ELASTIC-PLASTIC HULL PLATE RESPONSE TO
SLAMMING INDUCED PRESSURES

by

DOUGLAS J. HENKE

B.S. Naval Architecture and Marine Engineering
U.S. Coast Guard Academy (1985)

Submitted to the Department of
OCEAN ENGINEERING
In Partial Fulfillment of the Requirements For
the Degrees of

MASTER OF SCIENCE IN NAVAL ARCHITECTURE AND MARINE ENGINEERING

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUNE 1992


The Author hereby grants to MIT permission to reproduce and
to distribute copies of this thesis document in whole or in
part.

Signature of Author __________________________
Department of Ocean Engineering
May, 1992

Certified by __________________________
Tomasz Wierzbicki
Professor of Applied Mechanics
Thesis Supervisor

Certified by __________________________
Frank A. McClintock
Professor, Mechanical Engineering
Thesis Reader

Accepted by __________________________
A. Douglas Carmichael, Chairman
Departmental Graduate Committee
Department of Ocean Engineering
MAY 07, 1992
ELASTIC-PLASTIC HULL PLATE RESPONSE TO SLAMMING INDUCED PRESSURES

by

DOUGLAS J. HENKE

Submitted to the Department of Ocean Engineering and Mechanical Engineering in partial fulfillment of the requirements for the Degrees of Master of Science in Naval Architecture and Marine Engineering and Mechanical Engineering

ABSTRACT

The effect of slamming on ships' hulls in the ocean has, for many years, been the subject of much scrutiny. Many approximate methods have been devised for determining the expected pressure pulses and evaluating the response of hull plating. Optimizing hull plate design based on a range of failure criterion from first yield to ultimate failure requires insight into hull plate responses under a wide variety of wave induced loads.

There is a considerable range of slamming pulses on hull plating which cause either elastic response or elastic-plastic response where the plastic deformation is of the same order as the elastic deformation. Failure criteria for plate design can likely be from pulses causing this range of deformation.

This paper gives a detailed analysis of the elastic response of hull plating to slamming pulses. A finite difference model of a plate strip is developed which accounts for membrane effects. The analysis shows detailed stress and deflection response to various pulses. The resulting design plots give the combinations of typical pulses of 30 to 200 psi peak pressures and decay times of .001 to .5 sec. leading to incipient yield at 40000 and 60000 psi in various plate sizes.

The plate strip concept is then extended to the elastic-plastic range. The model is made of two rigid sections with deformable hinges at the center and ends. These hinges are comprised of layers of bar elements with elastic-plastic characteristics. Plots are given for permanent deflections from pulses of 2-6 times those which cause incipient yield. Unexpected resonances of 5%-30% occur in the elastic-plastic range, not noted in the elastic range.
ACKNOWLEDGEMENTS

I'd like to thank Professor Tomasz Wierzbicki for his encouragement, interest and assistance.

Professor Frank McClintock generously offered his time and expertise for refining this work.

Most of all, to my wife Kelly. Her unselfish support, strength, and patience throughout the entire course of study here at MIT made it possible for me to accomplish this thesis.
# TABLE OF CONTENTS

ABSTRACT ........................................................................................................... 2

ACKNOWLEDGEMENTS ....................................................................................... 3

NOTATION ............................................................................................................ 7

DIMENSIONLESS PARAMETERS ........................................................................... 8

1. INTRODUCTION .................................................................................................. 10

2. PLATE LOADING .................................................................................................. 12
   2.1. SLAMMING LOADS .......................................................................................... 12
   2.2. SLAMMING LOAD APPROXIMATIONS ............................................................... 12
       2.2.1. Load Prediction ............................................................................................ 12
       2.2.2. Time Domain Slamming Pulse Representation .............................................. 13
       2.2.3. Space Domain Slamming Pulse Representation ............................................... 13

3. PLATE RESPONSE ............................................................................................... 15
   3.1. PLATE SUPPORT AND BOUNDARIES ............................................................... 15
   3.2. GLOBAL AND LOCAL RESPONSES .................................................................. 15
   3.3. STRUCTURAL COMPLEXITIES WITH SLAMMING ........................................... 16
       3.3.1. Boundary Rotation ....................................................................................... 16
       3.3.2. Boundary Lateral Deflection ....................................................................... 16
       3.3.3. In Plane Deflection ...................................................................................... 17
       3.3.4. Initial Plate Curvature .................................................................................. 17
   3.4. RESPONSE ANALYSIS CONSIDERATIONS ..................................................... 17
       3.4.1. Necessity For Elastic-Plastic Analysis .......................................................... 17
       3.4.2. Inertia Effects ............................................................................................... 18

4. PLATE MODELLING ............................................................................................. 19
   4.1. MAXIMUM STRESS IN A CLAMPED PLATE ....................................................... 19
   4.2. PLATE STRIP .................................................................................................... 19
   4.3. PLATE STRIP BOUNDARY CONDITIONS .......................................................... 20

5. FORMULATION OF GOVERNING EQUATIONS OF MOTION ............................. 20
   5.1. PLATE STRIP DYNAMICS IN ELASTIC RANGE ................................................. 20
   5.2. NON-DIMENSIONAL FORM OF EQUATION OF MOTION ............................... 24

6. FINITE DIFFERENCE MODEL ............................................................................. 27
   6.1. FINITE DIFFERENCE MODEL OF PLATE STRIP .............................................. 27
       6.1.1. Finite Difference Equations .......................................................................... 27
       6.1.2. Finite Difference Boundary Conditions ......................................................... 29
   6.2. MODEL ACCURACY VALIDATION ..................................................................... 30
       6.2.1. Galerkin Solution .......................................................................................... 30
       6.2.2. Wave Speed and Dispersion Relationship ....................................................... 33
   6.3. USE OF FINITE DIFFERENCE MODEL .............................................................. 36
       6.3.1. Boundary Conditions .................................................................................... 36
       6.3.2. Convergence of Results ................................................................................. 37
   6.4. MOVING PULSE FRONT .................................................................................... 37
       6.4.1. Effect of Pulse Front Velocity ...................................................................... 37
7. RESULTS OF ELASTIC ANALYSIS .................................. 39
  7.1. INCIPIENT YIELD PULSE .................................. 39
  7.2. SCOPE OF APPLICATION OF DIMENSIONLESS RESULT .... 40
  7.3. FIRST YIELD PULSE CHART ................................ 41
  7.4. USE OF YIELD PULSE CHARTS ............................ 42

8. ELASTIC-PLASTIC RESPONSE ..................................... 43
  8.1. NECESSITY OF ELASTIC-PLASTIC ANALYSIS ................. 43
  8.2. ELASTIC-PLASTIC (E-P) MODEL ............................ 44
    8.2.1. Yankelevsky's Elastic-Plastic Beam Model .......... 44
    8.2.2. Modifications to Yankelevsky's Model .. 44
    8.2.3. Summary of Model Development ....................... 45
    8.2.4. End and Midstrip Stresses Based on Elastic Analysis 46
    8.2.5. Strain in Hinge Elements .......................... 49
  8.3. DYNAMICS OF THE ELASTIC-PLASTIC MODEL ................. 54
    8.3.1. Equation of Motion ................................ 54
    8.3.2. Tracking Stress In Each Element .................... 56
    8.3.3. E-P Model Response Algorithm ....................... 57
  8.4. E-P MODEL VALIDATION .................................... 59
    8.4.1. Static Response Evaluation ......................... 59
    8.4.2. Time Step and Number of Elements for Convergence 59
    8.4.3. Comparison to Finite Difference Model in Elastic Range 60
    8.4.4. Analysis of Sample Elastic-Plastic Responses ... 61
  8.5. EFFECT OF ADDED MASS ON RESPONSE ....................... 64
  8.6. FURTHER VALIDATION OF THE NEED FOR ELASTIC-PLASTIC VS. PLASTIC ANALYSIS ............................. 66
  8.7. RUNNING THE E-P MODEL PROGRAM FOR RESULTS .......... 67

9. ANALYSIS OF E-P MODEL RESULTS ................................ 70
  9.1. PEAK PERMANENT DEFLECTIONS ................................ 70
  9.2. COUNTER-INTUITIVE RESULTS ................................ 72
  9.3. BUCKLING CONSIDERATIONS ............................... 73

10. CONCLUSIONS ............................................. 74
  10.1. NEED FOR MODEL VALIDATION ............................ 74
  10.2. EVALUATION ........................................... 74

11. REFERENCES ............................................. 76

LIST OF FIGURES ............................................... 79

Appendix A. Sample of Recorded Slamming Pulse Pressure vs. Time .......... 90
Appendix B. Computer Program NONDIM.C. Response to Spatially Constant-Time Varying Loads .......... 92
Appendix C. Computer Program NONDIM.FULL.C. Response to Space and Time varying Loads .......... 105
Appendix D. Frequency Responses of Galerkin Solution and Finite Difference Model .......... 118
Appendix E. Computer Program STATIC.C. Static Response to Uniform Loads .......... 121
Appendix F. Convergence Data for Finite Difference Solution ........................................ 127
Appendix G. Moving Pulse vs. Spatially Constant Pulse End Stress Response ..................... 131
Appendix H. Moving Pulse vs. Spatially Constant Pulse 3-D Plot of Node Deflection ........... 137
Appendix I. Yield Pulse Charts .................. 142
Appendix J. Comparison of Dynamic and Statically Loaded Plate Strip Shapes .................... 147
Appendix K. Stress vs. Center Deflection in Plate Strip under Uniform Static Load ............ 151
Appendix L. Characteristic Length Fraction vs. Dimensionless Deflection ......................... 154
Appendix M. Computer Program PLASTIC.C. Elastic-Plastic Response ............................. 157
Appendix N. Static Response of E-P and Finite Difference Model to Uniform Loads .......... 173
Appendix O. E-P Model Dynamic Response with Various Numbers of Hinge Elements ........... 175
Appendix P. Finite Difference and E-P Model response to Yield Pulses ............................. 177
Appendix Q. E-P Model Response Data to Multiples of Sample Yield Pulse ......................... 190
Appendix R. Illustration of Added Mass Effect on E-P Model Response ............................... 202
Appendix S. Rigid-Plastic vs. Elastic-Plastic Response to Applied Pulse ............................ 204
Appendix T. Elastic-Plastic Analysis Results .................. 206
Appendix U. E-P Model Response Data to Peak Pulses .................................................. 228
Appendix V. Frequency Content of Applied Pulse ........................................ 230
NOTATION

\( \alpha \) Angular acceleration of E-P model half strip.
\( \beta \) Stiffness coefficient \( E/12(1-v^2) \)
\( c \) Wave velocity
CLF Characteristic length fraction
\( c_p \) Pulse front velocity
\( D \) Flexural rigidity \( (Eh^3/12(1-v^2)) \)
\( E \) Modulus of elasticity
\( \varepsilon \) Strain \( (\Delta L/L) \)
E-P Elastic-Plastic
\( \varepsilon_i \) Element strain in E-P model hinge.
\( \varepsilon_0 \) Plate strip membrane strain
EP Plastic tangent modulus
\( F_{b,\text{end}} \) CLF for clamped end bending stress
\( F_{b,\text{mid}} \) CLF for clamped end bending stress
\( F_m \) CLF for membrane stress
\( h \) Plate thickness
\( h_i \) Element thickness in E-P model
\( J \) Rotational mass moment of inertia
\( K \) Kurvature \( (\partial^2 w/\partial x^2) \)
\( K \) Wave number (Kappa)
\( k \) Stiffness of elastic supports.
\( L \) Plate strip length
\( \lambda \) Wave length
\( M \) Moment at any point \( x \).
\( v \) Poisson's ratio (assumed .3)
\( P \) Applied load.
\( P_0 \) Initial magnitude of pulse load at time = \( t_0 \).
\( P_0 \) Amplitude of load in Galerkin solution
\( R \) Ratio of plate strip length/thickness
\( \rho \) Density of plate material
\( R_G \) Residual in Galerkin solution
\( S \) Stretching force in plate strip.
\( \sigma \) Stress
\( \sigma_b \) Bending stress at the plate surface
\( \sigma_i \) Stress in the 'i'th element in an E-P model hinge.

\( \sigma_{\text{max}} \) Maximum stress through the thickness

\( \sigma_0 \) Membrane stress

\( \sigma_Y \) Incipient yield strength

\( t \) Time

\( \tau_{\text{as}} \) Space domain pulse decay constant

\( \tau_{\text{at}} \) Time domain pulse decay constant

\( t_0 \) Time of instant rise in pulse to maximum pressure.

\( u_i \) Displacement of the 'i'th element in E-P hinge.

\( V \) Shear force.

\( W \) Center deflection

\( W_f \) Permanent center deflection

\( w \) Lateral deflection

\( \omega_n \) Natural frequency

\( w_s \) Shape function for the Galerkin approximate solution.

\( x \) Distance and direction along plate strip from clamped end.

\( y \) Direction in plane and orthogonal to \( x \).

\( z \) Distance and direction perpendicular to plate middle surface.

\( z_i \) Distance and direction from E-P model hinge middle surface to the center of the 'i'th element.

**DIMENSIONLESS PARAMETERS**

\[ \tilde{w} = \frac{w}{h} \]

\[ \tilde{\omega} = \omega \frac{L^2 \sqrt{\rho h}}{D} \]

\[ \tilde{x} = \frac{x}{L} \]
\[
\tilde{P}(x,t) = \frac{P(x,t)}{\frac{D h}{L^4}} = \frac{P(x,t) \cdot R^4}{\beta}
\]

Load

\[
\tilde{\sigma} = \frac{\sigma}{\frac{D}{h L^2}} = \frac{\sigma \cdot R^2}{\beta}
\]

Stress

\[
\tilde{S} = \frac{S}{\frac{D}{L^2}}
\]

Stretching Force

\[
\tilde{\tau}_s = \frac{\tau_s}{L}
\]

Spacial Decay Constant

\[
\tilde{t} = \frac{\frac{t}{L^2 \sqrt{\frac{p}{h}}}}{R L \sqrt{\frac{p}{\beta}}}
\]

Time

\[
\tilde{\lambda} = \frac{\lambda}{L}
\]

Wave Length

\[
\tilde{c} = c R \sqrt{\frac{p}{\beta}}
\]

Wave Velocity
1. INTRODUCTION

There has been many analyses of hull structural response to slamming loads [1 thru 8]. Typically, either a hull section of stiffened panels or each plate bounded by the stiffeners is considered for analysis. The response of the single plate is considered here.

It turns out that plates can sustain loads well above that which cause incipient yield. Factors of these incipient yield loads can be used as design criterion [7]. It is therefore necessary to determine what types of loads cause incipient yield. Since large deflections occur, the effect of membrane forces must be taken into account to more accurately determine elastic response. This applies especially to thin plates where stretching plays a more significant role. It appears no closed form solutions are available for nonlinear elastic response to slamming pulse loading. Therefore, numerical analysis is a valid approach to determine the detailed behavior of a plate in response to slamming loads.

When large permanent deflections are expected, rigid-plastic behavior of material is often assumed. There are some closed form solutions for plate deflections with rigid-plastic behavior which have application to plate slamming [6,7,8,18]. This makes rigid-plastic analysis appealing though it is applicable only when plastic deformation is much larger than elastic deformation. However, failure criteria
for hull plating can be based on permanent deflections where both elastic and plastic deflections are of the same order. To extend the analyses and results of references [3 thru 8] in determining hull plate response, elastic-plastic analysis is necessary. There are no direct analytical solutions for elastic-plastic response to slamming induced loads. Therefore, numerical methods can be employed to evaluate elastic-plastic response.

The analysis performed here considers plate response to a wide range of slamming pulses for two cases: purely elastic response; where the effect of stretching is taken into account, and elastic-plastic response; where the amount of plasticity is of the same order as elasticity. In both cases, the plate is modelled as a plate strip. Finite difference methods are used to determine elastic response of plates. The results of the elastic analysis give the types of slamming pulses which cause incipient yield for various plate sizes. Elastic-plastic analysis is performed with a simplified model which has two rigid sections with a middle and two end hinges which behave as if elastic-plastic. The results of this analysis give the magnitude of permanent deflections to multiples of the pulses which cause incipient yield based on elastic analysis.

This study gives insight into the elastic behavior of plates under pulse loading and bridges the gap between rigid-plastic and elastic behavior of plates subjected to slamming pulses.
2. PLATE LOADING

2.1. SLAMMING LOADS

Wave impact loads on the hull plating of a ship vary from relatively low magnitude and long duration to high magnitude and short duration. [1 thru 7] The high magnitude loads result from slamming which occurs typically to the forward sections of hull plating. This may be due to a wave slamming onto the hull or to sudden submergence of the hull through the ocean surface in heavy weather or at high speeds. Slamming-induced loads can have a severe impact on the structural integrity of hulls. The effects of loads on hull plating must be understood in order to design competently for them.

2.2. SLAMMING LOAD APPROXIMATIONS.

2.2.1. Load Prediction.

Accurately predicting wave and slamming induced loads on hull plating is extremely difficult due to the complexities of non-linear random fluid-solid interactions between the hull and the ocean. However, one method used in determining wave loading on ships in an ocean environment has been statistical analysis [1,2,8] where probabilities of slamming occurrences and pulse magnitudes and durations are assessed. Recorded data has shown there is a wide variety of pulse magnitudes and durations occur [1,2,3,6,8]. In order to determine accurately the response of a plate to these
slamming loads, a detailed description of the pulse is necessary.

2.2.2. Time Domain Slamming Pulse Representation

Recorded data of slamming pressure at a point on the plate show a very sharp initial rise and a gradual exponential decay [3,6]. Typically, the long side of the plate is oriented longitudinally with the ship. It is assumed this pressure is constant in the longitudinal direction of the plate. A sample of slamming pressure vs.time at two points with different centerline offsets [6] is shown in Appendix (A). This form of pulse can be closely approximated by an exponential decay function in terms of time $t > t_0$, a time constant $\tau_t$, and a peak pressure $P_0$:

$$P(t) = P_0 \cdot \exp\left(\frac{(t-t_0)}{\tau_t}\right)$$

(1)

where $P(t) = 0$ for $t < t_0$. The variable $t$ denotes time, $\tau_t$ is the characteristic time decay constant and $P_0$ is the peak pressure at time $= t_0$. The time decay constant is the time it takes the pressure to reduce to $P_0/e$ (by 63.2%).

As a ship slams into the ocean the pressure pulse propagates outboard. As it propagates, its magnitude reduces due to the decreasing downward velocity of the ship. [6]. Therefore, the peak pressure $P_0$ is a function of $x$, a measure of outboard distance along the width of the plate.

2.2.3. Space Domain Slamming Pulse Representation
In order to establish the space representation of the pulse, the travelling pressure front velocity must be known. While this velocity will vary with space and time, it is here assumed constant for the purposes of determining plate response. The pulse front velocity, denoted as \( c_p \), enables the pulse to be represented in space and time. This is necessary for understanding the distribution of pressure on the plate. Based on experiments with a wedge shaped hull [6], the time decay constant, \( \tau_{at} \), stays constant for a given travelling pulse even though the pulse magnitude, \( P_0 \), changes. Therefore, \( \tau_{at} \) is assumed to be constant in approximating the slamming pulse. Just as the pulse has a decay constant in time, it also has an associated decay constant in space. This space decay constant is defined by the product of the time decay constant and the wave front velocity.

\[
\text{Spatial decay constant} = \tau_{at} \times c_p \quad (2)
\]

This is the distance along the plate where a fractional drop in pressure is 63.2%. It is a function of only pulse front velocity and \( \tau_{at} \). The pulse can now be defined as a function of space and time.

\[
P(x, t) = P_0 \exp\left(\frac{-(t \times c_p - x)}{\tau_{at} \times c_p}\right) \quad (3)
\]

where \( P(x, t) = 0 \) for \( x > t \times c_p \). To illustrate the association of space and time domain pulse representations, figure 2
shows a pulse of unit magnitude with \( c_p = 10 \text{ in/sec} \) and \( \tau_{ut} = .2 \text{ sec} \) in both the space and time domains evaluated at arbitrary points.

It is made apparent by figure 2 that the spatial variation in pressure across the plate is characterized by the spatial decay constant, \( \tau_{ut}c_p \). By equation (2), the spatial decay constant varies with velocity. Therefore, higher velocities will tend to a more constant pressure distribution across the plate. Assumptions have been made by many authors in predicting plate response that the pressure across the plate is constant [3, 4, 6, 7]. It will be shown that the actual spatial decay constant is critical in assuming constant pressure across the plate since \( \tau_{ut}c_p \) uniquely describes the spatial variation in pressure across the plate.

3. PLATE RESPONSE

3.1. PLATE SUPPORT AND BOUNDARIES

The hull plating on a ship is supported by frames, web stiffeners and longitudinal stiffeners. This supporting framework is typically comprised of "L", "T" or "I" beams. The hull plating is welded together smoothly and is supported at the boundaries by the framework. This forms, what is considered as, a fully clamped plate. [6, 7, 8]

3.2. GLOBAL AND LOCAL RESPONSES
The response to the pulse loading can be categorized as global and local. Global response involves the deflection of the framework due to the pulse acting on a large section of the hull. This includes many sections of stiffened plates. The local response is simply the plate deflection in reference to its boundaries. It is in the context of local response that plate slamming is considered in this paper.

3.3. STRUCTURAL COMPLEXITIES WITH SLAMMING

3.3.1. Boundary Rotation

There are several effects which occur during slamming that complicate the local plate analysis. While it seems reasonable to treat the plate as fully clamped around its entire boundary, this assumption may misrepresent actual conditions. The moving pulse travelling outboard over the hull surface may cause rotation of the longitudinal stiffener immediately in front of the moving pulse [6] as shown in figure 3. The amount of rotation of the stiffener depends on its torsional rigidity, length, stiffener spacing and plate stiffness.

