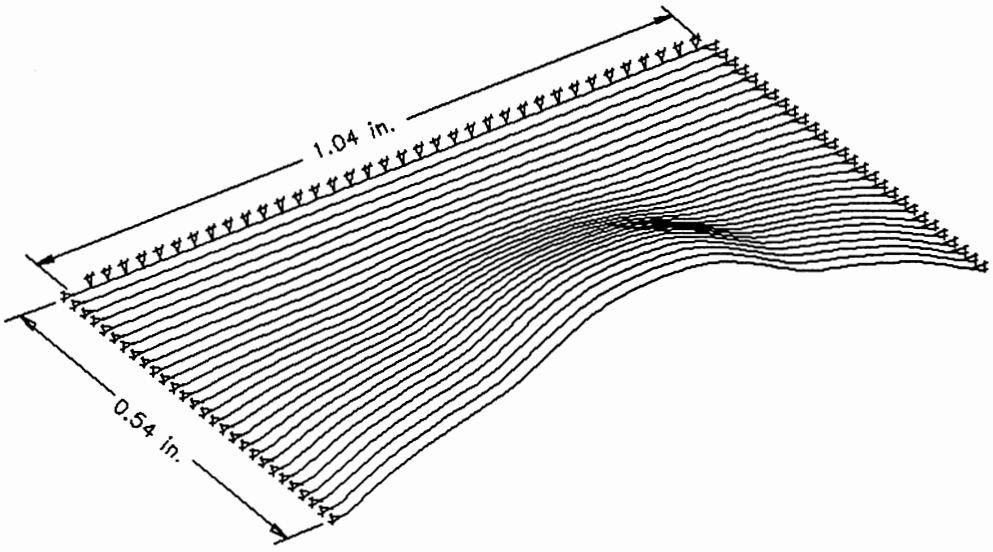


Figure D.30. Linear Plate Displacement at 3000 Time Frames;
Dimensional Time = 659.85 Microsec



Maximum Displacement 0.2628 Inches

Figure D.31. Linear Plate Displacement at 3100 Time Frames;
Dimensional Time = 680.54 Microsec



Maximum Displacement 0.2934 Inches

Figure D.32. Linear Plate Displacement at 3200 Time Frames;
Dimensional Time = 700.91 Microsec