3.3.2. Boundary Lateral Deflection

The sides of the plate; particularly the longer side, will deflect to some degree relative to the corners because the longitudinal stiffeners supporting the plate edge have finite stiffness. This amounts to an elastically supported plate with infinite stiffness at the corners and finite
stiffness along the sides. This stiffness will depend on the scantlings but is normally very high in relation to the plate stiffness.

3.3.3. In Plane Deflection

In plane deflection of the plate edges may occur when local deflection causes sufficient membrane stress about the plate. This effect is minimized in highly dynamic plate deflection because the in-plane inertial resistance from the surrounding plates is so large [7].

3.3.4. Initial Plate Curvature

Due the nature of hull geometry, there will be plates which are dished, twisted, or convex. In this case, small curvature shell theory may be appropriate. Additionally, there may be residual stresses due to welding from original construction. For generality, only flat plates are considered in this analysis.

The effects discussed above add considerable complexity to the problem of plate response analysis. It is important to recognize all of these effects. However, since quantifying these effects can be particular to each ship and overly complex given their limited impact, it is reasonable to model the hull plate response to slamming loads as a fully clamped initially flat plate.

3.4. RESPONSE ANALYSIS CONSIDERATIONS

3.4.1. Necessity For Elastic-Plastic Analysis
There has been a number of analyses which predicts the response of hull plating to slamming induced loads by assuming rigid-perfectly plastic behavior of metal [3,4,7,18]. It has been shown that when the plastic strain energy of deformation is of the same order of magnitude as the maximum elastic strain energy, the results of rigid-perfectly plastic behavior can differ significantly from elastic-plastic behavior [9]. Therefore, in evaluating permanent deflections which are not very large and are in the transition between elastic and plastic response, an elastic-plastic analysis is necessary.

3.4.2. Inertia Effects

If any pressure pulse has a sharp rise time and is of short duration relative to the natural frequency of the plate, the inertia of the plate must be taken into consideration. Assumptions have been made that if the duration of the pulse is larger than the natural period of the plate, the deflection can be considered static [7,8]. However, the pulse rise time contributes significantly to maximum deflection [10] because of inertia effects of the plate. Comparing pulses which eventually reach the same magnitude, the shorter the rise time, the greater the maximum deflection. This is indeed the case with slamming loads where rise times are typically 2-3 orders of magnitude shorter than the natural period of the plate [3,6]. Hence,
inertia effects play a large role in determining plate response to slamming loads.

4. PLATE MODELLING

4.1. MAXIMUM STRESS IN A CLAMPED PLATE.

Hull plate typically has a narrow aspect ratio to conform to longitudinal stiffeners. While this is not true in the case of transversely stiffened hulls, most hulls have longitudinal stiffeners. The maximum stress on a fully clamped plate under uniform load occurs at the mid point of the long side. The difference in clamped end stresses from uniform loading at this point for plates with aspect ratio greater than 2 is negligible. [11]. Figure 4. shows a plot of the ratio of moments, and therefore stresses, between a plate of infinite aspect ratio and finite aspect ratio at the long side midpoint due to a static uniform load.

4.2. PLATE STRIP

To accurately represent the behavior of these plates at the location of maximum stress, a plate strip was chosen as a model. The length of the plate strip is equal to the width of the plate and is itself of unit width. Figure 5 shows the plate strip representation on a fully clamped plate.

Using a plate strip, implies the aspect ratio of the plate (length/width) is infinity. Therefore, the behavior of this plate strip actually represents that of a fully clamped, infinitely long plate of finite width. Relating moments
directly to stress, it can be seen by Figure 4 the plate strip model applies very well to hull plating with aspect ratio greater than 2.0. The plate strip also gives a reasonably good representation for plates with aspect ratios of 1.7 where the ratio of stresses shown in figure 4 is 95%.

While this comparison is valid for a statically applied load, it is reasonably extended into the dynamic range for evaluating plate response.

4.3. PLATE STRIP BOUNDARY CONDITIONS

As stated previously, stiffener rotation and deflection at the plate boundary that can occur during slamming represent deviations from a fully clamped condition. Quantifying these effects will be difficult and ship-specific. They depend on plate dimensions and stiffener bending and torsional rigidity. Generally, these effects are considered to be small because of the relatively large stiffness of the plate support structure. Therefore, boundary rotation and in plane deflection effects will be disregarded in the following analysis. However, the plate strip is assumed to be resting on elastic supports to represent lateral deflection of the stiffeners at the ends. Figure 6 illustrates the plate strip model used to represent hull plate behavior.

5. FORMULATION OF GOVERNING EQUATIONS OF MOTION.

5.1. PLATE STRIP DYNAMICS IN ELASTIC RANGE.
Equations governing the behavior of the plate strip follow Kirchoff's hypotheses [12]: plane sections remain plane and orthogonal to the midsurface. Plane strain is assumed so stress in the lateral direction to the plate is ignored. Thus, with an infinitely long plate, only a one dimensional stress state needs to be considered in the plate strip. Figure 7 shows a differential element and conventions to illustrate the basis of governing equation formulation.

The following variables are defined:

- \( x \) Distance along plate strip from clamped end.
- \( M \) Moment due to curvature per unit width.
- \( S \) Stretching force per unit width.
- \( V \) Shear force per unit width.
- \( P \) Applied load per unit width.

The slopes of the plate strip (\( \theta \)) are assumed to be small enough such that \( \cos(\theta) \) is approximately unity. Defining the density as \( \rho \) and thickness as \( h \), it can be shown [13] that the governing equation of motion for the plate strip is

\[
\frac{\partial^2 M}{\partial x^2} + \rho h \frac{\partial^2 w}{\partial t^2} - S \frac{\partial^2 w}{\partial x^2} = P
\]

The area equals the thickness since unit width of the plate strip is assumed.

For an infinitely long plate, the in-plane strain that is perpendicular to the plate strip (\( y \) direction) is zero. Defining \( \varepsilon \) as strain, \( \sigma \) as stress, \( E \) as the modulus of elasticity, \( \nu \) as Poisson's ratio, and assuming plane stress, the following equations:
\[
\varepsilon_x = \frac{1}{E} (\sigma_x - \nu \sigma_y), \quad \varepsilon_y = \frac{1}{E} (\sigma_y - \nu \sigma_x)
\]  
(5)

give the relationship between stress and strain in the x direction. In the case of the plate strip, the stress in the strip is related to the stress by

\[
\sigma_x = \frac{E \varepsilon_x}{(1 - \nu^2)}
\]  
(6)

The moment per unit width can be determined by integrating the stress times moment arm through the thickness. Let \( z \) denote the lateral direction to the plate.

\[
M(x) = \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma(z)xz \, dz
\]  
(7)

Under elasticity and Kirchhoff's assumptions, the bending moment in the strip is proportional to the curvature, \( K = \frac{\partial^2 w}{\partial x^2} \). Also, the stress due to stretching is accounted for where \( \varepsilon_0 \) is the plate strip midsurface strain. Therefore the moment can be determined from the following equation:

\[
M(x) = \int_{-\frac{h}{2}}^{\frac{h}{2}} -E \left( \varepsilon_0 z - \frac{\partial^2 w}{\partial x^2} \right) \frac{\partial^2 w}{\partial x^2} \, dz = \frac{Eh^3}{12(1 - \nu^2)} \frac{\partial^2 w}{\partial x^2}
\]  
(8)

The coefficient, \( \frac{Eh^3}{12(1 - \nu^2)} \), preceding the curvature on the right hand side is typically referred to as the flexural rigidity, \( D \). Therefore, the relationship of curvature to moment is

\[
M(x) = DK(x); \quad K(x) = \frac{\partial^2 w(x)}{\partial x^2}
\]  
(9)
Using (9), (4) now becomes.

$$D \frac{\partial^4 w}{\partial x^4} + \rho A \frac{\partial^2 w}{\partial t^2} - S \frac{\partial^2 w}{\partial x^2} = P(x,t)$$  \hspace{1cm} (10)

Since the strain is assumed to vary linearly through the thickness of the plate strip, the total strain at any point through the thickness is the sum of strain due to curvature and strain due to membrane stretching. The membrane strain is simply $\varepsilon_0$, the midsurface strain. It is assumed the edges are restrained from in plane deflection. Therefore, $\varepsilon_0$ is defined as the ratio of the change in length of the plate strip to the unloaded plate strip length.

$$\varepsilon_0 = \frac{L(t) - L(0)}{L(0)}$$  \hspace{1cm} (11)

where $L(t)$ is the length of the plate strip at any time, $t$, and $L(0)$ is the length at time $0$; no load condition. Using the formula determining the length of a curve, the midsurface strain is defined as

$$\varepsilon_0 = \frac{\int_0^L \sqrt{1 + \left(\frac{\partial w}{\partial x}\right)^2} \, dx - L(0)}{L(0)}$$  \hspace{1cm} (12)

Taking a Taylor expansion of the integrand and dropping terms of $\partial w/\partial x$ which are to the 4th power or higher yields a very close approximation to average membrane strain in the plate strip.

$$\varepsilon_0 \approx \frac{1}{2L} \int_0^L \left(\frac{\partial w}{\partial x}\right)^2 \, dx$$  \hspace{1cm} (13)
This immediately yields the stretching force per unit width, \( S \), from the stress-strain relation (6).

\[
S = \frac{Eh}{2L(1-v^2)} \int_0^L \left( \frac{\partial^2 w}{\partial x^2} \right)^2 \, dx
\]  
(14)

Substituting for \( S \), (14) into (10) gives an integro-differential equation for the motion of the plate strip:

\[
D \frac{\partial^4 w}{\partial x^4} + \frac{\rho A}{E h^3} \frac{\partial^2 w}{\partial t^2} - \frac{EA}{2L(1-v^2)} \frac{\partial^2 w}{\partial x^2} \int_0^L \left( \frac{\partial w}{\partial x} \right)^2 \, dx = P(x,t)
\]  
(15)

5.2. NON-DIMENSIONAL FORM OF EQUATION OF MOTION

To help generalize results based on this equation, it should be made dimensionless. There are two parameters found to occur frequently in dimensional analysis. They are the length to thickness ratio and a stiffness term which drops out of \( D \), flexural rigidity.

\[
R = \frac{L}{h} \quad \text{and} \quad \beta = \frac{D}{E h^3} = \frac{D}{12(1-v^2)}
\]  
(16)

Additionally, the fundamental length variables, \( x \) and \( w \), are nondimensionalized as follows:

\[
\tilde{x} = \frac{x}{L} \quad \tilde{w} = \frac{w}{h}
\]  
(16a)

Time is made dimensionless by comparing it to the small deflection natural period of the plate strip with simply supported ends. This applies to the clamped end plate strip, as well, differing only by a constant. The natural period, \( T_{nat} \), of a simply supported plate strip is
\[ T_{\text{nat}} = \frac{2L^2}{\pi} \sqrt{\frac{\rho h}{D}} \]  

(17)

By dropping the constants in equation (17), dimensionless time can be defined as

\[ \tilde{t} = \frac{t}{L^2 \sqrt{\frac{\rho h}{D}}} = \frac{t}{RL \sqrt{\frac{\rho}{\beta}}} \]  

(18)

Applying these dimensionless parameters to the governing equation (10) yields:

\[ \frac{Dh}{L^4} \frac{\partial^4 \tilde{w}}{\partial x^4} + \frac{Dh}{L^4} \frac{\partial^2 \tilde{w}}{\partial t^2} - \frac{hS}{L^2} \frac{\partial^2 \tilde{w}}{\partial x^2} = P(x,t) \]  

(19)

This naturally leads to the dimensionless form of the stretching force, \( S \), and applied load, \( P \).

\[ \tilde{S} = \frac{S}{\left( \frac{D}{L^2} \right)} \quad \tilde{P}(x,t) = \frac{P(x,t)}{\left( \frac{Dh}{L^4} \right)} = \frac{P(x,t) \cdot R^4}{\beta} \]  

(20)

This results in the preliminary dimensionless form of the equation of motion.

\[ \frac{\partial^4 \tilde{w}}{\partial x^4} + \frac{\partial^2 \tilde{w}}{\partial t^2} - \tilde{S} \frac{\partial^2 \tilde{w}}{\partial x^2} = \tilde{P}(\tilde{x},\tilde{t}) \]  

(21)

Applying the dimensionless variables to equation (14), which quantifies the stretching force, gives

\[ \tilde{S} = 6 \int_0^1 \left( \frac{\partial \tilde{w}}{\partial \tilde{x}} \right)^2 d\tilde{x} \]  

(22)

This results in the final form of the dimensionless governing equation of motion.
Since this analysis includes evaluation of the stress in a plate strip modelled with elastic supports, both stress and the stiffness of the supports must be made dimensionless.

The total stress at any point in the strip can be decoupled into stress due to curvature and stress due to stretching of the strip. The maximum stress at any point along the length of the strip occurs at either the top or bottom because of bending and the assumption of linear strain variation through the thickness of the plate strip. The following stress components are defined: \( \sigma_b \) is the stress on the plate surface due to bending. \( \sigma_o \) is the uniform stress across the plate thickness due to membrane stretching. \( \sigma_{\text{max}} \) is the sum of these two stresses at the surface of the plate where the stress is maximum. From the Moment-Curvature relationship and equation (9), the maximum stress can be determined.

\[
\sigma_{\text{max}} = \sigma_b + \sigma_o = \frac{6D}{h^2} \left( \frac{\partial^2 w}{\partial x^2} \right) + \frac{S}{h} \tag{24}
\]

Applying the dimensionless forms of \( x, w, \) and \( S \) to this equation yields the dimensionless form of maximum stress, \( \sigma_{\text{max}} \).

\[
\sigma_{\text{max}} = 6 \left( \frac{D}{hL^2} \right) \frac{\partial^2 w}{\partial \tilde{x}^2} + \tilde{S} \quad \therefore \quad \tilde{\sigma}_{\text{max}} = \tilde{\sigma}_{\text{max}} = \frac{\sigma_{\text{max}} R^2}{\tilde{\beta}} \tag{25}
\]
Based on this equation, the resultant dimensionless maximum stress at any point, \( x \), can be defined.

\[
\tilde{\sigma}_{\text{max}} = 6 \left| \frac{\partial^2 \tilde{w}}{\partial x^2} \right| + \tilde{S}
\]  

(26)

Under the assumption of elastic supports modelled as linear elastic springs, the shear force at the clamped end is proportional to the deflection. Assuming the support spring has a stiffness, \( k \), the boundary condition of elastic supports can be imposed.

\[-D \frac{\partial^3 w}{\partial x^3} = kw(x=0,L)\]  

(27)

This relates the shear force at the ends to the deflection. Applying the dimensionless forms of \( w \) and \( x \) yields the dimensionless equivalent of equation (27):

\[-\frac{\partial^3 \tilde{w}}{\partial x^3} = \tilde{k} \tilde{w}(x=0,1) \text{ where } \tilde{k} = \frac{k}{\frac{D}{L^3}}\]

(28)

6. **FINITE DIFFERENCE MODEL.**

6.1. **FINITE DIFFERENCE MODEL OF PLATE STRIP.**


Classical finite difference techniques are employed to evaluate the time and space domain response of the plate strip. Several programs are written in the computer language "C" which are based on these techniques. Appendix B is the program, NONDIM.C, for plate strip response under the
assumption of a spatially constant load. Appendix C is the
program, NONDIM.FULL.C, for plate strip response where
applied force can vary independently in space and time.

This propagation problem uses an implicit time step
method which averages the spatial finite difference
approximations at time steps before and after the current
time step [14,15]. This implicit method is appealing since
it is unconditionally stable as can be shown by Von Neuman
analysis. That is, no spurious modes of solutions ever cause
numerical instabilities.

Letting the subscript $j$ denote the space coordinate
(node) and superscript $k$ denote the time step, the following
fundamental finite centered difference formulas are used:

\[ \frac{\partial w_j^k}{\partial x} = \frac{w_{j+1}^k - w_j^{k-1}}{2 \Delta x} \]  \hspace{1cm} (29)

\[ \frac{\partial^2 w_j^k}{\partial x^2} = \frac{w_{j-1}^{k-1} - 2 w_j^k + w_{j+1}^{k-1}}{\Delta t^2} \]  \hspace{1cm} (30)

\[ \frac{\partial^4 w_j^k}{\partial x^4} = \frac{w_{j-2}^{k-1} - 4 w_{j-1}^{k-1} + 6 w_j^k - 4 w_{j+1}^{k-1} + w_{j+2}^{k-1}}{\Delta x^4} \]  \hspace{1cm} (31)

Using equations (29), (30) and (31) in the implicit time
step method applied to equation (21) gives the following
formula for the $j$th spatial node and the $k$th time step.

\[ w_j^{k+1} = 2 w_j^k - w_j^{k-1} + (\Delta t)^2 \frac{\partial^2 w_j^k}{\partial x^2} \]
\[ + (\Delta t)^2 \left[ \frac{1}{2} \left( \frac{w_{j-1}^{k-1} - 2 w_j^{k-1} + w_{j+1}^{k-1}}{\Delta t^2} \right) + \frac{1}{2} \left( \frac{w_{j-1}^{k-1} - 2 w_j^{k-1} + w_{j+1}^{k-1}}{\Delta t^2} \right) \right] \]
\[-\frac{(\Delta t)^2}{2} \left\{ \frac{w_{j,2}^{k+1} - 4w_{j,1}^{k+1} + 6w_{j,1}^{k+1} - 4w_{j+1,1}^{k+1} + w_{j+2,1}^{k+1}}{\Delta x^4} \right\} \]

\[-\frac{(\Delta t)^2}{2} \left\{ \frac{w_{j,2}^{k+1} - 4w_{j-1,1}^{k+1} + 6w_{j,1}^{k+1} - 4w_{j+1,1}^{k+1} + w_{j+2,1}^{k+1}}{\Delta x^4} \right\} \]  \hspace{1cm} (32)

Note that terms from the k+1 time step appear on both sides of the equation. By invoking boundary conditions all deflections in time step k+1 can be determined through the solution of simultaneous equations.

The dimensionless stretching force, $S$, is determined by use of equation (22) and (29). Simpson's rule is employed to numerically integrate the square of the slope in equation (22). This is done at each time step so the dimensionless value of the stretching force can be used in equation (32) for time stepping.

6.1.2. Finite Difference Boundary Conditions.

With clamped ends, the slope of the plate strip at the first node and the last node must be zero. To impose this, (29) is used, where $N$ is the number of the last node, to give:

$$w_{1,1}^k = w_{N+1,1}^k , \quad w_{N,1}^{k+1} = w_{N+1,1}^{k+1}$$  \hspace{1cm} (32a)

The program is written such that initial nodal displacement and velocity can be specified. While both velocity and displacement are assumed to be zero in evaluating the slamming response, initial conditions are used to evaluate the accuracy of the model. Additionally, it is an option available to future users of the program.
The stiffness of the elastic supports must be specified. As mentioned previously, since the stiffness of the plate supports vary from ship to ship, the stiffness is made extremely large for simplicity. This represents a fully clamped immobile support. It is important to note, however, that the finite support stiffness is a variable entered by the user.

6.2. MODEL ACCURACY VALIDATION.

6.2.1. Galerkin Solution.

To ensure the accuracy of the finite difference program it should be compared to a known solution in some way. Since the problem is nonlinear, exact solutions which would validate the performance of the finite difference program over a wide variety of conditions are difficult, if not impossible, to obtain. Therefore, comparison to an approximate solution which applies to small and large deflections is a good method to check accuracy.

A very good approach is to approximate the steady response to a spatially constant, sinusoidally time varying load. That is, $P(x,t) = P_0 \sin(\omega t)$. Therefore, a Galerkin solution [16] is obtained for the plate strip response under this loading. This is done to check if the finite difference solution is inaccurate due to some programming error. Since the problem is non-linear, this comparison must be made for various magnitudes to ensure, not only the linear effects, but also the nonlinear effects are consistent with an
approximate solution. If both the finite difference and the Galerkin solution are similar, this lends a great deal of credence to the accuracy of the program, NONDIM.C, using finite difference techniques.

The assumed shape function used for the Galerkin solution is \( w_s(x,t) = \sin^2(\pi x/L) \sin(\omega t) \). This is modulated with the center deflection amplitude, \( W \). Therefore the shape of the plate strip is approximated by \( w(x,t) = Ww_s \). The variable \( \omega \) is a floating variable so frequency response can be determined. The shape function satisfies the clamped end boundary conditions and assumes a shape that is intuitively similar to the expected shape as shown in figure 8.

Letting \( w(x,t) = Ww_s \) and putting it into the governing equation of motion (15) will cause an inequality with the left and right hand side. The difference between these is called the residual [15,16]. If the residual is integrated over some domain, with the weighting function \( w_s \), the Galerkin solution minimizes the integrated residual by optimizing \( W \). This results in the closest possible solution to the differential equation over that domain using the shape function and center deflection, \( W \), as the single degree of freedom. Letting the residual be denoted as \( R_G \), the Galerkin method applied over one period of vibration must satisfy the following integral:

\[
\int_0^L \int_0^{2\pi} R_Gw_s \, dt \, dx = 0
\] (33)
The result of the Galerkin solution with \( w(x,t) = Ww_s \) and spatially constant load \( = P_0 \sin(\omega t) \) applied to equation (15) is

\[
WD^2 \left( \frac{\pi^4}{L^4} \right) - W \omega^2 \rho A \frac{3}{8} + \frac{W^3 E A A_3}{(1 - v^2) 32 L} - \frac{P_0}{2} = 0
\]  

The dimensionless form of equation (34) is

\[
\bar{W} 2 \pi^4 - \bar{W} \bar{\omega}_o \frac{3}{8} + \frac{W^3}{8} \frac{9}{8} \pi^4 - \frac{\bar{P}_o}{2} = 0.
\]  

where

\[
\bar{\omega} = \omega L^2 \sqrt{\frac{\rho h}{D}}
\]  

Note in equations (34) and (35); the first term is the stiffness term, the second is the inertia term, the third is the stretching term and the fourth is the applied force.

By comparing the frequency response of both solutions, a very good assessment can be made of their mutual consistency since the frequency response reveals a great deal about system through a wide range of dynamic behavior. Appendix D shows the frequency response of Galerkin solution and the finite difference solution. It is readily apparent how close the two responses are to each other. This is a strong argument for the accuracy of the finite difference program.

To give insight to the difference between the Galerkin solution and the finite difference solution, figure 9 shows each of their dimensionless deflection half shapes under static uniform dimensionless loading of 227. This load causes a dimensionless deflection of .5 for the finite difference solution. A "C" language program was developed...
using the same fundamental finite difference equations (29), (30), and (31) to evaluate the static loading of the plate strip. (Appendix E) Note the finite difference deflection has greater curvature at the ends and less in the middle than does the Galerkin approximate solution. In general, though, the approximation is very good. In the Figure 9, the percentage difference in center deflection is 1.6%.

6.2.2. Wave Speed and Dispersion Relationship.

While the frequency response comparison provides convincing evidence of accuracy in the global sense, a further measure can be taken to evaluate the accuracy of the finite difference program. The propagation of flexural waves in the plate strip should not only be detected by the finite difference approximation of the plate strip dynamics, but should accurately portray the wave velocities.

To test for this, the plate equilibrium equation is considered without the forcing term.

\[ D \frac{d^4 w}{dx^4} + \rho A \frac{d^2 w}{dx^2} - S \frac{d^2 w}{dx^2} = 0 \]  

(37)

Assuming a complex solution \( w(x,t) = e^{i\kappa(x-ct)} \) where \( \kappa \) is the wave number and \( c \) is the wave speed, the wave dispersion relation can be determined. This relates wave speed to wave length [17]. This relationship is

\[ c = \sqrt{\frac{\kappa^2 D + S}{\rho A}} \]  

(38)
For the purposes of verifying model wave response of small deflections, the stretching force $S$ can be ignored. This yields the linearized dispersion relationship.

$$
c = k h \sqrt{\frac{\beta}{\rho}} = \frac{2 \pi h \sqrt{\beta}}{\lambda} \tag{39}
$$

where $\lambda$ is the wave length. The dimensionless dispersion relationship, based on the the dimensionless parameters previously developed is

$$
\tilde{c} = \frac{2 \pi}{\lambda} \tag{40}
$$

The dimensionless wave period can now be defined as

$$
\tilde{T} = \frac{\tilde{\lambda}}{\tilde{c}} = \frac{\tilde{\lambda} \tilde{c}}{2 \pi} = \frac{2 \pi}{\tilde{c} \tilde{c}} \tag{41}
$$

Based on these relations, it can be seen that smaller amplitude waves propagate faster than larger amplitude waves. This can be used qualitatively assess the behavior of the finite difference model. In order to produce tangible waves in the plate strip, an initial velocity was imposed on a given node. This is equivalent to an impulse loading. Figure 10 shows a 3-dimensional plot of the plate strip dimensionless response to an impulse applied at one third the length of the strip. Two significant comments can be made about this figure. The speed of the waves relative to each other can be determined by the slope of the wave crest with respect to time. Note the smaller wave crests have a steeper
slope which indicates a greater velocity. This conforms to the dispersion relationship of equation (40). Additionally, the boundaries are clamped so wave reflection would be expected. This can be seen clearly after time = 0.002.

Figure 11 is a 3 dimensional plot of dimensionless deflection response to an impulse at the center node. Only half of the strip is shown for clarity. Note the curvature of the wave crests with time as wave length increases and speed decreases. Figure 12 shows a more detailed view of the same response for one tenth the time period. Over the range of this time period the crest of the waves are fairly straight. To relate the wave lengths and propagation velocities, the data from figure 12 is used to measure an arbitrary wave length and the time to propagate one wavelength. Due to wave dispersion, a peak to peak distance average was taken over a time period of one wave length propagation. The following table shows the dimensionless values that were measured and the velocity determined from both dispersion relationship and measured wavelength.

<table>
<thead>
<tr>
<th>Average Wavelength</th>
<th>Period</th>
<th>Measured Velocity</th>
<th>Dispersion Relationship Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.02893</td>
<td>0.00013</td>
<td>222.5</td>
<td>217.2</td>
</tr>
</tbody>
</table>

Table 1. Elastic Wave Propagation Data
The 2-3% difference between the measured velocity and the velocity determined from the dispersion relationship and measured wavelength is minimal.

Based on the close comparison of frequency response between the Galerkin approximate solution and the finite difference model, and its concurrence with the wave dispersion relationship, the finite difference program shows no indication of error due to a faulty algorithm.

6.3. USE OF FINITE DIFFERENCE MODEL

6.3.1. Boundary Conditions

The finite difference model is assumed to have elastic supports. The degree of elasticity of the support depends on the frame spacing of the ship, stiffener rigidity and plate stiffness. In order to determine the stiffness of the elastic support, the deflection of the center of the stiffener which bounds the plate (i.e. plate strip clamped end) must be related to the applied load. This amounts to solving a plate problem with the long edges having orthotropic properties and points elsewhere on the plate having isotropic properties. The anisotropy is from the stiffener along the long edge. The supporting framework is much stiffer than the plate but this varies from ship to ship. So it is reasonable that in order to maintain consistency and generality, the stiffness of the support is set to very large, $5 \times 10^{25}$, which is equivalent to a fully clamped, immobile boundary. The effect of finite stiffness
in the supports is left to future analysis with this program used as a tool.

6.3.2. Convergence of Results

It has been shown that the program for plate strip dynamics agrees well with approximate solutions and the wave dispersion relationship. For the purposes of determining slamming pulse response, adequate node spacing and time step size must be determined.

A dimensionless, spatially constant pulse of 2000 exp(-t/.005) was applied for various node spacings and time steps to achieve convergence of results. This pulse is in the high end of the magnitude range and the low end of the decay constant range. This will cause greatest accelerations and is therefore a very good trial pulse.

Appendix F shows plots of center deflection and clamped end stress vs. time for various node spacing and time steps. It can be seen that for this trial pulse of high magnitude, if the number of nodes is greater than 100 and the time step is smaller than .0002, then adequate convergence is achieved. In obtaining results with this program, the number of nodes used is 160 and the dimensionless time step is .0002.

6.4. MOVING PULSE FRONT

6.4.1. Effect of Pulse Front Velocity

There have been assumptions made in the analysis of plate response to slamming pulses that the pressure across
the plate in the direction of the pulse velocity is constant [3,6,7]. This allows the analysis to be more tractable, but this assumption needs justification. If the assumption is reasonable, then the finite difference program can take advantage of symmetry about the center of the plate strip and increase accuracy while saving time. Additionally, it removes pulse front velocity as another variable makes the analysis less general and more complex.

Therefore, the effect of varying pulse front velocities on plate response are considered using the finite difference program which allows applied force variation in both space and time.

As shown in figure 2, the spatial distribution of pressure is determined by the time decay constant, tau_t, and the pulse front velocity, c_p. The spatial decay constant, tau_t*c_p, completely determines the spatial distribution of pressure at any one time. Therefore, several runs of the finite difference program are made with varying velocities but the same pulse in the time domain. These results are compared to the spatially constant pulse response which is equivalent to pulse front velocity approaching infinity. Appendix G shows plots of these results. The stress in the clamped ends is used for the comparative measure as this is where the maximum stress will occur. By symmetry, the stress response at both clamped ends will be the same for the spatially constant pulse. The pulse with finite velocity produces asymmetric clamped end responses. Appendix H shows
3-D plots of stress in the plate strip vs. time for the spatially constant pulse and the same pulse with finite velocity. Note the symmetry in the response to the spatially constant pulse as compared to the response to finite velocity pulse. The latter has a delayed response at the second clamped end \((x=1.00)\) because of the finite amount of time for the pulse front to reach it.

It can be seen from appendices G and H that for a dimensionless spatial decay constant \((\tau a_U^p)\) greater than 1, the spatially constant approximation of the moving pressure pulse is quite accurate. Therefore, this can be considered a condition for the accuracy of the spatially constant approximation as it is applied to the finite difference program. The condition that \(\tau a_U^p\) be greater than 1 is quite reasonable since it applies to actual slamming conditions as indicated by records of pulse front velocities [6].

Based on the results above, the assumption of the spatially constant pulse is applied in the analysis of plate strip response to slamming induced pulses.

7. RESULTS OF ELASTIC ANALYSIS

7.1. INCIPIENT YIELD PULSE

Since the model and finite difference program are valid only in the elastic range, its best use is to find the combination of pulse magnitudes and decay constants which induce incipient yield. That is, when the first point on the surface of the plate just reaches yield strength. Any pulse
which causes this incipient yield on the plate surface is defined here as a "yield pulse." It is known that a pulse which causes incipient yield will cause no permanent deflection, by definition, but most importantly, it is a lower bound to plate yielding.

7.2. SCOPE OF APPLICATION OF DIMENSIONLESS RESULT

From actual yield pulse measurements, [1 thru 7], the time decay constants of the pulse can vary from as low as .001 to .5 seconds or more. Therefore, the finite difference program must be run through this range of decay constants and at various pulse magnitudes to apply to a variety of plates.

Since commercial steel is typically used as hull plate its properties were used in the program. The assumed density is .283 lbsm/in³. The program was run for yield strengths of 40000 psi and 60000 psi as this covers a wide range steels in use.

The three fundamental dimensionless parameters used in applying the results of the program are maximum stress, initial pulse magnitude and pulse time decay constant.

There will be some point during the response where stress on the plate surface is a maximum. This is found to occur at the clamped end in all instances. By equation (25), it can be seen that the dimensionless maximum stress applies to a unique value of $R (L/h)$ when $\sigma_{max}$ and $\beta$ are prescribed values. The value for $\beta$ is based on properties of steel and $\sigma_{max}$ is the yield strength which results in:
\[ R = \sqrt{\frac{\sigma_{\text{max}} \cdot \beta}{\sigma_{\text{max}}}} = \sqrt{\frac{\sigma_{\text{max}} \cdot \beta}{\sigma_{\text{yield}}}} \]  

(42)

Consequently, the initial pulse magnitude \( P_0 \) (psi) is determined by the magnitude of the dimensionless load which was applied. This is based on equation (20) and results in:

\[ P(0) = \frac{P(0) \cdot \beta}{R^4} \]  

(43)

Similarly, the time decay constant (in seconds) is determined from equation (18)

\[ \tau = \tau_{RL} \sqrt{\frac{\rho}{\beta}} \]  

(44)

Therefore, any result from the finite difference program applies to a unique value of \( R \), with the strip length (plate width) as the free parameter for general application.

7.3. FIRST YIELD PULSE CHART

The goal of this elastic analysis is to find the combination of pulse magnitudes and time decay constants which are yield pulses. It is most practical to categorize plates by \( R = L/h \).

To synthesize the data from numerous runs made with the finite difference program over a range of practical application, charts are assembled for each yield strength: 40000 and 60000 psi. As shown by equation (42), \( R \) is determined by the dimensionless maximum stress that comes from the program run. This will almost always yield values
of $R$ which are not integers. Interpolation was then necessary to get data for evenly spaced values of $R$.

Based on equation (44), $\tau_\text{ut}/L$ is the natural choice of variable to indicate what value of decay constant is associated with yield pulse magnitude. Using the parameters $P_0$, $\tau_\text{ut}/L$ and $R$, the results of the program runs are efficiently condensed into one chart for each yield strength. The pulse peak magnitude varies from 0 to 200 psi and $\tau_\text{ut}/L$ varies from 0 to .01. Appendix (I) shows log-log and normal plots for 40000 and 60000 psi yield stress. These plots give the combination of $P_0$ and $\tau_\text{ut}/L$ that cause incipient yield through a range of $R$ which are commonly found in ship construction.

7.4. USE OF YIELD PULSE CHARTS.

The yield pulse charts give a lower bound for plate yielding under a specified slamming pulse. Depending on the design requirements, which can be based on a wide range of failure criteria, the yield pulse charts will serve well as a guide when the expected slamming pulse characteristics are known.

These charts give a correspondence between Length, $R$, $\tau_\text{ut}$, $P_0$ and plate thickness, $h$. The choice of what variable has priority can be made by the user. Table 2 shows what unique output, denoted with $O$, can result from the chart based on specified input parameters, denoted as $I$. 

42
Table 2 Yield Pulse Chart Input/Output Relationships

<table>
<thead>
<tr>
<th>Case</th>
<th>$\tau_{ul}$</th>
<th>$P_0$</th>
<th>$h$</th>
<th>$R$</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>I</td>
</tr>
<tr>
<td>2</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>I</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>I</td>
<td>I</td>
<td>I</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>I</td>
<td>0</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>6</td>
<td>I</td>
<td>0</td>
<td>0</td>
<td>I</td>
<td>I</td>
</tr>
<tr>
<td>7</td>
<td>I</td>
<td>0</td>
<td>I</td>
<td>0</td>
<td>I</td>
</tr>
</tbody>
</table>

8. ELASTIC-PLASTIC RESPONSE

8.1. NECESSITY OF ELASTIC-PLASTIC ANALYSIS.

The load at initial yield from the elastic analysis serves as a lower limit to the load for general plate yielding. Based on prescribed failure criterion, it is of interest to the Naval Architect to know how much more pressure a plate can sustain beyond a yield pulse. To extend the elastic analysis, it is of interest to determine what the plate response and permanent deflections will be to multiples of the yield pulse; that is, when holding the time decay, $\tau_{ul}$, constant and multiplying the yield pulse magnitude. This is done with an elastic-plastic model for a range of plate dimensions and yield pulses applicable to slamming.

Approximating moderately large permanent deflections has been accomplished by assuming rigid-plastic hinge lines.[18]. However, it has been shown [9] when the plastic deformation is of the same order as the elastic, rigid-plastic assumptions may result in large errors. Therefore, an
8.2. ELASTIC-PLASTIC (E-P) MODEL.

8.2.1. Yankelevsky’s Elastic-Plastic Beam Model

A model was developed by D.Z. Yankelevsky for elastic-plastic beam deflections [19]. This model has simply supported ends. It is comprised of two rigid sections with a hinge in the center. The hinge is comprised of layers of bar elements which have prescribed elastic-plastic behavior. The resultant behavior of the whole hinge can then represent an elastic-plastic hinge.

As in the elastic analysis, it is desirable to apply the plate strip approximation for elastic-plastic behavior. Of course, the deflections will be greater. Plate strip theory applies provided the membrane effect in the longitudinal direction of the plate is negligible compared to the membrane effects in the transverse direction. Therefore, application of the plate strip into plastic analysis is limited to deflections where membrane stretching has only a moderate effect.

8.2.2. Modifications to Yankelevsky’s Model.

It is shown in the elastic analysis that the plate strip is essentially a stiffened beam by a factor of $1/(1-v^2)$. Therefore, in applying Yankelevsky’s model to plate
deflection, the clamped ends and increased stiffness must be accounted for.

To represent the behavior of clamped end condition, elastic-plastic hinges are added to each end of the model. The resultant model is shown pictorially in figure 13 with the bar element hinges at the center and at the ends.

8.2.3. Summary of Model Development

In plasticity, an upper bound to the fully plastic load is found by any assumed displacement field giving the plastic work. In the dynamics problem, and now by analogy, this concept of assumed displacement field to find stresses and resisting forces is applied to incremental elastic-plastic deformation. The strain field in the hinges is developed from elastic deflections of the plate strip. The strain magnitude is derived from the stress magnitude induced by a static uniform load on a purely elastic strip. A relationship is then made between the elastic center deflection and strains at the hinges. Based on these strains which are dependent on center deflection, the stresses in each element of the elastic-plastic model are incrementally tracked through elasticity and plasticity to determine moments and membrane forces.

The advantage of this formulation is that in the elastic range, the static behavior of the model matches exactly the behavior of the actual strip as determined by finite difference calculations. Also the elastic-plastic model
hinge enters plasticity at the same center deflection of the actual strip under static uniform load. After hinge elements enter into plasticity, the plastic strains become more concentrated and may no longer be representative of the actual hinge; however, the stresses which determine the motion of the strip remain accurate since the hinge elements have reached yield. Once the element enters back into elasticity, the linear relationship between stress and strain occurs in the E-P model and the actual strip. Therefore, this model is considered to model the dynamic behavior of the plate strip reasonably well.

8.2.4. End and Midstrip Stresses Based on Elastic Analysis.

In the E-P model formulation, the strains in the hinge elements are correlated to center deflection by elastic plate strip stress response to static uniform loads. Therefore, it is necessary to relate stresses at the end and middle of the finite difference plate strip to center deflections. Then the strains at the ends and in the middle are calculated from the stress-strain relationship. Uniform loading is assumed since the slamming pulse is assumed to be spatially constant as shown in the elastic analysis. Also, while this is a dynamic analysis, using static response is merely a common basis for all plate sizes. It is therefore necessary to make a comparison between the static and dynamic stress response of the actual plate strip. This is accomplished using the finite-difference programs for the elastic dynamic and static
response. As a test case, a yield pulse is applied to a plate strip 100 inches long. Appendix J shows a comparison between half of the symmetric plate strip shape during dynamic response and static response. Two times are considered for the dynamic response: when the end stress and when the midstrip stress are maximum. At these times, the shape of the strip is compared to the shape of the same strip under a static uniform load which causes the same center deflection. Based on these two extreme conditions, it can be presumed that the shape of the actual plate strip during elastic dynamic response stays in the neighborhood (and on both sides) of the statically deformed strip shape with the same center deflection. This is a good indication that the E-P model is representative of the average stress at the ends and in the middle of the actual strip. This concept is extended into plasticity where the stresses in the E-P model elements have elastic-plastic behavior and are incremented based on the change in element strains with each time step. These strains are based on stresses as a function of elastic center deflection.

When the center deflection is zero, the hinge elements are considered to be of zero length. Whenever deflection occurs, the extension or compression of the elements is due to pure bending at the hinge and stretching of all hinge elements. As in the elastic analysis of the continuous plate strip, these strain components are separable. Their sum
completely defines the displacement of the elements as shown in figure 14.

The magnitude of element displacement due to bending and stretching is based on the simple geometry of the model during deflection. The displacement for any element can be made strictly a function of center deflection, $W$, with the plate strip thickness and initial length fixed. Due to symmetry, only half of the elastic-plastic model need be considered. It is assumed the stretching displacement is shared equally between the end hinge element and half of the center hinge.

Letting $u_i$ be the displacement of the 'i'\textsuperscript{th} element and 'z', the distance from the midsurface, the displacement of each element is

$$u_i = u_b + u_o$$

(45)

where $u_b$ and $u_o$ are displacement due to bending and stretching respectively. Each is defined as follows:

$$u_b = \frac{W}{L} \cdot z$$

$$u_o = \sqrt{\left(\frac{L}{2}\right)^2 + W^2} - \frac{L}{2}$$

(46)

With element displacements known as a function of center deflection, it is necessary to correlate these displacements to stress in the actual strip as a function of center displacement. The program for elastic plate strip static response, STATIC.C, is used for this purpose. A Newton type root finding method is employed so convergence to a desired center deflection is obtained by varying the applied uniform
load. As a result, the maximum end bending, maximum mid-span bending and membrane stresses are determined for a range of dimensionless center deflection responses. Appendix K shows these results graphically. Bending stress through the thickness is easily determined due to its linear variation. An excellent 5th order polynomial curve fit is made for these stress-deflection relationships in two ranges of applicability: $0 < W/h \leq 1$ and $1 < W/h \leq 4$. Appendix K gives the polynomial coefficients for each type of stress as a function of center dimensionless deflection, $W/h$. This result actually gives the dimensionless stresses in an infinitely long, clamped plate based on elastic, nonlinear response to dimensionless uniform loads. The dimensionless values have applicability to all isotropic, constant thickness plates.

8.2.5. Strain in Hinge Elements

The strain in each element can be a function of center deflection. This is determined by element displacements and from associated stresses which are considered to be a function of center deflection as described above. With the strain known for each element as strictly a function of center deflection, the kinematic strain field is defined. From this strain field, elastic-plastic analysis may be accomplished.

Small strains are typically defined as some change in length per length; $\Delta l/l$. The element displacement can be correlated to $\Delta l$. The length over which the strain is based
must be determined so the stress-strain relationship of (6) holds true in the elastic range. The basis is defined to be some factor of \( L/2 \).

In contrast to stresses, the strains in a hinge can be decomposed to bending and stretching while in plasticity where it is assumed plane sections remain plane and perpendicular to the midsurface.

The strain of an element, say an end hinge element, can then be defined as follows

\[
\varepsilon_i = \frac{u_i}{F_{b,\text{end}} \times \frac{L}{2}} + \frac{u_i}{F_m \times \frac{L}{2}}
\]

where \( F_{b,\text{end}} \) and \( F_m \) are defined as "Characteristic Length Fractions (CLF)" for bending strain and membrane strain respectively. This implies that the CLF is that fractional length of the half strip which is the basis for strain associated with each element's displacement, \( u_i \). Recall that \( u_i \) is determined from center deflection. Actual stress based on static elastic response is also a function of center deflection as shown by the results of appendix K. Strain associated with actual stress is then determined by (6). Thus, for this model, actual strain becomes determined by center deflection alone. This actual strain must be equated to the E-P model strain by choosing the correct values of \( F_{b,\text{end}}, F_{b,\text{mid}} \) and \( F_m \). The magnitude of strain becomes inaccurate when the hinge is in plasticity due to the associated increased curvature. However, the resultant
stresses which determine the plate strip dynamics and final deflections are not affected by the strain inaccuracy but only by the specified yield characteristics of the material.