APPENDIX E

COMPUTER PROGRAM LISTINGS FOR SINGLE-POINT TENSION MEASUREMENTS

```
/*************************************************************************/
/***     Program WH1DF (Digital Filtering) -- Web Handling Program     ***/
/***     to take two records from the pulsing system, perform          ***/
/***     digital filtering on the raw signal, and calculate a time     ***/
/***     of flight DeltaT value on the pulse.  Additional records      ***/
/***     are similarly obtained, formed into a pseudoperiodic          ***/
/***     record, upon which the spectral density is calculated for     ***/
/***     the upstream microphone signal, generating a characteristic   ***/
/***     frequency value Webfreq.  The DeltaT and Webfreq results      ***/
/***     are used in the Ribbon Equation to yield a Tension value.     ***/
/*************************************************************************/
        #include                    <stdio.h>
        #include                    <stdlib.h>
        #include                    <dos.h>
        #include                    <graphics.h>
        #include                    <conio.h>
        #include                    <alloc.h>
        #include                    <string.h>
        #include                    <process.h>
        #include                    <math.h>
        #define       pi            3.14159L
        #define       cair          13500.0L
        #define       radius        1.25
        #define       rhoair        4.373e-5L

      FILE              *D50;
      char              DataStr[64],Key,dir;
      int               D50IOFlag,i,j,k,l;
      int               sl,NS,grdriver,grmode;
      int               ycoord,ncorr,ndel,nspan;
      int               reflin,mxptr,mnptr,addr,instr,coun,ncy;
      int               maxptr[20],minptr[20],Itemp,delTptr;
      double            SRdivider,temp,rate,sum,conv,deltaT,period;
      double            vel,Tc,Te,webfreq,kair,kweb;
      double            max[20],min[20],XDAT[1250],YDAT[1250];
      double            totmax,totmin,test,pktook,ratio;
      double            Fc,BW,Gain,rhoweb,X[1250],Y[1250];
      float             dist,Rtemp;

int SEG(int far *Var)
{       unsigned  seg;
        seg=FP_SEG(Var);
        return(seg);
}
int OFF(int far *Var)
{       unsigned   offs;
        offs=FP_OFF(Var);
        return(offs);
}
int D50Output(char *Str)
{       if(D50IOFlag) { rewind(D50); D50IOFlag=0; }
        if (fprintf(D50,"%s",Str) == EOF) return(1);
        if (fflush(D50) == EOF) return(1);
        return(0);
 }
int D50InputInt(int *Int)
{       char    Str[16];
        if (!D50IOFlag) { rewind(D50); D50IOFlag=1; }
        if (!fgets(Str,16,D50)) return(1);
        if (sscanf(Str,"%d",Int) != 1) return(1);
        return(0);
}
int D50InputStr(char *Str)
{       if (!D50IOFlag) { rewind(D50); D50IOFlag=1; }
        if (!fgets(Str,32,D50)) return(1);
        return(0);
}
```

```
void GetError()
{       char            Str[48];
        int             EN;
        clrscr() ; gotoxy(6,27); puts("\7");
        if (D50InputInt(&EN));
        if (D50InputStr(DataStr));
        if (D50InputStr(Str));
        printf("Driver Error Has Occurred !!\n\n");
        printf("DAS-50 Error Number => %d\n\n",EN);
        printf("Error => %s\n\n",Str);
        printf("On Command Line of => %s\n\n",DataStr);
        exit(1);
}
void Pixel1()
{       putpixel(k,i,1); j=i+5; putpixel(k,j,1);
        j=i+10; putpixel(k,j,1); j=i+15; putpixel(k,j,1);
}

void Pixel2()
{       putpixel(k,i,1); j=k+10; putpixel(j,i,1);
}

void GetKBDInfo(int Ind)
{       char            *Str,buffer[10];
        int             ii;
        buffer[0]=10;
        for (ii=1; ii<=9; ii++) { buffer[ii]=0; }
        Str=cgets(buffer); switch(Ind)  {
        case 1: sscanf(Str,"%1c%f",&Key,&dist); break;
        case 2: sscanf(Str,"%1c",&Key); break;
        case 3: sscanf(Str,"%f",&Rtemp); break;
        case 4: sscanf(Str,"%d",&Itemp); break;
        }
}

void GetWave(double SamplRate,int Num)
{       char            Str[48];
        int             Status;
                /*  Set the Number of Samples  */
        sprintf(Str,"SEt SAmples= %d\n",Num);
        if (D50Output(Str)) GetError();
                /*  Set the On Board Memory Address  */
        sprintf(Str,"SEt ADdress= 000\n");
        if (D50Output(Str)) GetError();
                /*  Set the Sampling Rate  */
        sprintf(Str,"SEt RAte= Int %e\n",SamplRate);
        if (D50Output(Str)) GetError();
                /*  Acquire the Record  */
        if (D50Output("ACquire\n")) GetError();

        Status=0;       /* initialize status indicator  */
        do { if(D50Output("GEt STatus\n")) GetError();
             if(D50InputInt(&Status)) GetError();  }
        while ((Status & 0x10) != 0x10);
}

void FilterRecord(double A[],double B[],double SamplRate,
                                        double Scale,int Num)
{       char            Str[48];
        int             D[2500],ii,jj;
        double          cy0,cy1,cy2,cx,T,prod,f1,f2,f3,f4;
                /*  Set the Memory Address  */
        sprintf(Str,"SEt ADdress= %d\n",000);
        if(D50Output(Str) != 0) GetError();
                /*  Transfer Data to Given Vector  */
        sprintf(Str,"TRAnsfer %d %d %d %d\n",SEG(D),OFF(D),Num,1);
        if (D50Output(Str) != 0) GetError();
```

```
        T=2.0/SamplRate; prod=pi*Fc*T;
        cy0=1.0+pi*T*BW+prod*prod; cy1=2.0*(1.0-prod*prod);
        cy2=-1.0+pi*T*BW-prod*prod; cx=pi*Gain*T*BW;
        A[0]=cx*(Scale*D[0]/2048.0)/cy0;
        B[0]=cx*(Scale*D[1]/2048.0)/cy0;
        A[1]=(cx*(Scale*D[2]/2048.0)+cy1*A[0])/cy0;
        B[1]=(cx*(Scale*D[3]/2048.0)+cy1*B[0])/cy0;
        for (ii=4, jj=2; ii<=Num; ii+=2, jj++)  {
                f1=Scale*D[ii-2]/2048.0;
                f2=Scale*D[ii-1]/2048.0;
                f3=Scale*D[ii]/2048.0;
                f4=Scale*D[ii+1]/2048.0;
                A[jj]=(cx*(f3-f1)+cy1*A[jj-1]+cy2*A[jj-2])/cy0;
                B[jj]=(cx*(f4-f2)+cy1*B[jj-1]+cy2*B[jj-2])/cy0;  }
}

void Correlate(double A[],double B[],double C[],int CorrSpan
                                        int CorrShifts,int Delay)
{       double          prod;
        int             r,rdiff,totrdiff,ii,jj;
        for (r=1; r<=CorrShifts; r++) {
                sum=0.0; rdiff=CorrSpan-r;
                totrdiff=CorrSpan+Delay+1-r;
                for (ii=Delay+1; ii<=totrdiff; ii++) {
                        jj=ii+r;
                        prod=A[ii]*B[jj];
                        sum=sum+prod;   }
                C[r]=sum/(float)rdiff;  }
}

void Fourea(double datar[],double datai[],int Nft,int isign)
{       int             mmax,istep,m;
        double          tempr,tempi,wr,wi,theta;

/*    perform the bit reversal   */
        j=1; for (i=1; i<=Nft; i++) {
                if ((i-j)<0) goto L10;
                else goto L20;
L10:
          { tempr=datar[j]; tempi=datai[j];
            datar[j]=datar[i]; datai[j]=datai[i];
            datar[i]=tempr; datai[i]=tempi; }
L20:
            m=Nft/2;
L30:
            if ((j-m)<=0) goto L40;
            j-=m; m=(m+1)/2; goto L30;
L40:
            j+=m;  }

/*    compute the butterflies   */
        mmax=1;
L50:
        if ((mmax-Nft)<0) goto L60;
           else goto L70;
L60:
        istep=2*mmax;
        for (m=1; m<=mmax; m++) {
            theta=pi*isign*(m-1)/mmax;
            wr=cos(theta); wi=sin(theta); i=m;
            while (i<=Nft) {
                  j=i+mmax;
                  tempr=wr*datar[j]-wi*datai[j];
                  tempi=wr*datai[j]+wi*datar[j];
                  datar[j]=datar[i]-tempr;
                  datai[j]=datai[i]-tempi;
```

```
                datar[i]=datar[i]+tempr;
                datai[i]=datai[i]+tempi;
                i+=istep;  }  }
            mmax=istep; goto L50;
L70:
          if (isign<0) goto L80;
          else  { for (i=1; i<=Nft; i++) {
                        datar[i]=datar[i]/Nft;
                        datai[i]=datai[i]/Nft; } }
L80: ;
}

void ScopeSetup(int addr)
{         char            Str[48];
          strcpy(Str,":disp:form 2");
          ieoutput(addr,Str,12);       /* split screen into two parts  */
          strcpy(Str,":view chan1;:view chan4");
          ieoutput(addr,Str,23);       /* view channel 1 and channel 4  */
          strcpy(Str,":tim:mode trig;del -5.0e-4;rang 5.0e-3;ref left");
          ieoutput(addr,Str,47);       /* set up the scope time base   */
          strcpy(Str,":trig:sour chan1;lev 0.4");
          ieoutput(addr,Str,25);
          strcpy(Str,":chan1:coup ac;offs -400.e-3;rang 1.6");
          ieoutput(addr,Str,32);       /* set up channel 1 sensitivity */
          strcpy(Str,":chan4:coup ac;offs -200e-3;rang 1.6");
          ieoutput(addr,Str,32);       /* set up channel 4 sensitivity */
          strcpy(Str,":acq:type norm;coun 1;poin 512;comp 100");
          ieoutput(addr,Str,39);       /* set up the acquire subsystem */
}

void ScopeAcquire(int addr)
{         char            Str[32];
          strcpy(Str,":dig chan1,chan4");
          ieoutput(addr,Str,16);       /*   digitize channels 1 and 4   */
 }

void SpecDens(double A[],double B[],double E[],double SamplRate,int Num)
{        int              ii;
         double           AR,AI,BR,BI,C,D,ScaleFactor;
         ScaleFactor=4.0/(SamplRate*(float)Num);
         for (ii=1; ii<=Num; ii++) {
                AR=(A[ii]+A[(Num+1)-ii])/2.0;
                AI=(B[ii]-B[(Num+1)-ii])/2.0;
                BR=(B[ii]+B[(Num+1)-ii])/2.0;
                BI=(A[ii]-A[(Num+1)-ii])/(-2.0);
                C=AR*BR+AI*BI; D=AR*BI-AI*BR;
                E[ii]=ScaleFactor*sqrt(C*C+D*D); }
}

void StepMotor(char dir, int coun)
{
          #define         ioprt1   688
          #define         ioprt2   689
          int             ii;
          outportb(ioprt2,dir);          /*  set the direction of rotation
          for (ii=1; ii<=coun; ii++) {   /*  loop until all counts stepped
              outportb(ioprt1,011);      /*  set the io port bit 0 high
              delay(04);                 /*  delay for 4 milli-seconds
              outportb(ioprt1,010);      /*  complete the cycle for a pulse
              delay(04);            }
          outportb(ioprt1,000);
 }
  /*************************************************************************/
  /**                                                                     **/
  /**     START THE MAIN PROGRAM FOR WEB TENSION MEASUREMENT              **/
  /**                                                                     **/
```

```
main()
{
ncy=0; ndel=0; addr=07;            /* address of HP 54501A Digital Scope*/
Fc=1000.0; BW=1500.0; Gain=2.0;    /* Digital Filter default parameters */
NS=2400; rate=4.0e+5;              /* initial sample number and rate  */

D50=fopen("$DAS50","r+");
if(D50Output("Clear\n")) GetError();
ieseg(0xd000);                     /* firmware address of IEEE488 code  */
ieabort(); iedevclr(addr);

clrscr() ;                         /* Use BIOS scroll to clear screen */
gotoxy(10,3);                      /* Set cursor to Row 1, Col 7 */
puts("  Program WH1DF - Single Point Measurement - Digital Filtering\n\n")
puts("   This program is for the operation of the tension measurement ");
puts("system developed through Dr. Richard Lowery of Oklahoma State   ");
puts("University for the purpose of On-Line Web Tension Analysis.  This");
puts("program acquires records from a Metrabyte DAS50 A-D board and   ");
puts("calculates the Cross Correlation and Spectral Density functions ");
puts("for the two signals.  This procedure provides a delta T and     ");
puts("characteristic frequency value which are used to calculate web  ");
puts("tension through use of the Ribbon Equation.  Digital filtering is");
puts("used such that filter parameters may be adjusted by the user to  ");
puts("obtain the best quality signal possible.  Default filter parameters"
puts("are:  Filter Gain = 2.0,  Filter Center Frequency = 1000 Hz, Filter"
puts("Bandwidth = 1500 Hz.\n\n");
puts("ENTER MATERIAL DENSITY (LBM/IN-IN)");
GetKBDInfo(3); puts(""); rhoweb=Rtemp;

  puts("        POSITIONING OF TRANSDUCER HEAD     \n");
  puts("     Position the transducer head to the desired intial");
  puts("location through interactive use of this computer.  Enter the");
  puts("desired adjustment in the form 'Lxx<cr>' for adjustment Left ");
  puts("xx inches or 'Rxx<cr>' for adjustment Right xx inches.  When ");
  puts("the move is completed, the computer will poll the user as    ");
  puts("follows: Enter 'R<cr>' to readjust or 'C<cr>' to continue.   ");
  puts("Entering R will allow repositioning left or right and entering");
  puts("C will allow for continuation of this program.\n");
L100:
      puts("  Position Adjustment:  Enter 'Lxx<cr>' to move");
      puts("        left or 'Rxx<cr>' to move right          \n");

  GetKBDInfo(1); puts(""); switch(Key)  {
  case 'r':                         /*Digital Output Bit 8 for Direction*/
  case 'R': dir=0xdf; break;       /*right - D/O-8 set to one for CW rot*/
  case 'l':
  case 'L': dir=0xde; break;      /*left - D/O-8 set to zero for CCW rot*/
  default: puts("Invalid Entry - Try Again\n"); goto L100;
  }
  coun=dist*35.0/(pi*radius);        /* Number of Stepper Motor pulses */
  StepMotor(dir,coun);          /*Move the Transducer the amount entered*/
L140:
      puts("  Enter 'R<cr>' to readjust or 'C<cr>' to continue \n");
      GetKBDInfo(2); puts(""); switch(Key)  {
      case 'r':
      case 'R': goto L100;
      case 'c':
      case 'C': break;
      default: puts("Invalid Entry - Try Again\n"); goto L140;
      }

  /* ------------------------------------------------------------*/
  /* Set the DAS-50 to take 2400 samples, alternating between    */
  /* Channels 0 & 1, in the Trace-After Trigger mode.  Note that*/
  /* all the following Print statements could be abbreviated     */
  /* and put into one print statement                            */
  /*------------------------------------------------------------*/
```

```
        if (D500utput("SEt CHannels=0&1\n")) GetError() ;
        if (D500utput("SEt RAnge=+-5V\n")) GetError() ;
        if (D500utput("SEt TRigger Mode=7\n")) GetError() ;
        if (D500utput("SEt TRig Volt=0.7\n")) GetError() ;
        if (D500utput("SEt STart=AFter\n")) GetError() ;

/*  Initialize graphics facility to exhibit an acquired waveform */
L250:   grdriver=5; grmode=3;
        initgraph(&grdriver,&grmode,"");
        setviewport(0,0,639,349,1);

/* Sample and Filter Data Set For Default or Modified Filter Parameters *
L999:   clearviewport(); GetWave(rate,NS);
        FilterRecord(XDAT,YDAT,rate,65.0,NS);

 /*  draw a grid on the screen   */
        gotoxy(1,0); lineto (638,0); lineto (638,260);
        lineto (1,260); lineto (1,0); line(1,130,638,130);

        for (i=0,k=0; i<=240; i+=20) {
                for (k=0; k<=639; k+=20) {
                        switch(k)  {
                        case 80:
                        case 160:
                        case 240:
                        case 320:
                        case 400:
                        case 480:
                        case 560: Pixel1(); Pixel2(); break;
                        default:  Pixel2();     } } }

 /*   put the data traces on the screen in screen coordinates   */
        for (i=ndel; i<=(ndel+640); i++) {
                k=i-ndel; ycoord=(int)XDAT[i];
                j=65-ycoord; putpixel(k,j,1);
                ycoord=(int)YDAT[i];
                j=195-ycoord; putpixel(k,j,1); }
        textheight(&Key); if (Key==8) i=2; else i=1;
        settextstyle(0,0,i);

/*  Present user options for waveform positioning and filtering */
  outtextxy(1,264,"Press Space Bar to Repeat Process");
  outtextxy(1,276,"Press 'l' or 'r' to shift pattern left or right");
  outtextxy(1,288,"Press 'i' or 'd' to increase or decrease sample rate")
  outtextxy(1,300,"Press 'f' to alter digital filter center frequency");
  outtextxy(1,312,"Press 'b' to alter digital filter bandwidth");
  outtextxy(1,324,"Press 'g' to alter digital filter gain");
  outtextxy(1,336,"Press 't' when ready to begin web tension test");

L400:   Key=getch(); switch(Key)  {

/***  Case of Left or Right Waveform Positioning ***/
        case 'l':
        case 'L': ndel=ndel+40; goto L999;
        case 'r':
        case 'R': ndel=ndel-20; if (ndel<0) ndel=0;
                        else goto L999; break;

/***  Case of Increase or Decrease of Sample Rate ***/
        case 'i':
        case 'I': rate=rate+1.0e+5; temp=rate; goto L999;
        case 'd':
        case 'D': rate=rate-1.0e+5; temp=rate; goto L999;

/***  Case of Space Bar -- Repeat Waveform Viewing ***/
        case ' ': goto L999;
```

```
/***  Case of Digital Filter Center Frequency Adjustment ***/
        case 'f':
        case 'F': clearviewport(); restorecrtmode();
        gotoxy(10,3); puts(" FILTER CENTER FREQUENCY ADJUSTMENT \n");
        puts("Enter 'lxxx' or 'rxxx' to adjust the filter");
        puts("center frequency left or right, respectively,");
        puts("by an amount xxx in Hertz\n\n");
L950:   GetKBDInfo(1); switch(Key)  {
                case 'r':
                case 'R': Fc=Fc+dist; goto L250;
                case 'l':
                case 'L': Fc=Fc-dist; goto L250;
                default:  puts("Invalid entry - try again\n");
                          goto L950; }

/***  Case of Digital Filter Bandwidth Adjustment ***/
        case 'b':
        case 'B': clearviewport(); restorecrtmode();
        gotoxy(10,3); puts("FILTER BANDWIDTH ADJUSTMENT \n");
        puts("Enter 'ixxx' or 'dxxx' to adjust the filter");
        puts("bandwidth increase or decrease, respectively,");
        puts("by an amount xxx in Hertz\n\n");
L1250:  GetKBDInfo(1); switch(Key)   {
                case 'i':
                case 'I': BW=BW+dist; goto L250;
                case 'd':
                case 'D': BW=BW-dist; goto L250;
                default:  puts("Invalid entry - try again\n");
                          goto L1250;   }

/***  Case of Digital Filter Gain Adjustment ***/
        case 'g':
        case 'G': clearviewport(); restorecrtmode();
        gotoxy(10,3); puts(" FILTER GAIN ADJUSTMENT ");
        puts("Enter new filter gain specification.\n");
        GetKBDInfo(3); Gain=Rtemp; goto L250;

/***  Case of Web Tension Test Desired ***/
        case 't':
        case 'T': goto L1500;
        default:  puts("'\x7'"); goto L400;
        }
L1500:    clearviewport(); moveto(20,10);
          outtext("CROSS  CORRELATION  TEST");

  /*   Set the Scope up for automatic waveform acquisition  */
         ScopeSetup(addr);

  /*   Select cross correlation shifts and span depending on rate */
/* SRdivider divides sample rate down for subsequent Fourier analysis */

    if (rate==2.0e+5){ncorr=200; nspan=360; NS=1296;SRdivider=4.0;}

    else if (rate==3.0e+5){ncorr=300;nspan=500;NS=1488;SRdivider=5.0;}

    else if (rate==4.0e+5){ncorr=360;nspan=600;NS=1840;SRdivider=8.0;}

    else if (rate==5.0e+5){ncorr=420;nspan=760;NS=2048;SRdivider=10.0;}

    else if (rate==6.0e+5){ncorr=500;nspan=900;NS=2272;SRdivider=10.0;}

    else if (rate==7.0e+5){ncorr=560;nspan=1000;NS=2496;SRdivider=11.0;}

    else if (rate==8.0e+5){ncorr=560;nspan=1000;NS=2496;SRdivider=12.0;}
```

```
    else { puts("  Sampling rate selected is not supported  \n");
                        goto END; }

/***   Obtain the data records according to NS and rate values  ***/
    GetWave(rate,NS); ScopeAcquire(addr);
    FilterRecord(XDAT,YDAT,rate,5.0,NS);     /*Perform Digital Filtering*/
    GetWave(rate,NS);
    FilterRecord(X,Y,rate,5.0,NS);          /*Average second record with*/
    for (i=1; i<NS/2; i++) {                       /*with first prior to*/
        XDAT[i]=XDAT[i]+X[i];                        /*cross correlation*/
        YDAT[i]=YDAT[i]+Y[i]; }
    Correlate(XDAT,YDAT,X,nspan,ncorr,ndel);       /*Perform Correlation*/

/*********   sort the cross correlation data for plotting   *********/
        test=fabs(X[2]);
        if(fabs(X[1])>test) sl=-1; else sl=1;
        i=2; j=0; k=0;
L3500:
        i++; if (i>=ncorr) goto L4500;
        if (sl>0) goto L3600; else goto L4000;
L3600:
        if ((X[i]-test)>=0.0) goto L3700; else goto L3800;
L3700:
        test=X[i]; mxptr=i; goto L3500;
L3800:
        k++; max[k]=test; maxptr[k]=mxptr; sl=-1; goto L3500;
L4000:
        if ((X[i]-test)<=0.0) goto L4100; else goto L4200;
L4100:
        test=X[i]; mnptr=i; goto L3500;
L4200:
        j++; min[j]=test; minptr[j]=mnptr; sl=1; goto L3500;

/* Find absolute maximum and minimum cross correlation values */
L4500:  totmax=max[1]; mxptr=maxptr[1];
        totmin=min[1]; mnptr=minptr[1];
        for (i=2; i<=k; i++) {
                if ((fabs(max[i])-fabs(totmax))<=0.0) continue;
                else {  totmax=max[i];
                        mxptr=maxptr[i]; }  }
        for (i=2; i<=j; i++) {
                if ((fabs(min[i])-fabs(totmin))<=0.0) continue;
                else {  totmin=min[i];
                        mnptr=minptr[i]; }  }

/* Find span of cross correlation function to aid in plotting */
        pktopk=fabs(totmax)+fabs(totmin);
        if (fabs(totmax)>=fabs(totmin)) ratio=fabs(totmax)/pktopk;
                else ratio=fabs(totmin)/pktopk; delTptr=mxptr;

/* Establish correlation function referenceline and graphics conversion */
        reflin=260.0*ratio; conv=200.0/pktopk; moveto(10,reflin);
        for (i=1; i<=ncorr; i++) {                  /*plot reference line*/
                j=i+10; lineto(j,reflin); }
        ycoord=X[1]*conv; j=reflin-ycoord; moveto(10,j);
        for (i=2; i<=ncorr; i++) {
        j=i+10; ycoord=X[i]*conv;                      /* plot the cross */
        k=reflin-ycoord; lineto(j,k); }            /*correlation function*/

    l=reflin+12; moveto(10,l); outtext("1");
    itoa(ncorr,DataStr,10); moveto(ncorr,l);
    outtext(DataStr); period=2.0/rate;              /*Provide graphical*/
    deltaT=period*delTptr; moveto(450,150);         /*view of the cross*/
    itoa(delTptr,DataStr,10);                          /*correlation*/
    outtext("Max Correlation at"); moveto(450,166);    /*function plus*/
```

```
    outtext("  Delay Point "); outtext(DataStr);       /*identified index*.
    moveto(450,186); gcvt(period,6,DataStr);          /*which is used in*.
    outtext("Period for this"); moveto(450,202);           /*DeltaT*/
    outtext("  Test is "); outtext(DataStr);             /*computation*/
    moveto(450,222); gcvt(deltaT,6,DataStr);
    outtext("Delta T for this"); moveto(450,238);
    outtext("  Test is "); outtext(DataStr);
    moveto(20,330); outtext("Press any Key to Continue");
    getch(); clearviewport();

/*  Sample waveform again at a reduced rate to perform FFT computa- */
/*  tions to obtain the wave characteristic frequency. Lower sample */
/*  rate will allow for a higher frequency domain resolution.       */

    moveto(320,40); outtext("FOURIER  ANALYSIS"); j=1;
    rate=rate/SRdivider; ndel=ndel/(int)SRdivider; NS=160+ndel;
    for (l=1; l<=7; l++) {                      /*Acquire seven records.*/
    GetWave(rate,NS);                            /*Assemble records into*/
    FilterRecord(X,Y,rate,5.0,NS);                     /*pseudoperiodic*/
    for (i=ndel; i<=((148+ndel)/2); i++, j++)  {            /*resultant*/
        XDAT[j]=X[i]; YDAT[j]=0.0;  }  }                /*record for FFT*/
                                                          /*processing*/
    sum=0.0; for (j=1; j<=512; j++) {
        sum=sum+XDAT[j]; }                            /*Subtract Average*/
    sum=sum/512.0; for (j=1; j<=512; j++) {            /*value from each*/
        XDAT[j]=XDAT[j]-sum;   }                         /*record point*/

/*    Use a cosine windowing function to smooth the data ends     */
        for (j=1; j<=25; j++) {
                XDAT[j]=XDAT[j]*cos(pi*(1.0-(j-1)/24.0)/2.0); }
        for (j=488; j<=512; j++) {
                XDAT[j]=XDAT[j]*cos(pi*(1.0-(512-j)/24.0)/2.0);}

/*   Pad the vector with zeros from 513 to 1024 to double resolution  */
        for (j=513; j<=1024; j++)  {
                XDAT[j]=0.0; YDAT[j]=0.0;  }

/***  Fourier Transform: XDAT = real vector; YDAT = imaginery vector ***/
        Fourea(XDAT,YDAT,1024,-1);
/** Compute Spectral Density for Fore Microphone Pseudoperiodic Record **
        SpecDens(XDAT,YDAT,X,rate,1024);

        test=X[20]; mxptr=20;                          /*Identify maximum*/
        for (i=21; i<=256; i++) {                      /*Spectral Density*/
                if (X[i]<test) continue;                /*value and index*/
                test=X[i]; mxptr=i;     }

        pktopk=fabs(test); conv=260.0/pktopk; moveto(40,300);
        for (i=1; i<=256; i++) {
                j=i*2+39; putpixel(j,300,1);          /*Plot the Spectral*
                ycoord=X[i]*conv; l=300-ycoord;        /*Density Function*
                lineto(j,l);  }


  webfreq=(rate/2048.0)*mxptr; moveto(34,310);
  outtext("k=1"); moveto(530,310); outtext("k=256");
  gcvt(rate,6,DataStr); moveto(180,320);
  outtext("SAMPLING RATE = "); outtext(DataStr);          /*Graphical*/
  outtext(" HZ "); temp=rate/2048.0;                     /*display with*/
  gcvt(temp,6,DataStr); moveto(70,335);                       /*support*/
  outtext("FOR 1024 PT FFT:  FREQUENCY INCREMENT = ");   /*information*/
  outtext(DataStr); outtext(" HZ ");
```

```
    itoa(mxptr,DataStr,10); moveto(320,60);
    outtext(" MAX MAGNITUDE AT K = "); outtext(DataStr);
    moveto(320,80); outtext(" WEB FREQ = ");
    gcvt(webfreq,6,DataStr); outtext(DataStr);
    outtext(" HZ "); getch(); restorecrtmode(); clrscr();

    kair=2.0*pi*webfreq/cair;                /*Wave Number for Air*/
    vel=(2.0/deltaT);                          /* Web Phase Speed */
    kweb=2.0*pi*webfreq/vel;                 /*Wave Number for Web*/
    temp=kweb*kweb-kair*kair;
    if (temp<0.0)  {puts("### BAD VALUE OF DELTAT OR WEBFREQ ###");
      puts(" No Tension Value Processed"); goto END;}

/** Use Ribbon Eqation to Provide Air Loading Compensated Tension Value
    Tc=(rhoweb+2.0*rhoair/sqrt(temp))*vel*vel/386.0;

/** Use Membrane Equation to Provide In-Vacuo Tension Estimate **/
    Te=rhoweb*vel*vel/386.0;

  printf(" Characteristic Frequency calculated to be %f\n\n",webfreq);
  printf(" Estimated Membrane Equation Tension = %f\n\n",Te);
  printf(" Calculated Ribbon Equation Tension = %f\n\n",Tc);
END:          ;
}
```

```
/*************************************************************************/
/***     Program WH1FC (Fast Correlation) -- Web Handling Program      ***/
/***     to take one record from the pulsing system, perform           ***/
/***     digital filtering on the raw signal, and perform fast         ***/
/***     correlation procedures resulting in a time of flight          ***/
/***     DeltaT value.  A 128 point spectral density is calculated    ***/
/***     based on the original sampled record from which a char-       ***/
/***     acteristic frequency value Webfreq is obtained.  The          ***/
/***     DeltaT and Webfreq results are used in the Ribbon             ***/
/***               equation to yield a Tension value.                  ***/
/*************************************************************************/
        #include                    <stdio.h>
        #include                    <stdlib.h>
        #include                    <dos.h>
        #include                    <graphics.h>
        #include                    <conio.h>
        #include                    <alloc.h>
        #include                    <string.h>
        #include                    <process.h>
        #include                    <math.h>
        #define       pi            3.14159L
        #define       cair          13500.0L
        #define       rhoair        4.373e-5L

      FILE              *D50;
      char              DataStr[64],Key,dir;
      int               D50IOFlag,i,j,k,l;
      int               NS,grdriver,grmode,NumCorr;
      int               NumDelay,Windo,NumRelMax,NumRelMin;
      int               RelMaxIndx[20],RelMinIndx[20];
      int               Itemp,delTptr,SRdivider,reflin;
      int               AbsMaxIndx,AbsMinIndx,MaxFreqIndx;
      double            RelMax[20],RelMin[20],freqincrement;
      double            rate,sum,deltaT,period,temp;
      double            vel,Tc,Te,webfreq,kair,kweb,Fc,BW,Gain;
      double            rhoweb,X[260],Z[260],RXY[260],AbsMax;
      double            XDAT[1250],YDAT[1250],WindoVal,AbsMin;
      double            MaxFreq;
      float             dist,Rtemp;

int SEG(int far *Var)
{       unsigned  seg;
        seg=FP_SEG(Var);
        return(seg);
}

int OFF(int far *Var)
{       unsigned   offs;
        offs=FP_OFF(Var);
        return(offs);
}

int D50Output(char *Str)
{       if(D50IOFlag) { rewind(D50); D50IOFlag=0; }
        if (fprintf(D50,"%s",Str) == EOF) return(1);
        if (fflush(D50) == EOF) return(1);
        return(0);
 }

int D50InputInt(int *Int)
{       char    Str[16];
        if (!D50IOFlag) { rewind(D50); D50IOFlag=1; }
        if (!fgets(Str,16,D50)) return(1);
        if (sscanf(Str,"%d",Int) != 1) return(1);
        return(0);
}
```

```
int D50InputStr(char *Str)
{       if (!D50IOFlag) { rewind(D50); D50IOFlag=1; }
        if (!fgets(Str,32,D50)) return(1);
        return(0);
}

void GetError()
{       int             EN;
        char            Str[48];
        clrscr() ; gotoxy(6,27); puts("\7");
        if (D50InputInt(&EN));
        if (D50InputStr(DataStr));
        if (D50InputStr(Str));
        printf("Driver Error Has Occurred !!\n\n");
        printf("DAS-50 Error Number => %d\n\n",EN);
        printf("Error => %s\n\n",Str);
        printf("On Command Line of => %s\n\n",DataStr);
        exit(1);
}

void Pixel1()
{       putpixel(k,i,1); j=i+5; putpixel(k,i,1);
        j=i+10; putpixel(k,j,1); j=i+15; putpixel(k,i,1);
}

void Pixel2()
{       putpixel(k,i,1); j=k+10; putpixel(i,i,1);
}

void FunctMaxMinVals(double F[],int Num,int *Nmax,int *Nmin,int ind)
/***  Subroutine to find Maximum and Minimum Values of a Function  ***/
{               double          Ftest;
                int             max,min,mxindx,mnindx,slope,kk;
    if (ind==2) Num=Num/2;                  /* Scan Half FFT Values */
    max=0; min=0;                 /*Scan all Correlation Values*/
    if (F[1]<0.0 && F[2]<0.0) {
        if (F[1]<=F[2]) { slope=1; min++;
                        RelMin[1]=F[1]; RelMinIndx[1]=min; }
                        else { slope=-1; max++;
                        RelMax[1]=F[1]; RelMaxIndx[1]=max; } }
    if (F[1]>0.0 && F[2]>0.0) {
        if (F[1]>=F[2]) { slope=-1; max++;
                        RelMax[1]=F[1]; RelMaxIndx[1]=max; }
                        else { slope=1; min++;
                        RelMin[1]=F[1]; RelMinIndx[1]=min; } }
    if (F[1]>0.0 && F[2]<0.0) { slope=-1; max++;
                        RelMax[1]=F[1]; RelMaxIndx[1]=max; }
    if (F[1]<0.0 && F[2]>0.0) { slope=1; min++;
                        RelMin[1]=F[1]; RelMinIndx[1]=min; }
    kk=2; Ftest=F[1];
    while (kk<Num) {
        kk++; if (slope>0) goto L1; else goto L2;
L1:
        if ((F[kk]-Ftest)>=0.0)    {  Ftest=F[kk];
                                   mxindx=kk; continue; }
        else  { max++; RelMax[max]=Ftest;
        RelMaxIndx[max]=mxindx; slope=-1; continue;  }
L2:
        if ((F[kk]-Ftest)<=0.0)    {  Ftest=F[kk];
                                   mnindx=kk; continue; }
        else  { min++; RelMin[min]=Ftest;
        RelMinIndx[min]=mnindx; slope=1; continue;   }
        }
        *Nmax=max; *Nmin=min;
}
```

```
void PlotCorr(double A[],int Num,int NumMax,int NumMin,double *AbsMax,
          int *AbsMaxIndx,double *AbsMin,int *AbsMinIndx,int *RefLine)
/***   Suboutine to plot the Cross Correlation Function   ***/
{       double          ratio,pktopk,conversion;
        double          totmax,totmin,mxindx,mnindx;
        int             ii,jj,kk,reflin;
        totmax=RelMax[1]; mxindx=RelMaxIndx[1];
        totmin=RelMin[1]; mnindx=RelMinIndx[1];
        if (NumMax<2) goto L1;
        for (ii=2; ii<=NumMax; ii++) {
                if (RelMax[ii]>totmax)
                { totmax=RelMax[ii]; mxindx=RelMaxIndx[ii]; } }
L1:     if (NumMin<2) goto L2;
        for (ii=2; ii<=NumMin; ii++) {
                if (RelMin[ii]<totmin)
                { totmin=RelMin[ii]; mnindx=RelMinIndx[ii]; } }
L2:     pktopk=fabs(totmax)+fabs(totmin);
        if (pktopk==0.0) {  puts("pktopk = zero");
                        ratio=0.6; pktopk=1.0; goto L3;  }
        if (fabs(totmax)>=fabs(totmin)) ratio=fabs(totmax)/pktopk;
                else ratio=fabs(totmin)/pktopk;
L3:     reflin=(int)(320.0*ratio);
        conversion=260.0/pktopk;
        *AbsMax=totmax; *AbsMaxIndx=mxindx;
        *AbsMin=totmin; *AbsMinIndx=mnindx;
        *RefLine=reflin; moveto(10,reflin);
        for (ii=1; ii<=Num; ii++) {
                jj=ii+10; lineto(jj,reflin); }
        jj=reflin-(int)(A[1]*conversion); moveto(10,ii);
        for (ii=2; ii<=Num; ii++) {
                jj=ii+10;
                kk=reflin-(int)(A[ii]*conversion);
                lineto(jj,kk);    }
}

void PlotSpect(double A[],int Num,double *AMax,int *AMaxIndx)
/***   Subroutine to plot the Spectral Density function   ***/
{               double          pktopk,test,conversion;
                int             slope,mult,ii,jj,ll;
        if (Num==64) mult=16; else if (Num==128) mult=8;
        else if (Num==256) mult=4; else if (Num==512) mult=2;
        else if (Num==1024) mult=1; test=A[1];
        for (ii=2; ii<=Num/2; ii++) {
                if (A[ii]<test) continue;
                else { test=A[ii]; ll=ii;  } }
        pktopk=fabs(test); if (pktopk==0.0) { conversion=1.0; }
        conversion=250.0/pktopk;
        *AMax=pktopk; *AMaxIndx=ll;
        moveto(40,300);
        for (ii=1; ii<=Num/2; ii++) {
                jj=ii*mult+39; putpixel(jj,300,1);
                ll=300-(int)(A[ii]*conversion);
                lineto(jj,ll);      }
}

void GetKBDInfo(int Ind)
{
        char            *Str,buffer[10];
        buffer[0]=10;
        for (i=1; i<=9; i++) { buffer[i]=0; }
        Str=cgets(buffer); switch(Ind)  {
        case 1: sscanf(Str,"%1c%f",&Key,&dist); break;
        case 2: sscanf(Str,"%1c",&Key); break;
        case 3: sscanf(Str,"%f",&Rtemp); break;
        case 4: sscanf(Str,"%d",&Itemp); break;
        }
}
```

```
void Zero(double A[],int Num)
{       int             ii;
        for (ii=1; ii<=Num; ii++)  {
                A[ii]=0.0;       }
}

void GetWave(double SamplRate,int Num)
{       int             Status;
        char            outstr[48];
                /*  Set the Number of Samples  */
        sprintf(outstr,"SEt SAmples= %d\n",Num);
        if (D50Output(outstr)) GetError();
                /*  Set the On Board Memory Address  */
        sprintf(outstr,"SEt ADdress= 000\n");
        if (D50Output(outstr)) GetError();
                /*  Set the Sampling Rate  */
        sprintf(outstr,"SEt RAte= Int %e\n",SamplRate);
        if (D50Output(outstr)) GetError();
                /*  Acquire the Record  */
        if (D50Output("ACquire\n")) GetError();

        Status=0;         /* initialize status indicator  */
        do { if(D50Output("GEt STatus\n")) GetError();
                if(D50InputInt(&Status)) GetError();  }
        while ((Status & 0x10) != 0x10);
  }

void FltrRecord(double A[],double B[],float SamplRate,float scale,int Num)
{       int             ii,jj,D[2500];
        char            outstr[48];
        double          cx,cy0,cy1,cy2;
        double          prod,T,f1,f2,f3,f4;
                /*  Set the Memory Address  */
        sprintf(outstr,"SEt ADdress= %d\n",000);
        if(D50Output(outstr) != 0) GetError();
                /*  Transfer Data to Given Vector  */
        sprintf(outstr,"TRAnsfer %d %d %d %d\n",SEG(D),OFF(D),Num,1);
        if (D50Output(outstr) != 0) GetError();
/* Define Bandpass Filter parameters and first two Filtered Data Values
        T=2.0/SamplRate; prod=pi*Fc*T;
        cy0=1.0+pi*T*BW+prod*prod; cy1=2.0*(1.0-prod*prod);
        cy2=-1.0+pi*T*BW-prod*prod; cx=pi*Gain*T*RW;
        A[0]=cx*(scale*D[0]/2048.0)/cy0;
        B[0]=cx*(scale*D[1]/2048.0)/cy0;
        A[1]=(cx*(scale*D[2]/2048.0)+cy1*A[0])/cy0;
        B[1]=(cx*(scale*D[3]/2048.0)+cy1*B[0])/cy0;
/* Filter remaining input data resulting in A[] and B[] Filtered Data */
        for (ii=4, jj=2; ii<Num; ii+=2, jj++)  {
                f1=scale*D[ii-2]/2048.0;
                f2=scale*D[ii-1]/2048.0;
                f3=scale*D[ii]/2048.0;
                f4=scale*D[ii+1]/2048.0;
                A[jj]=(cx*(f3-f1)+cy1*A[jj-1]+cy2*A[jj-2])/cy0;
                B[jj]=(cx*(f4-f2)+cy1*B[jj-1]+cy2*B[jj-2])/cy0;  }
}

void Fourea(double datar[],double datai[],int Nft,int sion)
{       int             mmax,istep,m;
        double          tempr,tempi,wr,wi,theta;

/*    perform the bit reversal    */

      j=1; for (i=1; i<=Nft; i++) {
                if ((i-j)<0) goto L10;
                else goto L20;
```

```
L10:
        {       tempr=datar[j]; tempi=datai[j]:
                datar[j]=datar[i]; datai[j]=datai[i]:
                datar[i]=tempr; datai[i]=tempi; }
L20:
          m=Nft/2;
L30:
          if ((j-m)<=0) goto L40;
          j-=m; m=(m+1)/2; goto L30;
L40:
          j+=m;  }

/*    compute the butterflies   */
      mmax=1;
L50:
      if ((mmax-Nft)>=0) goto L70;
      istep=2*mmax;
      for (m=1; m<=mmax; m++) {
          theta=pi*sign*(m-1)/mmax;
          wr=cos(theta); wi=sin(theta); i=m;
          while (i<=Nft) {
                j=i+mmax;
                tempr=wr*datar[j]-wi*datai[j]:
                tempi=wr*datai[j]+wi*datar[j]:
                datar[j]=datar[i]-tempr;
                datai[j]=datai[i]-tempi;
                datar[i]=datar[i]+tempr;
                datai[i]=datai[i]+tempi;
                i+=istep;  }  }
          mmax=istep; goto L50;
L70:
      if (sign>0)  { for (i=1; i<=Nft; i++) {
                       datar[i]=datar[i]/1.0;
                       datai[i]=datai[i]/1.0; } }
}

void SpecDens(double A[],double B[],double E[],double SamplRate,int Num)
{       int             ii;
        double          AR,AI,BR,BI,C,D,ScaleFactor;
        ScaleFactor=4.0/(SamplRate*(float)Num);
        for (ii=1; ii<=Num; ii++) {
                AR=(A[ii]+A[(Num+1)-ii])/2.0;
                AI=(B[ii]-B[(Num+1)-ii])/2.0;
                BR=(B[ii]+B[(Num+1)-ii])/2.0;
                BI=(A[ii]-A[(Num+1)-ii])/(-2.0):
                C=AR*BR+AI*BI;
                D=AR*BI-AI*BR;
                E[ii]=ScaleFactor*sqrt(C*C+D*D):   }
        /*  Smooth the Spectral Density Function  */
                A[1]=0.5*(E[1]+E[2]):
                A[Num]=0.5*(E[Num-1]+E[Num]):
        for (ii=2; ii<=Num-1; ii++) {
                A[ii]=0.25*(E[ii-1]+E[ii+1])+0.5*E[ii]: }
        for (ii=1; ii<=Num; ii++) {
                E[ii]=A[ii];      }
}

void Correlate(double A[],double B[],int Delay,int Ncorr)
{       int             TwoNcorr,ii,jj,kk;
        double          C[1030],D[1030],ScaleFactor,Real,Imag:
        TwoNcorr=2*Ncorr;
        for (jj=1,ii=Delay; jj<=Ncorr; ii++,jj++) {
                C[jj]=A[ii]; D[jj]=B[ii]; }
        for (jj=Ncorr+1; jj<=TwoNcorr; jj++) {
                C[jj]=0.0; D[jj]=0.0;  }
        Zero(A,TwoNcorr);
```

```
        Fourea(C,A,TwoNcorr,-1);
        Zero(B,TwoNcorr);
        Fourea(D,B,TwoNcorr,-1);
        for (ii=1; ii<=TwoNcorr; ii++) {
                Real=C[ii]*D[ii]+A[ii]*B[ii];
                Imag=C[ii]*B[ii]-A[ii]*D[ii];
                C[ii]=Real/(float)Ncorr;
                D[ii]=Imag/(float)Ncorr;        }
        Fourea(C,D,TwoNcorr,1);
        for (ii=1; ii<=Ncorr; ii++) {
/* ScaleFactor=(float)Ncorr/((float)Ncorr+1.0-(float)ii);   */
                ScaleFactor=(float)Ncorr+1.0-(float)ii;
        /*      A[ii]=ScaleFactor*D[ii];
                B[ii]=ScaleFactor*D[ii];     */
                A[ii]=D[ii]/ScaleFactor;
                B[ii]=D[ii]/ScaleFactor;   }
}

float Window(int Wtype,int Num,int Index)
/***   Subroutine to allow for variable data windowing    ***/
{               int             L;
                float           windoval;
        windoval=0.0;
        if (Wtype<1 || Wtype>6) goto END;
        if (Index<1 || Index>Num) goto END;
        windoval=1.0;
        if (Wtype==1) goto END;                  /* Rectangular Window */
        if (Wtype==2) { L=(Num-2)/10;
                if (Index<=L)                  /* Tapered Rect  Window */
        windoval=0.5*(1.0-cos(pi*(float)Index/(float)(L+1)));
                if (Index>(Num-L-2))
        windoval=0.5*(1.0-cos(pi*(float)(Num-Index-1)/(float)(L+1)));  }
        if (Wtype==3)                             /* Triangular Window */
        windoval=1.0-fabs(1.0-(float)(2*Index)/(float)Num);
        if (Wtype==4)                               /* Hanning Window */
        windoval=0.5*(1.0-cos(pi*(float)(2*Index)/(float)Num));
        if (Wtype==5)                               /* Hamming Window */
        windoval=0.54-0.46*cos(pi*(float)(2*Index)/(float)Num);
        if (Wtype==6)                              /* Blackman Window */
        windoval=0.42-0.5*cos(pi*(float)(2*Index)/(float)Num)
                        +0.08*cos(pi*(float)(4*Index)/(float)Num);
END:            return(windoval);
}
  /*************************************************************************/
  /**                                                                     **/
  /**     START THE MAIN PROGRAM FOR WEB TENSION MEASUREMENT              **/
  /**                                                                     **/
  /*************************************************************************/

main()
{
  Fc=1000.0; BW=1500.0; Gain=2.0;    /* Digital Filter default parameters
  NumDelay=0; NS=2400; rate=4.0e+5;     /* initial sample number and rate

  D50=fopen("$DAS50","r+");
  if(D50Output("Clear\n")) GetError();

  clrscr() ;                          /* Use BIOS scroll to clear screen */
  gotoxy(10,3);                       /* Set cursor to Row 1, Col 7 */
  puts("  Program WH1FC - Single Point Measurement - Fast Correlation\n\n
  puts("   This program is for the operation of the tension measurement")
  puts("system developed through Dr. Richard Lowery of Oklahoma State ");
  puts("University for the purpose of On-Line Web Tension Analysis.  This
  puts("program acquires one record from a Metrabyte DAS50 A-D board.");
  puts("Fourier transform techniques are used to calculate the Cross ");
  puts("Correlation function and Spectral Density function for the data")
```

```
  puts("record.  This procedure provides delta T and characteristic");
  puts("frequency values which are used to calculate web tension through"
  puts("use of the ribbon equation.  Digital filtering is used where");
  puts("filter parameters are user adjustable to allow for optimal signal
  puts("quality.  Default filter specifications are:  Filter Gain = 2.0,"
  puts("Filter Center Frequency= 1000 Hz,  Filter Bandwidth= 1500 Hz.\n\n
  puts("ENTER MATERIAL DENSITY (LBM/IN-IN)");
  GetKBDInfo(3); puts(""); rhoweb=Rtemp;
  puts("Enter type of window to use");
  GetKBDInfo(4); puts(""); Windo=Itemp;

  /* ------------------------------------------------------------*/
  /* Set the DAS-50 to take 2400 samples, alternating between   */
  /* Channels 0 & 1, in the Trace-After Trigger mode.  Note that*/
  /* all the following Print statements could be abbreviated    */
  /* and put into one print statement                           */
  /*------------------------------------------------------------*/

        if (D50Output("SEt CHannels=0&1\n")) GetError() ;
        if (D50Output("SEt RAnge=+-5V\n")) GetError() ;
        if (D50Output("SEt TRigger Mode=7\n")) GetError() ;
        if (D50Output("SEt TRig Volt=0.2\n")) GetError() ;
        if (D50Output("SEt STart=AFter\n")) GetError() ;

/***  Initialize graphics facility to exhibit an acquired waveform  ***/
L250:   grdriver=5; grmode=3;
        initgraph(&grdriver,&grmode,"");
        setviewport(0,0,639,349,1);

/* Sample and Filter Data Set for Default of Modified Filter Parameters *
L999:   clearviewport(); GetWave(rate,NS);
        FltrRecord(XDAT,YDAT,rate,65.0,NS);

 /*  draw a grid on the screen   */
        gotoxy(1,0); lineto (638,0); lineto (638,260);
        lineto (1,260); lineto (1,0); line(1,130,638,130);

        for (i=0,k=0; i<=240; i+=20) {
                for (k=0; k<=639; k+=20) {
                        switch(k)  {
                        case 80:
                        case 160:
                        case 240:
                        case 320:
                        case 400:
                        case 480:
                        case 560: Pixel1(); Pixel2(); break;
                        default:  Pixel2();     }  }  }

 /*   put the data traces on the screen in screen coordinates   */
        j=65-(int)XDAT[NumDelay]; putpixel(1,j,1);
        for (i=(NumDelay+1); i<=(NumDelay+640); i++) {
                k=i-NumDelay; j=65-(int)XDAT[i]; putpixel(k,j,1);   }
        j=195-(int)YDAT[NumDelay]; putpixel(1,j,1);
        for (i=(NumDelay+1); i<=(NumDelay+640); i++) {
                k=i-NumDelay; j=195-(int)YDAT[i]; putpixel(k,i,1);  }

        textheight(&Key); if (Key==8) i=2; else i=1;
        settextstyle(0,0,i);

  outtextxy(1,264,"Press Space Bar to Repeat Process");
  outtextxy(1,276,"Press 'l' or 'r' to shift pattern left or right");
  outtextxy(1,288,"Press 'i' or 'd' to increase or decrease sample rate"
  outtextxy(1,300,"Press 'f' to alter digital filter center frequency");
  outtextxy(1,312,"Press 'b' to alter digital filter bandwidth");
  outtextxy(1,324,"Press 'g' to alter digital filter gain");
  outtextxy(1,336,"Press 't' when ready to begin web tension test");
```

```
L400:   Key=getch(); switch(Key)  {

/***  Case of Left or Right Waveform Positioning  ***/
        case 'l':
        case 'L': NumDelay=NumDelay+40; goto L999;
        case 'r':
        case 'R': NumDelay=NumDelay-20;
                  if (NumDelay<0) NumDelay=0;
                  else goto L999; break;

/***  Case of Increase or Decrease of Sample Rate  ***/
        case 'i':
        case 'I': rate=rate+1.0e+5; goto L999;
        case 'd':
        case 'D': rate=rate-1.0e+5; goto L999;

/***  Case of Space Bar -- Repeat Waveform Viewing  ***/
        case ' ': goto L999;

/***  Case of Digital Filter Center Frequency Adjustment  ***/
        case 'f':
        case 'F': clearviewport(); restorecrtmode();
        gotoxy(10,3); puts(" FILTER CENTER FREQUENCY ADJUSTMENT \n");
        puts("Enter 'lxxx' or 'rxxx' to adjust the filter");
        puts("center frequency left or right, respectively,");
        puts("by an amount xxx in Hertz\n\n");
L950:   GetKBDInfo(1); switch(Key)  {
                case 'r':
                case 'R': Fc=Fc+dist; goto L250;
                case 'l':
                case 'L': Fc=Fc-dist; goto L250;
                default: puts("Invalid entry - try again\n");
                         goto L950;  }

/***  Case of Digital Filter Bandwidth Adjustment  ***/
        case 'b':
        case 'B': clearviewport(); restorecrtmode();
        gotoxy(10,3); puts("FILTER BANDWIDTH ADJUSTMENT \n");
        puts("Enter 'ixxx' or 'dxxx' to adjust the filter");
        puts("bandwidth increase or decrease, respectively.");
        puts("by an amount xxx in Hertz\n\n");
L1250:  GetKBDInfo(1); switch(Key)   {
                case 'i':
                case 'I': BW=BW+dist; goto L250;
                case 'd':
                case 'D': BW=BW-dist; goto L250;
                default: puts("Invalid entry - try again\n");
                         goto L1250;   }

/***  Case of Digital Filter Gain Adjustment  ***/
        case 'g':
        case 'G': clearviewport(); restorecrtmode();
        gotoxy(10,3); puts(" FILTER GAIN ADJUSTMENT ");
        puts("Enter new filter gain specification.\n");
        GetKBDInfo(3); Gain=Rtemp; goto L250;

/***  Case of Web Tension Test Desired  ***/
        case 't':
        case 'T': goto L1500;
        default:  puts("'\x7'"); goto L400;
        }

/*** Set cross correlation shifts NumCorr depending on sample rate ***/
L1500:  clearviewport();
        if (rate==2.0e+5)     { SRdivider=8; NumCorr=256; NS=1296; }
```

```
        if (rate==3.0e+5)      { SRdivider=12; NumCorr=512; NS=1600; }

        if (rate==4.0e+5)      { SRdivider=16; NumCorr=512; NS=2048; }

        if (rate==5.0e+5)      { SRdivider=20; NumCorr=512; NS=2496; }

        if (rate==6.0e+5)      { SRdivider=24; NumCorr=512; NS=2496; }

        if (rate>=7.0e+5) {  clearviewport(); restorecrtmode();
                puts("Sampling rate selected is not supported\n");
                        goto ProgEnd; }

****    Obtain the data record according to NS and rate values  ****/

    GetWave(rate,NS);                                    /*Acquire Data*/
    FltrRecord(XDAT,YDAT,rate,5.0,NS); j=1;             /*Filter Record*/
    for (i=1; i<=NS/2; i+=SRdivider,j++) {
        WindoVal=Window(Windo,64,j);                  /*Apply Windowing*/
        X[j]=WindoVal*XDAT[i];        }
    Correlate(XDAT,YDAT,NumDelay,NumCorr);                /*Correlate*/
    FunctMaxMinVals(XDAT,NumCorr,&NumRelMax,&NumRelMin,1);
    PlotCorr(XDAT,NumCorr,NumRelMax,NumRelMin,&AbsMax,
        &AbsMaxIndx,&AbsMin,&AbsMinIndx,&reflin);   /*Plot Correlation*/
    l=reflin+12; moveto(10,l); outtext("1");
    itoa(NumCorr,DataStr,10); moveto(NumCorr,l);
    outtext(DataStr); period=2.0/rate; j=10;
    for (i=1; i<=NumRelMax; i++) {
        moveto(440,j); outtext("Rel Max Val at");   /*Graphical Display*/
        itoa(RelMaxIndx[i],DataStr,10);               /*which notes max*/
        moveto(570,j); outtext(DataStr); j+=12; }   /*and min values of*/
    for (i=1; i<=NumRelMin; i++) {                  /*Cross Correlation*/
        moveto(440,j); outtext("Rel Min Val at");         /*function*/
        itoa(RelMinIndx[i],DataStr,10);
        moveto(570,j); outtext(DataStr); j+=12; }

        itoa(AbsMaxIndx,DataStr,10);              /*DeltaT value based on*
        deltaT=period*(float)AbsMaxIndx;     /*Maximum Correlation value*

    moveto(20,324); outtext("Max Correlation at Delay Point ");
    outtext(DataStr); moveto(360,324); gcvt(period,6,DataStr);
    outtext("Sample Period = "); outtext(DataStr);
    moveto(560,324); outtext("sec."); moveto(20,336);
    gcvt(deltaT,6,DataStr); outtext("Delta T for this Test is ");
    outtext(DataStr); moveto(380,336);
    outtext("Press any Key to Continue"); getch();

*      Sort through the Upstream Microphone waveform record to achieve */
*     a reduced sample rate record for FFT processing.  Obtain the     */
*     web characteristic frequency from the resultant Spectral Density */
*     Lower sample rate allows for higher frequency domain resolution. */

    rate=rate/((float)SRdivider*2.0);
    freqincrement=rate/128.0; j=1;
    sum=0.0; for (j=1; j<=64; j++) { sum=sum+X[j]; }
    sum=sum/64.0;                                       /*Compute record*/
    for (j=1; j<=64; j++) {     X[j]=X[j]-sum; }          /*average and*/
    for (j=65; j<=128; j++) { X[j]=0.0; }                /*subtract from*/
                                                           /*each point*/
    Zero(Z,128); Fourea(X,Z,128,-1);
    SpecDens(X,Z,RXY,rate,128);                        /*Compute Spectral*/
    clearviewport();                                   /*Density function*/
    PlotSpect(RXY,128,&MaxFreq,&MaxFreqIndx);
    moveto(34,310); outtext("k=1"); moveto(520,310);
    outtext("k=64"); gcvt(rate,6,DataStr);          /*Graphical Display*/
    moveto(180,320); outtext("SAMPLING RATE = ");        /*with support*/
    outtext(DataStr); outtext(" HZ ");                    /*information*/
```

```
    gcvt(freqincrement,6,DataStr); moveto(70,335);
    outtext("FOR 128 PT FFT:  FREQUENCY INCREMENT = ");
    outtext(DataStr); outtext(" HZ ");
    webfreq=(float)MaxFreqIndx*freqincrement;
    itoa(MaxFreqIndx,DataStr,10);
    moveto(350,60); outtext("K = ");                    /*Select Char.*/
    outtext(DataStr); moveto(420,60);                 /*Frequency value*/
    gcvt(webfreq,6,DataStr); outtext(" ; FREQ = ");       /*and enter.*/
    outtext(DataStr); outtext(" HZ "); moveto(350,100);
    outtext("Enter Web Freq Value"); GetKBDInfo(3);
    webfreq=Rtemp; restorecrtmode();

    kair=2.0*pi*webfreq/cair;                        /*Wave Number for Air*/
    vel=(2.0/deltaT);                                    /*Web Phase Speed*/
    kweb=2.0*pi*webfreq/vel;                        /*Wave Number for Web*/
    temp=kweb*kweb-kair*kair;
    if (temp<0.0)  {puts("### BAD VALUE OF DELTAT OR WEBFREQ ###");
             puts(" No Tension Value Processed"); goto ProgEnd; }
    sum=rhoweb+2.0*rhoair/sqrt(temp);

/**Use Ribbon Equation to Provide Air Loading Compensated Tension Value*/
    Te=rhoweb*vel*vel/386.0;

/** Use Membrane Equation to Provide In-Vacuo Tension Estimate **/
    Tc=sum*vel*vel/386.0;

    printf(" Characteristic Frequency calculated to be %f\n\n",webfreq);
    printf(" Estimated Membrane Equation Tension = %f\n\n",Te);
    printf(" Calculated Ribbon Equation Tension = %f\n\n",Tc);
ProgEnd:;
}
```

APPENDIX F

COMPUTER PROGRAM LISTINGS FOR MULTI-POINT/ PROFILE TENSION MEASUREMENTS

```
/*********************************************************************/
/***   Program WH3DF (Digital Filtering) -- Web Handling Program ***/
/***   to take records from the pulsing system, perform digital  ***/
/***   filtering on the raw signal, and calculate a time of      ***/
/***   flight DeltaT value for the record.  Additional records   ***/
/***   are similarly acquired at a reduced sample rate, formed   ***/
/***   into a pseudoperiodic record, upon which the Spectral     ***/
/***   Density function is calculated, resulting in a Character- ***/
/***   istic Frequency value.  DeltaT and Characteristic Freq.   ***/
/***   are converted to tension through the Ribbon Equation.  A  ***/
/***   A graphical summary of the Tension Profile for a web      ***/
/***   span is provided by the program.                          ***/
/*********************************************************************/

      #include                  <stdio.h>
      #include                  <stdlib.h>
      #include                  <dos.h>
      #include                  <graphics.h>
      #include                  <string.h>
      #include                  <alloc.h>
      #include                  <math.h>
      #include                  <process.h>
      #include                  <conio.h>
      #define   rhoair        4.373e-5L       /*  lbm/in-cubed  */
      #define   cair          13500.0L        /*     in/sec     */
      #define   radius        1.25            /*traverse near radius*/
      #define   pi            3.14159L

      FILE              *D50;
      char              DataStr[48],Key;
      int               i,j,k,l,dir,coun,instr;
      int               sl,NS,grdriver,grmode;
      int               ndel,fdel,nspan,reflin,maxindex;
      int               ycoord,xcoor,xsum,intind,FNS;
      int               ncy,cycles,ncorr,D50IOFlag,Itemp;
      double            Tmax,Tmin,rhoweb,frate,rate;
      double            refedge,totmax,vel,f2,f3;
      double            conv,period,deltaT,Tens,Tcalc,temp;
      double            webfreq,T,webwidth,transvel,trvel,Tensavg;
      int               maxptr[20],xcoord[10],deltaX[10];
      double            max[20],Tension[10],Fc,BW,Gain,SRdivider;
      double            X[1250],Y[1250],XDAT[1250],YDAT[1250];
      double            test,kweb,kair,sum;
      div_t             delw;
      float             dist,Rtemp;

int SEG(int far *Var)
{       unsigned        seg;
        seg=FP_SEG(Var);
        return(seg);
}

int OFF(int far *Var)
{       unsigned        offs;
        offs=FP_OFF(Var);
        return(offs);
}

int D50Output(char *Str)
{       if(D50IOFlag) {rewind(D50); D50IOFlag=0; }
        if(fprintf(D50,"%s",Str)==EOF) return(1);
        if(fflush(D50)==EOF) return(1);
        return(0);
}
```

```
int D50InputInt(int *Int)
{       char    TmpStr[32] ;
        if(!D50IOFlag) {rewind(D50); D50IOFlag=1; }
        if(!fgets(TmpStr,32,D50)) return(1);
        if(sscanf(TmpStr,"%d",Int) != 1) return(1);
        return(0);
}

int D50InputStr(char *Str)
{       if (!D50IOFlag) {rewind(D50); D50IOFlag=1;}
        if (!fgets(Str,32,D50)) return(1);
        return(0);
}

void GetError()
{       int             EN;
        char            Str[48];
        restorecrtmode(); clrscr() ;
        gotoxy(6,27); puts("\7");
        if(D50InputInt(&EN));
        if(D50InputStr(Str));
        if(D50InputStr(DataStr));
        puts(" Driver Error Has Occurred !!");
        printf(" DAS-50 Error Number => %d\n",EN);
        printf(" Error => %s\n",DataStr);
        printf(" On Command Line of => %s\n",Str);
        exit(1);
}

void Pixel1()
{       putpixel(k,i,1); j=i+5; putpixel(k,j,1);
        j=i+10; putpixel(k,j,1); j=i+15; putpixel(k,j,1);
}

void Pixel2()
{       putpixel(k,i,1); j=k+10; putpixel(j,i,1);
}

void GetKBDInfo(int Ind)
{       char            *Str;
        char            buffer[10];
        buffer[0]=10;
        for (i=1; i<=9; i++) {
                buffer[i]=0; }
        Str=cgets(buffer); switch(Ind)  {
        case 1: sscanf(Str,"%1c%f",&Key,&Rtemp); break;
        case 2: sscanf(Str,"%1c",&Key); break;
        case 3: sscanf(Str,"%f",&Rtemp); break;
        case 4: sscanf(Str,"%d",&Itemp); break;
        }
}

void GetWave(double SamplRate,int Num)
/***   Subroutine to Acquire a Data Record From DAS-50 A-D Board  ***/
{       int             Status;
        char            outstr[32];
      sprintf(outstr,"SEt SAmples=%d\n",Num);       /*set number of samp
      if (D50Output(outstr)) GetError();
      sprintf(outstr,"SEt RAte= Int %e\n",SamplRate);  /*set sampling r
      if (D50Output(outstr)) GetError();
      sprintf(outstr,"SEt ADdress= 000\n");      /*set on board mem addr
      if (D50Output(outstr)) GetError();
      if(D50Output("ACquire\n")) GetError();              /*acquire rec
      Status=0;                                  /*initialize status indica
                              /*device clears bits 3 & 4 when D/A compl
```

```
     do  {  if (D500utput("GEt STatus\n")) GetError();
            if (D50InputInt(&Status)) GetError();     }
            while((Status & 0x10) != 0x10);
}

void FilterRecord(double A[],double B[],double SamplRate,
                                        float Scale,int Num)
/***  Subroutine to Digitally Filter a Raw Data Record  ***/
{
        char            outstr[32];
        int             ii,jj,D[2500];
        double          cx,cy0,cy1,cy2,T,prod,f1,f2,f3,f4;
        strcpy(outstr,"SEt ADdress 000\n");  /*   set address of record
        if (D500utput(outstr)) GetError();
        sprintf(outstr,"TRAnsfer %d %d %d %d\n",SEG(D),OFF(D),
            Num,1);                              /*  transfer the data  */
        if(D500utput(outstr)) GetError();

        T=2.0/SamplRate; prod=pi*Fc*T;
        cy0=1.0+pi*T*BW+prod*prod; cy1=2.0*(1.0-prod*prod);
        cy2=-1.0+pi*T*BW-prod*prod; cx=pi*Gain*T*BW;
        A[0]=cx*(Scale*D[0]/2048.0)/cy0;
        B[0]=cx*(Scale*D[1]/2048.0)/cy0;
        A[1]=(cx*(Scale*D[2]/2048.0)+cy1*A[0])/cy0;
        B[1]=(cx*(Scale*D[3]/2048.0)+cy1*B[0])/cy0;
        for (ii=4, jj=2; ii<=Num; ii+=2, jj++)  {
                f1=Scale*D[ii-2]/2048.0;
                f2=Scale*D[ii-1]/2048.0;
                f3=Scale*D[ii]/2048.0;
                f4=Scale*D[ii+1]/2048.0;
                A[jj]=(cx*(f3-f1)+cy1*A[jj-1]+cy2*A[jj-2])/cy0;
                B[jj]=(cx*(f4-f2)+cy1*B[jj-1]+cy2*B[jj-2])/cy0;  }
}

void Correlate(double A[],double B[],int CorrSpan, int CorrShifts,int
/* Subroutine to compute Cross Correlation function                Delay)
{               double          C[1250],sum,prod;
                int             r,rdiff,totrdiff,ii,jj;
        for (r=1;r<=CorrShifts; r++) {
                rdiff=CorrSpan-r;
                totrdiff=CorrSpan+Delay+1-r;
                sum=0.0;
                for (ii=Delay; ii<=totrdiff; ii++) {
                        jj=ii+r;
                        prod=A[ii]*B[jj];
                        sum=sum+prod;    }
                C[r]=sum/(float)rdiff;  }
        for (ii=1; ii<=CorrShifts; ii++) {
                A[ii]=C[ii];    }
}

void Fourea(double datar[],double datai[],int Nft)
{       int                     mmax,istep,m;
        double                  tempr,tempi;
        double                  wr,wi,theta;

  /*    perform the bit reversal     */
        j=1;
        for (i=1; i<=Nft; i++) {
        if ((i-j)<0) goto L10;
        else goto L20;
L10:
   {    tempr=datar[j]; tempi=datai[j];
        datar[j]=datar[i]; datai[j]=datai[i];
        datar[i]=tempr; datai[i]=tempi;  }
```

```
L20:
        m=Nft/2;
L30:
        if ((j-m)<=0) goto L40;
        j-=m; m=(m+1)/2; goto L30;
L40:
        j+=m; }

/*      compute the butterflies   */
        mmax=1;
L50:
        if((mmax-Nft)>=0) goto L70;
        istep=2*mmax;
        for (m=1; m<=mmax; m++) {
                theta=-pi*(m-1)/mmax;
                wr=cos(theta);
                wi=sin(theta); i=m;
                while(i<=Nft) {
                        j=i+mmax;
                        tempr=wr*datar[j]-wi*datai[j];
                        tempi=wr*datai[j]+wi*datar[j];
                        datar[j]=datar[i]-tempr;
                        datai[j]=datai[i]-tempi;
                        datar[i]=datar[i]+tempr;
                        datai[i]=datai[i]+tempi;
                        i+=istep;  } }
                mmax=istep; goto L50;
L70: ;
}

void SpecDens(double A[],double B[],double E[],double SamplRate,int Num)
/* Subroutine to compute Spectral Density Function for A and B records */
{               int             ii;
                double          AR,AI,BR,BI,C,D,ScaleFactor;
        ScaleFactor=4.0/(SamplRate*(float)Num);
        for (ii=1; ii<=Num; ii++) {
                AR=(A[ii]+A[(Num+1)-ii])/2.0;
                AI=(B[ii]-B[(Num+1)-ii])/2.0;
                BR=(B[ii]+B[(Num+1)-ii])/2.0;
                BI=(A[ii]-A[(Num+1)-ii])/(-2.0);
                C=AR*BR+AI*BI;
                D=AR*BI-AI*BR;
                E[ii]=ScaleFactor*sqrt(C*C+D*D);  }
        A[1]=0.5*(E[1]+E[2]);
        A[Num]=0.5*(E[Num-1]+E[Num]);
        for (ii=2; ii<=Num-1; ii++) {
                A[ii]=0.25*(E[ii-1]+E[ii+1])+0.5*E[ii]; }
}

void StepMotor(int dir,int coun)
{
        #define         ioport1         688
        #define         ioport2         689
        int             jj;
        outportb(ioport2,dir);          /*set the direction of rotation*/
        for (jj=1; jj<=coun; jj++) {
        outportb(ioport1,011);            /* set the port bit 0 high   */
        delay(05);                        /* delay for 5 milli-seconds */
        outportb(ioport1,010);            /* reset port bit 0 to zero  */
        delay(05);    }                   /* delay for 5 milli-seconds */
        outportb(ioport1,000);            /* disable the stepper motor */
}

  /*************************************************************************
  /**                                                                     **
  /**      START THE MAIN PROGRAM FOR WEB TENSION MEASUREMENT             **
  /**                                                                     **
```

```
main()
{
D50IOFlag=0; rate=4.0e+5; ndel=0; NS=2400;
Gain=2.0; Fc=1000.0; BW=1500.0;     /* Default Digital Filter Specs. */
D50=fopen("$DAS50","r+");           /*open the path to the DAS50 board*/
if (D50Output("Clear\n")) GetError();

clrscr(); gotoxy(1,3);              /* Set cursor to Row 1, Col 7 */
puts(" Program WH3DF - Multi Point Measurement - Digital Filtering\n\n"
puts("   This program is for the operation of the tension measurement")
puts("system developed through Dr. Richard Lowery of Oklahoma State ");
puts("University for the purpose of On-Line Web Tension Analysis.  A");
puts("Metrabyte DAS-50 A-D board is used to acquire data records where"
puts("upon digital filtering is used to condition the signal.  Filter")
puts("parameters are user adjustable.  Cross Correlation and Spectral")
puts("Density functions are computed which yield time of flight and ");
puts("characteristic frequency information, respectively.  Ribbon   ");
puts("Equation is used to convert this information to tension values.")
puts("A stepper motor system moves the transducer head across the web "
puts("width to obtain a tension profile across the web width. Stepper")
puts("Motor increments may be evenly or unevenly spaced across the web"
puts("span.  A graphical tension summary is output showing the test");
puts("locations and associated tension values.\n");
puts(" ENTER THE WIDTH OF THE WEB BEING TESTED IN INCHES  ");
      GetKBDInfo(3); puts(""); webwidth=Rtemp;
puts(" ENTER THE WEB MATERIAL DENSITY IN LBM/SQUARE INCH ");
      GetKBDInfo(3); puts(""); rhoweb=Rtemp;
puts(" ENTER THE TRANSPORT VELOCITY IN FEET/MINUTE ");
      GetKBDInfo(3); puts(""); transvel=Rtemp;
puts("Press any key to continue . . ."); getch();

      clrscr(); gotoxy(1,3);
puts("         INITIAL  POSITIONING  OF  TRANSDUCER  HEAD \n\n");
puts("    Position the transducer head to the desired initial ");
puts("    location through interactive use of this computer.  ");
puts("    Enter the desired adjustment in the form 'LXX<cr>'  ");
puts("    for adjustment left XX inches or 'RXX<cr>' for adjust-");
puts("    ment right XX inches.  When the move is completed, the");
puts("    computer will poll the user as follows:  Enter 'R<cr>'");
puts("    to readjust or 'C<cr>' to continue.  Entering R will  ");
puts("    allow repositioning left or right and entering C will ");
puts("    allow for continuation of the Web Handling program.\n");
L100:
puts("    Initial Position Adjustment:  Enter 'LXX<cr>' to move ");
puts("              left or 'RXX<cr>' to move right             ");
GetKBDInfo(1); puts(""); dist=Rtemp; switch(Key)  {
case 'r':                                  /*right - set bit D/O-8 to*/
case 'R': dir=0xdf; break;                /*one for clockwise rotation*/
case 'l':                                   /*left  - set bit D/O-8 to*/
case 'L': dir=0xde; break;                    /*zero for CCW rotation*/
default: puts("Invalid Entry - Try Again\n"); goto L100;  }

      coun=dist*36.0/(radius*pi);          /*Move the Transducer Head*/
      StepMotor(dir,coun);                  /* by the amount entered */
L140:
      puts(" Enter 'R<cr>' to readjust or 'C<cr>' to continue ");
      GetKBDInfo(2); puts(""); switch(Key)   {
      case 'r':
      case 'R':  goto L100;                        /* Readjustment desired *
      case 'c':
      case 'C':  break;                            /* Continuation desired *
      default: puts("Invalid Entry - Try Again\n"); goto L140;
      }
L150:
      puts(" Enter distance between Transducer and Web Edge ");
      GetKBDInfo(3); puts(""); refedge=Rtemp;
      clrscr(); gotoxy(1,3);
```

```
  puts("                    STEPPER  MOTOR  OPTIONS  \n\n");
  puts("    Two options are available for Stepper Motor spacing ");
  puts("    during a test:  Equal or Unequal Intervals during web ");
  puts("    traversal.  Also, direction of traversal may be to the");
  puts("    left or right.  Enter the desired options as polled for");
  puts("    below.\n");
L160:
  puts("    DIRECTION OF TRAVERSAL:  Enter 'R<cr>' for traversal");
  puts("        right or 'L<cr>' for traversal left ");
  GetKBDInfo(2); puts(""); switch(Key)   {
        case 'r':
        case 'R':  dir=0xdf; break;          /*right-D/O code for CW rot.
        case 'l':
        case 'L':  dir=0xde; break;          /*left-D/O code for CCW rot.
        default: puts("Invalid Entry - Try Again\n\n"); goto L160;  }
L190:
  puts("    EQUAL OR UNEQUAL TRAVERSAL INTERVALS:  Enter 'E<cr>' ");
  puts("      equal intervals or 'U<cr>' for unequal intervals ");
  GetKBDInfo(2); puts(""); switch(Key)   {
  case 'u':                                  /*indicates unequal intervals
  case 'U':  intind=1; goto L220;
  case 'e':
  case 'E':  intind=0; break;                /* indicates equal intervals
  default: puts("Invalid Entry - Try Again \n\n"); goto L190;   }

    puts(" Enter the Interval Length 'XX' where XX is in inches ");
    GetKBDInfo(3); puts(""); dist=Rtemp; intind=0;
    xcoor=(int)(webwidth-refedge);                  /*Calculate number of
    delw= div(xcoor,(int)dist);                  /*tests if equal spacing
    cycles=delw.quot;
    if ((delw.rem/dist)>0.2) cycles=cycles+1;
    printf(" * * *   %d Tests Will Be Performed   * * *\n\n",cycles);
    printf(" * * *   At %f Inch Intervals   * * *\n\n",dist);
    delay(2000);
L220:
  /* ------------------------------------------------------------*/
  /* Set the DAS-50 to take NS samples, alternating between     */
  /* Channels 0 & 1, in the Trace-After Trigger mode.  Note that*/
  /* all the following Print statements could be abbreviated    */
  /* and put into one print statement                           */
  /*------------------------------------------------------------*/

        if (D50Output("SEt CHannels=0&1\n"))   GetError() ;
        if (D50Output("SEt RANge=+-5V \n"))    GetError() ;
        if (D50Output("SEt TRigger Mode=7\n")) GetError() ;
        if (D50Output("SEt TRig Volt=0.40\n")) GetError() ;
        if (D50Output("SEt STart=AFter\n"))    GetError() ;

/*** Initialize graphics facility to exhibit the acquired waveform ***/
L250:   grdriver=5; grmode=3;
        initgraph(&grdriver,&grmode,"");
        setviewport(0,0,639,349,1);
L270:   clearviewport();
        GetWave(rate,NS);                  /*Acquire waveform from DAS-50*/
        FilterRecord(XDAT,YDAT,rate,65.0,NS);      /*Filter the Record*/

/*****************  draw a grid on the screen   ********************/
        gotoxy(1,0); lineto (638,0); lineto (638,260);
        lineto (1,260); lineto (1,0); line(1,130,638,130);

        for (i=0,k=0; i<=240; i+=20) {
                for (k=0; k<=639; k+=20) {
                switch(k)  {
                case 80:
                case 160:
```

```
                case 240:
                case 320:
                case 400:
                case 480:
                case 560: Pixel1(); Pixel2(); break;
                default:  Pixel2();     } } }

/******* put the data traces on the screen in screen coordinates ******/
L66:    for (i=ndel; i<=(ndel+640); i++) {
                k=i-ndel; j=65-(int)XDAT[i]; putpixel(k,j,1);
                j=195-(int)YDAT[i]; putpixel(k,j,1);  }

/*********  Present User Options for Signal Manipulation  ************/
    outtextxy(1,264,"Press Space Bar to Repeat Process");
    outtextxy(1,276,"Press 'l' or 'r' to shift pattern left or right");
    outtextxy(1,288,"Press 'i' or 'd' to increase or decrease sample ra
    outtextxy(1,300,"Press 'f' to alter digital filter center frequency
    outtextxy(1,312,"Press 'b' to alter digital filter bandwidth");
    outtextxy(1,324,"Press 'g' to alter digital filter gain");
    outtextxy(1,336,"Press 't' when ready to begin web tension test");

L400:   Key=getch(); switch(Key)  {

/***   Case of Left or Right Waveform Positioning  ***/
        case 'l':
        case 'L':  ndel=ndel+40; goto L270;
        case 'r':
        case 'R':  ndel=ndel-20; if (ndel<0) ndel=0;
                        else goto L270; break;

/***   Case of Increase or Decrease Sampling Rate  ***/
        case 'i':
        case 'I':  rate=rate+1.0e+5; goto L270;
        case 'd':
        case 'D':  rate=rate-1.0e+5; goto L270;

/***   Case of Space Bar -- Repeat Waveform Viewing  ***/
        case ' ':  goto L270;

/***   Case of Digital Filter Center Frequency Adjustment  ***/
        case 'f':
        case 'F':  clearviewport(); restorecrtmode();
        gotoxy(10,3); puts("FILTER CENTER FREQUENCY ADJUSTMENT\n");
        puts("Enter 'lxxx' or 'rxxx' to adjust the filter");
        puts("center frequency left or right, respectively,");
        puts("by an amount xxx in Hertz.\n\n");
L320:   GetKBDInfo(1); switch(Key)  {
                case 'r':
                case 'R': Fc=Fc+Rtemp; goto L250;
                case 'l':
                case 'L': Fc=Fc-Rtemp; goto L250;
                default: puts("Invalid Entry - Try Again\n");
                         goto L320;  }

/***   Case of Digital Filter Bandwidth Adjustment   ***/
        case 'b':
        case 'B':  clearviewport(); restorecrtmode();
        gotoxy(10,3); puts("FILTER BANDWIDTH ADJUSTMENT\n");
        puts("Enter 'ixxx' or 'dxxx' to adjust the filter");
        puts("bandwidth -- increase or decrease, respectively.");
        puts("by an amount xxx in Hertz.\n\n");
L350:   GetKBDInfo(1); switch(Key)  {
                case 'i':
                case 'I': BW=BW+Rtemp; goto L250;
                case 'd':
                case 'D': BW=BW-Rtemp; goto L250;
```

```
                default: puts("Invalid Entry - Try Again\n");
                         goto L350;   }

/***   Case of Digital Filter Gain Adjustment   ***/
        case 'g':
        case 'G':  clearviewport(); restorecrtmode();
        gotoxy(10,3); puts("FILTER GAIN ADJUSTMENT\n");
        puts("Enter new filter gain specification.\n\n");
        GetKBDInfo(3); Gain=Rtemp; goto L250;

/***   Case of Web Tension Test Desired  ***/
        case 't':
        case 'T':  break;

/***   Case of Invalid Character Entry -- Repeat Entry  ***/
        default:  puts("\x7"); goto L400;  }

  /*   BEGIN THE WEB TENSION TEST -- DATA ACQUISTION   */
/*  Set correlation shifts and span depending on sample rate   */

    if (rate==2.0e+5) {ncorr=220; nspan=400; NS=1296; SRdivider=4.0; }

    else if (rate==3.0e+5){ncorr=280;nspan=500;NS=1600; SRdivider=5.0;}

    else if (rate==4.0e+5){ncorr=360;nspan=600;NS=1840; SRdivider=8.0;}

    else if (rate==5.0e+5){ncorr=420;nspan=760;NS=2048;SRdivider=10.0;}

    else if (rate==6.0e+5){ncorr=500;nspan=900;NS=2272;SRdivider=10.0;}

    else if (rate==7.0e+5){ncorr=560;nspan=1000;NS=2496;SRdivider=11.0;}

    else { outtextxy(200,40,"Sampling Rate Selected is not Supported");
                                goto L1160; }

    clearviewport(); restorecrtmode();
    gotoxy(10,3); l=0; ncy=1; deltaX[1]=refedge;
    frate=rate/SRdivider; period=2.0/rate;              /*Define Fourier
    fdel=ndel/(int)SRdivider; FNS=160+fdel;            /*sample rate and
    printf("  BEGINNING  WEB  TENSION  TEST \n\n");              /*delay
    printf("  Sampling Rate for the test is %e\n",rate);
    printf("  Sampling Period for the test is %e\n",period);
    printf("  Width of the web under test is %f\n",webwidth);
    printf("  Shoe is positioned %f inches from web edge\n\n",refedge);

/******   Take two waveform records and average the results  *******/
L650:
    l++; puts(" ....Acquiring Data Records....\n ");
    GetWave(rate,NS);                                  /*Record 1 acquired*/
    FilterRecord(XDAT,YDAT,rate,5.0,NS);                 /*Record Filtered*/
    GetWave(rate,NS);                                  /*Record 2 acquired*/
    FilterRecord(X,Y,rate,5.0,NS);                       /*Record Filtered*/
    for (j=1,i=ndel; i<(NS/2); j++,i++) {
        XDAT[j]=XDAT[j]+X[i]; YDAT[j]=YDAT[j]+Y[i]; }
    Correlate(XDAT,YDAT,nspan,ncorr,ndel);          /*Perform Correlation*/

/**Sort the Cross Correlation data to find the maximum correlation value
          test=fabs(XDAT[2]); j=2;
          if(fabs(XDAT[1])>test) sl=-1; else sl=1;
          i=2; k=0;
L700:
          i++;
          if (i>=ncorr) goto L780;
          if (sl<0) goto L750;
          if ((XDAT[i]-test)<0.0) goto L730;
          test=XDAT[i]; j=i; goto L700;
```

```
L730:
          k++; max[k]=test; maxptr[k]=j;
          sl=-1; goto L700;
L750:
          if ((XDAT[i]-test)>0.0) {  sl=1; goto L700;  }
          test=XDAT[i]; goto L700;
L780:
          totmax=max[1]; maxindex=maxptr[1];
          for (i=2; i<=k; i++) {
              if((fabs(max[i])-fabs(totmax))<=0.0) continue;
                  totmax=max[i];
                  maxindex=maxptr[i];     }

/***   Calculate delta T for this correlation function based on ***/
/***   index value corresponding to maximum correlation value  ***/

      deltaT=period*maxindex;
      printf(" Maximum Correlation Occured at Delay Point %d\n",maxindex);
      printf(" Delta T value for this correlation is %e\n\n",deltaT);
      printf(" .... Beginning FFT Analysis ....\n\n");

/*   Perform the FFT analysis on waveforms to obtain Web Frequency  */
/*   Set sample rate to reduced value (frate) to aid Freq Resolution */

    j=1; for (k=1; k<=7; k++) {                        /*Assemble Pseudo-*/
        GetWave(frate,FNS);                             /*periodic record*/
        FilterRecord(X,Y,frate,5.0,FNS);
        for (i=fdel; i<=((148+fdel)/2); i++,j++) {
                XDAT[j]=X[i]; YDAT[j]=0.0;  }   }

/*** Compute record average value and subtract from each data point ***/
        temp=0.0; for (j=1; j<=512; j++) {
                temp=temp+XDAT[j]; }
        temp=temp/512.0; for (j=1; j<=512; j++) {
                XDAT[j]=XDAT[j]-temp;  }

/***  Use Cosine windowing on the assembled zero biased record  ***/
        for (j=1; j<=25; j++) {
                XDAT[j]=XDAT[j]*cos(pi*(1.0-(float)(j-1)/24.0)/2.0); }
        for (j=488; j<=512; j++) {
                XDAT[j]=XDAT[j]*cos(pi*(1.0-(float)(512-j)/24.0)/2.0); }

/***   Pad the data vector from 513 to 1024 with zeros  ***/
        for (j=513; j<=1024; j++)  {
                XDAT[j]=0.0; YDAT[j]=0.0;  }

/* Fourier Transform: XDAT=real vector, YDAT=imaginery vector (zeroed) *
        Fourea(XDAT,YDAT,1024);

/** Compute Spectral Density Function from transformed data vectors ***/
        SpecDens(XDAT,YDAT,X,frate,1024);

/* Find maximum Spectral value to indicate the dominant Web Frequency */
        test=X[15]; k=15;
        for (i=16; i<=256; i++) {
                if (X[i]<test) continue;
                test=X[i]; k=i;       }

     printf(" Maximum Spectral Value Occured at Point %d\n",k);
     webfreq=(frate/2048.0)*k;
     printf(" Characteristic Frequency for this Test is %f\n",webfreq);

  /*    calculate the web tension based on the delta T value and  */
  /*   based on the web frequency calculated via the fft analysis */
```

```
  trvel=transvel*0.20;                  /*Transport velocity in in/sec*/
  vel=(2.0/deltaT)-trvel;                  /*Corrected Web Phase Speed*/
  kair=2.0*pi*webfreq/cair;                      /*Wave Number for Air*/
  Tens=rhoweb*vel*vel/386.0;           /*In-Vacuo Web Tension Estimate*/
  kweb=2.0*pi*webfreq/vel;                       /*Wave Number for Web*/
  f3=kweb*kweb-kair*kair;
  if (f3<0.0) {printf("### BAD DELTAT OR WEBFREQ VALUES OBTAINED ###\n");
               printf("      Program will skip this web location\n");
                        Tension[1]=0.0;   goto L1000; }
  f2=rhoweb+2.0*rhoair/sqrt(f3);

/*****  Calculated Tension Value Based on Ribbon Equation  ******/
  Tcalc=vel*vel*f2/386.0;
  Tension[1]=Tcalc;

  printf(" Membrane In-Vacuo Estimated Tension = %f\n",Tens);
  printf(" Calculated Ribbon Equation Tension = %f\n\n",Tension[1]);

L1000:  if (intind==0) goto L1010;        /*case of equal trav. intervals

  puts(" Enter 'Space Bar' to move shoe and retest, 'Esc' to exit\n");
  Key=getch(); if (Key!=' ') goto L1030;
  puts(" Enter desired amount of shoe movement in inches\n");
  GetKBDInfo(3); puts(""); dist=Rtemp; goto L1020;
L1010:
  ncy=ncy+1; if (ncy>cycles) goto L1030;
L1020:
  deltaX[1+1]=(int)dist; coun=dist*36.0/(radius*pi);
  StepMotor(dir,coun); goto L650;

  /*   PERFORM THE GRAPHIC SUMMARY OF THE TENSION TEST   */
L1030:
   sum=0.0; for (i=1; i<=1; i++) {
                    sum=sum+Tension[i];  }
   Tensavg=sum/(float)1;                    /*Compute Average Tension*/
   initgraph(&grdriver,&grmode,"");
   setviewport(0,0,639,349,1); clearviewport();
   outtextxy(60,8,"  WEB  TENSION  SUMMARY : ");
   outtextxy(300,8,"#"); circle(303,11,10);
   outtextxy(315,8,"  INDICATES STATION NUMBER ");

   moveto(1,90); lineto(1,340);
   lineto(2,340); lineto(2,90);          /*Draw the "Web Boundaries"*/
   moveto(637,90); lineto(637,340);      /*      on the screen      */
   lineto(638,340); lineto(638,90);
   i=1;
   outtextxy(20,30,"1"); circle(23,33,10);
   outtextxy(34,30,"  T=");                  /* Print out the tension */
   sprintf(DataStr,"%2.5f",Tension[1]);      /* value for station one */
   outtextxy(68,30,DataStr);
   i=i+1; if (i>1) goto L1100;
   outtextxy(230,30,"2"); circle(233,33,10);
   outtextxy(244,30,"  T=");                  /* Print out the tension */
   sprintf(DataStr,"%2.5f",Tension[2]);      /* value for station two */
   outtextxy(278,30,DataStr);
   i=i+1; if (i>1) goto L1100;
   outtextxy(440,30,"3"); circle(443,33,10);
   outtextxy(454,30,"  T=");                  /* Print out the tension */
   sprintf(DataStr,"%2.5f",Tension[3]);      /*value for station three*/
   outtextxy(488,30,DataStr);
   i=i+1; if (i>1) goto L1100;
   outtextxy(20,50,"4"); circle(23,53,10);
   outtextxy(34,50,"  T=");                  /* Print out the tension */
   sprintf(DataStr,"%2.5f",Tension[4]);      /*value for station four */
   outtextxy(68,50,DataStr);
   i=i+1; if (i>1) goto L1100;
```

```
      outtextxy(230,50,"5"); circle(233,53,10);
      outtextxy(244,50,"  T=");                  /* Print out the tension */
      sprintf(DataStr,"%2.5f",Tension[5]);     /*value for station five */
      outtextxy(278,50,DataStr);
      i=i+1; if (i>1) goto L1100;
      outtextxy(440,50,"6"); circle(443,53,10);
      outtextxy(454,50,"  T=");                  /* Print out the tension */
      sprintf(DataStr,"%2.5f",Tension[6]);     /* value for station six */
      outtextxy(488,50,DataStr);
      i=i+1; if (i>1) goto L1100;
      outtextxy(20,70,"7"); circle(23,73,10);
      outtextxy(34,70,"  T=");                  /* Print out the tension */
      sprintf(DataStr,"%2.5f",Tension[7]);     /*value for station seven*/
      outtextxy(68,70,DataStr);
      i=i+1; if (i>1) goto L1100;
      outtextxy(230,70,"8"); circle(233,73,10);
      outtextxy(244,70,"  T=");                  /* Print out the tension */
      sprintf(DataStr,"%2.5f",Tension[8]);     /*value for station eight*/
      outtextxy(278,70,DataStr);
      i=i+1; if (i>1) goto L1100;
      outtextxy(440,70,"9"); circle(443,73,10);
      outtextxy(454,70,"  T=");                   /*Print out the tension*/
      sprintf(DataStr,"%2.5f",Tension[9]);     /*value for station nine*/
        outtextxy(488,70,DataStr);
L1100:
      conv=630.0/webwidth; xcoord[0]=0; xsum=0;
      for (i=1; i<=1; i++)  {
            xcoor=(int)(deltaX[i]*conv);
            xcoord[i]=xcoord[i-1]+xcoor;
            xsum=xsum+deltaX[i];                /*  draw the station
            moveto(xcoord[i],250);              /*number on the screen
            sprintf(DataStr,"%d",i);
            outtext(DataStr);
            circle(xcoord[i]+3,253,10); }
      xcoord[1+1]=639;
      deltaX[1+1]=(int)webwidth-xsum;
      for (i=0; i<=1; i++)  {
            xcoor=(xcoord[i]+xcoord[i+1])/2;   /*indicate the number*
            moveto(xcoor,270);                /*of inches between the*
            sprintf(DataStr,"%d",deltaX[i+1]);  /*different stations*
            outtext(DataStr);  }
      outtextxy(200,290,"Delta X Values in Inches");
      moveto((xcoord[1]/2),294);
      lineto((xcoord[1]/2)+7,290);
      lineto((xcoord[1]/2)+7,298);
      lineto((xcoord[1]/2),294);
      lineto(190,294);
      moveto(((xcoord[1]+639)/2)+7,294);          /*Arrows to indicate*
      lineto(((xcoord[1]+639)/2),290);            /*meaning of results*
      lineto(((xcoord[1]+639)/2),298);
      lineto(((xcoord[1]+639)/2)+7,294);
      lineto(400,294);                          /*Display Total Web Width*
      sprintf(DataStr,"Total Web Width is %5.1f Inches",webwidth);
      outtextxy(180,310,DataStr);
      moveto(3,314); lineto(10,310);
      lineto(10,318); lineto(3,314); lineto(165,314);
      moveto(435,314); lineto(636,314);
      lineto(629,310); lineto(629,318); lineto(637,314);
      outtextxy(100,330,"Reference Edge");
      moveto(3,334); lineto(10,330);
      lineto(10,338); lineto(3,334); lineto(85,334);
      outtextxy(350,330,"Average Tension =");
      sprintf(DataStr,"%2.5f",Tensavg);
      outtextxy(500,330,DataStr);             /*Display Average Tension*
      if (1==1) goto L1140;
```

```
/***   Search for the max and min values of Tension[i] for scaling  ***/
        Tmax=Tension[1]; Tmin=Tension[1];
        for (i=2; i<=l; i++)  {
                if ((Tmax-Tension[i])>0.0) goto L1120;
                Tmax=Tension[i];
L1120:
                if ((Tmin-Tension[i])<0.0) continue;
                Tmin=Tension[i];  }

        conv=150.0/(Tmax-Tmin); xcoor=xcoord[1];  /*Graphical Conversion*
        ycoord=(int)((Tensavg-Tmin)*conv);       /*Plot Avg Tension Level*
        while (xcoor<=xcoord[l]) {
                j=230-ycoord; putpixel(xcoor,j,1); xcoor+=6;  }

/**********   Plot the actual tension values on the screen   ************

      for (i=1; i<=l; i++)  {
          ycoord=(int)((Tension[i]-Tmin)*conv);
          j=230-ycoord; moveto(xcoord[i],j); outtext("*"); }
      if (l>1) goto L1150;

/**********  if l=1 (one station only) then plot the tension   **********
L1140: moveto(xcoord[1],100);
       outtext("*");

/*******  Use a ReadLn command to protect the graphics display  *********
L1150: getch();

L1160: ; }
```