We can equate any component of strain to that component of 'actual' stress from elastic analysis. Based on dimensionless parameters,

$$\varepsilon_{\text{actual}} = \frac{a(W)}{12(1-v^2)R^2}$$  \hspace{1cm} (48)

where the applicable stress as a function of center deflection is defined in appendix K. Based on (46), (47) and (48), bending strains at the outer element of the E-P model and the 'actual' plate surface, as determined by finite difference, are equated to solve for $F_{b,\text{end}}$ and $F_{b,\text{mid}}$ as a function of dimensionless deflection, $W/h$.

$$F_{b,\text{end or mid}} = \frac{\overline{W}^{*24*(1-v^2)}}{\overline{a(W)}}$$  \hspace{1cm} (49)

The dimensionless stress function is the end or mid-span bending stress at the plate surface for the respective CLF.

Based on (46) and (47), a taylor expansion can be formed for the model membrane strain.

$$\varepsilon_o = \frac{\left[2\left(\frac{W}{L}\right)^2 - 2\left(\frac{W}{L}\right)^4 + 4\left(\frac{W}{L}\right)^6 - 10\left(\frac{W}{L}\right)^8\right]}{F_m}$$  \hspace{1cm} (50)

and using (48) where the stress function is membrane stress, the membrane CLF can be determined.

$$F_m = \frac{\left[2\left(\frac{W}{L}\right)^2 - 2\left(\frac{W}{L}\right)^4 + 4\left(\frac{W}{L}\right)^6 - 10\left(\frac{W}{L}\right)^8\right] * 12 \left(1-v^2\right)R^2}{\overline{a(W)}}$$  \hspace{1cm} (51)
A plot of the CLFs developed from finite difference determined stress vs. deflection data is shown in appendix L with the fifth order polynomial curve fit for each CLF. A measure used to check the accuracy is to solve for the linearized plate strip equation analytically and ensure concurrence for small deflections. The results for bending stress show the following relationship between center deflection and mid-span and clamped end bending stresses:

\[ \sigma_b(\text{end}) = 192 \tilde{W} \quad \text{and} \quad \sigma_b(\text{mid-span}) = 96 \tilde{W} \] (52)

To check for the accuracy of \( F_m \), the membrane strain in the linearized solution is determined by numerical integration. The second order approximation to strain as shown in (12) and (13) is applied to the linear solution so as \( W \) goes very small, the nonlinear effects disappear. The membrane strain is determined to be

\[ \varepsilon_o = \left[ \frac{\tilde{W}}{L} \right]^2 \times 2.4381 \] (53)

from which the dimensionless membrane stress is

\[ \tilde{\sigma}_o = \tilde{W}^2 \times 12 \times 2.4381 \] (54)

Using the stresses in (52) and (54), the correct CLFs can be determined for small deflections by (49) and (51). Comparison of results between the linear exact solution and the curve fit functions for the finite difference solution (appendix L) can be made. When center deflection is zero, the comparison of results is shown in Table 3.
<table>
<thead>
<tr>
<th>CLF</th>
<th>Exact Linear</th>
<th>Finite Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Bending</td>
<td>.11375</td>
<td>.114</td>
</tr>
<tr>
<td>Mid Bending</td>
<td>.2275</td>
<td>.228</td>
</tr>
<tr>
<td>Membrane</td>
<td>.74648</td>
<td>.746</td>
</tr>
</tbody>
</table>

Table 3. Comparison of CLF at $W = 0$ between Exact Solution and Finite Difference Solution.

This shows an excellent agreement between the exact and finite difference solutions for CLF. This result validates the accuracy of the for CLF and stress vs. $W/h$ while in the elastic range.

With CLF now defined over a range of $W/h$, (50) gives the membrane strain as a function of center deflection.

The bending strain in any element can now be defined as well.

$$
\varepsilon_b = \frac{4Wz}{F_b L^2}
$$

This applies to both end and mid-span bending strain. Adding the results of (50) and (55) give the total strain in any element.

A model has been produced which yields the strain at any point in the hinges based on center deflection. By design, it represents the stresses in the actual strip under a uniform static load in the elastic range with the same center deflection. It is important to note that the formulation of the model is based on elastic analysis. Note that at incipient yield, the model gives the exact solution to load required for yield; by definition.
Applying this model into the range of plastic deformation is considered to be a reasonable assumption for moderate plasticity. When the clamped end of the actual strip goes into plasticity, the actual local strain concentration increases. The model does not represent this increase in local curvature though the model does more closely represent the plasticity-limited stress. This supports the upper bound theorem. In the static case, for a given center deflection, the model can sustain more load than the actual strip. This translates to a stiffer model. The further into plasticity the hinges go, the less accurate is the model. The development of a fully plastic travelling hinge is not expected since the end hinges sustain only moderate plasticity in this study. Additionally, there is some error in assuming a shape of static response. This ignores the harmonics in the plate strip which can induce higher oscillating strains. However, plasticity would damp these oscillations. Therefore, this model is considered to be limited to moderate plasticity and deflections. Altogether, since deflections which only yield mild plasticity are of interest, this model should be a good engineering tool.

8.3. DYNAMICS OF THE ELASTIC-PLASTIC MODEL.

8.3.1. Equation of Motion.

Due to symmetry, only one rigid half strip need be considered. Since the clamped end is fixed, the rotation of
the mass about the end is considered where the sum of the moments acting on the half strip equal its angular acceleration, \( \alpha \), times the rotational mass moment of inertia, \( J \).

\[
\Sigma M = J \alpha
\]  
(56)

The sum of the moments acting about the clamped end of the half strip include bending moments at the ends of the rigid section, membrane force times its moment arm, \( W \), and the moment due to the applied load. These are shown in figure 15.

The mass moment of inertia about the hinge is found to be

\[
J = \frac{\text{mass} (L^2 + h^2)}{12} \text{ (lb-in}^2\text{)}
\]  
(57)

Neglecting the effect of thickness, since \( h^2 \ll L^2 \), and approximating the angle formed by the deflected half strip and the horizontal plane by \( \frac{W}{L/2} \), (56) can be applied to yield the governing equation of motion.

\[
\frac{\partial^2 W}{\partial t^2} = \frac{P(t) L^2}{8} - SW - M_{\text{mid}} - M_{\text{end}}
\]

\[
- \frac{\rho h (L^2)}{12}
\]  
(58)

The moments and stretching force are determined from the stresses in the hinge elements. Stretching force is simply the membrane stress multiplied by the thickness (assuming unit width). The moment at a hinge is determined by summing the element forces times moment arm as follows:
max el. # 
\[ M = \sum_{i=0}^{\text{max el. #}} (\sigma_i \cdot z_i \cdot h_i) \]  \hspace{1cm} (59)

where \( \sigma_i, z_i \) and \( h_i \) are the stress, distance from hinge center to element center, and thickness for each element.

A Central Difference method, as shown in (30), is used in time stepping for evaluating the response of this model. The general formula used for to determine center deflection at the next time step \( t^{k+1} \) is as follows:

\[
W^{k+1} = \Delta t^2 \left[ \frac{P(t) L^2}{8} - SW - M_{\text{mid}} - M_{\text{end}} \right] + 2W^k - W^{k-1} - \frac{\rho h L^2}{12} \]  \hspace{1cm} (60)

8.3.2. Tracking Stress In Each Element

Equations (50) and (55) are used to determine the membrane and stretching strain in each element. These are combined to yield the total strain. From the total strain at each time step, the stresses can be determined.

Since stress in plasticity is dependant on the history of the stress, it must be tracked incrementally. To determine stress at the next time step \( t^{k+1} \), the strain increment, \( \delta \varepsilon_i \), in each element must be known. This is determined from \( W^{k+1} \) and \( W^k \). The stress increment, \( \delta \sigma_i \), is then found by multiplying \( \delta \varepsilon_i \cdot E \) and added to the current stress. Special steps must be taken to account for yield strength and strain hardening since they characterize the limits of stress values based on strain. For example, if
some increment in element strain increases the stress beyond
tensile yield, it must be limited to that yield strength
specified as a function of strain. This is illustrated in
figure 16. This applies similarly for compressive yield.

This method of determining stresses is used in
evaluating the E-P model response.

8.3.3. E-P Model Response Algorithm

The program written in "C" language, PLASTIC.C, for the
dynamic response of the E-P model is shown appendix M. It is
written so that yield strength and ideal kinematic hardening
can be accounted for. The degree of kinematic hardening is
specified by the plastic tangent modulus, EP. Therefore,
when each element is between tensile and compressive yield it
will act perfectly elastic. When in yield, it will stay on
the idealized yield line until the strain increment reverses
direction back into elastic behavior. Figure 17 shows
typical yield limits on stress vs. strain coordinates with
kinematic hardening specified by the plastic tangent modulus.

Both tensile and compressive yield strength are
considered to be the same magnitude. This idealization of
yield behavior enables the tensile and compressive yield
lines to be defined mathematically as a function of strain.

With the behavior of each element characterized, the
equation of motion can be applied for each time step.
Equation (60) gives \( W \) at time step \( k+1 \), \( W_{k+1} \). Based on \( W_{k+1} \)
and \( W_k \), the strain increment can then be determined by
applying each to the sum of bending and membrane strain; (50) and (55). For each element the increment in stress is determined, as shown above, from the elastic modulus, $E$, plastic tangent modulus, $EP$, and yield strength, $\sigma_y$. When the stresses are known for each element at time $k+1$, the stretching force and moments in each hinge are determined as shown above in (59) from which $W_{k+2}$ can be determined by (60). This is the time stepping process which is used for E-P model response.

Since permanent deflection of the plate is of interest, some means of determining this must be devised. After plastic deformation ceases, the model will simply continue to oscillate freely under no load since the motion is elastic with no dissipative mechanism. The method chosen to determine final deflection is based on what would occur naturally. With no applied load, the internal forces must be in equilibrium for no motion to occur. Using a Newton type root finding method, the applied load is released and a series of changes in center deflection are imposed such that convergence to the point where the sum of the moments in the plate strip equals zero. This would be the static permanent deflection since, at this point, the internal forces are equilibrated with no load applied. This routine is incorporated into the program PLASTIC.C.

One benefit of using this routine is being able to evaluate static response. The convergence routine is implemented immediately such that the internal moments due to
hinge bending and stretching are in equilibrium with the moment imposed by the applied load.

8.4. E-P MODEL VALIDATION

8.4.1. Static Response Evaluation

To determine the accuracy and applicability of the E-P model it is necessary to evaluate both its static and dynamic response for qualitative and quantitative accuracy. Appendix N compares the static response of the E-P model to the finite difference model under uniform loads. Note that when yield strength is infinite, the finite difference and E-P models match perfectly, as they should.

For finite yield strength, perfectly plastic behavior is assumed at the yield strength specified.

8.4.2. Time Step and Number of Elements for Convergence

The explicit central difference method is used in time stepping and the time increment is based on the natural frequency of the model. The program PLASTIC.C defines the time step to be approximately 1/500 of the natural period. The deflection magnitude error for free vibration with a time step of 1/500 of the natural period is less than .01 percent [22]. This falls well within the range of accuracy needed for this analysis.

Yankelevsky [19] suggests that 10 elements per hinge is adequate to determine deflections in his model and 30 is very accurate for stresses. For all analyses here, 60 elements
per hinge are used. Appendix 0 illustrates the E-P model response to a strong pulse with various numbers of hinge elements. It can be seen that 60 elements is more than adequate to achieve convergence.

8.4.3. Comparison to Finite Difference Model in Elastic Range

Within the elastic range, the E-P model dynamics can be defined exactly by the governing differential equation of motion. By applying the dimensionless parameters from elastic analysis, it can be shown the governing equation of elastic motion is as follows:

\[ \frac{\partial^2 \tilde{W}}{\partial t^2} + \alpha \tilde{W} + \chi \tilde{W}^3 = \frac{3}{2} \tilde{P} \]  \hfill (61)

where the parameters \( \alpha \) and \( \chi \) are:

\[ \chi = \frac{288(1-v^2)}{F_m} \quad \alpha = 48(1-v^2)\left[\frac{1}{F_{b,\text{end}}} + \frac{1}{F_{b,\text{end}}}\right] \]  \hfill (62)

The governing equation has the form of a hardening spring which would be expected due to membrane stretching. The dimensionless frequency of natural vibration for small deflections can be easily determined from \( \alpha \). Assuming the values of the end and mid-span bending CLFs at \( W=0 \), the dimensionless natural frequency of the E-P model is exactly 24. The dimensionless natural frequency of the linearized plate strip was found by solving the governing equation of motion, (21), with no load or stretching term. Using finite differences to solve this as a classical eigenvalue problem,
80 nodes were used for the plate strip. The result yielded a natural dimensionless frequency of 22.35. Note that this is within roundoff error of the finite difference model frequency response. Compared with the E-P model, the 7% difference is due to the increased stiffness of the E-P model.

It is prudent to compare the response of the E-P model to that of the finite difference model under yield pulses in the elastic range. This is done with 3 plate strips having length to thickness ratios, R, of 40, 70 and 100 which spans the range of applicability for hull plating. For each plate strip, a yield pulse with high magnitude and low magnitude is applied. Appendix P shows the comparison between the two models' stress and center deflection responses to these pulses. Note that for R = 100 the variation from smooth response is greater due to the decreasing stiffness. It was suggested previously that the static response shape of the uniform load is between the extremes of dynamic response. As shown by the stress responses, the E-P model gives a good mean response which is the best to be hoped for in a one degree of freedom model.

8.4.4. Analysis of Sample Elastic-Plastic Responses

To ensure the program operates well in the dynamic plastic range, the details of response must be analyzed. The approach taken for analysis is to apply multiples of some yield pulse magnitude for the plate strip from elastic
analysis with the time decay held constant. As in elastic analysis, the assumption of a spatially constant pulse is made. Appendix Q contains detailed response data for a plate strip with R = 60, yield strength = 40000 psi and no strain hardening. This strip is subjected to multiples of yield pulse $106 \exp(-t/0.008)$ psi with the decay constant 0.008 sec. unchanged. That is, the magnitude, 106, is multiplied by 2,3,4,5, and 6.

From the results, there are many points worth noting. The amplitude of vibration decreases with time which indicates a non-conservative force is present. This, of course, is the dissipation of energy by plastic deformation. Note that the frequency of vibration increases at higher magnitudes of deflection due to the effect of membrane force. The permanent deflections determined by convergence to internal equilibrium are shown in Table 4.

<table>
<thead>
<tr>
<th>Yield Pulse Multiple</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Final Dimensionless Deflection</td>
<td>.006699</td>
<td>.065687</td>
<td>.264133</td>
<td>.579371</td>
<td>.888501</td>
</tr>
</tbody>
</table>

Table 4. Final Permanent Deflections of Sample Plate Strip Subject to Multiples of a Yield Pulse.

The stress-strain histories and stress versus time plots for the top element in the end hinge are also shown in appendix Q. These are shown for responses to 2,4, and 6 times the yield pulse. The top element was chosen since it
undergoes the most strain of all hinge elements. For clarity, only a portion of the response indicated by the deflection vs. time plot is shown. As defined in the program, the element has elastic perfectly-plastic behavior. The loops formed in the stress-strain histories continue to close with time as plastic energy dissipates.

Also in appendix Q are 3-D plots of all end hinge elements’ stresses vs. time for 2, 4, and 6 times the yield pulse. This clearly shows the increased onset of residual stresses with increasing pulse magnitudes. Note that during an increase in amplitude, the elements go into yield one after the next, but immediately upon the change in direction of motion, all elements in yield return to elasticity simultaneously. This is precisely what should occur in plastic vibration.

Finally, the response to 4 times the yield pulse is shown where the elements have a plastic tangent modulus of 3*e+6. As expected there is an increasing slope in plasticity due to the plastic tangent modulus.

These results, along with the elastic dynamic and static analysis of this model, give very good indications of its validity and accuracy. With only one degree of freedom, there is only one geometric form of permanent deflections as opposed to the infinite possibilities with the continuous plate strip. For the E-P model, however, it has been shown [18] that permanent plate deflections assume a roof top hinge line configuration as shown in figure 18.
With rigid-plastic analysis and applying the upper bound theorem, it can be shown that the minimum upper bound to plasticity induced by a uniform load is the roof top configuration with a single line in the middle. Hence, the E-P model with hinges at the ends and in the middle serves as an excellent kinematic model.

Based on this support, the E-P model is considered to be a valid tool for elastic-plastic analysis where plasticity does not predominate.

8.5. EFFECT OF ADDED MASS ON RESPONSE

One would think that the added mass of water would greatly suppress the vibration of the plate. By definition, the added mass of a body accelerating in an ideal inviscid fluid results from the resisting force which is proportional to the body's acceleration. That is to say, acceleration is the causal influence and the resisting force is the effect. It is considered added mass because the force is proportional to acceleration. In the E-P model, the force is prescribed. Therefore added mass is not a consideration since force on the plate would be changed by added mass. However, since the force is prescribed, added mass should not be considered until the prescribed force is negligible.

It turns out that the pressure-time history of the slamming pulses in [6] shows negligible oscillations which result from added mass of plate acceleration. The effect of added mass can be neglected during the period when the
applied force is substantial; time \( t < 4 \tau_0 \). It can also be shown that, typically, the velocity of the plate during vibration is small relative to the velocity of the hull as it slams the ocean surface. Hence the additional velocity of vibration has little effect on the applied force from slamming.

Consider a point in the plate strip response when the applied load is near zero and the plate strip is at a deflection peak in its motion where velocity is zero. Added mass is treated by simply adding mass to the body. In this case, the effect is increasing \( J \), the mass moment of inertia.

At this peak where velocity is zero and applied load is negligible, if the mass were suddenly increased, the only effect would be a decreased frequency of vibration. This is true even when there is still plastic energy to be dissipated when at this starting peak. Consider that at the peak there is a set amount of strain energy about to be changed into kinetic energy. For some subsequent segment of displacement, the plate strip will be behaving elastically which implies the total energy is contained in the strain and kinetic energy. The strain energy will be strictly a function of center deflection. This is a result of a single degree of freedom model. Therefore, while in elasticity, the kinetic energy must be a function of center deflection. So up to the point where plastic energy starts to dissipate, the kinetic and the strain energy will be determined by the center deflection, \( W \), and not the mass. Similarly, the dissipation
of plastic energy is strictly a function of $W$ so kinetic energy must also be a function of center deflection; not mass. Therefore, kinetic energy will go to zero at the same point regardless of the magnitude of the mass moment of inertia. This implies, the next stationary point in deflection will be dependant only on deflection. The result of this deflection dependency is simply that mass will only slow the process down and not change the final result. An example of this is shown in appendix R. Notice that the deflection peaks corresponding to the two responses have the same magnitude as explained above.

There is, however some mechanism which eventually causes the cessation of vibration. It is assumed this is attributed to viscous effects of water and is neglected. Viscous forces act tangentially to the plate and have a small effect on the dynamics. There is also viscous dissipation of energy in the fluid medium around the plate since the fluid is not ideal. Finally, energy is lost due to the production of nonrecoverable pressure waves from vibration.

8.6. FURTHER VALIDATION OF THE NEED FOR ELASTIC-PLASTIC VS. PLASTIC ANALYSIS.

As stated earlier, it has been shown that for responses where plastic deformation is of the same order as the elastic deformation, the assumption of rigid-plasticity may give large inaccuracies and elastic-plastic analysis is necessary [9]. This can be verified using the E-P model by specifying
an elastic modulus that is much higher than that used for steel, $30\times 10^6$ psi. This allows the material behavior to approach rigid-perfectly plastic behavior and still be tractable for the E-P model program, PLASTIC.C. As the elastic modulus, $E$, grows, the maximum elastic energy potential decreases if the yield strength stays the same. As $E$ approaches infinity, the elastic potential goes to zero which is the case in rigid-plastic behavior. Therefore, increasing $E$ simulates the behavior which approaches rigid-plasticity. Appendix S shows an example of the response of an E-P model to the same load with various values of the elastic modulus, $E$. It can be seen that as $E$ goes much larger than that for steel, the error in permanent deflection is nearly 90%. This is a very strong argument for the need of elastic-plastic analysis for this range and type of applied pulse.

8.7. RUNNING THE E-P MODEL PROGRAM FOR RESULTS.

The extent of plate strip permanent deflection can be determined for any given pulse by using the E-P model for elastic-plastic analysis. From elastic analysis, the combinations of magnitudes and time decay constants to cause first yield are known. These results are extended to determine the response of the plate to multiples of all these yield pulses. This applies to a variety of plate length to thickness ratios, $R$. Using the yield pulse as a basis is a
natural extension of the elastic analysis. It also provides a practical basis for comparison and trend analysis.

The slamming pulse is assumed to be spatially constant as is the case in the elastic analysis. The same restriction applies where the dimensionless spatial decay constant, $c_p\tau_{ut}/L$, of the pulse should be greater than 1 to restrict inaccuracies in this approximation. A 2-dimensional hydrodynamic analysis of wedge slamming by Wagner [20] shows that the pressure at the pulse front is proportional to the square of the pulse front velocity. It is expressed in the following equation:

$$P(0) = \frac{1}{2} \rho (c_p)^2 \frac{\pi^2}{4(\tan \theta)^2}$$

(63)

where $\theta$ is angle between the hull and water surface. This suggests that as pulse magnitude pressure increases, the spatial decay constant, $c_p\tau_{ut}$, will increase even greater. This supports the spatially constant pressure approximation for stronger pulses.

In order to stay within the limits of the moderate plasticity, the multiples of the yield pulse to be applied are 2,3,4,5, and 6 with the time decay, $\tau_{ut}$, held constant. Six was found to be a limit on maximum response and permanent deflections where plastic and elastic deformations are of the same order. Additionally, the range of 5 different multiples allows for interpolation in trend and specific performance analysis.
Although the program of elastic-plastic analysis, PLASTIC.C, is able to accommodate the plastic tangent modulus, it is assumed to be zero as a base line performance criterion.