APPENDIX G

COMPUTER PROGRAM LISTING FOR GENERATION OF PULSE PRESSURE AND VELOCITY CONTOUR DATA

```
c
c     Program Pulse.f -- This program is for integration of the
c     two dimensional inviscid, unsteady state, nonisentropic,
c     compressible flow equations for a problem grid that simulates
c     a shock tube in proximity to a web surface.  An initial weak
c     shock condition in Region 6 propagates upward and reflects
c     from a rigid interface.  Pressure and Velocity contours may
c     be acquired at specified times during the propagation process.
c
      dimension df(10,21,21),uf(10,21,21),vf(10,21,21),ef(10,21,21)
      dimension da(21,21),ua(21,21),va(21,21),ea(21,21)
      dimension dxl(21),uxl(21),vxl(21),exl(21),dxr(21),uxr(21)
      dimension vxr(21),exr(21),dyl(21),uyl(21),vyl(21),eyl(21)
      dimension dyr(21),uyr(21),vyr(21),eyr(21),dxc(21),uxc(21)
      dimension vxc(21),exc(21),dtl(21),utl(21),vtl(21),etl(21)
      dimension dtr(21),utr(21),vtr(21),etr(21),dwl(21),uwl(21)
      dimension vwl(21),ewl(21),dwr(21),uwr(21),vwr(21),ewr(21)
      dimension dzl(21),uzl(21),vzl(21),ezl(21),dzr(21),uzr(21)
      dimension vzr(21),ezr(21),dsl(21),usl(21),vsl(21),esl(21)
      dimension dsr(21),usr(21),vsr(21),esr(21),dql(21),uql(21)
      dimension vql(21),eql(21),dqr(21),uqr(21),vqr(21),eqr(21)
      dimension drl(21),url(21),vrl(21),erl(21),drr(21),urr(21)
      dimension vrr(21),err(21),dpl(21),upl(21),vpl(21),epl(21)
      dimension dpr(21),upr(21),vpr(21),epr(21)
      dimension dtime(330),time(330),vel(21,21),pres(21,21)
      character*12 PFILE, VFILE
      common /dat/gamma,omega,xKn,l,N
c
c     define the non dimensional quantities to be used.
c     dp,up,vp,ep -- field values.  ds,us,vs,es -- shock gradient values.
c     dn,un,vn,en -- full shock values.   wplusc -- max advection speed.
c     sigma -- stability parameter.   omega -- stability parameter.
c     delxd -- dimensional space increment.  icount -- iteration counter.
c     delxnd -- nondimensional space increment.  N -- no. of space increm
c     sumt -- nondimensional time sum.  gamma -- air constant.
c     intc1,intc2,intc3,etc -- interval for contour evaluation.
c
      N=21
      gamma=1.4
      omega=1.345
      delxd=0.01
      delxnd=0.05
      ind=0
      data sigma,wplusc,patm,cair/0.5,2.3,14.7,13500.0/
      data pn,ps,pp,un,us,up/2.0,1.5,1.0,0.0,0.0,0.0/
      data dn,ds,dp,vn,vs,vp/1.63,1.33,1.0,1.0,0.558,0.0/
      data intc1,intc2,intc3,intc4,icmax/60,90,110,120,260/
      data intc5,intc6,intc7,intc8/160,220,240,260/
      data icount,sumt/1,0.0/
c
c     define initial conditions for all coupling boundary values.
c
      data dxc/21*1.33/,uxc/21*0.0/,vxc/21*0.558/,exc/21*3.96/
      data dxl/21*1.0/,uxl/21*0.0/,vxl/21*0.0/,exl/21*2.5/
      data dxr/21*1.0/,uxr/21*0.0/,vxr/21*0.0/,exr/21*2.5/
      data dyl/21*1.0/,uyl/21*0.0/,vyl/21*0.0/,eyl/21*2.5/
      data dyr/21*1.0/,uyr/21*0.0/,vyr/21*0.0/,eyr/21*2.5/
      data dwl/21*1.0/,uwl/21*0.0/,vwl/21*0.0/,ewl/21*2.5/
      data dwr/21*1.0/,uwr/21*0.0/,vwr/21*0.0/,ewr/21*2.5/
      data dtl/21*1.0/,utl/21*0.0/,vtl/21*0.0/,etl/21*2.5/
      data dtr/21*1.0/,utr/21*0.0/,vtr/21*0.0/,etr/21*2.5/
      data dzl/21*1.0/,uzl/21*0.0/,vzl/21*0.0/,ezl/21*2.5/
      data dzr/21*1.0/,uzr/21*0.0/,vzr/21*0.0/,ezr/21*2.5/
      data dsl/21*1.0/,usl/21*0.0/,vsl/21*0.0/,esl/21*2.5/
```

```
      data dsr/21*1.0/,usr/21*0.0/,vsr/21*0.0/,esr/21*2.5/
      data dql/21*1.0/,uql/21*0.0/,vql/21*0.0/,eql/21*2.5/
      data dqr/21*1.0/,uqr/21*0.0/,vqr/21*0.0/,eqr/21*2.5/
      data drl/21*1.0/,url/21*0.0/,vrl/21*0.0/,erl/21*2.5/
      data drr/21*1.0/,urr/21*0.0/,vrr/21*0.0/,err/21*2.5/
      data dpl/21*1.0/,upl/21*0.0/,vpl/21*0.0/,epl/21*2.5/
      data dpr/21*1.0/,upr/21*0.0/,vpr/21*0.0/,epr/21*2.5/
c
c     define initial conditions for all field values.
c
      data (((df(l,i,j),l=1,10),i=1,21),j=1,21)/4410*1.0/
      data (((uf(l,i,j),l=1,10),i=1,21),j=1,21)/4410*0.0/
      data (((vf(l,i,j),l=1,10),i=1,21),j=1,21)/4410*0.0/
      data (((ef(l,i,j),l=1,10),i=1,21),j=1,21)/4410*2.5/
      es=ds*(us*us+vs*vs)/2.0+ps/(gamma-1.0)
      ep=dp*(up*up+vp*vp)/2.0+pp/(gamma-1.0)
      en=dn*(un*un+vn*vn)/2.0+pn/(gamma-1.0)
c
c     initialize time value and open file for time information.
c
      time(icount)=0.0
      write(6,21)icount,time(icount)
21    format(' ICOUNT = ',i3,'   TIME = ',f6.4,/)
      open(9, file='ptime.dat', status='unknown')
c
c     Initialize Region 6 for Full Shock and Shock Gradient.
c
220   do 230 i=1,N
      do 230 j=2,N-1
      df(6,i,j)=dn
      uf(6,i,j)=un
      vf(6,i,j)=vn
      ef(6,i,j)=en
230   continue
      do 240 i=1,N
      df(6,i,N)=ds
      uf(6,i,N)=us
      vf(6,i,N)=vs
      ef(6,i,N)=es
240   continue
c
c
c     Start the loop in the l number of regions --  1 <= l <= 10.
c     begin definition of the corner points of each region
c
300   afix=0.0
310   do 1800 l=1,10
      if (l.gt.1) go to 320
c
c     compute Courant Number xKn for this nth iteration
c     compute nondimensional time and dimensional time values.
c
      xKn=sigma/wplusc
      sumt=sumt+xKn
      time(icount)=(delxnd/1.41421)*sumt
      write(6,22)time(icount)
22    format(' NON-DIMENSIONAL  TIME  VALUE  IS  NOW ',f8.6)
      deldt=xKn*delxd*sqrt(gamma)/(1.4142*cair)
      dtime(icount)=sumt*delxd*sqrt(gamma)/(1.41421*cair)
      write(6,93)dtime(icount)
93    format(' DIMENSIONAL  TIME  VALUE  IS  NOW  ',f12.8)
320   go to (330,350,370,390,410,430,450,470,490,510) l
330   write(9,335)icount,time(icount),dtime(icount),deldt
335   format(i4,2x,f12.8,2x,f15.13,2x,f15.13)
```

```
      write(6,13)
13    format(' REGION  1  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 1 corner locations.
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(1,1,1),uf(1,1,1),
     1vf(1,1,1),ef(1,1,1),dyl(1),uyl(1),vyl(1),eyl(1),df(1,1,2),
     2uf(1,1,2),vf(1,1,2),ef(1,1,2),df(1,2,1),uf(1,2,1),vf(1,2,1),
     3ef(1,2,1),dxc(1),uxc(1),vxc(1),exc(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(1,1,N),uf(1,1,N),
     1vf(1,1,N),ef(1,1,N),dyl(N),uyl(N),vyl(N),eyl(N),df(1,1,N),
     2uf(1,1,N),-vf(1,1,N),ef(1,1,N),df(1,2,N),uf(1,2,N),vf(1,2,N),
     3ef(1,2,N),df(1,1,N-1),uf(1,1,N-1),vf(1,1,N-1),ef(1,1,N-1),2)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(1,N,1),uf(1,N,1),
     1vf(1,N,1),ef(1,N,1),df(1,N-1,1),uf(1,N-1,1),vf(1,N-1,1),
     2ef(1,N-1,1),df(1,N,2),uf(1,N,2),vf(1,N,2),ef(1,N,2),dyr(1),
     3uyr(1),vyr(1),eyr(1),dxc(N),uxc(N),vxc(N),exc(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(1,N,N),uf(1,N,N),
     1vf(1,N,N),ef(1,N,N),df(1,N-1,N),uf(1,N-1,N),vf(1,N-1,N),
     2ef(1,N-1,N),df(1,N,N),uf(1,N,N),-vf(1,N,N),ef(1,N,N),dyr(N),
     3uyr(N),vyr(N),eyr(N),df(1,N,N-1),uf(1,N,N-1),vf(1,N,N-1),
     4ef(1,N,N-1),2)
c     Evaluate Region 1 edge locations.
      do 340 i=2,N-1
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(1,i,1),uf(1,i,1),
     1vf(1,i,1),ef(1,i,1),df(1,i-1,1),uf(1,i-1,1),vf(1,i-1,1),
     2ef(1,i-1,1),df(1,i,2),uf(1,i,2),vf(1,i,2),ef(1,i,2),df(1,i+1,1),
     3uf(1,i+1,1),vf(1,i+1,1),ef(1,i+1,1),dxc(i),uxc(i),vxc(i),
     4exc(i),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(1,i,N),uf(1,i,N),
     1vf(1,i,N),ef(1,i,N),df(1,i-1,N),uf(1,i-1,N),vf(1,i-1,N),
     2ef(1,i-1,N),df(1,i,N),uf(1,i,N),-vf(1,i,N),ef(1,i,N),df(1,i+1,N),
     3uf(1,i+1,N),vf(1,i+1,N),ef(1,i+1,N),df(1,i,N-1),uf(1,i,N-1),
     4vf(1,i,N-1),ef(1,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(1,N,i),uf(1,N,i),
     1vf(1,N,i),ef(1,N,i),df(1,N-1,i),uf(1,N-1,i),vf(1,N-1,i),
     2ef(1,N-1,i),df(1,N,i+1),uf(1,N,i+1),vf(1,N,i+1),ef(1,N,i+1),
     3dyr(i),uyr(i),vyr(i),eyr(i),df(1,N,i-1),uf(1,N,i-1),vf(1,N,i-1),
     4ef(1,N,i-1),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(1,1,i),uf(1,1,i),
     1vf(1,1,i),ef(1,1,i),dyl(i),uyl(i),vyl(i),eyl(i),df(1,1,i+1),
     2uf(1,1,i+1),vf(1,1,i+1),ef(1,1,i+1),df(1,2,i),uf(1,2,i),
     3vf(1,2,i),ef(1,2,i),df(1,1,i-1),uf(1,1,i-1),vf(1,1,i-1),
     4ef(1,1,i-1),1)
340   continue
      go to 530
350   write(6,14)
14    format(' REGION  2  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 2 corner locations.
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(2,N,1),uf(2,N,1),
     1vf(2,N,1),ef(2,N,1),df(2,N-1,1),uf(2,N-1,1),vf(2,N-1,1),
     2ef(2,N-1,1),df(2,N,2),uf(2,N,2),vf(2,N,2),ef(2,N,2),dyl(1),
     3uyl(1),vyl(1),eyl(1),dxl(N),uxl(N),vxl(N),exl(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(2,N,N),uf(2,N,N),
     1vf(2,N,N),ef(2,N,N),df(2,N-1,N),uf(2,N-1,N),vf(2,N-1,N),
     2ef(2,N-1,N),df(2,N,N),uf(2,N,N),-vf(2,N,N),ef(2,N,N),dyl(N),
     3uyl(N),vyl(N),eyl(N),df(2,N,N-1),uf(2,N,N-1),vf(2,N,N-1),
     4ef(2,N,N-1),2)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(2,1,1),uf(2,1,1),
     1vf(2,1,1),ef(2,1,1),dzl(1),uzl(1),vzl(1),ezl(1),df(2,1,2),
     2uf(2,1,2),vf(2,1,2),ef(2,1,2),df(2,2,1),uf(2,2,1),vf(2,2,1),
     3ef(2,2,1),dxl(1),uxl(1),vxl(1),exl(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(2,1,N),uf(2,1,N),
     1vf(2,1,N),ef(2,1,N),dzl(N),uzl(N),vzl(N),ezl(N),df(2,1,N),
```

```
     2uf(2,1,N),-vf(2,1,N),ef(2,1,N),df(2,2,N),uf(2,2,N),vf(2,2,N),
     3ef(2,2,N),df(2,1,N-1),uf(2,1,N-1),vf(2,1,N-1),ef(2,1,N-1),2)
c     Evaluate Region 2 edge locations.
      do 360 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(2,i,N),uf(2,i,N),
     1vf(2,i,N),ef(2,i,N),df(2,i-1,N),uf(2,i-1,N),vf(2,i-1,N),
     2ef(2,i-1,N),df(2,i,N),uf(2,i,N),-vf(2,i,N),ef(2,i,N),df(2,i+1,N),
     3uf(2,i+1,N),vf(2,i+1,N),ef(2,i+1,N),df(2,i,N-1),uf(2,i,N-1),
     4vf(2,i,N-1),ef(2,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(2,N,i),uf(2,N,i),
     1vf(2,N,i),ef(2,N,i),df(2,N-1,i),uf(2,N-1,i),vf(2,N-1,i),
     2ef(2,N-1,i),df(2,N,i+1),uf(2,N,i+1),vf(2,N,i+1),ef(2,N,i+1),
     3dyl(i),uyl(i),vyl(i),eyl(i),df(2,N,i-1),uf(2,N,i-1),vf(2,N,i-1),
     4ef(2,N,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(2,i,1),uf(2,i,1),
     1vf(2,i,1),ef(2,i,1),df(2,i-1,1),uf(2,i-1,1),vf(2,i-1,1),
     2ef(2,i-1,1),df(2,i,2),uf(2,i,2),vf(2,i,2),ef(2,i,2),df(2,i+1,1),
     3uf(2,i+1,1),vf(2,i+1,1),ef(2,i+1,1),dxl(i),uxl(i),vxl(i),
     4exl(i),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(2,1,i),uf(2,1,i),
     1vf(2,1,i),ef(2,1,i),dzl(i),uzl(i),vzl(i),ezl(i),df(2,1,i+1),
     2uf(2,1,i+1),vf(2,1,i+1),ef(2,1,i+1),df(2,2,i),uf(2,2,i),
     3vf(2,2,i),ef(2,2,i),df(2,1,i-1),uf(2,1,i-1),vf(2,1,i-1),
     4ef(2,1,i-1),1)
360   continue
      go to 530
370   write(6,16)
16    format(' REGION  3  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 3 corner locations.
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(3,1,1),
     1uf(3,1,1),vf(3,1,1),ef(3,1,1),dyr(1),uyr(1),vyr(1),eyr(1),
     2df(3,1,2),uf(3,1,2),vf(3,1,2),ef(3,1,2),df(3,2,1),uf(3,2,1),
     3vf(3,2,1),ef(3,2,1),dxr(1),uxr(1),vxr(1),exr(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(3,1,N),uf(3,1,N),
     1vf(3,1,N),ef(3,1,N),dyr(N),uyr(N),vyr(N),eyr(N),df(3,1,N),
     2uf(3,1,N),-vf(3,1,N),ef(3,1,N),df(3,2,N),uf(3,2,N),vf(3,2,N),
     3ef(3,2,N),df(3,1,N-1),uf(3,1,N-1),vf(3,1,N-1),ef(3,1,N-1),2)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(3,N,1),uf(3,N,1),
     1vf(3,N,1),ef(3,N,1),df(3,N-1,1),uf(3,N-1,1),vf(3,N-1,1),
     2ef(3,N-1,1),df(3,N,2),uf(3,N,2),vf(3,N,2),ef(3,N,2),dzr(1),uzr(1),
     3vzr(1),ezr(1),dxr(N),uxr(N),vxr(N),exr(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(3,N,N),
     1uf(3,N,N),vf(3,N,N),ef(3,N,N),df(3,N-1,N),uf(3,N-1,N),vf(3,N-1,N),
     2ef(3,N-1,N),df(3,N,N),uf(3,N,N),-vf(3,N,N),ef(3,N,N),dzr(N),
     3uzr(N),vzr(N),ezr(N),df(3,N,N-1),uf(3,N,N-1),vf(3,N,N-1),
     4ef(3,N,N-1),2)
c     Evaluate Region 3 edge locations.
      do 380 i=2,N-1
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(3,1,i),
     1uf(3,1,i),vf(3,1,i),ef(3,1,i),dyr(i),uyr(i),vyr(i),eyr(i),
     2df(3,1,i+1),uf(3,1,i+1),vf(3,1,i+1),ef(3,1,i+1),df(3,2,i),
     3uf(3,2,i),vf(3,2,i),ef(3,2,i),df(3,1,i-1),uf(3,1,i-1),vf(3,1,i-1),
     4ef(3,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(3,i,1),
     1uf(3,i,1),vf(3,i,1),ef(3,i,1),df(3,i-1,1),uf(3,i-1,1),vf(3,i-1,1),
     2ef(3,i-1,1),df(3,i,2),uf(3,i,2),vf(3,i,2),ef(3,i,2),df(3,i+1,1),
     3uf(3,i+1,1),vf(3,i+1,1),ef(3,i+1,1),dxr(i),uxr(i),vxr(i),exr(i),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(3,i,N),
     1uf(3,i,N),vf(3,i,N),ef(3,i,N),df(3,i-1,N),uf(3,i-1,N),vf(3,i-1,N),
     2ef(3,i-1,N),df(3,i,N),uf(3,i,N),-vf(3,i,N),ef(3,i,N),df(3,i+1,N),
     3uf(3,i+1,N),vf(3,i+1,N),ef(3,i+1,N),df(3,i,N-1),uf(3,i,N-1),
     4vf(3,i,N-1),ef(3,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(3,N,i),uf(3,N,i),
```

```
     1vf(3,N,i),ef(3,N,i),df(3,N-1,i),uf(3,N-1,i),vf(3,N-1,i),
     2ef(3,N-1,i),df(3,N,i+1),uf(3,N,i+1),vf(3,N,i+1),ef(3,N,i+1),
     3dzr(i),uzr(i),vzr(i),ezr(i),df(3,N,i-1),uf(3,N,i-1),vf(3,N,i-1),
     4ef(3,N,i-1),1)
380   continue
      go to 530
390   write(6,17)
17    format(' REGION  4  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 4 corner locations.
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(4,N,N),
     1uf(4,N,N),vf(4,N,N),ef(4,N,N),df(4,N-1,N),uf(4,N-1,N),vf(4,N-1,N),
     2ef(4,N-1,N),dxl(N),uxl(N),vxl(N),exl(N),df(4,N,N),-uf(4,N,N),
     3vf(4,N,N),ef(4,N,N),df(4,N,N-1),uf(4,N,N-1),vf(4,N,N-1),
     4ef(4,N,N-1),4)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(4,1,N),
     1uf(4,1,N),vf(4,1,N),ef(4,1,N),dtl(N),utl(N),vtl(N),etl(N),dxl(1),
     2uxl(1),vxl(1),exl(1),df(4,2,N),uf(4,2,N),vf(4,2,N),ef(4,2,N),
     3df(4,1,N-1),uf(4,1,N-1),vf(4,1,N-1),ef(4,1,N-1),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(4,1,1),uf(4,1,1),
     1vf(4,1,1),ef(4,1,1),dtl(1),utl(1),vtl(1),etl(1),df(4,1,2),
     2uf(4,1,2),vf(4,1,2),ef(4,1,2),df(4,2,1),uf(4,2,1),vf(4,2,1),
     3ef(4,2,1),dsl(1),usl(1),vsl(1),esl(1),1)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(4,N,1),uf(4,N,1),
     1vf(4,N,1),ef(4,N,1),df(4,N-1,1),uf(4,N-1,1),vf(4,N-1,1),
     2ef(4,N-1,1),df(4,N,2),uf(4,N,2),vf(4,N,2),ef(4,N,2),df(4,N,1),
     3-uf(4,N,1),vf(4,N,1),ef(4,N,1),dsl(N),usl(N),vsl(N),esl(N),4)
c     Evaluate Region 4 edge locations.
      do 400 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(4,i,N),
     1uf(4,i,N),vf(4,i,N),ef(4,i,N),df(4,i-1,N),uf(4,i-1,N),vf(4,i-1,N),
     2ef(4,i-1,N),dxl(i),uxl(i),vxl(i),exl(i),df(4,i+1,N),uf(4,i+1,N),
     3vf(4,i+1,N),ef(4,i+1,N),df(4,i,N-1),uf(4,i,N-1),vf(4,i,N-1),
     4ef(4,i,N-1),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(4,N,i),
     1uf(4,N,i),vf(4,N,i),ef(4,N,i),df(4,N-1,i),uf(4,N-1,i),vf(4,N-1,i),
     2ef(4,N-1,i),df(4,N,i+1),uf(4,N,i+1),vf(4,N,i+1),ef(4,N,i+1),
     3df(4,N,i),-uf(4,N,i),vf(4,N,i),ef(4,N,i),df(4,N,i-1),uf(4,N,i-1),
     4vf(4,N,i-1),ef(4,N,i-1),4)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(4,1,i),
     1uf(4,1,i),vf(4,1,i),ef(4,1,i),dtl(i),utl(i),vtl(i),etl(i),
     2df(4,1,i+1),uf(4,1,i+1),vf(4,1,i+1),ef(4,1,i+1),df(4,2,i),
     3uf(4,2,i),vf(4,2,i),ef(4,2,i),df(4,1,i-1),uf(4,1,i-1),
     4vf(4,1,i-1),ef(4,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(4,i,1),uf(4,i,1),
     1vf(4,i,1),ef(4,i,1),df(4,i-1,1),uf(4,i-1,1),vf(4,i-1,1),
     2ef(4,i-1,1),df(4,i,2),uf(4,i,2),vf(4,i,2),ef(4,i,2),df(4,i+1,1),
     3uf(4,i+1,1),vf(4,i+1,1),ef(4,i+1,1),dsl(i),usl(i),vsl(i),esl(i),1)
400   continue
      go to 530
410   write(6,27)
27    format(' REGION  5  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 5 corner locations.
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(5,1,N),
     1uf(5,1,N),vf(5,1,N),ef(5,1,N),df(5,1,N),-uf(5,1,N),vf(5,1,N),
     2ef(5,1,N),dxr(1),uxr(1),vxr(1),exr(1),df(5,2,N),uf(5,2,N),
     3vf(5,2,N),ef(5,2,N),df(5,1,N-1),uf(5,1,N-1),vf(5,1,N-1),
     4ef(5,1,N-1),3)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(5,N,N),uf(5,N,N),
     1vf(5,N,N),ef(5,N,N),df(5,N-1,N),uf(5,N-1,N),vf(5,N-1,N),
     2ef(5,N-1,N),dxr(N),uxr(N),vxr(N),exr(N),dtr(N),utr(N),vtr(N),
     3etr(N),df(5,N,N-1),uf(5,N,N-1),vf(5,N,N-1),ef(5,N,N-1),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(5,1,1),uf(5,1,1),
```

```
     1vf(5,1,1),ef(5,1,1),df(5,1,1),-uf(5,1,1),vf(5,1,1),ef(5,1,1),
     2df(5,1,2),uf(5,1,2),vf(5,1,2),ef(5,1,2),df(5,2,1),uf(5,2,1),
     3vf(5,2,1),ef(5,2,1),dsr(1),usr(1),vsr(1),esr(1),3)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(5,N,1),uf(5,N,1),
     1vf(5,N,1),ef(5,N,1),df(5,N-1,1),uf(5,N-1,1),vf(5,N-1,1),
     2ef(5,N-1,1),df(5,N,2),uf(5,N,2),vf(5,N,2),ef(5,N,2),dtr(1),
     3utr(1),vtr(1),etr(1),dsr(N),usr(N),vsr(N),esr(N),1)
c     Evaluate Region 5 edge locations.
      do 420 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(5,i,N),uf(5,i,N),
     1vf(5,i,N),ef(5,i,N),df(5,i-1,N),uf(5,i-1,N),vf(5,i-1,N),
     2ef(5,i-1,N),dxr(i),uxr(i),vxr(i),exr(i),df(5,i+1,N),uf(5,i+1,N),
     3vf(5,i+1,N),ef(5,i+1,N),df(5,i,N-1),uf(5,i,N-1),vf(5,i,N-1),
     4ef(5,i,N-1),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(5,1,i),
     1uf(5,1,i),vf(5,1,i),ef(5,1,i),df(5,1,i),-uf(5,1,i),vf(5,1,i),
     2ef(5,1,i),df(5,1,i+1),uf(5,1,i+1),vf(5,1,i+1),ef(5,1,i+1),
     3df(5,2,i),uf(5,2,i),vf(5,2,i),ef(5,2,i),df(5,1,i-1),uf(5,1,i-1),
     4vf(5,1,i-1),ef(5,1,i-1),3)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(5,N,i),uf(5,N,i),
     1vf(5,N,i),ef(5,N,i),df(5,N-1,i),uf(5,N-1,i),vf(5,N-1,i),
     2ef(5,N-1,i),df(5,N,i+1),uf(5,N,i+1),vf(5,N,i+1),ef(5,N,i+1),
     3dtr(i),utr(i),vtr(i),etr(i),df(5,N,i-1),uf(5,N,i-1),vf(5,N,i-1),
     4ef(5,N,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(5,i,1),uf(5,i,1),
     1vf(5,i,1),ef(5,i,1),df(5,i-1,1),uf(5,i-1,1),vf(5,i-1,1),
     2ef(5,i-1,1),df(5,i,2),uf(5,i,2),vf(5,i,2),ef(5,i,2),df(5,i+1,1),
     3uf(5,i+1,1),vf(5,i+1,1),ef(5,i+1,1),dsr(i),usr(i),vsr(i),esr(i),1)
420   continue
      go to 530
430   write(6,33)
33    format(' REGION  6  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 6 corner locations.
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(6,1,N),uf(6,1,N),
     1vf(6,1,N),ef(6,1,N),df(6,1,N),-uf(6,1,N),vf(6,1,N),ef(6,1,N),
     2dxc(1),uxc(1),vxc(1),exc(1),df(6,2,N),uf(6,2,N),vf(6,2,N),
     3ef(6,2,N),df(6,1,N-1),uf(6,1,N-1),vf(6,1,N-1),ef(6,1,N-1),3)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(6,N,N),
     1uf(6,N,N),vf(6,N,N),ef(6,N,N),df(6,N-1,N),uf(6,N-1,N),vf(6,N-1,N),
     2ef(6,N-1,N),dxc(N),uxc(N),vxc(N),exc(N),df(6,N,N),-uf(6,N,N),
     3vf(6,N,N),ef(6,N,N),df(6,N,N-1),uf(6,N,N-1),vf(6,N,N-1),
     4ef(6,N,N-1),4)
c     Evaluate Region 6 edge locations.
      do 440 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(6,i,N),
     1uf(6,i,N),vf(6,i,N),ef(6,i,N),df(6,i-1,N),uf(6,i-1,N),vf(6,i-1,N),
     2ef(6,i-1,N),dxc(i),uxc(i),vxc(i),exc(i),df(6,i+1,N),uf(6,i+1,N),
     3vf(6,i+1,N),ef(6,i+1,N),df(6,i,N-1),uf(6,i,N-1),vf(6,i,N-1),
     4ef(6,i,N-1),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(6,N,i),
     1uf(6,N,i),vf(6,N,i),ef(6,N,i),df(6,N-1,i),uf(6,N-1,i),vf(6,N-1,i),
     2ef(6,N-1,i),df(6,N,i+1),uf(6,N,i+1),vf(6,N,i+1),ef(6,N,i+1),
     3df(6,N,i),-uf(6,N,i),vf(6,N,i),ef(6,N,i),df(6,N,i-1),uf(6,N,i-1),
     4vf(6,N,i-1),ef(6,N,i-1),4)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(6,1,i),
     1uf(6,1,i),vf(6,1,i),ef(6,1,i),df(6,1,i),-uf(6,1,i),vf(6,1,i),
     2ef(6,1,i),df(6,1,i+1),uf(6,1,i+1),vf(6,1,i+1),ef(6,1,i+1),
     3df(6,2,i),uf(6,2,i),vf(6,2,i),ef(6,2,i),df(6,1,i-1),uf(6,1,i-1),
     4vf(6,1,i-1),ef(6,1,i-1),3)
440   continue
      go to 530
450   write(6,37)
37    format(' REGION  7  BEING  PROCESSED ')
```

```
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 7 corner locations.
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(7,N,1),
     1uf(7,N,1),vf(7,N,1),ef(7,N,1),df(7,N-1,1),uf(7,N-1,1),vf(7,N-1,1),
     2ef(7,N-1,1),df(7,N,2),uf(7,N,2),vf(7,N,2),ef(7,N,2),dzl(1),uzl(1),
     3vzl(1),ezl(1),dwl(N),uwl(N),vwl(N),ewl(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(7,N,N),
     1uf(7,N,N),vf(7,N,N),ef(7,N,N),df(7,N-1,N),uf(7,N-1,N),vf(7,N-1,N),
     2ef(7,N-1,N),df(7,N,N),uf(7,N,N),-vf(7,N,N),ef(7,N,N),dzl(N),
     3uzl(N),vzl(N),ezl(N),df(7,N,N-1),uf(7,N,N-1),vf(7,N,N-1),
     4ef(7,N,N-1),2)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(7,1,1),uf(7,1,1),
     1vf(7,1,1),ef(7,1,1),dql(1),uql(1),vql(1),eql(1),df(7,1,2),
     2uf(7,1,2),vf(7,1,2),ef(7,1,2),df(7,2,1),uf(7,2,1),vf(7,2,1),
     3ef(7,2,1),dwl(1),uwl(1),vwl(1),ewl(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(7,1,N),uf(7,1,N),
     1vf(7,1,N),ef(7,1,N),dql(N),uql(N),vql(N),eql(N),df(7,1,N),
     2uf(7,1,N),-vf(7,1,N),ef(7,1,N),df(7,2,N),uf(7,2,N),vf(7,2,N),
     3ef(7,2,N),df(7,1,N-1),uf(7,1,N-1),vf(7,1,N-1),ef(7,1,N-1),2)