Two yield strengths were assumed: 40000 and 60000 psi. This spans the range of the vast majority of materials used in ship production. Also, the effect of yield strength on permanent deflection can be determined.

The dimensionless parameters used in the elastic analysis also apply to elastic-plastic responses. A result can be generalized based on these parameters provided the yield strength and plastic tangent modulus (in this case 0) are the same. Accordingly, only one program run for a given $R=\frac{L}{h}$ and pulse is necessary. That is, for a given dimensionless final deflection, $\frac{W_f}{h}$, the trial $P_o$ and $\frac{\tau_o}{L}$ apply to all plates with the same $R$.

The final results from the E-P model give permanent deflections from applying multiples of the yield pulse. These results are shown in graphical form in appendix T. A sample is shown at the beginning of appendix T to facilitate interpretation of the graphical results. The results of the elastic analysis gives combinations of pulse initial magnitude, $P_o$, and $\frac{\tau_o}{L}$, which describe yield pulses for a given length, $L$, and length to thickness ratio, $R$, at yield strengths of 40000 and 60000 psi. The final results from the E-P model in appendix T have the yield pulse $P_o$ on the abcissa. The associated $\frac{\tau_o}{L}$ for every point, $P_o$, on the
abcissa is determined from the elastic analysis results for the applicable $R=L/h$ and yield strength, $\sigma_y$. That is, the yield pulse is given on the abcissa and is characterized by $P_o$ of the yield pulse. The E-P model results are permanent deflections from multiples of the yield pulse indicated on the abscissa.

9. ANALYSIS OF E-P MODEL RESULTS

9.1. PEAK PERMANENT DEFLECTIONS

The multiples of yield pulse were applied to the same range of plate dimensions and yield strength as in the elastic analysis. Length to thickness ratios vary from 30 to 120 and the yield strengths assumed are 40000 and 60000 psi.

A phenomenon immediately noticeable in the results of appendix T are local peaks in the response lines of constant yield pulse multiples. Note that these peaks occur at smaller yield pulse magnitudes as the length to thickness ratio increases. Also the peaks occur at higher magnitudes for models with yield strength of 60000 psi than 40000 psi for a given length to thickness ratio. For more insight into the phenomenon, the detailed response is considered around a local peak. The distinct peak considered is on the response line for 6 times the yield pulse for $R = 60$ and yield strength = 60000 psi. This is shown as the sample graph at the beginning of appendix T. The applied pulse for this peak is $254*\exp(-t/.06)$. Pulses with responses on both sides of this peak are $318*\exp(-t/.049)$ and $178*\exp(-t/.108)$. For
these three pulses, the pulse shape, center deflection vs.
time and external work vs. time are shown in appendix U.
Those pulses causing peak permanent deflections will be
referred to as "peak pulses." Note the peak pulse external
energy input is clearly higher than those for adjacent
pulses. Also the corresponding center deflection magnitude
is greater, as expected, since it causes a peak deflection.

The total momentum of a pulse of the form $P_0 \exp(-t/t_{au})$
can be defined as the integral with respect to time
from $t_o$ to infinity. This is simply equal to $P_0*t_{au}$. For
the three pulses considered above, the total momentum for
each in units of lbf*sec is: 15.24 for peak pulse, 15.582 for
the higher magnitude pulse, and 19.224 for the lower
magnitude pulse. With the lowest momentum value for the peak
pulse, there is no correlation between the momentum of the
pulse and peak permanent deflection. Indeed, this is
contrary to expectations.

Another approach to understanding this phenomenon is to
consider the frequency content of the applied pulse by taking
its Fourier transform. There is an analytical solution to
the continuous Fourier transform of the form of applied pulse
assumed here. This is shown graphically in appendix V. It
was found that for all values of $R$ which show peak permanent
deflections, the ratio of model's natural frequency to
$(1/t_{au})$ of that model's peak pulse is $1.3 \leq \omega_n * t_{au} \leq 1.45$.
This is a narrow range with respect to the range of
significant frequency components. It clearly demonstrates
the relation between peak pulse frequency content and the natural frequency of the model. Note there is a maximum in the sine coefficient of the pulse at \( \omega/(1/\tau_{ut}) = 1 \). In this area, the sine components are in phase with the motion of the model during its initial rise.

To support the relationship of peak pulse frequency content to model natural frequency, consider the results from appendix T of final deflection vs. multiples of yield pulse for \( R = 100 \) and yield strength = 40000 psi. Note that as the multiple increases, the permanent deflection peak shifts to the right. This corresponds to decreasing \( \tau_{ut} \) of the yield pulse which, by the above analysis, relates to a higher model natural frequency. Although this is the same model, recall, that the frequency of natural vibration increases with greater deflections in the plastic range due to membrane stresses. This can be seen in center deflection vs. time for multiples of a yield pulse in appendix Q. This insight verifies the relationship between pulse frequency content and model natural frequency.

9.2. COUNTER-INTUITIVE RESULTS.

One unusual permanent deflection result occurred for length to thickness ratio of 120. For yield strength of 60000 psi, note that there is a range between \( P_o = 20 \) and 45 where 3 times the yield pulse gives less final deflection then 2 times the yield pulse. While this is counter intuitive, it has been shown [21] that certain pulses cause
negative permanent deflections. Though this is not the case here, it demonstrates highly counter-intuitive behavior is possible in final deflection and that the E-P model is approaching the range where this can occur.

9.3. BUCKLING CONSIDERATIONS

The possibility of buckling was not considered in this analysis. It could possibly occur when considerable membrane plastic strain is induced during the initial pulse response. This permanent membrane strain can cause compression in the plate strip during vibration since the strip length enlarged while the supports remain fixed. It can be shown that the dimensionless compressive membrane force necessary to induce the first mode of buckling is $4 \pi^2$ or approximately 39.5. This value of dimensionless membrane force was approached with the higher multiples of yield pulse for $R = 80$ thru 120.

It is possible that elastic buckling would induce greater permanent deflections. The compressive membrane force is limited by the buckling load. Therefore, compressive membrane plasticity cannot occur if the buckling membrane stress is less than the compressive yield strength. Hence, only tensile membrane plasticity is possible which could cause greater permanent deflections.

Another possibility is that since compressive membrane plasticity may not occur, the bending moment and hence,
bending plasticity compensate to absorb the energy otherwise consumed by compressive plasticity.

Obviously, the effect of buckling needs much more consideration. Though it is within the scope of structural response to slamming, it is not addressed in this analysis.

10. CONCLUSIONS

10.1. NEED FOR MODEL VALIDATION.

This treatment of response to slamming induced load should be validated from detailed pulse pressure and deflection data obtained on the plating of an ocean going ship; particularly when permanent deflections are induced. The extent of varying boundary conditions and their effects should be measured as well.

10.2. EVALUATION.

This elastic and elastic-plastic analysis offers detailed insight to the structural response to slamming induced loads. The plate strip is considered to be an adequate representation for typical hull plating. This deviates from accuracy for smaller plate length to width ratios and larger deflections where longitudinal bending and membrane effects essentially strengthen the strip. To this extent, the plate strip is a conservative model. Conversely, the E-P model gives a static upper bound to the load in plasticity and may therefore under-predict final deflections.
This potential is offset by the effects of finite aspect ratio as stated above.

The finite difference elastic analysis provides accurate results under the assumed boundary conditions and to the extent the mathematical model represents actual behavior. Since the plate is typically much longer than thicker, ignoring the shear deformation and rotational inertia, which is accounted for in Bernoulli beam theory, is warranted.

Minor modifications can be made to the program, NONDIM.FULL.C, to account for boundary rotation, and in plane deflection. Also, the program is designed for elastic supports. A study to determine the effects of each of these on yield pulse would make a more complete analysis at little cost. However, each of these effects are specific to the ship and are assumed to have a relatively minor effect on the deflections. Therefore, in practice, the yield pulse charts can serve as a good tool for the designer.

For a more complete structural analysis, low cycle fatigue at the boundaries should be investigated. The finite difference program, NONDIM.C, can give detailed and accurate stress histories at the edge of the plate to facilitate fatigue analysis. The E-P model may used for cyclic plasticity though it underpredicts the extent of plastic strain deformation since it is based on elastic behavior. Hence, the E-P model is not recommended for use in fatigue analysis.
The effect of dynamic buckling appears to be the limiting factor for the results and model applicability. The buckling load was based on a beam behavior. This applies well to plates of high length to width ratios. However, the two dimensional effects and the deformed plate would strengthen the resistance to buckling to some degree. Further research in this area is necessary to understand its applicability to results for plates with length to thickness ratios, \( R \), of 80 or greater.

The E-P model is an improvement on Yankelevsky’s beam model and an extension of the plate strip concept to larger deflections. Basing the strain in the strip on nonlinear elastic analysis was therefore necessary. This results in a model which is overly stiff in plasticity as explained previously. However, this effect is compensated for by longitudinal bending and membrane effects at moderate deflections which are neglected. Therefore, the permanent deflections from multiples of yield pulse are considered be a reasonably accurate tool for the design of pulse loaded plates.

11. REFERENCES.


LIST OF FIGURES
1. Slamming Pressure Pulse Representation
2. Space and Time Domain Pulse Representation
3. Plate Boundary Rotation from Pulse Loading
4. Moment/Stress Comparison between Plates with Finite and Infinite Aspect Ratio under Uniform Load.
5. Plate Strip in relation to Plate
6. Plate Strip Model
7. Differential Element of Plate Strip
8. Shape Function for Galerkin Solution
9. Comparison of Finite Difference Model and Galerkin Solution Deflected Shapes due to Static Load
10. 3-D Plot of Plate Strip Response to Impulse at 1/3 Strip Length
11. 3-D Plot of Half Strip. Response to Impulse at Center Node. Time Range 0-.005.
12. 3-D Plot of Half Strip. Response to Impulse at Center Node. Time Range 0-.0005.
13. Elastic-Plastic Plate Strip Model
14. Hinge Element Displacement Decomposition
15. Moments and Forces acting on E-P Model
16. Illustration of Yield Stress Limitation on E-P model element stress
17. Illustration of Yield lines with Kinematic Hardening.
18. Roof Top Hinge Line Configuration
Figure 1 Slamming Pressure Pulse representation.

Figure 2. Space and Time Representation of Pulse with 
Velocity = 10 in/sec, Tau_t = .2 sec & Po = 1 psi.
Figure 3. Plate Boundary Rotation with Traveling Pulse.

Figure 4. Ratio of moments of infinitely long plate to plate with finite aspect ratio at long side midpoint under uniform load.
Figure 5. Plate Strip Representation

Figure 6. Plate Strip Model of Hull Plate

Figure 7. Differential Element of Plate Strip
Figure 8. Shape Function for Galerkin Approximate Solution.

\[ w_3(x,t) = \sin^2(\pi x/L) \sin(\omega t) \]

Figure 9. Shape of Half-Strip. Static Response to dimensionless load 227 which causes 0.5 deflection, W/h.
Shape of Half Strip vs. Time in Response to
Initial Condition of only Center node having
Dimensionless Velocity of .01. No Load Applied

Figure 11
Shape of Half-Strip vs. Time. Response to initial condition of only center node having Dimensionless velocity of 0.1. No Load Applied.
Figure 13. Elastic-Plastic Plate Strip Model

Total Hinge Element Displacement = Bending Component + Stretching Component

Figure 14. Hinge Element Displacement Decomposition
Figure 15. Moments about the clamped end for the E-P Model

Figure 16. Illustration of Yield Strength Limitation on E-P Model Element Stress
Figure 17. Illustration of Yield lines with Kinematic Hardening

Figure 18. Roof Top Hinge Line Configuration.
Appendix A. Sample of Recorded Slamming Pulse Pressure vs. Time.
Recorded Slamming Pulse Pressure-Time History from Slamming of V-shaped Hull Section [6].

V-Shaped hull section used in experiments in [6]
Appendix B. Computer Program NONDIM.C. Response to Spatially Constant-Time Varying Loads

Plate Strip Elastic Response to Spatially Constant, Time varying loads using Finite Difference Techniques.
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/* This program evaluates the Dimensionless response of the plate strip with clamped ends which undergoes spatially constant, time varying loading. The supports are modeled as linear elastic springs which allow only lateral deflection. There is no inplane deflection and no rotation allowed by the support. An implicit finite-difference scheme is used which is unconditionally stable. Due to symmetry, only half of the plate strip in considered in this evaluation.

There are essentially three concentric loops which comprise the algorithm of this program. The outer loop varies the Magnitude of the exponentially decaying pulse. The next loop inward varies the time decay constant of the pulse. The final inner loop cycles through the time integration under the load specified by the outer loop.

The deflection, w, has a nodal or space coordinate (j) and a time coordinate (k). Hence, the deflection at the jth node and at the kth time step is referred to as W(j)(k).

Since the numerical scheme is implicit, the finite difference equations must be solved to yield the kth+1 time step. This yields a pentadiagonal matrix which is solved for every step.

This program also allows for initial velocity and deflection to be specified. This must be done by changing the function: init_v0 and inti_w0 after the main().

The '0' th node is the clamped end node and the node numbered 'nodemax' corresponds to the center of the plate strip.

There are three output files. nd.s.data, nd.w.data, and nd.shape.data. These are maximum stress, maximum deflection and stress deflection history. Concerning the latter program; either stress or deflection output can be specified as described in main(). */

/** V A R I A B L E   D E C L A R A T I O N S **/

    double delt;    /* time step */
    double tim=0.0;    /* cumulative time */
    double tmax;    /* Duration of propagation time */
    double delx;    /* spacing between spatial nodes */
    double sO;    /* stretching force for current time */
    double pO;    /* current force magnitude */
    double kk;    /* spring constant of elastic support */
    double stressm;    /* maximum stress for current time */
    double histress;    /* max stress for all time */
    double hitime;    /* time of maximum global stress */
    double stress0;    /* stress at end of strip */
    double stress5;    /* stress at mid-span of strip */
    double mag;    /* force at time 0 */
    double tau;    /* tau in exponential decay */
    double wmax=0.0;    /* Max W(mid-strip) for duration of pulse propagation */
    double wmaxtim=0.0;    /* time of W(mid-strip) max */
    int stressloc;    /* location of maximum stress */
int nodemax; /* node number of last node*/
int i,j,k=0; /* dummy variable used in loops*/
int plotflag; /* Indicator for printing 3-D plot data, if desired */
int colnum=100; /* Number of columns in the data matrix which will hold stress or deflection data */
double *wk9; /* displacement array for previous time step */
double *wk0; /* displacement array of current time step */
double *wk1; /* displacement array of k+1 time(unknown) */
double *v0; /* velocity array at time = 0 (I.C.) */
double *load; /* load vector in solving for W(j)(k+1) */
double *ubar; /* used in pentadiagonal decomp as dummy variable*/
double *zbar; /* used in pentadiagonal decomp as dummy variable */
double *slopes; /* vector for slope at each node */
double **a; /* pentadiagonal matrix to solve for W(j)(k+1) */
double **up; /* upper triangular in pentadiagonal decomp */
double **low; /* lower triangular in pentadiagonal decomp */
double **yofxt; /* data matrix of nodes disp vs time */

/** FUNCTION PROTOTYPES **/ 
double* calocl(int);
/* allocate memory for a vector */
double** caloc2(int, /* total number of rows in matrix */
int); /* total number of columns in a matrix */
/* calculate stretching force */
double calc_s (double *wk0, /* vector of displacements */
  double *slopes,
  double delx, /* 'x' spacing */
  int nodemax); /* node number of last node at mid-strip*/
/* initialize displacement vector for time = 0 */
void init_w0 (double *wk0, /* displacement vector at time=0 */
  int nodemax); /* node number of last node at mid-strip */
/* initialize velocity vector for time = 0 */
void init_v0 (double *v0, /* velocity vector at time=0 */
  int nodemax); /* node number of last node at mid-strip */
/* Applied force function*/
double pforce (double time, /* time */
  double mag, /* Pulse Magnitude at time=0 */
  double tau); /* Pulse time decay const. */
/* Form the "A" pentadiagonal matrix */
void make_a (double **a, /* 'a' matrix */
  double delx, /* 'x' spacing */
  double delt, /* time spacing */
  double s0, /* stretch force */
  double p0, /* applied load */
  double kk, /* elastic support stiffness coefficient */
  int nodemax); /* node number of last node (mid-strip) */
/* Form Load Vector for the first time step */
void loadvec0 (double *load, /* Load vector */
  double *wk0, /* displacement vector */
  /* allocate memory for a vector */
  /* total number of rows in matrix */
  /* total number of columns in a matrix */
  /* calculate stretching force */
  /* vector of displacements */
  /* 'x' spacing */
  /* node number of last node at mid-strip*/
  /* displacement vector at time=0 */
  /* node number of last node at mid-strip */
  /* velocity vector at time=0 */
  /* node number of last node at mid-strip */
  /* Pulse Magnitude at time=0 */
  /* Pulse time decay const. */
  /* "A" pentadiagonal matrix */
  /* 'a' matrix */
  /* 'x' spacing */
  /* time spacing */
  /* stretch force */
  /* applied load */
  /* elastic support stiffness coefficient */
  /* node number of last node (mid-strip) */
  /* Load vector */
  /* displacement vector */
  /* Allocate memory for a vector */
  /* Total number of rows in matrix */
  /* Total number of columns in a matrix */
  /* Calculate stretching force */
  /* Vector of displacements */
  /* 'x' spacing */
  /* Node number of last node at mid-strip */
  /* Displacement vector at time=0 */
  /* Node number of last node at mid-strip */
  /* Velocity vector at time=0 */
  /* Node number of last node at mid-strip */
  /* Applied force function */
  /* Time */
  /* Pulse Magnitude at time=0 */
  /* Pulse time decay constant */
  /* "A" pentadiagonal matrix */
  /* 'a' matrix */
  /* 'x' spacing */
  /* Time spacing */
  /* Stretch force */
  /* Applied load */
  /* Elastic support stiffness coefficient */
  /* Node number of last node (mid-strip) */
  /* Load vector */
  /* Displacement vector */
double *v0, /* velocity vector */
double delx, /* x spacing */
double delt, /* time spacing */
double s0, /* stretch force */
double p0, /* applied load */
double kk, /* elastic support stiffness coefficient */
int nodemax); /* node number of last node (mid-
strip)*/

/* Form Load Vector for the Kth time step */
void loadvec (double *load, /* Load vector */
double *wk9, /* previous disp. vector */
double *wk0, /* current disp. vector */
double delx, /* x spacing */
double delt, /* time spacing */
double s0, /* stretch force */
double p0, /* applied load */
double kk, /* elastic support stiffness coefficient */
int nodemax); /* node number of last node (mid-
strip)*/

/* Solve for W(j)(k+1) with pentadiagonal matrix */
void pentasolve(double** a, /* Pentadiagonal matrix */
double *wk1, /* unknown vector; W(j)(k+1) */
double *load, /* load vector for matrix eqn. */
int nodemax, /* node number of last node */
double **up, /* upper triangular in matrix reduction */
double **low, /* lower triangular in matrix reduction */
double *ubar, /* dummy variable used in reduction */
double *zbar); /* dummy variable used in reduction */
/* Find maximum stress in Plate Strip */
void findstress(double *wk0, /* vector of displacements */
double s0, /* Stretching force */
double delx, /* strip node spacing */
int nodemax, /* Node number of max node */
double *stressm, /* maximum stress */
int *stressloc); /* location of max stress */

FILE *fp;
FILE *fp1;
FILE *fp2;

main()
{
    /* START MAIN */

    fp=fopen("/mit/djhenke/nd.s.data","a");
    fp1=fopen("/mit/djhenke/nd.w.data","a");
    fp2=fopen("/mit/djhenke/nd.shape.data","w");


*/ INPUT DATA */
printf("Input the duration (time) of propagation\n");
scanf("%lf", &tmax);
printf("Input the number of the last spatial node. Must be an even
number\n");
scanf("%d", &nodemax);
printf("Input the time step\n");
scanf("%lf",&delt);
printf("Input stiffness of the elastic support\n");
scanf("%lf",&kk);
printf("If you want 3-D plot data for stress, type 1\n");
printf("If you want 3-D plot data for deflection, type 2\n");
printf("If you don't want 3-D plot data, type 0\n");
scanf("%d",&plotflag);

/* Print Header to Output Files */
fprintf(fp,"#nodes =\t%d	time	incre=	%lf\n",nodemax,delt);
fprintf(fp,"P(0)\tTau(d)\tMaxStess\tMaxStime\n");
fprintf(fp1,"#nodes =\t%d	time	incre-\t%lf\n",nodemax,delt);
fprintf(fp1,"P(0)\tTau(d)\tW(mid)\tmaxW(mid)\ttime\n");
fclose(fp);
fclose(fp1);

/***
Allocate memory for vectors and matrices ***/
wk9=calocl(nodemax+1);
wk0=calocl(nodemax+1);
wkl=calocl(nodemax+1);
v0= calocl(nodemax+1);
load=calocl(nodemax+1);
ubar=calocl(nodemax+1);
zbar=calocl(nodemax+1);
up=caloc2(nodemax+1,3);
slope = calocl(nodemax+1);
low=caloc2(nodemax+1,2);
a=caloc2(nodemax+1,5);

if(plotflag ==1 || plotflag == 2)
{
yofxt=caloc2(nodemax+2,colnum+2); /* memory for a data matrix */
/* initialize first column of data array with the node number */
for (i=0;i<=nodemax;i+=1)
   yofxt[i+1][0]=(double)i/2.0/(double)nodemax;
}

delx = .5/(double)nodemax; /* node spacing in strip */

/* The following loops are for varying the pulse magnitude at time = 0 and
   for varying the time decay constant of the pulse. */
for(mag=100.0;mag<=721.01;mag+=2000.0)
{
   for (tau = .01; tau <= .011; tau+=.1)
   {
      /* BEGIN LOOP FOR VARYING TAU(D) */