c     Evaluate Region 7 edge locations.
      do 460 i=2,N-1
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(7,i,1),
     1uf(7,i,1),vf(7,i,1),ef(7,i,1),df(7,i-1,1),uf(7,i-1,1),vf(7,i-1,1),
     2ef(7,i-1,1),df(7,i,2),uf(7,i,2),vf(7,i,2),ef(7,i,2),df(7,i+1,1),
     3uf(7,i+1,1),vf(7,i+1,1),ef(7,i+1,1),dwl(i),uwl(i),vwl(i),ewl(i),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(7,N,i),
     1uf(7,N,i),vf(7,N,i),ef(7,N,i),df(7,N-1,i),uf(7,N-1,i),vf(7,N-1,i),
     2ef(7,N-1,i),df(7,N,i+1),uf(7,N,i+1),vf(7,N,i+1),ef(7,N,i+1),
     3dzl(i),uzl(i),vzl(i),ezl(i),df(7,N,i-1),uf(7,N,i-1),vf(7,N,i-1),
     4ef(7,N,i-1),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(7,i,N),
     1uf(7,i,N),vf(7,i,N),ef(7,i,N),df(7,i-1,N),uf(7,i-1,N),
     2vf(7,i-1,N),ef(7,i-1,N),df(7,i,N),uf(7,i,N),-vf(7,i,N),
     3ef(7,i,N),df(7,i+1,N),uf(7,i+1,N),vf(7,i+1,N),ef(7,i+1,N),
     4df(7,i,N-1),uf(7,i,N-1),vf(7,i,N-1),ef(7,i,N-1),2)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(7,1,i),uf(7,1,i),
     1vf(7,1,i),ef(7,1,i),dql(i),uql(i),vql(i),eql(i),df(7,1,i+1),
     2uf(7,1,i+1),vf(7,1,i+1),ef(7,1,i+1),df(7,2,i),uf(7,2,i),
     3vf(7,2,i),ef(7,2,i),df(7,1,i-1),uf(7,1,i-1),vf(7,1,i-1),
     4ef(7,1,i-1),1)
460   continue
      go to 530
470   write(6,41)
41    format(' REGION  8  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 8 corner locations.
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(8,1,1),uf(8,1,1),
     1vf(8,1,1),ef(8,1,1),dzr(1),uzr(1),vzr(1),ezr(1),df(8,1,2),
     2uf(8,1,2),vf(8,1,2),ef(8,1,2),df(8,2,1),uf(8,2,1),vf(8,2,1),
     3ef(8,2,1),dwr(1),uwr(1),vwr(1),ewr(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(8,1,N),uf(8,1,N),
     1vf(8,1,N),ef(8,1,N),dzr(N),uzr(N),vzr(N),ezr(N),df(8,1,N),
     2uf(8,1,N),-vf(8,1,N),ef(8,1,N),df(8,2,N),uf(8,2,N),vf(8,2,N),
     3ef(8,2,N),df(8,1,N-1),uf(8,1,N-1),vf(8,1,N-1),ef(8,1,N-1),2)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(8,N,1),uf(8,N,1),
     1vf(8,N,1),ef(8,N,1),df(8,N-1,1),uf(8,N-1,1),vf(8,N-1,1),
     2ef(8,N-1,1),df(8,N,2),uf(8,N,2),vf(8,N,2),ef(8,N,2),
     3dqr(1),uqr(1),vqr(1),eqr(1),dwr(N),uwr(N),vwr(N),ewr(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(8,N,N),uf(8,N,N),
     1vf(8,N,N),ef(8,N,N),df(8,N-1,N),uf(8,N-1,N),vf(8,N-1,N),
     2ef(8,N-1,N),df(8,N,N),uf(8,N,N),-vf(8,N,N),ef(8,N,N),dqr(N),
     3uqr(N),vqr(N),eqr(N),df(8,N,N-1),uf(8,N,N-1),vf(8,N,N-1),
     4ef(8,N,N-1),2)
c     Evaluate Region 8 edge locations.
```

```
      do 480 i=2,N-1
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(8,1,i),
     1uf(8,1,i),vf(8,1,i),ef(8,1,i),dzr(i),uzr(i),vzr(i),ezr(i),
     2df(8,1,i+1),uf(8,1,i+1),vf(8,1,i+1),ef(8,1,i+1),df(8,2,i),
     3uf(8,2,i),vf(8,2,i),ef(8,2,i),df(8,1,i-1),uf(8,1,i-1),vf(8,1,i-1),
     4ef(8,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(8,i,1),
     1uf(8,i,1),vf(8,i,1),ef(8,i,1),df(8,i-1,1),uf(8,i-1,1),vf(8,i-1,1),
     2ef(8,i-1,1),df(8,i,2),uf(8,i,2),vf(8,i,2),ef(8,i,2),df(8,i+1,1),
     3uf(8,i+1,1),vf(8,i+1,1),ef(8,i+1,1),dwr(i),uwr(i),vwr(i),ewr(i),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(8,i,N),
     1uf(8,i,N),vf(8,i,N),ef(8,i,N),df(8,i-1,N),uf(8,i-1,N),vf(8,i-1,N),
     2ef(8,i-1,N),df(8,i,N),uf(8,i,N),-vf(8,i,N),ef(8,i,N),df(8,i+1,N),
     3uf(8,i+1,N),vf(8,i+1,N),ef(8,i+1,N),df(8,i,N-1),uf(8,i,N-1),
     4vf(8,i,N-1),ef(8,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(8,N,i),uf(8,N,i),
     1vf(8,N,i),ef(8,N,i),df(8,N-1,i),uf(8,N-1,i),vf(8,N-1,i),
     2ef(8,N-1,i),df(8,N,i+1),uf(8,N,i+1),vf(8,N,i+1),ef(8,N,i+1),
     3dqr(i),uqr(i),vqr(i),eqr(i),df(8,N,i-1),uf(8,N,i-1),vf(8,N,i-1),
     4ef(8,N,i-1),1)
480   continue
      go to 530
490   write(6,47)
47    format(' REGION  9  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 9 corner locations.
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(9,N,N),uf(9,N,N),
     1vf(9,N,N),ef(9,N,N),df(9,N-1,N),uf(9,N-1,N),vf(9,N-1,N),
     2ef(9,N-1,N),dwl(N),uwl(N),vwl(N),ewl(N),dtl(N),utl(N),vtl(N),
     3etl(N),df(9,N,N-1),uf(9,N,N-1),vf(9,N,N-1),ef(9,N,N-1),1)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(9,N,1),uf(9,N,1),
     1vf(9,N,1),ef(9,N,1),df(9,N-1,1),uf(9,N-1,1),vf(9,N-1,1),
     2ef(9,N-1,1),df(9,N,2),uf(9,N,2),vf(9,N,2),ef(9,N,2),
     3dtl(1),utl(1),vtl(1),etl(1),drl(N),url(N),vrl(N),erl(N),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(9,1,1),uf(9,1,1),
     1vf(9,1,1),ef(9,1,1),dpl(1),upl(1),vpl(1),epl(1),df(9,1,2),
     2uf(9,1,2),vf(9,1,2),ef(9,1,2),df(9,2,1),uf(9,2,1),vf(9,2,1),
     3ef(9,2,1),drl(1),url(1),vrl(1),erl(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(9,1,N),uf(9,1,N),
     1vf(9,1,N),ef(9,1,N),dpl(N),upl(N),vpl(N),epl(N),dwl(1),uwl(1),
     2vwl(1),ewl(1),df(9,2,N),uf(9,2,N),vf(9,2,N),ef(9,2,N),
     3df(9,1,N-1),uf(9,1,N-1),vf(9,1,N-1),ef(9,1,N-1),1)
c     Evaluate Region 9 edge locations.
      do 500 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(9,i,N),
     1uf(9,i,N),vf(9,i,N),ef(9,i,N),df(9,i-1,N),uf(9,i-1,N),vf(9,i-1,N),
     2ef(9,i-1,N),dwl(i),uwl(i),vwl(i),ewl(i),df(9,i+1,N),uf(9,i+1,N),
     3vf(9,i+1,N),ef(9,i+1,N),df(9,i,N-1),uf(9,i,N-1),vf(9,i,N-1),
     4ef(9,i,N-1),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(9,N,i),
     1uf(9,N,i),vf(9,N,i),ef(9,N,i),df(9,N-1,i),uf(9,N-1,i),vf(9,N-1,i),
     2ef(9,N-1,i),df(9,N,i+1),uf(9,N,i+1),vf(9,N,i+1),ef(9,N,i+1),
     3dtl(i),utl(i),vtl(i),etl(i),df(9,N,i-1),uf(9,N,i-1),vf(9,N,i-1),
     4ef(9,N,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(9,i,1),uf(9,i,1),
     1vf(9,i,1),ef(9,i,1),df(9,i-1,1),uf(9,i-1,1),vf(9,i-1,1),
     2ef(9,i-1,1),df(9,i,2),uf(9,i,2),vf(9,i,2),ef(9,i,2),df(9,i+1,1),
     3uf(9,i+1,1),vf(9,i+1,1),ef(9,i+1,1),drl(i),url(i),vrl(i),erl(i),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(9,1,i),uf(9,1,i),
     1vf(9,1,i),ef(9,1,i),dpl(i),upl(i),vpl(i),epl(i),df(9,1,i+1),
     2uf(9,1,i+1),vf(9,1,i+1),ef(9,1,i+1),df(9,2,i),uf(9,2,i),vf(9,2,i),
     3ef(9,2,i),df(9,1,i-1),uf(9,1,i-1),vf(9,1,i-1),ef(9,1,i-1),1)
500   continue
      go to 530
```

```
510   write(6,53)
53    format(' REGION  10  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
c     Evaluate Region 10 corner locations.
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(10,1,N),
     1uf(10,1,N),vf(10,1,N),ef(10,1,N),dtr(N),utr(N),vtr(N),etr(N),
     2dwr(1),uwr(1),vwr(1),ewr(1),df(10,2,N),uf(10,2,N),vf(10,2,N),
     3ef(10,2,N),df(10,1,N-1),uf(10,1,N-1),vf(10,1,N-1),ef(10,1,N-1),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(10,1,1),uf(10,1,1),
     1vf(10,1,1),ef(10,1,1),dtr(1),utr(1),vtr(1),etr(1),df(10,1,2),
     2uf(10,1,2),vf(10,1,2),ef(10,1,2),df(10,2,1),uf(10,2,1),vf(10,2,1),
     3ef(10,2,1),drr(1),urr(1),vrr(1),err(1),1)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(10,N,1),uf(10,N,1),
     1vf(10,N,1),ef(10,N,1),df(10,N-1,1),uf(10,N-1,1),vf(10,N-1,1),
     2ef(10,N-1,1),df(10,N,2),uf(10,N,2),vf(10,N,2),ef(10,N,2),
     3dpr(1),upr(1),vpr(1),epr(1),drr(N),urr(N),vrr(N),err(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(10,N,N),uf(10,N,N),
     1vf(10,N,N),ef(10,N,N),df(10,N-1,N),uf(10,N-1,N),vf(10,N-1,N),
     2ef(10,N-1,N),dwr(N),uwr(N),vwr(N),ewr(N),dpr(N),upr(N),vpr(N),
     3epr(N),df(10,N,N-1),uf(10,N,N-1),vf(10,N,N-1),ef(10,N,N-1),1)
c     Evaluate Region 10 edge locations.
      do 520 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(10,i,N),
     1uf(10,i,N),vf(10,i,N),ef(10,i,N),df(10,i-1,N),uf(10,i-1,N),
     2vf(10,i-1,N),ef(10,i-1,N),dwr(i),uwr(i),vwr(i),ewr(i),
     3df(10,i+1,N),uf(10,i+1,N),vf(10,i+1,N),ef(10,i+1,N),df(10,i,N-1),
     4uf(10,i,N-1),vf(10,i,N-1),ef(10,i,N-1),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(10,1,i),
     1uf(10,1,i),vf(10,1,i),ef(10,1,i),dtr(i),utr(i),vtr(i),etr(i),
     2df(10,1,i+1),uf(10,1,i+1),vf(10,1,i+1),ef(10,1,i+1),df(10,2,i),
     3uf(10,2,i),vf(10,2,i),ef(10,2,i),df(10,1,i-1),uf(10,1,i-1),
     4vf(10,1,i-1),ef(10,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(10,i,1),uf(10,i,1),
     1vf(10,i,1),ef(10,i,1),df(10,i-1,1),uf(10,i-1,1),vf(10,i-1,1),
     2ef(10,i-1,1),df(10,i,2),uf(10,i,2),vf(10,i,2),ef(10,i,2),
     3df(10,i+1,1),uf(10,i+1,1),vf(10,i+1,1),ef(10,i+1,1),drr(i),
     4urr(i),vrr(i),err(i),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(10,N,i),uf(10,N,i),
     1vf(10,N,i),ef(10,N,i),df(10,N-1,i),uf(10,N-1,i),vf(10,N-1,i),
     2ef(10,N-1,i),df(10,N,i+1),uf(10,N,i+1),vf(10,N,i+1),ef(10,N,i+1),
     3dpr(i),upr(i),vpr(i),epr(i),df(10,N,i-1),uf(10,N,i-1),
     4vf(10,N,i-1),ef(10,N,i-1),1)
520   continue
c
c     Evaluate interior locations for Regions 1 through 10.
c
530   do 540 i=2,N-1
      do 540 j=2,N-1
      call field(da(i,j),ua(i,j),va(i,j),ea(i,j),df(1,i,j),uf(1,i,j),
     1vf(1,i,j),ef(1,i,j),df(1,i-1,j),uf(1,i-1,j),vf(1,i-1,j),
     2ef(1,i-1,j),df(1,i,j+1),uf(1,i,j+1),vf(1,i,j+1),ef(1,i,j+1),
     3df(1,i+1,j),uf(1,i+1,j),vf(1,i+1,j),ef(1,i+1,j),df(1,i,j-1),
     4uf(1,i,j-1),vf(1,i,j-1),ef(1,i,j-1),1)
540   continue
c
c     determine the boundary conditions for next time step
c
600   goto (610,630,650,670,690,710,730,750,770,790) 1
c
c     Coupling Boundaries yl and yr associated with Region 1.
c
610   do 620 i=1,N
      dyl(i)=da(1,i)
      uyl(i)=ua(1,i)
```

```
      vyl(i)=va(1,i)
      eyl(i)=ea(1,i)
      dyr(i)=da(N,i)
      uyr(i)=ua(N,i)
      vyr(i)=va(N,i)
      eyr(i)=ea(N,i)
620   continue
      go to 900
c
c     Coupling Boundaries xl and zl associated with Region 2.
c
630   do 640 i=1,N
      dxl(i)=da(i,1)
      uxl(i)=ua(i,1)
      vxl(i)=va(i,1)
      exl(i)=ea(i,1)
      dzl(i)=da(1,i)
      uzl(i)=ua(1,i)
      vzl(i)=va(1,i)
      ezl(i)=ea(1,i)
640   continue
      go to 900
c
c     Coupling Boundaries xr and zr associated with Region 3.
c
650   do 660 i=1,N
      dxr(i)=da(i,1)
      uxr(i)=ua(i,1)
      vxr(i)=va(i,1)
      exr(i)=ea(i,1)
      dzr(i)=da(N,i)
      uzr(i)=ua(N,i)
      vzr(i)=va(N,i)
      ezr(i)=ea(N,i)
660   continue
      go to 900
c
c     Coupling Boundaries tl and sl associated with Region 4.
c
670   do 680 i=1,N
      dtl(i)=da(1,i)
      utl(i)=ua(1,i)
      vtl(i)=va(1,i)
      etl(i)=ea(1,i)
      dsl(i)=da(i,1)
      usl(i)=ua(i,1)
      vsl(i)=va(i,1)
      esl(i)=ea(i,1)
680   continue
      go to 900
c
c     Coupling Boundaries tr and sr associated with Region 5.
c
690   do 700 i=1,N
      dtr(i)=da(N,i)
      utr(i)=ua(N,i)
      vtr(i)=va(N,i)
      etr(i)=ea(N,i)
      dsr(i)=da(i,1)
      usr(i)=ua(i,1)
      vsr(i)=va(i,1)
      esr(i)=ea(i,1)
700   continue
      go to 900
```

```
c
c     Coupling Boundary xc associated with Region 6.
c
710   do 720 i=1,N
      dxc(i)=da(i,N)
      uxc(i)=ua(i,N)
      vxc(i)=va(i,N)
      exc(i)=ea(i,N)
720   continue
      go to 900
c
c     Coupling Boundaries wl and ql associated with Region 7.
c
730   do 740 i=1,N
      dwl(i)=da(i,1)
      uwl(i)=ua(i,1)
      vwl(i)=va(i,1)
      ewl(i)=ea(i,1)
      dql(i)=da(1,i)
      uql(i)=ua(1,i)
      vql(i)=va(1,i)
      eql(i)=ea(1,i)
740   continue
      go to 900
c
c     Coupling Boundaries wr and qr associated with Region 8.
c
750   do 760 i=1,N
      dwr(i)=da(i,1)
      uwr(i)=ua(i,1)
      vwr(i)=va(i,1)
      ewr(i)=ea(i,1)
      dqr(i)=da(N,i)
      uqr(i)=ua(N,i)
      vqr(i)=va(N,i)
      eqr(i)=ea(N,i)
760   continue
      go to 900
c
c     Coupling Boundaries pl and rl associated with Region 9.
c
770   do 780 i=1,N
      dpl(i)=da(1,i)
      upl(i)=ua(1,i)
      vpl(i)=va(1,i)
      epl(i)=ea(1,i)
      drl(i)=da(i,1)
      url(i)=ua(i,1)
      vrl(i)=va(i,1)
      erl(i)=ea(i,1)
780   continue
      go to 900
c
c     Coupling Boundaries pr and rr associated with Region 10.
c
790   do 800 i=1,N
      dpr(i)=da(N,i)
      upr(i)=ua(N,i)
      vpr(i)=va(N,i)
      epr(i)=ea(N,i)
      drr(i)=da(i,1)
      urr(i)=ua(i,1)
      vrr(i)=va(i,1)
      err(i)=ea(i,1)
```

```
800   continue
c
c     set da, ua, va, and ea values at this time step equal to
c     df, uf, vf, and ef values for the next time step.
c
900   do 910 i=1,N
      do 910 j=1,N
      df(l,i,j)=da(i,j)
      uf(l,i,j)=ua(i,j)
      vf(l,i,j)=va(i,j)
      ef(l,i,j)=ea(i,j)
910   continue
c
c     Search active region for maximum advection speed fixmax.
c     afix is maximum value of fixmax for regions 1 through 10.
c
      fixmax=0.0
      do 920 i=1,N
      do 920 j=1,N
      usquar=ua(i,j)*ua(i,j)
      vsquar=va(i,j)*va(i,j)
      vel(i,j)=sqrt(usquar+vsquar)
      vpres=da(i,j)*(usquar+vsquar)/2.0
      pres(i,j)=(gamma-1.0)*(ea(i,j)-vpres)
      ss=gamma*pres(i,j)/da(i,j)
      if (ss.gt.0.0) go to 930
      ss=0.0
930   ss=sqrt(ss)+vel(i,j)
      if (ss.le.fixmax) go to 920
      fixmax=ss
920   continue
      if (fixmax.le.afix) go to 940
      afix=fixmax
c
c     Check for count corresponding to a contour search.
c
940   if (icount.eq.intc1) go to 950
      if (icount.eq.intc2) go to 950
      if (icount.eq.intc3) go to 950
      if (icount.eq.intc4) go to 950
      if (icount.eq.intc5) go to 950
      if (icount.eq.intc6) go to 950
      if (icount.eq.intc7) go to 950
      if (icount.eq.intc8) go to 950
      go to 1800
c
c     Search the entire grid for maximum pressure and velocity.
c
950   if (l.ne.1) go to 960
      vmax=0.0
      pmax=0.0
      do 970 m=1,10
      do 970 j=1,N
      do 970 i=1,N
      veloc=sqrt(uf(m,i,j)*uf(m,i,j)+vf(m,i,j)*vf(m,i,j))
      vpress=df(m,i,j)*veloc*veloc/2.0
      press=(gamma-1.0)*(ef(m,i,j)-vpress)
      if (vmax.lt.veloc) vmax=veloc
      if (pmax.lt.press) pmax=press
970   continue
      write(6,213)pmax,vmax
213   format('Max Pressure is ',f7.4,'   Max Velocity is ',f7.4,/,
     1' Enter cpmax and cvmax values for contour search ')
      read(*,*)cpmax,cvmax
```

```
      write(6,215)cpmax,cvmax
215   format('CPMAX val read was',f6.3,' CVMAX val read was',f6.3)
      write(6,217)
217   format('Enter delp and delv values')
      read(*,*)delp,delv
      write(6,219)delp,delv
219   format('delp val read was',f6.3,'  delv val read was',f6.3)
c
960   write(6,182)icount,l
182   format('Iteration=',i3,'  Region=',i2,'Enter Pressure filename')
      read(5,183)PFILE
183   format(A12)
      write(6,184)icount,l
184   format('iteration=',i3,'  region=',i2,'enter Velocity filename')
      read(5,183)VFILE
c
1100  open(8, file=PFILE, status='unknown')
      write(8,83)delp,cpmax
83    format(f6.3,3x,f7.4)
c
c     pcntr is contour pressure level -- check grid with respect to
c     both x and y to see if any grid bracketing occurs.
c
1105  pcntr=pp+delp
1110  piso=0.0
      ipx=0
      ipy1=0
      ipy2=0
c
      do 1140 i=1,N
      do 1140 j=1,N
      do 1140 k=1,2
      if (j.eq.1) go to 1210
      if (k.eq.2) go to 1160
      test1=pres(i,j)
      test2=pres(i,j-1)
      go to 1170
1160  test1=pres(j,i)
      test2=pres(j-1,i)
1170  if (test1.gt.pcntr) go to 1180
      if (test2.gt.pcntr) go to 1190
      go to 1210
1180  if (test2.gt.pcntr) go to 1210
1190  ipx=i
      ipy1=j
      ipy2=j-1
      piso=float(ipy2)+(pcntr-test2)/(test1-test2)
      if (k.eq.2) go to 1200
      px=float(ipx)
c
c     write pressure contour information to the output file.
c
      write(8,1)pcntr,px,piso
1     format(f8.5,5x,f8.5,5x,f8.5)
      go to 1210
1200  px=float(ipx)
      write(8,2)pcntr,piso,px
2     format(f8.5,5x,f8.5,5x,f8.5)
1210  piso=0.0
1140  continue
c
c     increment the contour pressure value and repeat process.
c
      pcntr=pcntr+delp
```

```
      if (pcntr.le.cpmax) go to 1110
      close(8)
c
      open(9, file=VFILE, status='unknown')
      write(9,84)delv,cvmax
84    format(f6.3,3x,f7.4)
c
c     vcntr is the velocity contour value -- check the grid with
c     respect to x and y to bracket any contour velocity level.
c
1250  vcntr=delv
1260  viso=0.0
      ivx=0
      ivy1=0
      ivy2=0
      do 1280 i=1,N
      do 1280 j=1,N
      do 1280 k=1,2
      if (j.eq.1) go to 1340
      if (k.eq.2) go to 1290
      test1=vel(i,j)
      test2=vel(i,j-1)
      test3=va(i,j)
      test4=va(i,j-1)
      test5=ua(i,j)
      test6=ua(i,j-1)
      go to 1300
1290  test1=vel(j,i)
      test2=vel(j-1,i)
      test3=va(j,i)
      test4=va(j-1,i)
      test5=ua(j,i)
      test6=ua(j-1,i)
1300  if (test1.gt.vcntr) go to 1310
      if (test2.gt.vcntr) go to 1320
      go to 1340
1310  if (test2.gt.vcntr) go to 1340
1320  vx=float(i)
      vy1=float(j)
      vy2=float(j-1)
      viso=vy2+(vcntr-test2)/(test1-test2)
      yvel=test4+(viso-vy2)*(test3-test4)/(vy1-vy2)
      xvel=test6+(viso-vy2)*(test5-test6)/(vy1-vy2)
      if (k.eq.2) go to 1330
c
c     write contour information to the appropriate data file.
c
      write(9,211)vcntr,vx,viso,xvel,yvel
211   format(5(f6.3,2x))
      go to 1340
1330  write(9,211)vcntr,viso,vx,xvel,yvel
1340  viso=0.0
1280  continue
c
c     increment velocity contour value and repeat process.
c
      vcntr=vcntr+delv
      if (vcntr.le.cvmax) go to 1260
      close(9)
c
c     increment iteration count, set maximum advection speed for
c     the next iteration, and go back to the program start.
c
1800  continue
```

```
      if (icount.eq.icmax) go to 1860
1850  wplusc=afix
      icount=icount+1
      write(6,19)icount
19    format(' ICOUNT VALUE INCREMENTED TO ',i3)
      go to 300
1860  continue
      close(9)
      stop
      end
c
c
c     Subroutine Field computes the fluid properties based on finite
c     differencing relations developed through Rusanov procedure.
c
      subroutine field(da,ua,va,ea,d1,u1,v1,e1,d2,u2,v2,e2,
     1d3,u3,v3,e3,d4,u4,v4,e4,d5,u5,v5,e5,ind)
      common /dat/gamma,omega,xKn,l,N
c
c     compute advection speed for each grid location.
c
      t1=u1*u1+v1*v1
      t2=u2*u2+v2*v2
      t3=u3*u3+v3*v3
      t4=u4*u4+v4*v4
      t5=u5*u5+v5*v5
      p1=(gamma-1.0)*(e1-d1*t1/2.0)
      p2=(gamma-1.0)*(e2-d2*t2/2.0)
      p3=(gamma-1.0)*(e3-d3*t3/2.0)
      p4=(gamma-1.0)*(e4-d4*t4/2.0)
      p5=(gamma-1.0)*(e5-d5*t5/2.0)
      if (p1.gt.0.0) go to 2000
      t1=sqrt(t1)
      go to 2010
2000  t1=sqrt(gamma*p1/d1+t1)
2010  if (p2.gt.0.0) go to 2020
      t2=sqrt(t2)
      go to 2030
2020  t2=sqrt(gamma*p2/d2+t2)
2030  if (p3.gt.0.0) go to 2040
      t3=sqrt(t3)
      go to 2050
2040  t3=sqrt(gamma*p3/d3+t3)
2050  if (p4.gt.0.0) go to 2060
      t4=sqrt(t4)
      go to 2070
2060  t4=sqrt(gamma*p4/d4+t4)
2070  if (p5.gt.0.0) go to 2080
      t5=sqrt(t5)
      go to 2090
2080  t5=sqrt(gamma*p5/d5+t5)
2090  continue
c
c     If ind=1 then no rigid boundary is present -- normal.
c
      if (ind.ne.1) go to 2100
      r=2.0
      a=1.0
      b=1.0
      c=1.0
      d=1.0
      e=1.0
      f=1.0
      g=1.0
```

```
      h=1.0
      go to 2130
c
c     If ind=2 then flow occurs below a horizontal rigid boundary.
c
2100  if (ind.ne.2) go to 2110
      r=4.0
      a=1.0
      b=0.0
      c=1.0
      d=0.0
      e=1.0
      f=0.0
      g=1.0
      h=2.0
      go to 2130
c
c     If ind=3 then flow occurs right of a vertical rigid boundary.
c
2110  if (ind.ne.3) go to 2120
      r=4.0
      a=0.0
      b=1.0
      c=0.0
      d=1.0
      e=0.0
      f=1.0
      g=2.0
      h=1.0
      go to 2130
c
c     If ind=4 then flow occurs left of a vertical rigid boundary.
c
2120  if (ind.ne.4) go to 2130
      r=4.0
      a=0.0
      b=1.0
      c=0.0
      d=1.0
      e=2.0
      f=1.0
      g=0.0
      h=1.0
c
2130  sum=1.0-omega*xKn*(t1/r+(a*t2+b*t3+c*t4+d*t5)/8.0)
c
c     Compute density value at the grid location.
c
      da=d1*sum+omega*xKn*(a*t2*d2+b*t3*d3+c*t4*d4+d*t5*d5 +
     1(a*d2+b*d3+c*d4+d*d5)*t1)/8.0 - xKn*(-e*d2*u2+f*d3*v3+
     2g*d4*u4-h*d5*v5)/2.82842
c
c     Compute stagnation energy value at the grid location.
c
      ea=e1*sum+omega*xKn*(a*t2*e2+b*t3*e3+c*t4*e4+d*t5*e5 +
     1(a*e2+b*e3+c*e4+d*e5)*t1)/8.0 - xKn*(-e*(e2+p2)*u2+
     2f*(e3+p3)*v3+g*(e4+p4)*u4-h*(e5+p5)*v5)/2.82842
c
      if (ind.eq.3) then
      ua=0.0
      g=0.0
      goto 2160
      else if (ind.eq.4) then
      ua=0.0
```

```
      e=0.0
      goto 2160
      else
c
c     compute x directional velocity at the grid location.
c
      ua=(d1*u1*sum+omega*xKn*(a*t2*d2*u2+b*t3*d3*u3+c*t4*d4*u4+
     1d*t5*d5*u5 + (a*d2*u2+b*d3*u3+c*d4*u4+d*d5*u5)*t1)/8.0 -
     2xKn*(-e*(p2+d2*u2*u2)+f*d3*u3*v3+g*(p4+d4*u4*u4)-
     3h*d5*u5*v5)/2.82842)/da
      endif
c
2160  if (ind.eq.2) then
      va=0.0
      h=0.0
      goto 2180
      else
c
c     compute y directional velocity at the grid location.
c
      va=(d1*v1*sum+omega*xKn*(a*t2*d2*v2+b*t3*d3*v3+c*t4*d4*v4+
     1d*t5*d5*v5 + (a*d2*v2+b*d3*v3+c*d4*v4+d*d5*v5)*t1)/8.0 -
     2xKn*(-e*d2*u2*v2+f*(p3+d3*v3*v3)+g*d4*u4*v4-h*(p5+
     3d5*v5*v5))/2.8284)/da
      endif
c
2180  continue
      return
      end
```