      /***
Start procedure for first time step ***/
tim=0.0;
histress = 0.0;
itime = 0.0;

96
wmax=0.0;
wmaxtim=0.0;

/* initialize starting displacement vector*/
init_w0(wk0,nodemax);
/* re-zero "previous" vector */
init_w0(wk9,nodemax);
/* initialize starting velocity vector */
init_v0(v0,nodemax);

/* calculate stretching force */
s0=calc_s(wk0,slope,delx,nodemax);
/* Calculate applied force at time = 0 */
p0=pforce(0,mag,tau);

/** The solution to W(j) (k+l) is from [A]W(k+l)=load ***/
/* Form the 'A' matrix */
make_a(a,delx,delt,s0,p0,kk,nodemax);

for (i=0;i<=nodemax;i=1)
{
    if (For the first time step the stiffness matrix is twice that of the remaining time steps)
        for(j=0;j<4;j+=1)
            a[i][j]=-2.0;

/** Assign values to first load vector */
loadvec0(load,wk0,v0,delx,delt,s0,p0,kk,nodemax);
/** Solve for displacements at first time step; W(j)(1)**/
pentasolve(a,wkl, load, nodemax, up, low, ubar, zbar);

/*******BEGIN TIME STEPPING FOR REMAINING DURATION***********/
while (tim<tmax) /* while the time limit is not reached */
{
    /* STRESS STRESS STRESS STRESS STRESS */
    /* find maximum stress at current time */
    findstress(wk0,s0,delx,nodemax,&stressm,&stressloc);
    /* keep track of max stress for all time */
    if(stressm>histress) /* Check if stress is maximum and record */
    {
        histress=stressm;
        hitime=tim;
    }

    /* Calculate Stress at clamped end of strip*/
stress0=12.0*fabs((wk0[1]-wk0[0]))/delx/delx +s0;
    /* Calculate Stress at mid strip*/
stress5=12.0*fabs((wk0[nodemax-1]-wk0[nodemax]))/delx/delx +s0;

    /** print data to screen **/
printf("Time= %f \t W= %f \t Max stress= %f \n", tim, wk0[nodemax], stressm);

/**
The following 'if' section is for sending data to the 2-D data array at evenly spaced intervals. The rows of the array are for each node and each column contains the node data at a particular time. The variable 'colnum' determined how many columns of data there will be. It is printed at time intervals of tmax/colnum. The value for 'colnum' should be assigned in the variable declarations above the main.
**/

/*/ if plot data is desired */
if(plotflag == 1 || plotflag == 2)
{
    if (tim/tmax>=(double)k/(double)colnum)
    {
        /* The first row of each column (row #0) is the time */
        yofxt[0][k+l]=tim;
    }
/** This is for sending nodal Deflection to data array */
if ( plotflag == 2)
{
    for (i=0;i<nodemax;i+=1)
    yofxt[i+l][k+l]=wk0[i];
}
/** This is for sending nodal Stress to data array */
if ( plotflag == 1)
{
    yofxt[1][k+l]=stress0;
    yofxt[nodemax+1][k+l]=stress5;
    for (i=1;i<nodemax-1;i+=1)
    yofxt[i+1][k+l]=6.0/fabs((wk0[i-1]-2.0*wk0[i]+wk0[i+1]))/delx/delx+s0;
}
/** k+=1; /* increment the column number */
}


/*/ check if current w(mid) is max w(mid) */
if (wmax<wk0[nodemax])
{
    wmax=wk0[nodemax];
    wmaxtim=tim;
}


/*/ SHIFT DISPLACEMENT vector values to previous step*/
for(i=0;i<nodemax;i+=1)
{

98
wk9[i]=wk0[i];
wk0[i]=wk1[i];

/* INCREMENT TIME */
tim+=delt;

/* calculate stretching force */
s0=calc_s(wk0,slope,delx,nodemax);

/* Calculate applied force at current time */
p0=pforce(tim,mag,tau);

/* Form the a matrix to determine W(j)(k+1) */
make_a(a,delx,delt,s0,p0,kk,nodemax);

/* Form the load vector in solving for W(j)(k+1) */
loadvec(load,wk9,wk0,delx,delt,s0,p0,kk,nodemax);

/* Solve for W(j)(k+1) in pentadiagonal matrix reduction */
pentasolve(a,wk1,load,nodemax,up,low,ubar,zbar);

} /* End While Loop for Time Stepping */

/* PRINT RESULTS TO SCREEN AND FILES */
printf("For P(0) = %f, Tau(d) = %f
",mag,tau);
printf(" Max Stress = %lf At time %f\n",histress,hitime);
printf(" Max W(mid) = %lf At time %f\n",wmax,wmaxtim);
fp=fopen("/mit/djhenke/nd.s.data","a");
fp1=fopen("/mit/djhenke/nd.w.data","a");
fprintf(fp,"%f\t%f\t%f\n",mag,tau,histress,hitime);
fprintf(fp1,"%f\t%f\t%f\n",mag,tau,wmax,wmaxtim);
close(fp);
close(fp1);

/* Print 3-d plot data to file */
if(plotflag == 1 || plotflag == 2)
{
    fprintf(fp2,"THE FIRST COLUMN IS DISTANCE ALONG THE STRIP\n");
    fprintf(fp2,"THE FIRST ROW CONTAINS THE TIMES FOR EACH COLUMN OF\nDATA\n");
    for (i=0;i<k;i++)
        fprintf(fp2,"%f\t",yofxt[0][i]);
    fprintf(fp2,"\n");
    for (i=1;i<=nodemax+1;i++)
        for (j=0;j<=k;j++)
            fprintf(fp2,"%e\t",yofxt[i][j]);
    fprintf(fp2,"\n");
}
close(fp2);

} /* end the tau loop */
} /* end the p(0) (or 'mag') loop */
*/
END OF MAIN */

/*
FUNCTIONS________*****/

/* Find maximum stress in Plate Strip */
void findstress(double *wk0, /* vector of displacements */
  double s0, /* Stretching force */
  double delx, /* strip node spacing */
  int nodemax, /* Node number of max node */
  double *stressm, /* maximum stress */
  int *stressloc) /* location of max stress */
{
  int i;
  double varl, var2;
  var2 = fabs((2.0*wk0[i-1]-2.0*wk0[i]+wk0[i+1])/delx/delx); /* curvature at end of strip */
  *stressloc = 0; /* first node number */
  /* Compare curvature at end of half-strip to that between the ends */
  for (i = 1; i < nodemax-1; i++)
  { varl = (wk0[i-1]-2.0*wk0[i]+wk0[i+1])/delx/delx;
    if(fabs(varl) > var2)
    { var2 = fabs(varl);
      *stressloc = i;
    }
  }
  /* Midstrip curvature*/
  varl = fabs((2.0*(wk0[nodemax-1]-wk0[nodemax-1])*delx/delx); /* node number of last node */
  if(varl > var2)
  { var2 = varl;
    *stressloc = nodemax;
  }
  *stressm = (6.0*var2 + s0);
}

/*******************************
/* Solve for W(j)(k+l) with pentadiagonal matrix */
void pentasolve(double** a, /* Pentadiagonal matrix */
  double *wk1, /* unknown vector; W(j)(k+l) */
  double *load, /* load vector for matrix eqn. */
  int n, /* node number of last node */
  double **up, /* upper triangular in matrix reduction */
  double **low, /* lower triangular in matrix reduction */
  double *ubar, /* dummy variable in reduction */
  double *zbar) /* dummy variable in reduction */
{
  int i, j; /* dummy variables for loops */
  double v1, v2; /* variables to discretize formulae */
  up[0][0] = a[0][2];
  up[0][1] = a[0][3];
  for (i = 0; i < n; i++)
  { up[i][2] = a[i][4];
    low[1][1] = a[1][1] / up[0][0];
    up[1][0] = a[1][2] - up[0][1]*low[1][1];
    up[1][1] = a[1][3] - up[0][2]*low[1][1];
    load[1][1] = load[1][1] - load[0][1]*low[1][1];
    for (i = 2; i < n; i++)
low[i][0]=a[i][0]/up[i-2][0];
    zbar[i]=a[i][1]-up[i-2][1]*low[i][0];
    ubar[i]=a[i][2]-up[i-2][2]*low[i][0];
    load[i]=load[i]-low[i][0]*load[i-2];
    low[i][1]=zbar[i]/up[i-1][0];
    up[i][0]=ubar[i]-up[i-1][1]*low[i][1];
    up[i][1]=a[i][3]-up[i-1][2]*low[i][1];
    load[i]=load[i]-load[i-1]*low[i][1];
}
wk1[n]=load[n]/up[n][0];
v1=(load[n-1]-wk1[n]*up[n-1][1]);
wkl[n-1]=v1/(up[n-1][0]);
for(i=n-2;i>=0;i-=1)
{
    v1=wk1[i+2]*up[i][2];
    v2=wk1[i+1]*up[i][1];
    wk1[i]=(load[i]-v1-v2)/up[i][0];
}
/* END FUNCTION */
/**************************************************************************
/* Form Load Vector for the Kth time step */
void loadvec (double *load, /* Load vector */
    double *wk9, /*previous disp. vector */
    double *wk0, /* current disp. vector */
    double delx, /* x spacing */
    double delt, /* time spacing */
    double s0, /* stretch force */
    double p0, /* applied load */
    double kk, /* elastic support stiffness coefficient */
    int n) /* node number of last node (mid-strip) */
{
    int j;  /* dummy variable for loops */
    double v1,v2; /* dummy variables for long formulae */
    /** ROW 0 **/
    v1=1.0/2.0/pow(delx,4.0)*(wk9[0]*(pow(delx,3.0)*kk+6.0)-
        wk9[1]*8.0+wk9[2]*2.0);
    v2=p0+s0/2.0/delx/delx*(wk9[0]-2.0*wk9[1]+wk9[2]);
    load[0]=2.0*wk0[0]-wk9[0]+delt*delt*(v2-v1);
    /** ROW 1 **/
    v1=1.0/2.0/pow(delx,4.0)*(-4.0*(wk9[0]+wk9[2])+7.0*wk9[1]+wk9[3]);
    v2=p0+s0/2.0/delx/delx*(wk9[0]-2.0*wk9[1]+wk9[2]);
    load[1]=2.0*wk0[1]-wk9[1]+delt*delt*(v2-v1);
    /** ROW 2 THRU N-2 ***/
    for(j=2;j<=n-2;j+=1)
    {
        v1=1.0/2.0/pow(delx,4.0)*(wk9[j-2]+wk9[j+2]-4.0*(wk9[j-1]+wk9[j+1])+6.0*wk9[j]);
        v2=p0+s0/2.0/delx/delx*(wk9[j-1]-2.0*wk9[j]+wk9[j+1]);
        load[j]= 2.0*wk0[j]-wk9[j]+delt*delt*(v2-v1);
    }
    /** ROW N-1 **/
    v1=1.0/2.0/pow(delx,4.0)*(-4.0*(wk9[n]+wk9[n-2])+7.0*wk9[n-1]+wk9[n-3]);
    v2=p0+s0/2.0/delx/delx*(wk9[n]-2.0*wk9[n-1]+wk9[n-2]);
    load[n-1]=2.0*wk0[n-1]-wk9[n-1]+delt*delt*(v2-v1);
    /** ROW N **/
    v1=1.0/pow(delx,4.0)*(wk9[n-2]-4.0*wk9[n-1]+3.0*wk9[n]);
v2=p0+s0/delx/delx*(wk9[n-1]-wk9[n]);
load[n]=2.0*wk0[n]-wk9[n]+delt*delt*(v2-v1);
} /* END FUNCTION */

/***************************************************************************/

/* Form Load Vector for the first time step */
void loadvec0 (double *load, /* Load vector */
   double *wk0, /* displacement vector */
   double *v0, /* velocity vector */
   double delx, /* x spacing */
   double delt, /* time spacing */
   double s0, /* stretch force */
   double p0, /* applied load */
   double kk, /* elastic support stiffness coefficient*/
   int nodemax) /* node number of last node (mid-strip)*/{
   int j,n; /* dummy variable for loops */
   double v1,v2; /* dummy variables for long formulae */
   /** ROW # 0 **/
   v1=delt*(-v0[0]*(pow(delx,3.0)*kk/1.0+6.0)+v0[1]*8.0-
   v0[2]*2.0)/pow(delx,4.0);
   v2=p0+s0*delt*(v0[0]-v0[1])/delx/delx;
   load[0]=2.0*wk0[0]+2.0*delt*v0[0]+delt*delt*(v2-v1);
   /** ROW # 1 **/
   v1=delt*(4.0*v0[0]+v0[2])/-7.0*v0[1]-v0[3]) /pow(delx,4.0);
   v2=p0+s0*delt*(-v0[0]+2.0*v0[1]-v0[2])/delx/delx;
   load[1]=2.0*wk0[1]+2.0*delt*v0[1]+delt*delt*(v2-v1);
   /** ROW 2 THRU N-2 **) /
   for(j=2;j<=nodemax-2;j+=1){
   v1=delt/pow(delx,4.0)*(-v0[j-2]+4.0*v0[j-1]+v0[j+1])-6.0*v0[j]-
   v0[j+2]);
   v2=p0+s0*delt*(-v0[j-1]+2.0*v0[j]-v[j+1])/delx/delx;
   load[j]=2.0*(wk[j]+v0[j]*delt)+delt*delt*(v2-v1);
   }
   /** ROW # N-1 **/
   n=nodemax; /* to reduce character length of formula */
   v1=delt*(4.0*(v0[n-2]+v0[n])--7.0*v0[n-1]-v0[n-3]) /pow(delx,4.0);
   v2=p0+s0*delt*(-v0[n-2]+2.0*v0[n-1]-v0[n])/delx/delx;
   load[n-1]=2.0*wk[n]-2.0*delt*v0[n-1]+delt*delt*(v2-v1);
   /** ROW # N **/
   v1=2.0*delt/pow(delx,4.0)*(-v0[n-2]+4.0*v0[n-1]-3.0*v0[n]);
   v2=p0+s0*delt*2.0*(v0[n]-v0[n-1])/delx/delx;
   load[n]=2.0*(wk[n]+v0[n]*delt)+delt*delt*(v2-v1);
   }
   /* end function */
/***************************************************************************/

/* Assign values to the 'a' matrix */
void make_a (double **a,
   /* 'a' matrix */
   double delx, /* x spacing */
   double delt, /* time spacing */
   double s0, /* stretch force */
   double p0, /* applied load */
   double kk, /* elastic support stiffness coefficient */
   int nodemax) {
   int i,j; /* dummy variables for loops */
   double v1; /* dummy variables to manage long formulae */
/** ROW # 0 **/
vl=s0/(delx*delx) + kk/2.0/delx + 3.0*1.0/pow(delx, 4.0);
a[0][2]= 1.0 + delt*delt*vl;
v1=(-s0)/delx/delx-4.0*1.0/pow(delx, 4.0);
a[0][3]=delt*delt*vl;
a[0][4]=1.0*delt*delt/pow(delx, 4.0);
/** ROW # 1 **/
a[1][1]=a[0][3]/2.0;
vl=s0/delx/delx+7.0/2.0*1.0/pow(delx, 4.0);
a[1][2]=1.0+delt*delt*vl;
a[1][3]=a[1][1];
a[1][4]=delt*delt*1.0/2.0/pow(delx, 4.0);
/** ROW 2 THRU ROW N-2 **/
for (i=2;i<nodemax-2;i+=1)
{
    a[i][0]=1.0*delt*delt/2.0/pow(delx, 4.0);
a[i][1]=a[i][1];
    vl=s0/delx/delx+3.0/pow(delx, 4.0);
a[i][2]=1.0+delt*delt*vl;
a[i][3]=a[i][1];
a[i][4]=a[i][0];
}
/** ROW N-1 **/
a[nodemax-1][0]=a[1][4];
a[nodemax-1][1]=a[1][1];
a[nodemax-1][2]=a[1][2];
a[nodemax-1][3]=a[nodemax-1][1];
/** ROW N **/
a[nodemax][0]=a[0][4];
a[nodemax][1]=a[0][3];
a[nodemax][2]=a[2][2];
} /* end function */
身为函数
/* CALCULATE THE APPLIED FORCE AT THE GIVEN TIME */
double pforce (double tim, /* time */
        double mag, /* P(0) */
        double tau) /* tau(d) in decay */
{
    double force;
    force=mag*exp(-tim/tau);
    return(force);
} /* end function */
身为函数
/* CALCULATE STRETCHING FORCE */
double calc_s (double *wk0, /* vector of dispacements */
        double *slope, /* vector of slopes at nodes*/
        double delx, /* 'x' spacing */
        int nodemax) /* node number of last node at mid-strip */
{ 
    int i; /* dummy variable for loops */
    double smsum; /* used to accumulate value of integral of slope^2*/
    double s; /* stretching force */

    slope[0]=0.0; /* slope = 0 at clamped end */
smsum=0.0;
slope[nodemax]=0.0; /* slope = 0 at mid-strip */
for (i=1;i<=nodemax-1;i+=1) /* calc slope at each node*/
    slope[i] = (wk0[i+1]-wk0[i-1])/(2.0*delx);
/* apply simpson's rule to integrate slope^2 */
for (i=1;i<=nodemax-2;i+=2)
    {snum += 4.0*slope[i]*slope[i];
     snum += 2.0*slope[i+1]*slope[i+1];
    }
snum = 4.0*slope[nodemax-1]*slope[nodemax-1];
snum = snum*delx/3.0; /* snum is now value of integral */
s=12.0*snum; /* Non-Dim stretching force by formula */
return(s);
} /*END FUNCTION*/
}  

void init_v0 (double *v0, /* velocity vector at time=0 */
    int nodemax) /* node number of last node at mid-strip */
{ /* END FUNCTION*/

void init_w0 ( double *wk0, /* displacement vector;time=0 */
    int nodemax) /* node number of last node at mid-strip */
{ int i;
    double varl;
    for (i=0;i<=nodemax;i+=1)
        wk0[i]=0.0;
} /* END FUNCTION*/

/* ALLOCATE MEMORY FOR A 1-D ARRAY */
double * calocl ( int row)
{ double * vec;
    vec = (double *)calloc(sizeof(double),row);
    return(vec);
}

/* ALLOCATE MEMORY FOR A 2-D ARRAY */
double ** caloc2(int row,int col)
{ int i;
    double ** mat;
    mat=(double**)calloc(sizeof(double*),row);
    for (i=0;i<=row-1;i+=1)
        mat[i]=(double*)calloc(sizeof(double),col);
    return(mat);
} /* END FUNCTION*/

104
Appendix C. Computer Program NONDIM.FULL.C. Response to Space and Time varying Loads
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/******* PROGRAM NAME: NONDIM.FULL.C *******/
/*
This program evaluates the Dimensionless response of the plate
strip with clamped ends which undergoes spatial and time varying
loading. An implicit finite-difference scheme is used which is
unconditionally stable.
The supports are modelled as linear elastic springs which allow only
deflection in the lateral direction (i.e. it can deflect).
This program was developed to compare the response of spatially
varying loading to spatially constant loading which are both
time varying.
Spatial Nodes are numbered from 0 to 'nodemax'. 'nodemax' is specified
by the user. The '0' th node is the first clamped end node and the
node numbered 'nodemax' corresponds to the second clamp end.
The deflection, W, has space or nodal coordinates (j) and time
coordinate k. Hence the deflection at the jth node and at the kth
time step is referred to as W(j)(k).
The loading can be specified in the function at the end of this
program; pforce().
This program also allows for initial velocity and deflection to be
specified. This must be done by changing the functions: init_v0
and init_w0.
When indicated by the user, memory for a large two dimensional array is
dynamically allocated to record nodal stress or displacement data
as a function of time.*/

/** V A R I A B L E  D E C L A R A T I O N S **/
double delt; /* time step */
double time=0.0; /* cummulative time */
double tmax; /* Duration of propagation time */
double delx; /* spacing between spatial nodes */
double s0; /* stretching force for current time */
double k; /* spring constant of elastic support */
double stressmid; /* stress at mid strip */
double stressm; /* maximum stress for current time */
double histress; /* max stress for all time */
double hitime; /* time of maximum global stress*/
double stress0; /* stress at end of strip*/
double stress5; /* stress at mid-span of strip*/
int stressloc; /* location of maximum stress*/
int nodemax; /* node number of last node */
int colnum=100; /* number of columns 2-D data array */
int plotflag; /* Indicator for printing 3-D plot data, if desired */
int i,j,k=0; /* dummy variable used in loops*/
double *p0; /* current force magnitude ( =f(x,t) ) */
double *wk9; /* displacement array for previous time step */
double *wk0; /* displacement array of current time step */
double *wk1; /* displacement array of k+1 time (unknown)*/
double *v0; /* velocity array at time = 0 (I.C.)*/
double *load; /* load vector in solving for W(j)(k+1)*
double *ubar; /* used in pentadiagonal decomp as dummy variable */
double *zbar; /* used in pentadiagonal decomp as dummy variable */
double *slopes; /* vector for slope at each node*/
double **a; /* pentadiagonal matrix to solve for W(j)(k+1)*/
double **up; /* upper triangular in pentadiagonal decomp */
double **low; /* lower triangular in pentadiagonal decomp*/
double **yofxt; /* data matrix of nodes disp vs time*/

/** FUNCTION PROTOTYPES /**
/* allocate memory for a vector */
double* calocl(int);
/* allocate memory for a matrix */
double** caloc2(int,/* total number of rows in matrix */
int); /* total # of columns in a matrix */
/* calculate stretching force */
double calc_s (double *wk00, /* vector of displacements */
double *slopes, /* vector; dw/dx at each node */
double delx, /* 'x' spacing */
int nodemax); /* node number of last right end node */
/* initialize displacement vector for time = 0 */
void init_w0 (double *wk00, /* displacement at time=0 */
int nodemax); /* node number of last right end node */
/* initialize velocity vector for time = 0 */
void init_v0 (double *vO0, /* velocity vector at time=0 */
int nodemax); /* node number of last right end node */
/* Applied force function */
void pforce(double *pO, /* applied force vector=> force at each node */
double *time, /* time */
int nodemax); /* nodenumber of right end node */
/* Form the "A" pentadiagonal matrix */
void make_a (double**a, /* "a" matrix */
double delx, /* x spacing */
double delt, /* time spacing */
double sO, /* stretch force */
double kk, /* elastic support stiffness coefficient*/
int nodemax); /* node number of last right end node */
/* Form Load Vector for the first time step */
void loadvec0 (double *load, /* Load vector */
double *wk0, /* displacement vector */
double *vO0, /* velocity vector*/
double delx, /* x spacing */
double delt, /* time spacing */
double sO, /* stretch force */
double *pO, /* applied load */
double kk, /* elastic support stiffness coefficient*/
int nodemax); /* node number of last right end node*/
/* Form Load Vector for the Kth time step */
void loadvec (double *load, /* Load vector */
double *wk9, /* previous disp. vector */
double *wk0, /* current disp. vector */
double delx, /* x spacing */)
double delt, /* time spacing */
double s0, /* stretch force */
double *p0, /* applied load */
double kk, /* elastic support stiffness coefficient*/
int nodemax); /* node number of last right end node */
/* Solve for \( W(j)(k+1) \) with pentadiagonal matrix */
void pentasolve(double** a, /* Pentadiagonal matrix */
    double *wk1, /* unknown vector; \( W(j)(k+1) \) */
    double *load, /* load vector for matrix eqn. */
    int nodemax, /* node number of last right end node */
    double **up, /* upper triangular in matrix reduction*/
    double **low, /* lower triangular in matrix reduction*/
    double *ubar, /* variable used in reduction */
    double *zbar); /* variable used in reduction */
/* Find maximum stress in Plate Strip */
void findstress(double *wk0, /* vector of displacements */
    double s0, /* Stretching force */
    double delx, /* strip node spacing */
    int nodemax, /* Node number of max node */
    double *stressm, /* maximum stress */
    int *stressloc); /* location of max stress */
FILE *fp;
FILE *fpl;
/************************************************************************
main()
{ /* START MAIN */
    fp=fopen("/mit/djhenke/nd.gen.data","w");
    fpl=fopen("/mit/djhenke/nd.shape.data","w");/* INPUT DATA */
    printf("Input the number of the last spatial node. Must be an even
        number\n");
    scanf("%d",&nodemax);
    printf("Input the time step\n");
    scanf("%lf",&delt);
    printf("Input the duration (time) of propagation\n");
    scanf("%lf",&tmax);
    printf("Input the stiffness of the supports\n");
    scanf("%lf",&kk);
    printf("If you want 3-D plot data for stress, type 1\n");
    printf("If you want 3-D plot data for deflection, type 2\n");
    printf("If you don't want 3-D plot data, type 0\n");
    scanf("%d",&plotflag);
    fprintf(fp,"#nodes\tper strip\t%d\tTime Increment\t%lf\n",nodemax,delt);
    fprintf(fp,"Time\tW(mid)\tStressMid\tFrontEndStress\tBackEndStress\n");
    fprintf(fpl,"Node\tDeflection\n");
/**
ALLOCATE MEMORY FOR MATRICES AND ARRAYS
**/
wk9 = caloc1(nodemax+1);
wk0 = caloc1(nodemax+1);
wkl = caloc1(nodemax+1);
vl = caloc1(nodemax+1);
load = caloc1(nodemax+1);
ubar = caloc1(nodemax+1);
zbar = caloc1(nodemax+1);
up = caloc2(nodemax+1,3);
slope = caloc1(nodemax+1);
p0 = caloc1(nodemax+1);
low = caloc2(nodemax+1,2);
a = caloc2(nodemax+1,5);

if(plotflag == 1 || plotflag == 2)
{
  /* allocate memory for a data matrix */
yofxt = caloc2(nodemax+2,colnnum+2);
  /* initialize first column of data array with the node number */
  for (i=0; i<nodemax; i++)
    yofxt[i+1][0] = (double)i/(double)nodemax;
}

delx = 1.0/(double)nodemax;  /* node spacing in strip */

/******* Start procedure for first time step **********/
/* initialize starting displacement vector*/
init_w0(wk0, nodemax);
/* initialize starting velocity vector */
init_v0(v0, nodemax);
/* calculate stretching force */
s0 = calc_s(wk0, slope, delx, nodemax);
/* Calculate applied force at each node at time = 0 */
pforce(p0, 0, nodemax);
/*The solution to W(j)(k+1) is from [A]{W(k+1)}-{load} */
/*where j is space variable and k is time variable */
/* W is deflection, [A] is the 'stiffness' matrix */
/* load is the 'load' vector */
make_a(a, delx, delt, s0, kk, nodemax); /* Form the "A" matrix */
/* With fourth order partial differential equations */
/* the [A] matrix that is formed is a pentadiagonal */

for (i=0; i<nodemax; i++)
{
  /* For the first time step the stiffness matrix is twice */
  /* that of the remaining time steps */
  for (j=0; j<=4; j++)
    a[i][j] = 2.0;
}

/*** Assign values to first load vector ***/
loadvec0(load, wk0, vl, v0, delx, delt, s0, p0, kk, nodemax);
/*** Solve for displacements at first time step; W(j)(1) ***/
pentasolve(a, wk1, load, nodemax, up, low, ubar, zbar);
/*** BEGIN TIME STEPPING FOR FOR REMAINING DURATION *******/
while (tim < tmax) /* while the time limit is not reached */ {
    /* STRESS STRESS STRESS STRESS STRESS */
    /* find maximum stress at current time */
    findstress(wk0, sO, delx, nodemax, &stressm, &stressloc);
    /* keep track of max stress for all time */
    if (stressm > histress) {
        histress = stressm;
        hitime = tim;
    }

    /* Calculate Stress at first clamped end of strip*/
    stress0 = 12.0 * (wk0[1] - wk0[0]) / delx / delx + sO;

    /* Calculate Stress at second clamped end of strip*/
    stress5 = 12.0 * (wk0[nodemax-1] - wk0[nodemax]) / delx / delx + sO;

    /* Calculate Stress at midpoint of strip*/
    stressmid = 6.0 * (wk0[nodemax/2+l] + wk0[nodemax/2-1] -
                      2.0 * wk0[nodemax/2]) / delx / delx + sO;

    /* print data to screen and files */
    printf("Time %f, Center %lf ", tim, wk0[nodemax/2]);
    printf("Max Stress=%f node %d
", stressm, stressloc, p0[nodemax/2]);
    fprintf(fp,"%lf\t%lf\t%lf\t", tim, wk0[nodemax/2], stressmid);
    fprintf(fp,"%lf\t%lf\n", stress0, stress5);

    /* The following 'if'section is for sending data to the 2-D data array
    at evenly spaced time intervals. The rows of the array are for
    each node and each column contains the node data at a particular
    time. The variable 'colmum' determines how many columns of data
    there will be. This number should be in the variable declarations
    above the main() so memory can be allocated dynamically. */

    if (plotflag == 1 || plotflag == 2) /* if plot data is desired */ {
        if (tim / tmax > (double) k / (double) colnum) {
            /* The first row of each column (row #0) is the time */
            yofxt[0][k+1] = tim;
            /*********************************************************/
            /* This is for sending nodal Deflection to data array */
            if (plotflag == 2) {
                for (i = 0; i <= nodemax; i++)
                    yofxt[i+1][k+1] = wk0[i];
            }
            /*********************************************************/
            /* This is for sending nodal Stress to data array */
            if (plotflag == 1)
{ 
  yofxt[1][k+1]=stress0;
  yofxt[nodemax+1][k+1]=stress5;
  for (i=1;i<=nodemax-1;i+=1) 
    yofxt[i+1][k+1]=6.0*(wk0[i-1]- 2.0*wk0[i] +wk0[i+1]) 
    /delx/delx+s0;
}

/*****************************************/
  k+=1;  /* increment the column number */
}

/*SHIFT DISPLACEMENT vector values to previous step*/
for(i=0;i< nodemax;i+=l)
{
  wk9[i]=wk0[i];
  wk0[i]=wk1[i];
}

/* increment time */
tim+=delt;

/* calculate stretching force */
s0=calc_s(wk0,slope,delx,nodemax);

/* Calculate applied force at current time*/
pforce(p0,tim,nodemax);

/* Form the [A] matrix to determine W(j) (k+1)*/
make_a(a,delx,delt,s0,kk,nodemax);

/* Form the load vector in solving for W(j) (k+1)*/
loadvec(load,wk9,wk0,delx,delt,s0,p0,kk,nodemax);

/* Solve for W(j) (k+1) in pentadiagonal matrix reduction*/
pentasolve(a,wk1,load,nodemax,up,low,ubar,zbar);
}

/* End While Loop for Time Stepping */

printf("Max Stess overall is %f Occured at time %f\n",histress,hitime);
fprintf(fp,"Max Stess overall is %f Occured at time 
%f\n",histress,hitime);

/* Print 3-d plot data to file */
if(plotflag == 1 || plotflag == 2)
{
  fprintf(fpl,"THE FIRST COLUMN IS DISTANCE ALONG THE STRIP\n");
  fprintf(fpl,"THE FIRST ROW CONTAINS THE TIMES FOR EACH COLUMN OF 
DATA\n");
  for (i=0;i<=k;i+=1)
    fprintf(fpl, "%f\t",yofxt[0][i]);
  fprintf(fpl, "\n");
}
/* Print the deletion or stress data from yofxt matrix. 
   Note increment is i+=4 so only 1/4 the data is printed 
to file. */
for (i=1;i<=nodemax+1;i+=4)
{
    for (j=0;j<=k;j+=l)
    {
        fprintf(fp1, "%e\t", yofxt[i][j]);
        fprintf(fp1, "\n");
    }
}
fclose(fp);
fclose(fpl);

} /* END OF MAIN */
***********************************************************************

/***** FUNCTIONS **********/
void findstress(double * wk0, /* vector of displacements */
    double s0, /* Stretching force */
    double delx, /* strip node spacing */
    int nodemax, /* Node number of max node */
    double * stressm, /* maximum stress */
    int * stressloc) /* location of max stress */
{
    int i;
    double var1,var2;
    var2=fabs((2.0*wk0[1]-2.0*wk0[0])/delx/delx); /* curvature at left end of 
        strip */
    *stressloc=0; /* first node number (0) is location*/
    /* Compare curvature at end of half-strip to that between the 
        ends */
    for (i=1;i<=nodemax-1;i+=1)
    {
        var1=(wk0[i-1]-2.0*wk0[i]+wk0[i+1])/delx/delx;
        if(fabs(var1)>var2)
        {
            var2=fabs(var1);
            *stressloc=i;
        }
    }
    var1=fabs(2.0*(wk0[nodemax-1]-wk0[nodemax])/delx/delx); /*rightend 
        curvature*/
    if(var1>var2)
    {
        var2=var1;
        *stressloc=nodemax;
    }
    *stressm=(6.0*var2+s0);
}
***************************************************************************/

/***** Solve for W(j)(k+1) with pentadiagonal matrix *****
void pentasolve(double** a, /* Pentadiagonal matrix */
    double *wk1, /* unknown vector; W(j)(k+1) */
# Load Vector for Matrix Equation

```c
double *load;  /* load vector for matrix eqn. */
int n;  /* node number of last node */
double **up;  /* upper triangular in matrix reduction*/
double **low;  /* lower triangular in matrix reduction*/
double * ubar, /* dummy variable used in reduction */
double * zbar) /* dummy variable used in reduction */
{
    int i,j;  /* dummy variables for loops */
double v1,v2; /* variables to discretize complex formulae*/

    up[0][0]=a[0][2];
    up[0][1]=a[0][3];
    for(i=0;i<=n;i++)
    up[i][2]=a[i][4];
    low[1][1]=a[1][1]/up[0][0];
    up[1][0]=a[1][2]-up[0][1]*low[1][1];
    up[1][1]=a[1][3]-up[0][2]*low[1][1];
    load[1]=load[1]-load[0]*low[1][1];
    for(i=2;i<=n;i++)
    {
        low[i][0]=a[i][0]/up[i-2][0];
        zbar[i]=a[i][1]-up[i-2][1]*low[i][0];
        ubar[i]=a[i][2]-up[i-2][2]*low[i][0];
        load[i]=load[i]-load[i-1]*low[i][1];
        low[i][1]=zbar[i]/up[i-1][0];
        up[i][0]=ubar[i]-up[i-1][1]*low[i][1];
        up[i][1]=a[i][3]-up[i-1][2]*low[i][1];
        load[i]=load[i]-load[i-1]*low[i][1];
    }

    wk1[n]=load[n]/up[n][0];
v1=(load[n-1]-wk1[n]*up[n-1][1]);
wk1[n-1]=v1/(up[n-1][0]);
    for(i=n-2;i>=0;i--)
    {
        v1=wk1[i+2]*up[i][2];
        v2=wk1[i+1]*up[i][1];
        wk1[i]=(load[i]-v1-v2)/up[i][0];
    }
}

END FUNCTION

END FUNCTION

-- Form Load Vector for the Kth time step --
void loadvec (double *load,
    double *wk9, /* previous disp. vector */
    double *wk0, /* current disp. vector */
    double delx, /* x spacing */
    double delt, /* time spacing */
    double s0, /* stretch force */
    double *p0, /* applied load */
    double kk, /* elastic support stiffness coefficient */
    int n) /* node number of last node */
{
    int j;  /* dummy variable for loops */
double v1,v2; /* dummy variables for long formulae */

    v1=1.0/2.0/pow(delx,4.0)*(wk9[0]-(pow(delx,3.0)*kk+6.0)-
    wk9[1]*8.0+wk9[2]*2.0);   
    v2=p0[0]*s0*(-wk9[0]+wk9[1])/delx/delx;  
    load[0]=2.0*wk0[0]-wk9[0]+delt*delt*(v2-v1);  

END FUNCTION
```

113
void loadvec0 (double *load, /* Load vector */ double *wk0, /* displacement vector */ double *v0, /* velocity vector */ double delx, /* x spacing */ double delt, /* time spacing */ double s0, /* stretch force */ double *p0, /* applied load */ double *vO, /* elastic support stiffness coefficient */ double *wkO, /* displacement vector */ double *vO, /* velocity vector */ double delt, /* time spacing */ double sO, /* stretch force */ double *pO, /* applied load */ double *vO, /* elastic support stiffness coefficient */ int nodemax) /* node number of last node (mid-strip)*/
{
    int j, n; /* dummy variable for loops */
    double v1, v2; /* dummy variable for long formulae */
    /* ** ROW # 0 **/
    v1 = delt * (-v0[0] * (pow(delx, 3.0) * kk/1.0+6.0) + v0[1] * 8.0 - v0[2] * 2.0) / pow(delx, 4.0);
    v2 = pO[0] + s0 * delt * (v0[0] - v0[1]) / delx/delx;
    load[0] = 2.0 * wk0[0] + 2.0 * delt * v0[0] + delt * delt * (v2 - v1);
    /* ** ROW # 1 **/
    v1 = delt * (4.0 * (v0[0] + v0[2]) - 7.0 * v0[1] - v0[3]) / pow(delx, 4.0);
    v2 = pO[1] + sO * delt * (-v0[0] + 2.0 * v0[1] - v0[2]) / delx/delx;
    load[1] = 2.0 * wk0[1] + 2.0 * delt * v0[1] + delt * delt * (v2 - v1);
    /* ** ROW # N-1 **/
    v1 = delt * (-v0[n-2] + 4.0 * (v0[n-1] - v0[n+1]) - 6.0 * v0[n] - v0[n+2]) / pow(delx, 4.0);
    v2 = p[n-1] + sO * delt * (-v0[n] + 2.0 * v0[n-1] - v0[n-2]) / delx/delx;
    load[n-1] = 2.0 * wk0[n-1] + 2.0 * delt * v0[n-1] + delt * delt * (v2 - v1);
    for(j=2; j<nodemax-2; j++)
    {
        /* ** ROW # N-2 **/
        v1 = delt * (-v0[j-2] + 4.0 * (v0[j-1] + v0[j+1]) - 6.0 * v0[j] - v0[j+2]) / pow(delx, 4.0);
        v2 = pO[j] + sO * delt * (-v0[j-1] + 2.0 * v0[j] - v0[j+1]) / delx/delx;
        load[j] = 2.0 * wk0[j] + v0[j] * delt + delt * delt * (v2 - v1);
    }
    /* ** ROW # N **/
    v1 = delt * (-v0[n+1] + 2.0 * (v0[n+2] - v0[n+3]) / pow(delx, 4.0);
    v2 = p[n+1] + sO * delt * (-v0[n] + 2.0 * v0[n-1] - v0[n-2]) / delx/delx;
    load[n] = 2.0 * wk0[n] + 2.0 * delt * v0[n] + delt * delt * (v2 - v1);
    /* END FUNCTION */
    /* END FUNCTION */
}
/** ROW 0 */
vl=delt*(-v0[n]*pow(delx,3.0)*kk/l.0+6.0)+v[0][n-2]/pow(delx,4.0);
v2=p0[n]+s0*2.0*delt*(v0[n]-v0[n-1])/delx/delx;
load[n]=2.0*(wk0[n]+v0[n]*delt)+delt*delt*(v2-v1);
} /* end function */

/********************************************
/* Assign values to the 'a' matrix */
void make_a (double**a, /* "a" matrix */
    double delx, /* x spacing */
    double delt, /* time spacing */
    double s0, /* stretch force */
    double kk, /* elastic support stiffness coefficient*/
    int nodemax)
{
    int i,j; /* dummy variables for loops */
    double vl; /* dummy variables for long formulae */
    /** ROW 0 **/
    vl=s0/(delx*delx) + kk/2.0/delx + 3.0*1.0/pow(delx,4.0);
    a[0][2]= 1.0 + delt*delt*vl;
    vl= (-s0)/(delx*delx)+7.0/2.0*1.0/pow(delx,4.0);
    a[0][4]=delt*delt/pow(delx,4.0);
    /** ROW 1 **/
    a[1][1]=a[0][3]/2.0;
    vl=s0/delx*delx+3.0/pow(delx,4.0);
    for (i=2;i<=nodemax-2;i+=1)
    {
        a[i][0]=1.0*delt*delt/2.0/pow(delx,4.0);
        a[i][1]=a[i][1];
        a[i][2]=1.0+delt*delt*vl;
        a[i][3]=a[i][1];
        a[i][4]=a[i][0];
    }
    /** ROW N-1 **/
    a[nodemax-1][0]=a[1][4];
    a[nodemax-1][1]=a[1][1];
    a[nodemax-1][2]=a[1][2];
    a[nodemax-1][3]=a[nodemax-1][1];
    /** ROW N **/
    a[nodemax][0]=a[0][4];
    a[nodemax][1]=a[0][3];
    a[nodemax][2]=1.0+delt*delt*(vl+kk/2.0/delx);
} /* end function */

/********************************************
/* CALCULATE STRETCHING FORCE */
double calc_s (double *wk0, /* vector of dispacements */
    double *slope, /* vector of slopes at nodes*/
    double delx, /* 'x' spacing */
    int nodemax) /* node number of last node at mid-strip */
int i;    /* dummy variable for loops */
double smsum;   /* used to accumulate value of integral of slope^2*/
double s;       /* stretching force      */

slope[0]=0.0;  /* slope = 0 at clamped end   */
smsum=0.0;
slope[nodemax]=0.0;   /* slope = 0 at mid-strip    */
for (i=1;i<nodemax-1;i+=1)    /* calc slope at each node   */
   slope[i] = (wk0[i+1]-wk0[i-1])/(2.0*delx);

   /* apply simpson's rule to integrate slope^2 */
for (i=1;i<nodemax-2;i+=2)    
   
   smsum = smsum + 4.0*slope[i]*slope[i];
   smsum = smsum + 2.0*slope[i+1]*slope[i+1];

   smsum = smsum +4.0*slope[nodemax-1]*slope[nodemax-1];
   smsum=smsum*delx/3.0;    /* smsum is now value of integral */
   s=6.0*smsum;   /* Non-Dim stretching force from formula */
   return(s);
}    /*END FUNCTION*/

/**************************************
/* INITIALIZE VELOCITY VECTOR AT TIME = 0 */
void init_v0 ( double *v0, /* velocity vector at time=0 */
   int nodemax)    /* node number of last node at mid-strip */
{
   int i;
   for (i=0;i<nodemax;i+=1)
      v0[i]=0.01;
}    /* END FUNCTION*/

/**************************************
/* INITIALIZE DISPLACEMENT VECTOR AT TIME = 0 */
void init_w0 ( double *wk0, /* displacement;time=0 */
   int nodemax)    /* node number of last node at mid-strip */
{
   int i,j;
   double dum;
   for (i=0;i<nodemax;i+=1)
      wk0[i]=0.0;
}

/**************************************
/* ALLOCATE MEMORY FOR A 1-D ARRAY */
double * calcol ( int row)
{ double * vec;
  vec = (double *)calloc(sizeof(double),row);
  return(vec);
}

/**************************************
/* ALLOCATE MEMORY FOR A 2-D ARRAY */
double ** calcol2(int row,int col)
{    int i;
    double ** mat;
  mat=(double**)calloc(sizeof(double*),row);
for (i=0;i<=row-1;i+=1)
    mat[i]=(double*)calloc(sizeof(double),col);
return(mat);

/***********************************************************/
/* CALCULATE THE APPLIED FORCE AT THE GIVEN TIME */
void pforce (double *pO, /* pointer to applied force vector */
    double tim, /* time */
    int nodemax) /* # of last node */
{
    int i,j,k;
    double vs=5.0; /* pulse front velocity */
    double pp=700.0; /* peak magnitude of pulse at pulse front */
    double tt=.01; /* time decay constant of pulse */
    for(i=0;i<nodemax;i+=1)
        if((double)i/(double)nodemax > vs*tim)
            pO[i]= 0.0;
        else
            pO[i]=pp*exp(-(tim-(double)i/(double)nodemax/vs )/.tt);
} /*end function*/
Appendix D. Frequency Responses of Galerkin Solution and Finite Difference Model
Galerkin solution to Plate Strip Dimensionless Frequency Response
Under Uniform Dimensionless Loading.
Load = P * sin(omega * time) psi.  P = Magnitude of Load.
Frequency Response of Finite-Difference Model of Plate Strip
Under Dimensionless Uniform Loading. Response based on Center Deflection.
Load = P* \sin(\omega \cdot t) \text{ psi}. \ P = \text{Magnitude of Load.}

\begin{itemize}
  \item P = 200
  \item P = 50
  \item P = 0 (natural vibration)
  \item P = 100
\end{itemize}
**PROGRAM NAME: STATIC.C**

This program is designed to give the static response of the plate strip under a static uniform load. Central finite difference technique is used to solve for final deflection.