APPENDIX H

COMPUTER PROGRAM LISTING FOR GENERATION OF LINEAR MEMBRANE/PLATE RESPONSE

```
c
c     Program WebResp.f -- Program computes plate or membrane response
c     to pressure levels generated during pulse modeling.  This is the
c     same as in program Pulse.f so notation may be checked there.
c     Dimensional time step is divided by ten before being used in the
c     membrane/plate equations.
c
      dimension df(10,21,21),uf(10,21,21),vf(10,21,21),ef(10,21,21)
      dimension da(21,21),ua(21,21),va(21,21),ea(21,21)
      dimension dxl(21),uxl(21),vxl(21),exl(21),dxr(21),uxr(21)
      dimension vxr(21),exr(21),dyl(21),uyl(21),vyl(21),eyl(21)
      dimension dyr(21),uyr(21),vyr(21),eyr(21),dxc(21),uxc(21)
      dimension vxc(21),exc(21),dtl(21),utl(21),vtl(21),etl(21)
      dimension dtr(21),utr(21),vtr(21),etr(21),dwl(21),uwl(21)
      dimension vwl(21),ewl(21),dwr(21),uwr(21),vwr(21),ewr(21)
      dimension dzl(21),uzl(21),vzl(21),ezl(21),dzr(21),uzr(21)
      dimension vzr(21),ezr(21),dsl(21),usl(21),vsl(21),esl(21)
      dimension dsr(21),usr(21),vsr(21),esr(21),dql(21),uql(21)
      dimension vql(21),eql(21),dqr(21),uqr(21),vqr(21),eqr(21)
      dimension drl(21),url(21),vrl(21),erl(21),drr(21),urr(21)
      dimension vrr(21),err(21),dpl(21),upl(21),vpl(21),epl(21)
      dimension dpr(21),upr(21),vpr(21),epr(21),WP(115)
      dimension dtime(3404),time(3404),vel(21,21),pres(21,21)
      double precision WP1(60,115),WM1(60,115),W(60,115)
c     dimension WP1(60,115),WM1(60,115),W(60,115)
      character*12 DFILE
      common /dat/gamma,omega,xKn,l,N
c
c     see listing Pulse.f for definition of variables.
c
      N=21
      Nx=113
      Ny=56
      gamma=1.4
      omega=1.345
      Gc=386.0
      dx=0.05
      delx=0.01
      ind=0
      Tens=1.0
      D=1.0e-4
      rhoweb=3.25e-5
      factr1=D/(delx*delx*delx*delx)
      factr2=Tens/(delx*delx)
      factr4=Tens*Gc/rhoweb
      data sigma,wplusc,patm,cair/0.5,2.4,14.7,13500.0/
      data pn,ps,pp,dn,ds,dp/2.0,1.5,1.0,1.63,1.33,1.0/
      data vn,vs,vp,un,us,up/1.0,0.558,0.0,0.0,0.0,0.0/
      data intc1,intc2,intc3,intc4/100,2300,2500,2700/
      data intc5,intc6,intc7,intc8/2900,3100,3500,3500/
      data intc9,intc10,intc11,intc12/3500,3500,3500,3500/
      data icmax/101/
      data WP(1),WP(2),WP(112),WP(113)/1.0,1.0,1.0,1.0/
      data m,sumt/1,0.0/
      data dxc/21*1.33/,uxc/21*0.0/,vxc/21*0.64/,exc/21*3.96/
      data dxl/21*1.0/,uxl/21*0.0/,vxl/21*0.0/,exl/21*2.5/
      data dxr/21*1.0/,uxr/21*0.0/,vxr/21*0.0/,exr/21*2.5/
      data dyl/21*1.0/,uyl/21*0.0/,vyl/21*0.0/,eyl/21*2.5/
      data dyr/21*1.0/,uyr/21*0.0/,vyr/21*0.0/,eyr/21*2.5/
      data dwl/21*1.0/,uwl/21*0.0/,vwl/21*0.0/,ewl/21*2.5/
      data dwr/21*1.0/,uwr/21*0.0/,vwr/21*0.0/,ewr/21*2.5/
      data dtl/21*1.0/,utl/21*0.0/,vtl/21*0.0/,etl/21*2.5/
      data dtr/21*1.0/,utr/21*0.0/,vtr/21*0.0/,etr/21*2.5/
      data dzl/21*1.0/,uzl/21*0.0/,vzl/21*0.0/,ezl/21*2.5/
```

```
      data dzr/21*1.0/,uzr/21*0.0/,vzr/21*0.0/,ezr/21*2.5/
      data dsl/21*1.0/,usl/21*0.0/,vsl/21*0.0/,esl/21*2.5/
      data dsr/21*1.0/,usr/21*0.0/,vsr/21*0.0/,esr/21*2.5/
      data dql/21*1.0/,uql/21*0.0/,vql/21*0.0/,eql/21*2.5/
      data dqr/21*1.0/,uqr/21*0.0/,vqr/21*0.0/,eqr/21*2.5/
      data drl/21*1.0/,url/21*0.0/,vrl/21*0.0/,erl/21*2.5/
      data drr/21*1.0/,urr/21*0.0/,vrr/21*0.0/,err/21*2.5/
      data dpl/21*1.0/,upl/21*0.0/,vpl/21*0.0/,epl/21*2.5/
      data dpr/21*1.0/,upr/21*0.0/,vpr/21*0.0/,epr/21*2.5/
      data (((df(l,i,j),l=1,10),i=1,21),j=1,21)/4410*1.0/
      data (((uf(l,i,j),l=1,10),i=1,21),j=1,21)/4410*0.0/
      data (((vf(l,i,j),l=1,10),i=1,21),j=1,21)/4410*0.0/
      data (((ef(l,i,j),l=1,10),i=1,21),j=1,21)/4410*2.5/
c
c     WP1 are membrane/plate deflections at time n+1 and are
c     the values solved for each iteration.  WM1 are membrane/
c     plate deflections at time n-1.  WP values are Web Pressure
c     values which are input to the membrane/plate equations.
c     W values are plate/membrane deflections at time n.
c
      data ((WP1(j,i), j=1,60), i=1,115)/6900*0.0/
      data ((WM1(j,i), j=1,60), i=1,115)/6900*0.0/
      data ((W(j,i), j=1,60), i=1,115)/6900*0.0/
      es=ds*(us*us+vs*vs)/2.0+ps/(gamma-1.0)
      ep=dp*(up*up+vp*vp)/2.0+pp/(gamma-1.0)
      en=dn*(un*un+vn*vn)/2.0+pn/(gamma-1.0)
      time(m)=0.0
      dtime(m)=0.0
      write(6,21)m,time(m)
21    format(' ICOUNT = ',i3,'   TIME = ',f6.4,/)
      open(9, file='time.dat', status='unknown')
c
c     initialize the field values of region 6
c
220   do 230 i=1,N
      do 230 j=2,N-1
      df(6,i,j)=dn
      uf(6,i,j)=un
      vf(6,i,j)=vn
      ef(6,i,j)=en
230   continue
      do 240 i=1,N
      df(6,i,N)=ds
      uf(6,i,N)=us
      vf(6,i,N)=vs
      ef(6,i,N)=es
240   continue
c
c
c     start the loop in the l number of regions
c     begin definition of the corner points of each region
c
300   afix=0.0
310   do 1800 l=1,10
      if (l.gt.1) go to 320
      xKn=sigma/wplusc
      sumt=sumt+xKn
c
c     deldt -- dimensional time step.  deldtp and deldtm are
c     dimensional time step for plate and membrane analysis
c     where deldt has been divided by ten.
c
      deldt=xKn*delx*sqrt(gamma)/(1.41421*cair)
      deldtp=deldt/10.0
```

```
      deldtm=deldt/10.0
      time(m)=(dx/1.41421)*sumt
      write(6,22)time(m)
22    format(' NON-DIMENSIONAL  TIME  VALUE  IS  NOW ',f12.9)
      dtime(m)=sumt*delx*sqrt(gamma)/(1.4142*cair)
      write(6,93)dtime(m)
93    format(' DIMENSIONAL  TIME  VALUE  IS  NOW  ',f15.12)
320   go to (330,350,370,390,410,430,450,470,490,510) l
330   write(9,91)m,time(m),dtime(m),deldt
91    format(i4,2x,f10.6,2x,f15.12,2x,f15.12)
      write(6,13)
13    format(' REGION  1  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      k=47
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(1,1,1),uf(1,1,1),
     1vf(1,1,1),ef(1,1,1),dyl(1),uyl(1),vyl(1),eyl(1),df(1,1,2),
     2uf(1,1,2),vf(1,1,2),ef(1,1,2),df(1,2,1),uf(1,2,1),vf(1,2,1),
     3ef(1,2,1),dxc(1),uxc(1),vxc(1),exc(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(1,1,N),uf(1,1,N),
     1vf(1,1,N),ef(1,1,N),dyl(N),uyl(N),vyl(N),eyl(N),df(1,1,N),
     2uf(1,1,N),-vf(1,1,N),ef(1,1,N),df(1,2,N),uf(1,2,N),vf(1,2,N),
     3ef(1,2,N),df(1,1,N-1),uf(1,1,N-1),vf(1,1,N-1),ef(1,1,N-1),2)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(1,N,1),uf(1,N,1),
     1vf(1,N,1),ef(1,N,1),df(1,N-1,1),uf(1,N-1,1),vf(1,N-1,1),
     2ef(1,N-1,1),df(1,N,2),uf(1,N,2),vf(1,N,2),ef(1,N,2),dyr(1),
     3uyr(1),vyr(1),eyr(1),dxc(N),uxc(N),vxc(N),exc(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(1,N,N),uf(1,N,N),
     1vf(1,N,N),ef(1,N,N),df(1,N-1,N),uf(1,N-1,N),vf(1,N-1,N),
     2ef(1,N-1,N),df(1,N,N),uf(1,N,N),-vf(1,N,N),ef(1,N,N),dyr(N),
     3uyr(N),vyr(N),eyr(N),df(1,N,N-1),uf(1,N,N-1),vf(1,N,N-1),
     4ef(1,N,N-1),2)
      do 340 i=2,N-1
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(1,i,1),uf(1,i,1),
     1vf(1,i,1),ef(1,i,1),df(1,i-1,1),uf(1,i-1,1),vf(1,i-1,1),
     2ef(1,i-1,1),df(1,i,2),uf(1,i,2),vf(1,i,2),ef(1,i,2),df(1,i+1,1),
     3uf(1,i+1,1),vf(1,i+1,1),ef(1,i+1,1),dxc(i),uxc(i),vxc(i),
     4exc(i),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(1,i,N),uf(1,i,N),
     1vf(1,i,N),ef(1,i,N),df(1,i-1,N),uf(1,i-1,N),vf(1,i-1,N),
     2ef(1,i-1,N),df(1,i,N),uf(1,i,N),-vf(1,i,N),ef(1,i,N),df(1,i+1,N),
     3uf(1,i+1,N),vf(1,i+1,N),ef(1,i+1,N),df(1,i,N-1),uf(1,i,N-1),
     4vf(1,i,N-1),ef(1,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(1,N,i),uf(1,N,i),
     1vf(1,N,i),ef(1,N,i),df(1,N-1,i),uf(1,N-1,i),vf(1,N-1,i),
     2ef(1,N-1,i),df(1,N,i+1),uf(1,N,i+1),vf(1,N,i+1),ef(1,N,i+1),
     3dyr(i),uyr(i),vyr(i),eyr(i),df(1,N,i-1),uf(1,N,i-1),vf(1,N,i-1),
     4ef(1,N,i-1),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(1,1,i),uf(1,1,i),
     1vf(1,1,i),ef(1,1,i),dyl(i),uyl(i),vyl(i),eyl(i),df(1,1,i+1),
     2uf(1,1,i+1),vf(1,1,i+1),ef(1,1,i+1),df(1,2,i),uf(1,2,i),
     3vf(1,2,i),ef(1,2,i),df(1,1,i-1),uf(1,1,i-1),vf(1,1,i-1),
     4ef(1,1,i-1),1)
340   continue
      go to 530
350   write(6,14)
14    format(' REGION  2  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      k=25
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(2,N,1),uf(2,N,1),
     1vf(2,N,1),ef(2,N,1),df(2,N-1,1),uf(2,N-1,1),vf(2,N-1,1),
     2ef(2,N-1,1),df(2,N,2),uf(2,N,2),vf(2,N,2),ef(2,N,2),dyl(1),
     3uyl(1),vyl(1),eyl(1),dxl(N),uxl(N),vxl(N),exl(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(2,N,N),uf(2,N,N),
     1vf(2,N,N),ef(2,N,N),df(2,N-1,N),uf(2,N-1,N),vf(2,N-1,N),
```

```
     2ef(2,N-1,N),df(2,N,N),uf(2,N,N),-vf(2,N,N),ef(2,N,N),dyl(N),
     3uyl(N),vyl(N),eyl(N),df(2,N,N-1),uf(2,N,N-1),vf(2,N,N-1),
     4ef(2,N,N-1),2)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(2,1,1),uf(2,1,1),
     1vf(2,1,1),ef(2,1,1),dzl(1),uzl(1),vzl(1),ezl(1),df(2,1,2),
     2uf(2,1,2),vf(2,1,2),ef(2,1,2),df(2,2,1),uf(2,2,1),vf(2,2,1),
     3ef(2,2,1),dxl(1),uxl(1),vxl(1),exl(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(2,1,N),uf(2,1,N),
     1vf(2,1,N),ef(2,1,N),dzl(N),uzl(N),vzl(N),ezl(N),df(2,1,N),
     2uf(2,1,N),-vf(2,1,N),ef(2,1,N),df(2,2,N),uf(2,2,N),vf(2,2,N),
     3ef(2,2,N),df(2,1,N-1),uf(2,1,N-1),vf(2,1,N-1),ef(2,1,N-1),2)
      do 360 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(2,i,N),uf(2,i,N),
     1vf(2,i,N),ef(2,i,N),df(2,i-1,N),uf(2,i-1,N),vf(2,i-1,N),
     2ef(2,i-1,N),df(2,i,N),uf(2,i,N),-vf(2,i,N),ef(2,i,N),df(2,i+1,N),
     3uf(2,i+1,N),vf(2,i+1,N),ef(2,i+1,N),df(2,i,N-1),uf(2,i,N-1),
     4vf(2,i,N-1),ef(2,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(2,N,i),uf(2,N,i),
     1vf(2,N,i),ef(2,N,i),df(2,N-1,i),uf(2,N-1,i),vf(2,N-1,i),
     2ef(2,N-1,i),df(2,N,i+1),uf(2,N,i+1),vf(2,N,i+1),ef(2,N,i+1),
     3dyl(i),uyl(i),vyl(i),eyl(i),df(2,N,i-1),uf(2,N,i-1),vf(2,N,i-1),
     4ef(2,N,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(2,i,1),uf(2,i,1),
     1vf(2,i,1),ef(2,i,1),df(2,i-1,1),uf(2,i-1,1),vf(2,i-1,1),
     2ef(2,i-1,1),df(2,i,2),uf(2,i,2),vf(2,i,2),ef(2,i,2),df(2,i+1,1),
     3uf(2,i+1,1),vf(2,i+1,1),ef(2,i+1,1),dxl(i),uxl(i),vxl(i),
     4exl(i),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(2,1,i),uf(2,1,i),
     1vf(2,1,i),ef(2,1,i),dzl(i),uzl(i),vzl(i),ezl(i),df(2,1,i+1),
     2uf(2,1,i+1),vf(2,1,i+1),ef(2,1,i+1),df(2,2,i),uf(2,2,i),
     3vf(2,2,i),ef(2,2,i),df(2,1,i-1),uf(2,1,i-1),vf(2,1,i-1),
     4ef(2,1,i-1),1)
360   continue
      go to 530
370   write(6,16)
16    format(' REGION  3  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      k=69
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(3,1,1),
     1uf(3,1,1),vf(3,1,1),ef(3,1,1),dyr(1),uyr(1),vyr(1),eyr(1),
     2df(3,1,2),uf(3,1,2),vf(3,1,2),ef(3,1,2),df(3,2,1),uf(3,2,1),
     3vf(3,2,1),ef(3,2,1),dxr(1),uxr(1),vxr(1),exr(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(3,1,N),uf(3,1,N),
     1vf(3,1,N),ef(3,1,N),dyr(N),uyr(N),vyr(N),eyr(N),df(3,1,N),
     2uf(3,1,N),-vf(3,1,N),ef(3,1,N),df(3,2,N),uf(3,2,N),vf(3,2,N),
     3ef(3,2,N),df(3,1,N-1),uf(3,1,N-1),vf(3,1,N-1),ef(3,1,N-1),2)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(3,N,1),uf(3,N,1),
     1vf(3,N,1),ef(3,N,1),df(3,N-1,1),uf(3,N-1,1),vf(3,N-1,1),
     2ef(3,N-1,1),df(3,N,2),uf(3,N,2),vf(3,N,2),ef(3,N,2),dzr(1),uzr(1),
     3vzr(1),ezr(1),dxr(N),uxr(N),vxr(N),exr(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(3,N,N),
     1uf(3,N,N),vf(3,N,N),ef(3,N,N),df(3,N-1,N),uf(3,N-1,N),vf(3,N-1,N),
     2ef(3,N-1,N),df(3,N,N),uf(3,N,N),-vf(3,N,N),ef(3,N,N),dzr(N),
     3uzr(N),vzr(N),ezr(N),df(3,N,N-1),uf(3,N,N-1),vf(3,N,N-1),
     4ef(3,N,N-1),2)
      do 380 i=2,N-1
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(3,1,i),
     1uf(3,1,i),vf(3,1,i),ef(3,1,i),dyr(i),uyr(i),vyr(i),eyr(i),
     2df(3,1,i+1),uf(3,1,i+1),vf(3,1,i+1),ef(3,1,i+1),df(3,2,i),
     3uf(3,2,i),vf(3,2,i),ef(3,2,i),df(3,1,i-1),uf(3,1,i-1),vf(3,1,i-1),
     4ef(3,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(3,i,1),
     1uf(3,i,1),vf(3,i,1),ef(3,i,1),df(3,i-1,1),uf(3,i-1,1),vf(3,i-1,1),
     2ef(3,i-1,1),df(3,i,2),uf(3,i,2),vf(3,i,2),ef(3,i,2),df(3,i+1,1),
```

```
     3uf(3,i+1,1),vf(3,i+1,1),ef(3,i+1,1),dxr(i),uxr(i),vxr(i),exr(i),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(3,i,N),
     luf(3,i,N),vf(3,i,N),ef(3,i,N),df(3,i-1,N),uf(3,i-1,N),vf(3,i-1,N),
     2ef(3,i-1,N),df(3,i,N),uf(3,i,N),-vf(3,i,N),ef(3,i,N),df(3,i+1,N),
     3uf(3,i+1,N),vf(3,i+1,N),ef(3,i+1,N),df(3,i,N-1),uf(3,i,N-1),
     4vf(3,i,N-1),ef(3,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(3,N,i),uf(3,N,i),
     1vf(3,N,i),ef(3,N,i),df(3,N-1,i),uf(3,N-1,i),vf(3,N-1,i),
     2ef(3,N-1,i),df(3,N,i+1),uf(3,N,i+1),vf(3,N,i+1),ef(3,N,i+1),
     3dzr(i),uzr(i),vzr(i),ezr(i),df(3,N,i-1),uf(3,N,i-1),vf(3,N,i-1),
     4ef(3,N,i-1),1)
380   continue
      go to 530
390   write(6,17)
17    format(' REGION  4  BEING  PROCESSED')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(4,N,N),
     luf(4,N,N),vf(4,N,N),ef(4,N,N),df(4,N-1,N),uf(4,N-1,N),vf(4,N-1,N),
     2ef(4,N-1,N),dxl(N),uxl(N),vxl(N),exl(N),df(4,N,N),-uf(4,N,N),
     3vf(4,N,N),ef(4,N,N),df(4,N,N-1),uf(4,N,N-1),vf(4,N,N-1),
     4ef(4,N,N-1),4)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(4,1,N),
     luf(4,1,N),vf(4,1,N),ef(4,1,N),dtl(N),utl(N),vtl(N),etl(N),dxl(1),
     2uxl(1),vxl(1),exl(1),df(4,2,N),uf(4,2,N),vf(4,2,N),ef(4,2,N),
     3df(4,1,N-1),uf(4,1,N-1),vf(4,1,N-1),ef(4,1,N-1),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(4,1,1),uf(4,1,1),
     1vf(4,1,1),ef(4,1,1),dtl(1),utl(1),vtl(1),etl(1),df(4,1,2),
     2uf(4,1,2),vf(4,1,2),ef(4,1,2),df(4,2,1),uf(4,2,1),vf(4,2,1),
     3ef(4,2,1),dsl(1),usl(1),vsl(1),esl(1),1)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(4,N,1),uf(4,N,1),
     1vf(4,N,1),ef(4,N,1),df(4,N-1,1),uf(4,N-1,1),vf(4,N-1,1),
     2ef(4,N-1,1),df(4,N,2),uf(4,N,2),vf(4,N,2),ef(4,N,2),df(4,N,1),
     3-uf(4,N,1),vf(4,N,1),ef(4,N,1),dsl(N),usl(N),vsl(N),esl(N),4)
      do 400 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(4,i,N),
     luf(4,i,N),vf(4,i,N),ef(4,i,N),df(4,i-1,N),uf(4,i-1,N),vf(4,i-1,N),
     2ef(4,i-1,N),dxl(i),uxl(i),vxl(i),exl(i),df(4,i+1,N),uf(4,i+1,N),
     3vf(4,i+1,N),ef(4,i+1,N),df(4,i,N-1),uf(4,i,N-1),vf(4,i,N-1),
     4ef(4,i,N-1),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(4,N,i),
     luf(4,N,i),vf(4,N,i),ef(4,N,i),df(4,N-1,i),uf(4,N-1,i),vf(4,N-1,i),
     2ef(4,N-1,i),df(4,N,i+1),uf(4,N,i+1),vf(4,N,i+1),ef(4,N,i+1),
     3df(4,N,i),-uf(4,N,i),vf(4,N,i),ef(4,N,i),df(4,N,i-1),uf(4,N,i-1),
     4vf(4,N,i-1),ef(4,N,i-1),4)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(4,1,i),
     luf(4,1,i),vf(4,1,i),ef(4,1,i),dtl(i),utl(i),vtl(i),etl(i),
     2df(4,1,i+1),uf(4,1,i+1),vf(4,1,i+1),ef(4,1,i+1),df(4,2,i),
     3uf(4,2,i),vf(4,2,i),ef(4,2,i),df(4,1,i-1),uf(4,1,i-1),
     4vf(4,1,i-1),ef(4,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(4,i,1),uf(4,i,1),
     1vf(4,i,1),ef(4,i,1),df(4,i-1,1),uf(4,i-1,1),vf(4,i-1,1),
     2ef(4,i-1,1),df(4,i,2),uf(4,i,2),vf(4,i,2),ef(4,i,2),df(4,i+1,1),
     3uf(4,i+1,1),vf(4,i+1,1),ef(4,i+1,1),dsl(i),usl(i),vsl(i),esl(i),1)
400   continue
      go to 530
410   write(6,27)
27    format(' REGION  5  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(5,1,N),
     luf(5,1,N),vf(5,1,N),ef(5,1,N),df(5,1,N),-uf(5,1,N),vf(5,1,N),
     2ef(5,1,N),dxr(1),uxr(1),vxr(1),exr(1),df(5,2,N),uf(5,2,N),
     3vf(5,2,N),ef(5,2,N),df(5,1,N-1),uf(5,1,N-1),vf(5,1,N-1),
     4ef(5,1,N-1),3)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(5,N,N),uf(5,N,N),
```

```
     1vf(5,N,N),ef(5,N,N),df(5,N-1,N),uf(5,N-1,N),vf(5,N-1,N),
     2ef(5,N-1,N),dxr(N),uxr(N),vxr(N),exr(N),dtr(N),utr(N),vtr(N),
     3etr(N),df(5,N,N-1),uf(5,N,N-1),vf(5,N,N-1),ef(5,N,N-1),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(5,1,1),uf(5,1,1),
     1vf(5,1,1),ef(5,1,1),df(5,1,1),-uf(5,1,1),vf(5,1,1),ef(5,1,1),
     2df(5,1,2),uf(5,1,2),vf(5,1,2),ef(5,1,2),df(5,2,1),uf(5,2,1),
     3vf(5,2,1),ef(5,2,1),dsr(1),usr(1),vsr(1),esr(1),3)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(5,N,1),uf(5,N,1),
     1vf(5,N,1),ef(5,N,1),df(5,N-1,1),uf(5,N-1,1),vf(5,N-1,1),
     2ef(5,N-1,1),df(5,N,2),uf(5,N,2),vf(5,N,2),ef(5,N,2),dtr(1),
     3utr(1),vtr(1),etr(1),dsr(N),usr(N),vsr(N),esr(N),1)
      do 420 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(5,i,N),uf(5,i,N),
     1vf(5,i,N),ef(5,i,N),df(5,i-1,N),uf(5,i-1,N),vf(5,i-1,N),
     2ef(5,i-1,N),dxr(i),uxr(i),vxr(i),exr(i),df(5,i+1,N),uf(5,i+1,N),
     3vf(5,i+1,N),ef(5,i+1,N),df(5,i,N-1),uf(5,i,N-1),vf(5,i,N-1),
     4ef(5,i,N-1),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(5,1,i),
     1uf(5,1,i),vf(5,1,i),ef(5,1,i),df(5,1,i),-uf(5,1,i),vf(5,1,i),
     2ef(5,1,i),df(5,1,i+1),uf(5,1,i+1),vf(5,1,i+1),ef(5,1,i+1),
     3df(5,2,i),uf(5,2,i),vf(5,2,i),ef(5,2,i),df(5,1,i-1),uf(5,1,i-1),
     4vf(5,1,i-1),ef(5,1,i-1),3)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(5,N,i),uf(5,N,i),
     1vf(5,N,i),ef(5,N,i),df(5,N-1,i),uf(5,N-1,i),vf(5,N-1,i),
     2ef(5,N-1,i),df(5,N,i+1),uf(5,N,i+1),vf(5,N,i+1),ef(5,N,i+1),
     3dtr(i),utr(i),vtr(i),etr(i),df(5,N,i-1),uf(5,N,i-1),vf(5,N,i-1),
     4ef(5,N,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(5,i,1),uf(5,i,1),
     1vf(5,i,1),ef(5,i,1),df(5,i-1,1),uf(5,i-1,1),vf(5,i-1,1),
     2ef(5,i-1,1),df(5,i,2),uf(5,i,2),vf(5,i,2),ef(5,i,2),df(5,i+1,1),
     3uf(5,i+1,1),vf(5,i+1,1),ef(5,i+1,1),dsr(i),usr(i),vsr(i),esr(i),1)
420   continue
      go to 530
430   write(6,33)
33    format(' REGION  6  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(6,1,N),uf(6,1,N),
     1vf(6,1,N),ef(6,1,N),df(6,1,N),-uf(6,1,N),vf(6,1,N),ef(6,1,N),
     2dxc(1),uxc(1),vxc(1),exc(1),df(6,2,N),uf(6,2,N),vf(6,2,N),
     3ef(6,2,N),df(6,1,N-1),uf(6,1,N-1),vf(6,1,N-1),ef(6,1,N-1),3)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(6,N,N),
     1uf(6,N,N),vf(6,N,N),ef(6,N,N),df(6,N-1,N),uf(6,N-1,N),vf(6,N-1,N),
     2ef(6,N-1,N),dxc(N),uxc(N),vxc(N),exc(N),df(6,N,N),-uf(6,N,N),
     3vf(6,N,N),ef(6,N,N),df(6,N,N-1),uf(6,N,N-1),vf(6,N,N-1),
     4ef(6,N,N-1),4)
      do 440 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(6,i,N),
     1uf(6,i,N),vf(6,i,N),ef(6,i,N),df(6,i-1,N),uf(6,i-1,N),vf(6,i-1,N),
     2ef(6,i-1,N),dxc(i),uxc(i),vxc(i),exc(i),df(6,i+1,N),uf(6,i+1,N),
     3vf(6,i+1,N),ef(6,i+1,N),df(6,i,N-1),uf(6,i,N-1),vf(6,i,N-1),
     4ef(6,i,N-1),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(6,N,i),
     1uf(6,N,i),vf(6,N,i),ef(6,N,i),df(6,N-1,i),uf(6,N-1,i),vf(6,N-1,i),
     2ef(6,N-1,i),df(6,N,i+1),uf(6,N,i+1),vf(6,N,i+1),ef(6,N,i+1),
     3df(6,N,i),-uf(6,N,i),vf(6,N,i),ef(6,N,i),df(6,N,i-1),uf(6,N,i-1),
     4vf(6,N,i-1),ef(6,N,i-1),4)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(6,1,i),
     1uf(6,1,i),vf(6,1,i),ef(6,1,i),df(6,1,i),-uf(6,1,i),vf(6,1,i),
     2ef(6,1,i),df(6,1,i+1),uf(6,1,i+1),vf(6,1,i+1),ef(6,1,i+1),
     3df(6,2,i),uf(6,2,i),vf(6,2,i),ef(6,2,i),df(6,1,i-1),uf(6,1,i-1),
     4vf(6,1,i-1),ef(6,1,i-1),3)
440   continue
      go to 530
450   write(6,37)
```

```
37    format(' REGION  7  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      k=3
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(7,N,1),
     1uf(7,N,1),vf(7,N,1),ef(7,N,1),df(7,N-1,1),uf(7,N-1,1),vf(7,N-1,1),
     2ef(7,N-1,1),df(7,N,2),uf(7,N,2),vf(7,N,2),ef(7,N,2),dzl(1),uzl(1),
     3vzl(1),ezl(1),dwl(N),uwl(N),vwl(N),ewl(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(7,N,N),
     1uf(7,N,N),vf(7,N,N),ef(7,N,N),df(7,N-1,N),uf(7,N-1,N),vf(7,N-1,N),
     2ef(7,N-1,N),df(7,N,N),uf(7,N,N),-vf(7,N,N),ef(7,N,N),dzl(N),
     3uzl(N),vzl(N),ezl(N),df(7,N,N-1),uf(7,N,N-1),vf(7,N,N-1),
     4ef(7,N,N-1),2)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(7,1,1),uf(7,1,1),
     1vf(7,1,1),ef(7,1,1),dql(1),uql(1),vql(1),eql(1),df(7,1,2),
     2uf(7,1,2),vf(7,1,2),ef(7,1,2),df(7,2,1),uf(7,2,1),vf(7,2,1),
     3ef(7,2,1),dwl(1),uwl(1),vwl(1),ewl(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(7,1,N),uf(7,1,N),
     1vf(7,1,N),ef(7,1,N),dql(N),uql(N),vql(N),eql(N),df(7,1,N),
     2uf(7,1,N),-vf(7,1,N),ef(7,1,N),df(7,2,N),uf(7,2,N),vf(7,2,N),
     3ef(7,2,N),df(7,1,N-1),uf(7,1,N-1),vf(7,1,N-1),ef(7,1,N-1),2)
      do 460 i=2,N-1
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(7,i,1),
     1uf(7,i,1),vf(7,i,1),ef(7,i,1),df(7,i-1,1),uf(7,i-1,1),vf(7,i-1,1),
     2ef(7,i-1,1),df(7,i,2),uf(7,i,2),vf(7,i,2),ef(7,i,2),df(7,i+1,1),
     3uf(7,i+1,1),vf(7,i+1,1),ef(7,i+1,1),dwl(i),uwl(i),vwl(i),ewl(i),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(7,N,i),
     1uf(7,N,i),vf(7,N,i),ef(7,N,i),df(7,N-1,i),uf(7,N-1,i),vf(7,N-1,i),
     2ef(7,N-1,i),df(7,N,i+1),uf(7,N,i+1),vf(7,N,i+1),ef(7,N,i+1),
     3dzl(i),uzl(i),vzl(i),ezl(i),df(7,N,i-1),uf(7,N,i-1),vf(7,N,i-1),
     4ef(7,N,i-1),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(7,i,N),
     1uf(7,i,N),vf(7,i,N),ef(7,i,N),df(7,i-1,N),uf(7,i-1,N),
     2vf(7,i-1,N),ef(7,i-1,N),df(7,i,N),uf(7,i,N),-vf(7,i,N),
     3ef(7,i,N),df(7,i+1,N),uf(7,i+1,N),vf(7,i+1,N),ef(7,i+1,N),
     4df(7,i,N-1),uf(7,i,N-1),vf(7,i,N-1),ef(7,i,N-1),2)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(7,1,i),uf(7,1,i),
     1vf(7,1,i),ef(7,1,i),dql(i),uql(i),vql(i),eql(i),df(7,1,i+1),
     2uf(7,1,i+1),vf(7,1,i+1),ef(7,1,i+1),df(7,2,i),uf(7,2,i),
     3vf(7,2,i),ef(7,2,i),df(7,1,i-1),uf(7,1,i-1),vf(7,1,i-1),
     4ef(7,1,i-1),1)
460   continue
      go to 530
470   write(6,41)
41    format(' REGION  8  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      k=91
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(8,1,1),uf(8,1,1),
     1vf(8,1,1),ef(8,1,1),dzr(1),uzr(1),vzr(1),ezr(1),df(8,1,2),
     2uf(8,1,2),vf(8,1,2),ef(8,1,2),df(8,2,1),uf(8,2,1),vf(8,2,1),
     3ef(8,2,1),dwr(1),uwr(1),vwr(1),ewr(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(8,1,N),uf(8,1,N),
     1vf(8,1,N),ef(8,1,N),dzr(N),uzr(N),vzr(N),ezr(N),df(8,1,N),
     2uf(8,1,N),-vf(8,1,N),ef(8,1,N),df(8,2,N),uf(8,2,N),vf(8,2,N),
     3ef(8,2,N),df(8,1,N-1),uf(8,1,N-1),vf(8,1,N-1),ef(8,1,N-1),2)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(8,N,1),uf(8,N,1),
     1vf(8,N,1),ef(8,N,1),df(8,N-1,1),uf(8,N-1,1),vf(8,N-1,1),
     2ef(8,N-1,1),df(8,N,2),uf(8,N,2),vf(8,N,2),ef(8,N,2),
     3dqr(1),uqr(1),vqr(1),eqr(1),dwr(N),uwr(N),vwr(N),ewr(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(8,N,N),uf(8,N,N),
     1vf(8,N,N),ef(8,N,N),df(8,N-1,N),uf(8,N-1,N),vf(8,N-1,N),
     2ef(8,N-1,N),df(8,N,N),uf(8,N,N),-vf(8,N,N),ef(8,N,N),dqr(N),
     3uqr(N),vqr(N),eqr(N),df(8,N,N-1),uf(8,N,N-1),vf(8,N,N-1),
     4ef(8,N,N-1),2)
      do 480 i=2,N-1
```

```
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(8,1,i),
     1uf(8,1,i),vf(8,1,i),ef(8,1,i),dzr(i),uzr(i),vzr(i),ezr(i),
     2df(8,1,i+1),uf(8,1,i+1),vf(8,1,i+1),ef(8,1,i+1),df(8,2,i),
     3uf(8,2,i),vf(8,2,i),ef(8,2,i),df(8,1,i-1),uf(8,1,i-1),vf(8,1,i-1),
     4ef(8,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(8,i,1),
     1uf(8,i,1),vf(8,i,1),ef(8,i,1),df(8,i-1,1),uf(8,i-1,1),vf(8,i-1,1),
     2ef(8,i-1,1),df(8,i,2),uf(8,i,2),vf(8,i,2),ef(8,i,2),df(8,i+1,1),
     3uf(8,i+1,1),vf(8,i+1,1),ef(8,i+1,1),dwr(i),uwr(i),vwr(i),ewr(i),1)
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(8,i,N),
     1uf(8,i,N),vf(8,i,N),ef(8,i,N),df(8,i-1,N),uf(8,i-1,N),vf(8,i-1,N),
     2ef(8,i-1,N),df(8,i,N),uf(8,i,N),-vf(8,i,N),ef(8,i,N),df(8,i+1,N),
     3uf(8,i+1,N),vf(8,i+1,N),ef(8,i+1,N),df(8,i,N-1),uf(8,i,N-1),
     4vf(8,i,N-1),ef(8,i,N-1),2)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(8,N,i),uf(8,N,i),
     1vf(8,N,i),ef(8,N,i),df(8,N-1,i),uf(8,N-1,i),vf(8,N-1,i),
     2ef(8,N-1,i),df(8,N,i+1),uf(8,N,i+1),vf(8,N,i+1),ef(8,N,i+1),
     3dqr(i),uqr(i),vqr(i),eqr(i),df(8,N,i-1),uf(8,N,i-1),vf(8,N,i-1),
     4ef(8,N,i-1),1)
480   continue
      go to 530
490   write(6,47)
47    format(' REGION  9  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(9,N,N),uf(9,N,N),
     1vf(9,N,N),ef(9,N,N),df(9,N-1,N),uf(9,N-1,N),vf(9,N-1,N),
     2ef(9,N-1,N),dwl(N),uwl(N),vwl(N),ewl(N),dtl(N),utl(N),vtl(N),
     3etl(N),df(9,N,N-1),uf(9,N,N-1),vf(9,N,N-1),ef(9,N,N-1),1)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(9,N,1),uf(9,N,1),
     1vf(9,N,1),ef(9,N,1),df(9,N-1,1),uf(9,N-1,1),vf(9,N-1,1),
     2ef(9,N-1,1),df(9,N,2),uf(9,N,2),vf(9,N,2),ef(9,N,2),
     3dtl(1),utl(1),vtl(1),etl(1),drl(N),url(N),vrl(N),erl(N),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(9,1,1),uf(9,1,1),
     1vf(9,1,1),ef(9,1,1),dpl(1),upl(1),vpl(1),epl(1),df(9,1,2),
     2uf(9,1,2),vf(9,1,2),ef(9,1,2),df(9,2,1),uf(9,2,1),vf(9,2,1),
     3ef(9,2,1),drl(1),url(1),vrl(1),erl(1),1)
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(9,1,N),uf(9,1,N),
     1vf(9,1,N),ef(9,1,N),dpl(N),upl(N),vpl(N),epl(N),dwl(1),uwl(1),
     2vwl(1),ewl(1),df(9,2,N),uf(9,2,N),vf(9,2,N),ef(9,2,N),
     3df(9,1,N-1),uf(9,1,N-1),vf(9,1,N-1),ef(9,1,N-1),1)
      do 500 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(9,i,N),
     1uf(9,i,N),vf(9,i,N),ef(9,i,N),df(9,i-1,N),uf(9,i-1,N),vf(9,i-1,N),
     2ef(9,i-1,N),dwl(i),uwl(i),vwl(i),ewl(i),df(9,i+1,N),uf(9,i+1,N),
     3vf(9,i+1,N),ef(9,i+1,N),df(9,i,N-1),uf(9,i,N-1),vf(9,i,N-1),
     4ef(9,i,N-1),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(9,N,i),
     1uf(9,N,i),vf(9,N,i),ef(9,N,i),df(9,N-1,i),uf(9,N-1,i),vf(9,N-1,i),
     2ef(9,N-1,i),df(9,N,i+1),uf(9,N,i+1),vf(9,N,i+1),ef(9,N,i+1),
     3dtl(i),utl(i),vtl(i),etl(i),df(9,N,i-1),uf(9,N,i-1),vf(9,N,i-1),
     4ef(9,N,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(9,i,1),uf(9,i,1),
     1vf(9,i,1),ef(9,i,1),df(9,i-1,1),uf(9,i-1,1),vf(9,i-1,1),
     2ef(9,i-1,1),df(9,i,2),uf(9,i,2),vf(9,i,2),ef(9,i,2),df(9,i+1,1),
     3uf(9,i+1,1),vf(9,i+1,1),ef(9,i+1,1),drl(i),url(i),vrl(i),erl(i),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(9,1,i),uf(9,1,i),
     1vf(9,1,i),ef(9,1,i),dpl(i),upl(i),vpl(i),epl(i),df(9,1,i+1),
     2uf(9,1,i+1),vf(9,1,i+1),ef(9,1,i+1),df(9,2,i),uf(9,2,i),vf(9,2,i),
     3ef(9,2,i),df(9,1,i-1),uf(9,1,i-1),vf(9,1,i-1),ef(9,1,i-1),1)
500   continue
      go to 530
510   write(6,53)
53    format(' REGION  10  BEING  PROCESSED ')
c     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
      call field(da(1,N),ua(1,N),va(1,N),ea(1,N),df(10,1,N),
     1uf(10,1,N),vf(10,1,N),ef(10,1,N),dtr(N),utr(N),vtr(N),etr(N),
     2dwr(1),uwr(1),vwr(1),ewr(1),df(10,2,N),uf(10,2,N),vf(10,2,N),
     3ef(10,2,N),df(10,1,N-1),uf(10,1,N-1),vf(10,1,N-1),ef(10,1,N-1),1)
      call field(da(1,1),ua(1,1),va(1,1),ea(1,1),df(10,1,1),uf(10,1,1),
     1vf(10,1,1),ef(10,1,1),dtr(1),utr(1),vtr(1),etr(1),df(10,1,2),
     2uf(10,1,2),vf(10,1,2),ef(10,1,2),df(10,2,1),uf(10,2,1),vf(10,2,1),
     3ef(10,2,1),drr(1),urr(1),vrr(1),err(1),1)
      call field(da(N,1),ua(N,1),va(N,1),ea(N,1),df(10,N,1),uf(10,N,1),
     1vf(10,N,1),ef(10,N,1),df(10,N-1,1),uf(10,N-1,1),vf(10,N-1,1),
     2ef(10,N-1,1),df(10,N,2),uf(10,N,2),vf(10,N,2),ef(10,N,2),
     3dpr(1),upr(1),vpr(1),epr(1),drr(N),urr(N),vrr(N),err(N),1)
      call field(da(N,N),ua(N,N),va(N,N),ea(N,N),df(10,N,N),uf(10,N,N),
     1vf(10,N,N),ef(10,N,N),df(10,N-1,N),uf(10,N-1,N),vf(10,N-1,N),
     2ef(10,N-1,N),dwr(N),uwr(N),vwr(N),ewr(N),dpr(N),upr(N),vpr(N),
     3epr(N),df(10,N,N-1),uf(10,N,N-1),vf(10,N,N-1),ef(10,N,N-1),1)
      do 520 i=2,N-1
      call field(da(i,N),ua(i,N),va(i,N),ea(i,N),df(10,i,N),
     1uf(10,i,N),vf(10,i,N),ef(10,i,N),df(10,i-1,N),uf(10,i-1,N),
     2vf(10,i-1,N),ef(10,i-1,N),dwr(i),uwr(i),vwr(i),ewr(i),
     3df(10,i+1,N),uf(10,i+1,N),vf(10,i+1,N),ef(10,i+1,N),df(10,i,N-1),
     4uf(10,i,N-1),vf(10,i,N-1),ef(10,i,N-1),1)
      call field(da(1,i),ua(1,i),va(1,i),ea(1,i),df(10,1,i),
     1uf(10,1,i),vf(10,1,i),ef(10,1,i),dtr(i),utr(i),vtr(i),etr(i),
     2df(10,1,i+1),uf(10,1,i+1),vf(10,1,i+1),ef(10,1,i+1),df(10,2,i),
     3uf(10,2,i),vf(10,2,i),ef(10,2,i),df(10,1,i-1),uf(10,1,i-1),
     4vf(10,1,i-1),ef(10,1,i-1),1)
      call field(da(i,1),ua(i,1),va(i,1),ea(i,1),df(10,i,1),uf(10,i,1),
     1vf(10,i,1),ef(10,i,1),df(10,i-1,1),uf(10,i-1,1),vf(10,i-1,1),
     2ef(10,i-1,1),df(10,i,2),uf(10,i,2),vf(10,i,2),ef(10,i,2),
     3df(10,i+1,1),uf(10,i+1,1),vf(10,i+1,1),ef(10,i+1,1),drr(i),
     4urr(i),vrr(i),err(i),1)
      call field(da(N,i),ua(N,i),va(N,i),ea(N,i),df(10,N,i),uf(10,N,i),
     1vf(10,N,i),ef(10,N,i),df(10,N-1,i),uf(10,N-1,i),vf(10,N-1,i),
     2ef(10,N-1,i),df(10,N,i+1),uf(10,N,i+1),vf(10,N,i+1),ef(10,N,i+1),
     3dpr(i),upr(i),vpr(i),epr(i),df(10,N,i-1),uf(10,N,i-1),
     4vf(10,N,i-1),ef(10,N,i-1),1)
520   continue
c
530   do 540 i=2,N-1
      do 540 j=2,N-1
      call field(da(i,j),ua(i,j),va(i,j),ea(i,j),df(1,i,j),uf(1,i,j),
     1vf(1,i,j),ef(1,i,j),df(1,i-1,j),uf(1,i-1,j),vf(1,i-1,j),
     2ef(1,i-1,j),df(1,i,j+1),uf(1,i,j+1),vf(1,i,j+1),ef(1,i,j+1),
     3df(1,i+1,j),uf(1,i+1,j),vf(1,i+1,j),ef(1,i+1,j),df(1,i,j-1),
     4uf(1,i,j-1),vf(1,i,j-1),ef(1,i,j-1),1)
540   continue
c
c     determine the boundary conditions for next time step
c
600   go to (610,630,650,670,690,710,730,750,770,790) l
610   do 620 i=1,N
      dyl(i)=da(1,i)
      uyl(i)=ua(1,i)
      vyl(i)=va(1,i)
      eyl(i)=ea(1,i)
      dyr(i)=da(N,i)
      uyr(i)=ua(N,i)
      vyr(i)=va(N,i)
      eyr(i)=ea(N,i)
620   continue
      go to 900
630   do 640 i=1,N
      dxl(i)=da(i,1)
```

```
      uxl(i)=ua(i,1)
      vxl(i)=va(i,1)
      exl(i)=ea(i,1)
      dzl(i)=da(1,i)
      uzl(i)=ua(1,i)
      vzl(i)=va(1,i)
      ezl(i)=ea(1,i)
640   continue
      go to 900
650   do 660 i=1,N
      dxr(i)=da(i,1)
      uxr(i)=ua(i,1)
      vxr(i)=va(i,1)
      exr(i)=ea(i,1)
      dzr(i)=da(N,i)
      uzr(i)=ua(N,i)
      vzr(i)=va(N,i)
      ezr(i)=ea(N,i)
660   continue
      go to 900
670   do 680 i=1,N
      dtl(i)=da(1,i)
      utl(i)=ua(1,i)
      vtl(i)=va(1,i)
      etl(i)=ea(1,i)
      dsl(i)=da(i,1)
      usl(i)=ua(i,1)
      vsl(i)=va(i,1)
      esl(i)=ea(i,1)
680   continue
      go to 900
690   do 700 i=1,N
      dtr(i)=da(N,i)
      utr(i)=ua(N,i)
      vtr(i)=va(N,i)
      etr(i)=ea(N,i)
      dsr(i)=da(i,1)
      usr(i)=ua(i,1)
      vsr(i)=va(i,1)
      esr(i)=ea(i,1)
700   continue
      go to 900
710   do 720 i=1,N
      dxc(i)=da(i,N)
      uxc(i)=ua(i,N)
      vxc(i)=va(i,N)
      exc(i)=ea(i,N)
720   continue
      go to 900
730   do 740 i=1,N
      dwl(i)=da(i,1)
      uwl(i)=ua(i,1)
      vwl(i)=va(i,1)
      ewl(i)=ea(i,1)
      dql(i)=da(1,i)
      uql(i)=ua(1,i)
      vql(i)=va(1,i)
      eql(i)=ea(1,i)
740   continue
      go to 900
750   do 760 i=1,N
      dwr(i)=da(i,1)
      uwr(i)=ua(i,1)
      vwr(i)=va(i,1)
```

```
      ewr(i)=ea(i,1)
      dqr(i)=da(N,i)
      uqr(i)=ua(N,i)
      vqr(i)=va(N,i)
      eqr(i)=ea(N,i)
760   continue
      go to 900
770   do 780 i=1,N
      dpl(i)=da(1,i)
      upl(i)=ua(1,i)
      vpl(i)=va(1,i)
      epl(i)=ea(1,i)
      drl(i)=da(i,1)
      url(i)=ua(i,1)
      vrl(i)=va(i,1)
      erl(i)=ea(i,1)
780   continue
      go to 900
790   do 800 i=1,N
      dpr(i)=da(N,i)
      upr(i)=ua(N,i)
      vpr(i)=va(N,i)
      epr(i)=ea(N,i)
      drr(i)=da(i,1)
      urr(i)=ua(i,1)
      vrr(i)=va(i,1)
      err(i)=ea(i,1)
800   continue
c
c     set da, ua, va, and ea values at this time step equal to
c     df, uf, vf, and ef values for the next time step.
c
900   do 910 i=1,N
      do 910 j=1,N
      df(1,i,j)=da(i,j)
      uf(1,i,j)=ua(i,j)
      vf(1,i,j)=va(i,j)
      ef(1,i,j)=ea(i,j)
910   continue
c
      fixmax=0.0
      do 920 i=1,N
      do 920 j=1,N
      usquar=ua(i,j)*ua(i,j)
      vsquar=va(i,j)*va(i,j)
      vel(i,j)=sqrt(usquar+vsquar)
      vpres=da(i,j)*(usquar+vsquar)/2.0
      pres(i,j)=(gamma-1.0)*(ea(i,j)-vpres)
      ss=gamma*pres(i,j)/da(i,j)
      if (ss.gt.0.0) go to 930
      ss=0.0
930   ss=sqrt(ss)+vel(i,j)
      if (ss.le.fixmax) go to 920
      fixmax=ss
920   continue
      if (fixmax.le.afix) go to 940
      afix=fixmax
c
c     perform the plate analysis for time frame m.
c     do not perform this analysis until all regions
c     have been evaluated for time frame m.
c
940   go to (950,950,950,990,990,990,950,950,990,970) l
950   do 960 i=1,N
```

```
      WP(k)=pres(i,N)
      k=k+1
960   continue
      go to 990
970   WP(24)=(WP(23)+WP(25))/2.0
      WP(46)=(WP(45)+WP(47))/2.0
      WP(68)=(WP(67)+WP(69))/2.0
      WP(90)=(WP(89)+WP(91))/2.0
c
      write(6,971)
971   format('WORKING ON THE PLATE/MEMBRANE EQUATIONS')
666   factr3=Gc*deldtp*deldtp/rhoweb
      factr5=factr4*deldtm*deldtm
      if (m.gt.801) then
      do 985 i=1,Nx
      WP(i)=1.0
985   continue
      endif
c
c     LINEAR MEMBRANE EQUATIONS -- REMOVE COMMENTS TO USE
c
c      do 995 k=1,10
c      do 980 i=3,Nx-2
c      WP1(1,i)=-WM1(1,i)+2.0*W(1,i)+factr5*(W(1,i+1)+
c     1W(1,i-1)+2.0*W(2,i)-4.0*W(1,i))/(delx*delx)+factr5*
c     2(abs(WP(i)-1.0)*patm-rhoweb)/Tens
c980   continue
c
c      do 1025 j=2,Ny-2
c      factr0=exp(-float(j)*float(j)/800.0)
c      do 1025 i=3,Nx-2
c      WP1(j,i)=-WM1(j,i)+2.0*W(j,i)+factr5*(W(j,i+1)+
c     1W(j,i-1)+W(j+1,i)+W(j-1,i)-4.0*W(j,i))/(delx*delx)+
c     2factr5*(abs(WP(i)-1.0)*patm*factr0-rhoweb)/Tens
c1025  continue
c
c     LINEAR PLATE EQUATIONS
c
      do 995 k=1,10
      do 980 i=3,Nx-2
      WP1(1,i)=-WM1(1,i)-factr3*(W(1,i)*(20.0*factr1+2.0*
     1factr2-2.0/factr3)-(W(1,i+1)+W(1,i-1))*(8.0*factr1+
     2factr2)-W(2,i)*16.0*factr1+(W(2,i+1)+W(2,i-1))*4.0*
     3factr1+(W(1,i+2)+W(1,i-2)+2.0*W(3,i))*factr1-
     4(WP(i)-1.0)*patm+rhoweb)
980   continue
c
      do 1010 i=3,Nx-2
      WP1(2,i)=-WM1(2,i)-factr3*(W(2,i)*(20.0*factr1+2.0*
     1factr2-2.0/factr3)-(W(2,i+1)+W(2,i-1))*(8.0*factr1+
     2factr2)-(W(1,i)+W(3,i))*8.0*factr1+(W(3,i+1)+W(3,i-1)
     3+W(1,i+1)+W(1,i-1))*2.0*factr1+(W(2,i+2)+W(2,i-2)+
     4W(2,i)+W(4,i))*factr1-(WP(i)-1.0)*patm+rhoweb)
1010  continue
c
      do 1300 j=3,Ny-2
      factr0=exp(-float(j)*float(j)/800.0)
      do 1300 i=3,Nx-2
      WP1(j,i)=-WM1(j,i)-factr3*(W(j,i)*(20.0*factr1+2.0*
     1factr2-2.0/factr3)-(W(j,i+1)+W(j,i-1))*(8.0*factr1+
     2factr2)-(W(j+1,i)+W(j-1,i))*8.0*factr1+(W(j+1,i+1)+
     3W(j+1,i-1)+W(j-1,i+1)+W(j-1,i-1))*2.0*factr1+(W(j,i+2)
     4+W(j,i-2)+W(j+2,i)+W(j-2,i))*factr1-(WP(i)-1.0)*
     5patm*factr0+rhoweb)
```

```
1300  continue
c
c     Update working vectors before next iteration.
c
      do 1835 j=1,Ny-2
      do 1835 i=3,Nx-2
      WM1(j,i)=W(j,i)
      W(j,i)=WP1(j,i)
1835  continue
995   continue
c
c     Check for interval for writing deflections to output file.
c
990   if (m.eq.intc1) go to 1100
      if (m.eq.intc2) go to 1100
      if (m.eq.intc3) go to 1100
      if (m.eq.intc4) go to 1100
      if (m.eq.intc5) go to 1100
      if (m.eq.intc6) go to 1100
      if (m.eq.intc7) go to 1100
      if (m.eq.intc8) go to 1100
      if (m.eq.intc9) go to 1100
      if (m.eq.intc10) go to 1100
      if (m.eq.intc11) go to 1100
      if (m.eq.intc12) go to 1100
      go to 1800
c
1100  if (l.ne.10) go to 1800
      write(6,1750)
1750  format('Enter membrane or plate deflection file name')
      read(5,1760)DFILE
1760  format(A12)
      open(8, file=DFILE, status='unknown')
      do 1770 j=1,Ny-2
      do 1770 i=3,Nx-2,4
      write(8,1780)i,WP1(j,i),i+1,WP1(j,i+1),i+2,WP1(j,i+2),
     1i+3,WP1(j,i+3)
1780  format(4(i3,1x,f16.9,1x))
1770  continue
      close(8)
c
1800  continue
      if (m.eq.icmax) go to 1860
1850  wplusc=afix
      m=m+1
      write(6,19)m
19    format(' ICOUNT VALUE INCREMENTED TO ',i5)
      go to 300
1860  continue
      close(9)
      stop
      end
c
c
      subroutine field(da,ua,va,ea,d1,u1,v1,e1,d2,u2,v2,e2,
     1d3,u3,v3,e3,d4,u4,v4,e4,d5,u5,v5,e5,ind)
      common /dat/gamma,omega,xKn,l,N
c
      t1=u1*u1+v1*v1
      t2=u2*u2+v2*v2
      t3=u3*u3+v3*v3
      t4=u4*u4+v4*v4
      t5=u5*u5+v5*v5
      p1=(gamma-1.0)*(e1-d1*t1/2.0)
```

```
      p2=(gamma-1.0)*(e2-d2*t2/2.0)
      p3=(gamma-1.0)*(e3-d3*t3/2.0)
      p4=(gamma-1.0)*(e4-d4*t4/2.0)
      p5=(gamma-1.0)*(e5-d5*t5/2.0)
      if (p1.gt.0.0) go to 2000
      t1=sqrt(t1)
      go to 2010
2000  t1=sqrt(gamma*p1/d1+t1)
2010  if (p2.gt.0.0) go to 2020
      t2=sqrt(t2)
      go to 2030
2020  t2=sqrt(gamma*p2/d2+t2)
2030  if (p3.gt.0.0) go to 2040
      t3=sqrt(t3)
      go to 2050
2040  t3=sqrt(gamma*p3/d3+t3)
2050  if (p4.gt.0.0) go to 2060
      t4=sqrt(t4)
      go to 2070
2060  t4=sqrt(gamma*p4/d4+t4)
2070  if (p5.gt.0.0) go to 2080
      t5=sqrt(t5)
      go to 2090
2080  t5=sqrt(gamma*p5/d5+t5)
2090  continue
      if (ind.ne.1) go to 2100
      r=2.0
      a=1.0
      b=1.0
      c=1.0
      d=1.0
      e=1.0
      f=1.0
      g=1.0
      h=1.0
      go to 2130
2100  if (ind.ne.2) go to 2110
      r=4.0
      a=1.0
      b=0.0
      c=1.0
      d=0.0
      e=1.0
      f=0.0
      g=1.0
      h=2.0
      go to 2130
2110  if (ind.ne.3) go to 2120
      r=4.0
      a=0.0
      b=1.0
      c=0.0
      d=1.0
      e=0.0
      f=1.0
      g=2.0
      h=1.0
      go to 2130
2120  if (ind.ne.4) go to 2130
      r=4.0
      a=0.0
      b=1.0
      c=0.0
      d=1.0
```

```
      e=2.0
      f=1.0
      g=0.0
      h=1.0
2130  sum=1.0-omega*xKn*(t1/r+(a*t2+b*t3+c*t4+d*t5)/8.0)
c
      da=d1*sum+omega*xKn*(a*t2*d2+b*t3*d3+c*t4*d4+d*t5*d5 +
     1(a*d2+b*d3+c*d4+d*d5)*t1)/8.0 - xKn*(-e*d2*u2+f*d3*v3+
     2g*d4*u4-h*d5*v5)/2.82842
c
      ea=e1*sum+omega*xKn*(a*t2*e2+b*t3*e3+c*t4*e4+d*t5*e5 +
     1(a*e2+b*e3+c*e4+d*e5)*t1)/8.0 - xKn*(-e*(e2+p2)*u2+
     2f*(e3+p3)*v3+g*(e4+p4)*u4-h*(e5+p5)*v5)/2.82842
c
      if (ind.eq.3) then
      ua=0.0
      g=0.0
      go to 2160
      else if (ind.eq.4) then
      ua=0.0
      e=0.0
      go to 2160
      else
c
      ua=(d1*u1*sum+omega*xKn*(a*t2*d2*u2+b*t3*d3*u3+c*t4*d4*u4+
     1d*t5*d5*u5 + (a*d2*u2+b*d3*u3+c*d4*u4+d*d5*u5)*t1)/8.0 -
     2xKn*(-e*(p2+d2*u2*u2)+f*d3*u3*v3+g*(p4+d4*u4*u4)-
     3h*d5*u5*v5)/2.82842)/da
      endif
c
2160  if (ind.eq.2) then
      va=0.0
      h=0.0
      go to 2180
      else
c
      va=(d1*v1*sum+omega*xKn*(a*t2*d2*v2+b*t3*d3*v3+c*t4*d4*v4+
     1d*t5*d5*v5 + (a*d2*v2+b*d3*v3+c*d4*v4+d*d5*v5)*t1)/8.0 -
     2xKn*(-e*d2*u2*v2+f*(p3+d3*v3*v3)+g*d4*u4*v4-h*(p5+
     3d5*v5*v5))/2.8284)/da
      endif
c
2180  continue
      return
      end
```

APPENDIX I

PHOTOGRAPHS OF EXPERIMENTAL TENSION MEASUREMENT SYSTEM

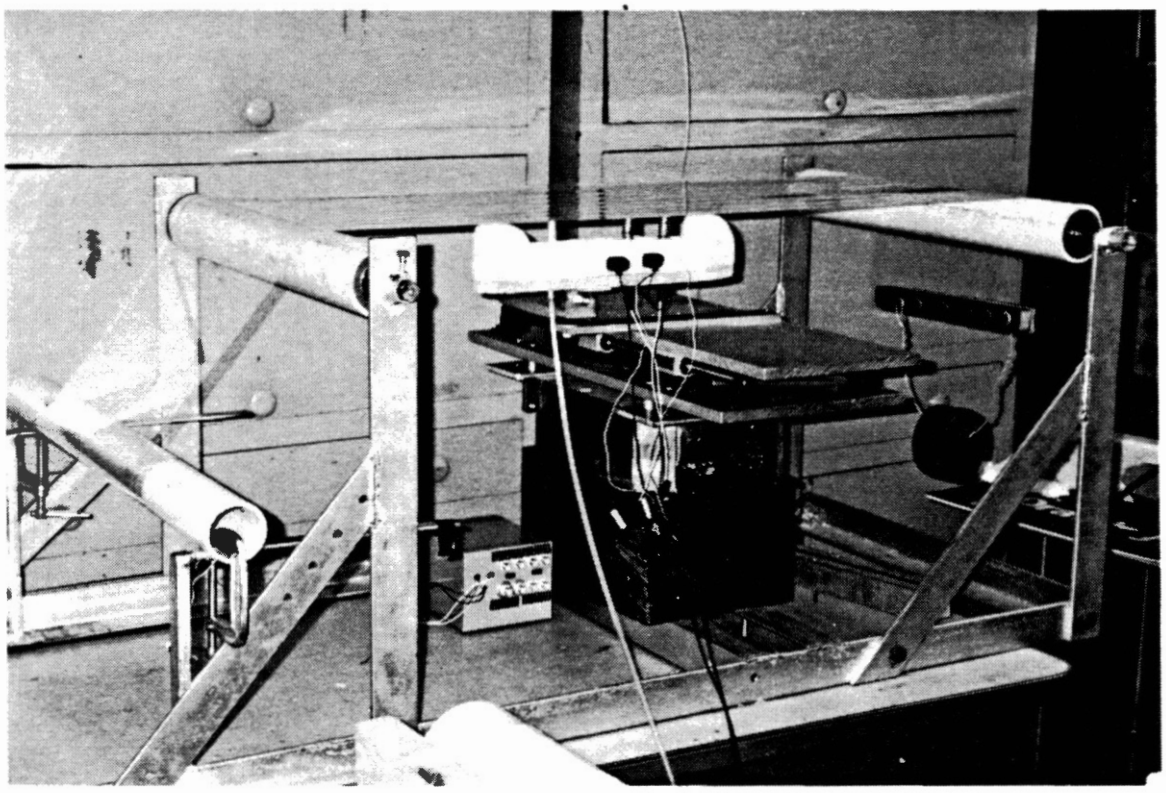


Figure I.1. Static Test Frame and Ballscrew Platform for Laboratory Tests

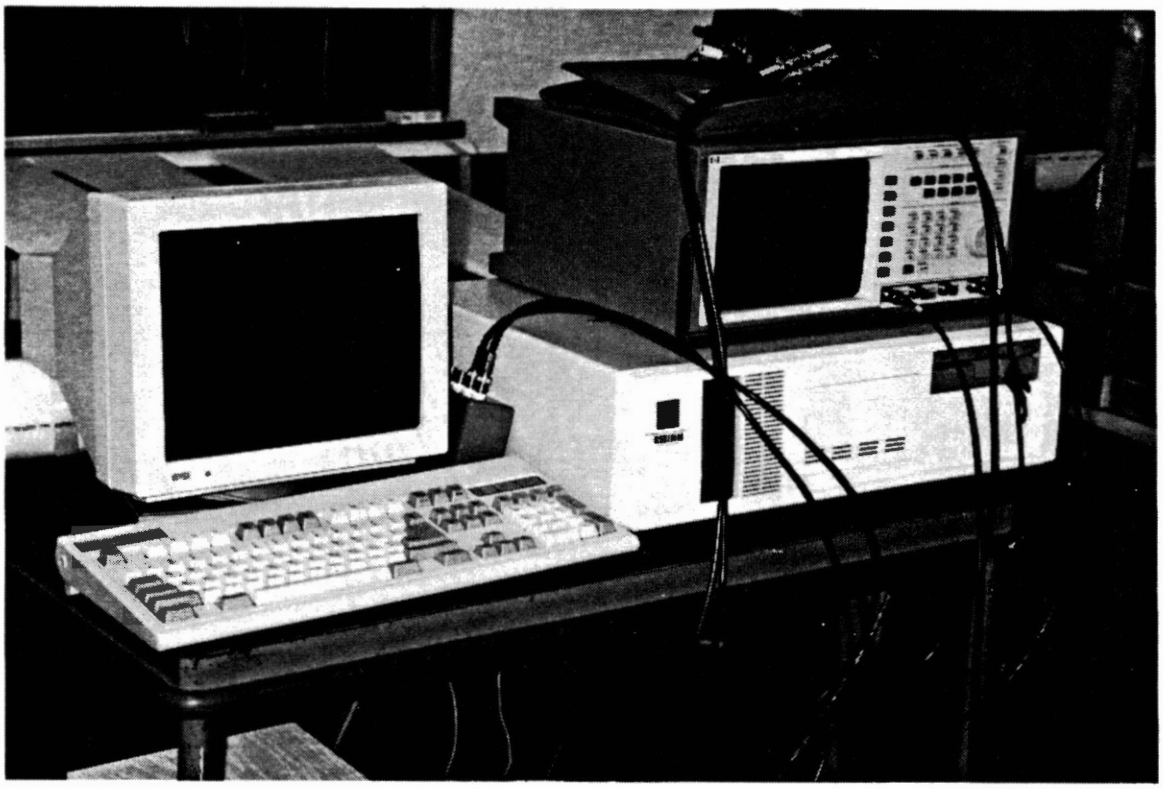


Figure I.2. IBM Compatible Computer and HP54501A Digital Oscilloscope Facility

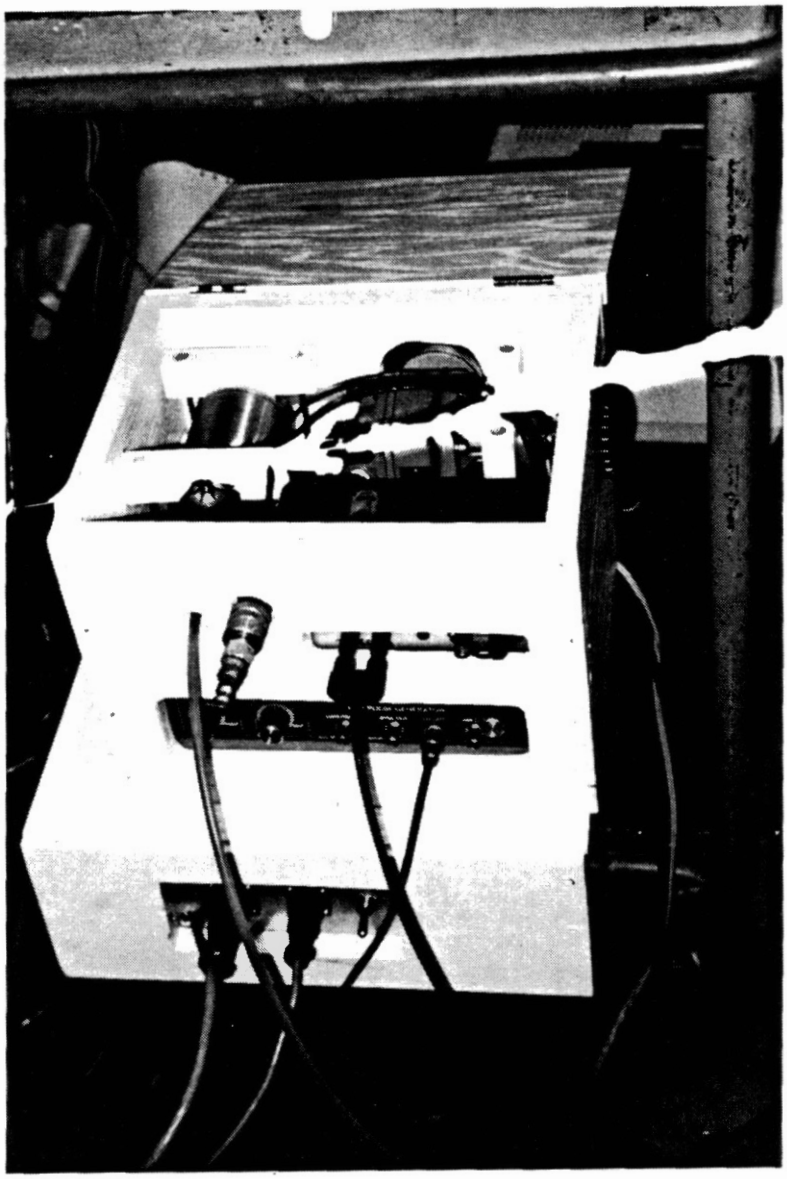


Figure I.3. Solenoid Valve Pulser (Top) and Stepper Motor Controller (Bottom)

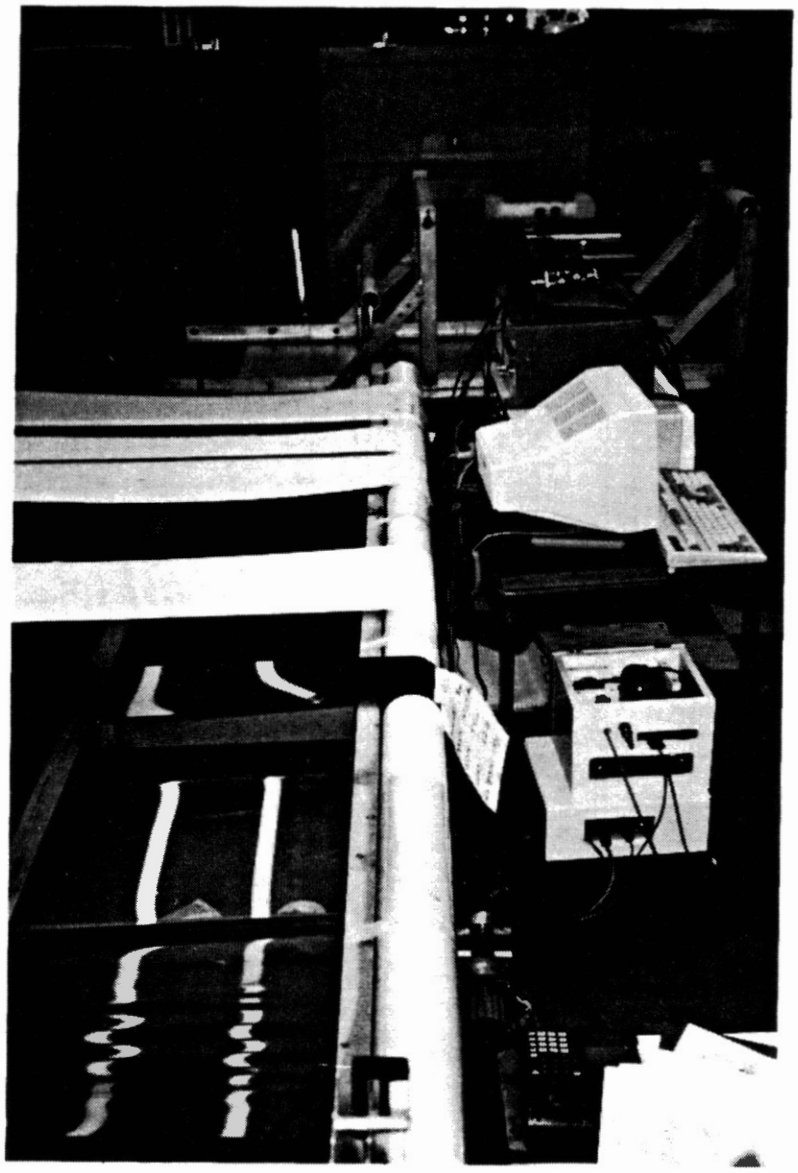


Figure I.4. Long Static Frame Apparatus for Testing of Traverse and Multiple Web Samples

VITA 2

Curtis M. Vickery

Candidate for the Degree of

Doctor of Philosophy

Thesis: USE OF PNEUMATIC PULSE STIMULUS FOR INDUCEMENT OF LARGE AMPLITUDE FLEXURAL WAVEFORMS IN WEB MATERIALS FOR THE PURPOSE OF LOCAL TENSION MEASUREMENT

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Blackwell, Oklahoma, September 27, 1958, the son of Dr. and Mrs. Rollin Vickery.

Education: Graduated from Braman High School, Braman, Oklahoma, in May, 1976; received the Bachelor of Science degree in Mechanical Engineering from Oklahoma State University in May, 1981; received the Master of Science degree in Electrical Engineering from Oklahoma State University in May, 1983; completed requirements for the Doctor of Philosophy degree from Oklahoma State University in May, 1992.

Professional Experience: Lecturer, School of Mechanical and Aerospace Engineering, Oklahoma State University; conducted lecture/laboratory courses in Instrumentation, Dynamic Systems, and Vibrations, June, 1990, to present. Laboratory Manager, School of Mechanical and Aerospace Engineering, Oklahoma State University; responsible for purchase of materials and supervision of student laboratory technicians, June, 1987, to June, 1990. Technical Staff Member, Sandia National Laboratories, Albuquerque, New Mexico, Division 7265QA Systems Test Equipment Design; performed electrical and electronic design for development and maintenance of automated test systems, March, 1983, to May, 1987. Research/Teaching Assistant, School of Mechanical and Aerospace Engineering, Oklahoma State University; developed instrumentation to assist in CDC floppy disk drive read/write head vibrational analysis, August, 1981, to March, 1983.