Convergence must be obtained between the stretching force in the plate strip used to solve the finite difference equations and the stretching force calculated after the solution is obtained. Therefore, the criterion used for convergence is when the difference between the two stretching forces is less than some small value. This value is assigned to the variable 'tolmin' in variable declarations.

The finite difference equations yield a pentadiagonal matrix which must be solved to yield the solution.

The output file contains the the nodal displacements.

---

double *w;         /* Displacement vector */
double *slope;    /* Slope vector for each node*/
double **a;       /* 'A' matrix */
double **up;      /* upper triangular in matrix reduction */
double **low;     /* lower triangular in matrix reduction */
double *ubar;     /* dummy variable used in reduction */
double *zbar;     /* dummy variable used in reduction */
double *load;     /* load vector (applied force) used in reduction */
double s;         /* stretching force used in reduction of matrix*/
double tol;       /* tolerance variable for successive values of stretching force*/
double tolmin=1e-6; /* tolerance for successive valus of stretching force*/
double scalcd;    /* Calculated S based on displacement*/
double delx;      /*delta x */
double stressm;   /* max stress at end */
int nodemax;      /* number of max node (mid strip) */
int j;            /* dummy variable */
double p;         /* Applied force */

****FUNCTION PROTOTYPES************

/* allocate memory for a vector */
double* calocl(int);

/* allocate memory for a matrix */
double** caloc2(int, /* total number of rows in matrix */
                int); /* total columns in a matrix */

/* calculate stretching force */
double calc_s (double *w, /* vector of dispacements */
              double *slope,/* vector of nodal slope (dw/dx) */
              double delx, /* 'x' spacing */
int nodemax); /* node number of last node at mid-strip */

/* form the stiffness matrix */
void init_a (double **a, /* stiffness matrix */
         double s, /* stretching force used to solve
         for deflection */
         double delx, /* delta x; nodal spacing */
         int node); /* number of last node */

/* Solve for W(j) with pentadiagonal matrix */
void penta_solve(double** a, /* Pentadiagonal matrix */
         double *w, /* unknown vector; W(j) */
         double *load, /* load vector for matrix */
         int nodemax, /* node number of last node */
         double **up, /* upper triangular in matrix reduction*/
         double **low, /* lower triangular in matrix reduction*/
         double *ubar, /* dummy variable in reduction */
         double *zbar); /* dummy variable in reduction */

/* Initialize the load vector */
void init_load (double p,
         double *load,
         int nodemax);

FILE *fp;
main()
{
    fp=fopen("/mit/djhenke/w.static.shape","w");
    printf("Input # of the last nodes. Must be an even number!!\n");
    scanf("%d", &nodemax);
    printf("Input the value of the Force\n");
    scanf("%lf", &p);

    /* node spacing */
    delx=.5/(double)nodemax;

    /* Allocate memory for vectors and matrices */
    w = calocl(nodemax+1);
    a = caloc2(nodemax+1,5);
    slope = calocl(nodemax+1);
    ubar=calocl(nodemax+1);
    zbar=calocl(nodemax+1);
    load=calocl(nodemax+1);
    up=caloc2(nodemax+1,3);
    low=caloc2(nodemax+1,2);

    s=0;
    tol=10;

    /* BEGIN CONVERGENCE CYCLE */
    while(fabs(tol)>tolmin)
    {
        /* INITIALIZE THE STIFFNESS MATRIX */
        init_a(a,s,delx,nodemax);
        /* INITIALIZE THE LOAD VECTOR */
        init_load(p,load,nodemax);
        /* SOLVE FOR THE DISPLACEMENT */
pentasolve(a,w,load,nodemax,up,low,ubar,zbar);
/* NOW DETERMINE STRETCHING FORCE */
scalcd=calc_s(w,slope,delx,nodemax);
/* FIND MAX CLAMPED END STRESS */
stressm=12.0*w[1]/delx/delx+scalcd;
/* FIND DIFFERENCE IN SUCCESSIVE STRETCH FORCES */
tol=s-scalcd;
/* ASSUME A NEW STRETCHING FORCE */
s=(s+scalcd)/2.0;
printf("S = %f W[nodemax] = %f Max stress = %f\n",scalcd,w[nodemax],stressm);
}
/* PRINT DATA */
init_load(p,load,nodemax);
for(j=0; j<nodemax; j+=1)
   fprintf(fp, "%d\t%15.131f\t%lf\n",j,w[j],load[j]);
fclose(fp);
} /* end main */
/*****************************/
/* Solve for W(j) (k+1) with pentadiagonal matrix */
void pentasolve(double** a,/* Pentadiagonal matrix */
   double *wkl, /* unknown vector; W(j) (k+1) */
   double *load, /* load vector for matrix eqn. */
   int n, /* node number of last node */
   double **up, /* upper triangular in matrix reduction*/
   double **low, /* lower triangular in matrix reduction*/
   double *ubar, /* dummy variable used in reduction */
   double *zbar) /* dummy variable used in reduction */
{
   int i,j; /* dummy variables for loops */
   double v1,v2; /* variables to discretize formulae */
   up[1][0]=a[1][2];
   up[1][1]=a[1][3];
   for(i=1;i<=n;i+=1)
      up[i][2]=a[i][4];
   low[2][1]=a[2][1]/up[1][0];
   up[2][0]=a[2][2]-up[1][1]*low[2][1];
   up[2][1]=a[2][3]-up[1][2]*low[2][1];
   for(i=3;i<=n;i+=1)
   {
      low[i][0]=a[i][0]/up[i-2][0];
      zbar[i]=a[i][1]-up[i-2][1]*low[i][0];
      ubar[i]=a[i][2]-up[i-2][2]*low[i][0];
      load[i]=load[i]-low[i][0]*load[i-2];
      low[i][1]=zbar[i]/up[i-1][0];
      up[i][0]=ubar[i]-up[i-1][1]*low[i][1];
      up[i][1]=a[i][3]-up[i-1][2]*low[i][1];
      load[i]=load[i]-load[i-1]*low[i][1];
   }
   wkl[n]=load[n]/up[n][0];
   v1=(load[n-1]-wkl[n]*up[n-1][1]);
   wkl[n-1]=v1/(up[n-1][0]);
   for(i=n-2;i>=1;i-=1)
   {
      v1=wkl[i+2]*up[i][2];
      v2=wkl[i+1]*up[i][1];
      wkl[i]=(load[i]-v1-v2)/up[i][0];
      
   
```
void init_load (double p, double *load, int nodemax)
{
    int i;
    for (i=0;i<=nodemax;i++)
        load[i]=p;
}

void init_a(double **a, double s, double delx, int node)
{
    int i;
    double k4,k2;
    double a0,a1,a2;
    k4=pow(delx, 4.0);
    k2=delx*delx;
    a[1][2]=7.0/k4+s*2.0/k2;
    a[1][3]=-4.0/k4-s/k2;
    a[1][4]=1.0/k4;
    a[2][1]=-4.0/k4-s/k2;
    a[2][2]=6.0/k4+2.0*s/k2;
    a[2][3]=a[2][1];
    a[2][4]=1.0/k4;
    a0=1.0/k4;
    a1=-4.0/k4-s/k2;
    a2=6.0/k4+2.0*s/k2;
    for(i=3;i<=node-2;i++)
    {
        a[i][0]=a0;
        a[i][1]=a1;
        a[i][2]=a2;
        a[i][3]=a1;
        a[i][4]=a0;
    }
    a[node-1][0]=a0;
    a[node-1][1]=a1;
    a[node-1][2]=7.0/k4+s*2.0/k2;
    a[node-1][3]=a1;
    a[node][0]=2.0*a0;
    a[node][1]=2.0*a1;
    a[node][2]=a2;
} /*END FUNCTION */

double calc_s (double *wk0, double *slopes, double delx, int nodemax) /* node number of last node at mid-strip */
int i; /* dummy variable for loops */
double smsum; /* used to accumulate value of integral of slope^2 */
double s; /* stretching force */

slope[0]=0.0; /* slope = 0 at clamped end */
smsum=0.0;
slope[nodemax]=0.0; /* slope = 0 at mid-strip */
for (i=1;i<=nodemax-1;i+=1) /* calc slope at node */
    slope[i] = (wk0[i+1]-wk0[i-1])/(2.0*delx);

/* apply simpson's rule to integrate slope^2 */
for (i=1;i<=nodemax-2;i+=2)
    { smsum = smsum + 4.0*slope[i]*slope[i];
        smsum = smsum + 2.0*slope[i+1]*slope[i+1];
    }
    smsum = smsum + 4.0*slope[nodemax-1]*slope[nodemax-1];
    smsum = smsum*delx/3.0; /* smsum is now value of integral */
    s=12.0*smsum; /* Non-Dim stretching force by formula */
    return(s);
} /*END FUNCTION*/

double * calocl(int row)
{ double * vec;
    vec = (double *)calloc(sizeof(double),row);
    return(vec);
}

double ** caloc2(int row,int col)
{ int i;
    double ** mat;
    mat=(double**)calloc(sizeof(double*),row);
    for (i=0;i<=row-1;i+=1)
        mat[i]=(double*)calloc(sizeof(double),col);
    return(mat);
}
Appendix F. Convergence Data for Finite Difference Solution

End Stress and Center Deflection vs. Time with Varying Time Step and Node Spacing.
Convergence Evaluation by Varying Number of Nodes in Plate Strip.
Center Deflection Response to Spatially Constant Dimensionless Pulse; $2000 \times \exp(-t/0.005)$. Time Step = .0005
Convergence Evaluation by Varying Number of Nodes in Plate Strip.
Clamped End Stress Response to Spatially Constant Dimensionless
Pulse; 2000*exp(-t/.005). Time step = .0005
Convergence Evaluation by Varying Time Step Size ($\Delta t$).
Clamped End Stress Response to Spatially Constant Dimensionless Pulse $2000 \exp(-t/0.005)$. Number of Nodes = 160.
Appendix G. Moving Pulse vs.Spatially Constant Pulse End Stress Response.
Plate Strip Clamped End Dimensionless Stress Response.
Comparison of Responses to Travelling Pulse and Spatially Constant Pulse.
Dimensionless Time Domain Pulse: $1000 \exp(-t/0.01)$.
Dimensionless Pulse Front Velocity = 200.

Pulse Dimensionless
Decay Constant, $\tau \cdot \xi_p = 2.0$

First End Response to Travelling Pulse

End Response to Spatially Constant Pulse.

Second End Response to Travelling Pulse
Plate Strip Clamped End Dimensionless Stress Response.
Comparison of Responses to Travelling Pulse and Spatially Constant Pulse.
Dimensionless Time Domain Pulse: $1000 \exp(-t/0.01)$.
Dimensionless Pulse Front Velocity = 150.

End Response to Spatially Constant Pulse.

Second End Response to Travelling Pulse

Pulse Dimensionless
Decay Constant, $\tau^{*p} = 1.5$

First End Response to Travelling Pulse

Dimensionless Stress / 100

Dimensionless Time

0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
Plate Strip Clamped End Dimensionless Stress Response.
Comparison of Responses to Travelling Pulse and Spatially Constant Pulse.
Dimensionless Time Domain Pulse: $1000 \exp(-t/0.01)$.
Dimensionless Pulse Front Velocity = 100.
Plate Strip Clamped End Dimensionless Stress Response.
Comparison of Responses to Travelling Pulse and Spatially Constant Pulse.
Dimensionless Time Domain Pulse: 1000 exp(-t/0.01).
Dimensionless Pulse Front Velocity = 50.

End Response to Spatially Constant Pulse.
Second End Response to Travelling Pulse

Pulse Dimensionless
Decay Constant, \( \tau \cdot c_p = 0.5 \)

Dimensionless Stress / 100

Dimensionless Time

0 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
Plate Strip Clamped End Dimensionless Stress Response.
Comparison of Responses to Travelling Pulse and Spatially Constant Pulse.
Dimensionless Time Domain Pulse: 1000 exp(-t/0.01).
Dimensionless Pulse Front Velocity = 25.

First End Response to Travelling Pulse
Second End Response to Travelling Pulse
End Response to Spatially Constant Pulse

Pulse Dimensionless
Decay Constant, $\tau_1 c_p = 0.25$
Appendix H. Moving Pulse vs. Spatially Constant Pulse

3-D Plot of Node Deflection.
Stress Along Top of Plate Strip vs. Time in Response to Spatially Constant Dimensionless Pulse $700 \exp(-t/0.01)$

480 Nodes. Time Step = 0.0001

Time Range: 0.0 - 0.15
Stress Along Top of Plate Strip vs. Time in Response to Travelling Dimensionless Pulse: $700 \exp(-t/0.01)$
Pulse Front Dimensionless Velocity = 100.
480 Nodes. Time Step = .00005.
Time Range: 0.0 - .05
Appendix I. Yield Pulse Charts

Pulse Magnitude vs. Time Decay Constant/Length which causes incipient yield for various values of $R$ (Length/thickness)
Incipient Yield Pulse Characteristics.

\[ \tau_{\text{length}} \text{ vs. } P_0 \text{ which results in } 40000 \text{ psi peak stress in Commercial Steel Plate Strip for various values of } R \text{ (length/thickness)}. \]

Applied Pressure Pulse is \( P_0 \cdot \exp(-\text{time/}\tau) \)
Incipient Yield Pulse Characteristics.

$t/\text{length}$ vs. $P_0$ which results in

60000 psi peak stress in Commercial Steel Plate Strip

for various values of $R$ (length/thickness).

Applied Pressure Pulse is $P_0 \exp(-t/\tau)$
Incipient Yield Pulse Characteristics.
Log-Log of tau/length vs. Po which results in
40000 psi peak stress in Commercial Steel Plate Strip
for various values of R (length/thickness)

Applied Pressure Pulse is Po*exp(-time/tau)
Incipient Yield Pulse Characteristics.
Log-Log of \( \tau / \)length vs. Po which results in
6000 psi peak stress in Commercial Steel Plate Strip
for various values of R (length/thickness)

Applied Pressure Pulse is \( P_0 \cdot \exp(-t/\tau) \)
Appendix J. Comparison of Dynamic and Statically Loaded Plate Strip Shapes

Shape of Half Strip in response to pulse compared to Half Strip Static Uniform Load with same center deflection. Comparison made at times of Maximum End Stress and Maximum Mid-Strip Stress.
Shape of Half Plate Strip at time of maximum end stress under applied pulse $219.78 \exp(-t/0.0032)$ psi compared with shape of plate strip under static uniform load causing the same center deflection. 

$R=100$, $L=100$ in. $h=1$ in. 160 nodes.
Shape of Half Plate Strip at time of maximum mid-strip stress under applied pulse $219.78 \exp(-t/0.0032)$ psi compared with shape of plate strip under static uniform load causing the same center deflection.

$R=100$  $L=100$ in.  $h=1$ in.  160 nodes.
Difference in deflection between plate half strip under pulse
219.78 \exp(-t/0.0032) \text{ psi} and plate strip under uniform static load
at the same center deflection. \( R=100 \). \( L=100 \text{ in.} \) \( h=1 \text{ in.} \) 160 nodes.

Node Number of Half Strip. 0(end) - 160(mid)
Appendix K. Stress vs. Center Deflection in Plate Strip under Uniform Static Load
Dimensionless Stress vs Center Deflection of Plate Strip determined from Finite Difference Model

Maximum End Bending Stress

Maximum Mid-Strip Bending Stress

Membrane Stress

Dimensionless Stress vs Center Deflection (W/h)
Curve fit to Dimensionless Stresses vs. Center Deflection of actual plate strip approximated by finite difference techniques

The form of the equation is as follows.

\[ \sigma = a \tilde{w}^5 + b \tilde{w}^4 + c \tilde{w}^3 + d \tilde{w}^2 + e \tilde{w} + f \]

<table>
<thead>
<tr>
<th>Stress type</th>
<th>Range</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max End Bending</td>
<td>0 ≤ W/h ≤ 1</td>
<td>-1.4045</td>
<td>-4.1441</td>
<td>49.036</td>
<td>-.53009</td>
<td>192.04</td>
<td>0</td>
</tr>
<tr>
<td>Max Mid Bending</td>
<td>0 ≤ W/h ≤ 1</td>
<td>.50446</td>
<td>2.9024</td>
<td>-13.278</td>
<td>.37634</td>
<td>95.963</td>
<td>0</td>
</tr>
<tr>
<td>Membrane</td>
<td>0 ≤ W/h ≤ 1</td>
<td>7.9159 e-2</td>
<td>1.5124 e-2</td>
<td>-3.5663 e-2</td>
<td>29.266 e-3</td>
<td>-1.297 e-3</td>
<td>0</td>
</tr>
<tr>
<td>Max End Bending</td>
<td>1 &lt; W/h ≤ 4</td>
<td>.25813</td>
<td>-4.1612</td>
<td>25.899</td>
<td>59.805</td>
<td>130.36</td>
<td>22.969</td>
</tr>
<tr>
<td>Membrane</td>
<td>1 &lt; W/h ≤ 4</td>
<td>2.1534 e-2</td>
<td>-.36438</td>
<td>2.3901</td>
<td>24.624</td>
<td>3.8348</td>
<td>-1.1820</td>
</tr>
</tbody>
</table>
Appendix L. Characteristic Length Fraction vs. Dimensionless Deflection
Curve fit to CLF vs. Dimensionless Center Deflection for determining strain in hinge elements of the Elastic-Plastic Model

The form of the equation is as follows.

\[
CLF = a\tilde{w}^5 + b\tilde{w}^4 + c\tilde{w}^3 + d\tilde{w}^2 + e\tilde{w} + f
\]

<table>
<thead>
<tr>
<th>CLF type</th>
<th>Range</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>End</td>
<td>$0 \leq W/h \leq 4$</td>
<td>2.4961</td>
<td>-3.4504</td>
<td>1.7529</td>
<td>-3.5912</td>
<td>8.3223</td>
<td>.114</td>
</tr>
<tr>
<td>Bending</td>
<td>$0 \leq W/h \leq 4$</td>
<td>-2.582</td>
<td>1.2651</td>
<td>-1.2493</td>
<td>4.1685</td>
<td>-5.5262</td>
<td>.228</td>
</tr>
<tr>
<td>Mid</td>
<td>$0 \leq W/h \leq 4$</td>
<td>-8.8117</td>
<td>8.6681</td>
<td>-1.9148</td>
<td>-3.6421</td>
<td>3.0604</td>
<td>.746</td>
</tr>
<tr>
<td>Membrane</td>
<td>$0 \leq W/h \leq 4$</td>
<td>e-4</td>
<td>e-3</td>
<td>e-2</td>
<td>e-2</td>
<td>e-3</td>
<td>e-3</td>
</tr>
</tbody>
</table>
Appendix M. Computer Program PLASTIC.C. Elastic-Plastic Response
```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

/******PROGRAM NAME: PLASTIC.C *******/
/**** This is the program for running the simplified elastic-plastic model
with hinges comprised of elastic-plastic bar elements. There are
several output files:
a) 'pdata.end.hinge' this file gives the stress and strain vs. time of
top and bottom elements of the end hinge. This can yield stress
vs. strain as well.
b) 'pdata.mid.hinge' this file gives the stress and strain vs. time of
top and bottom elements of the middle hinge. This also yields
stress-strain.
c) 'pdata.all.strs' This output a large matrix of data which gives
stress in all end hinge elements vs. time. This is for 3-D
plotting.
d) 'pdata.yoft' This yields center deflection vs. time in response to
the various pulse magnitudes.
e) 'pdata.y.final' This yields final permanent deflections to the
various pulses.
There is an array of variables for each element in each hinge to keep
track of their strain and stress. The elastic modulus (E) and
plastic tangent modulus of the elements can be specified in the
variable declarations.
The Applied load (yield pulse) length, thickness and yield stress of the
model is read from a data file as described below.
Final deflection is determined by using a newton method for converging
onto the deflection with causes Zero moment on the plate strip and
therefore its final deflection. So at some point in time, the
Newton method overrides the applied force and momentum of the
strip and induces deflection which will converge on that which
results in zero moment on the strip with no applied force. This
means the residual bending forces and membrane forces balance.
The number of elements in each hinge must be specified. Data on the
applied pulse may be read in from a file called 'yielddata'. This
is the yield pulse which causes first yield as determined from the
finite difference elastic analysis. The format of the yield pulse
data file is as follows: Length (tab) Magnitude(t-0) (tab) Tau(t)
(tab) yieldstress (tab) thickness (return)
For each yield pulse read, Tau(t) stays constant and the Magnitude is
multiplied by 2,3,4,5, and 6 times and runs the program with each
of these Multiples to determine the response. That is, to see the
response of the plate strip under pulses that are multiples of the
yield pulse.
*/

/** V A R I A B L E   D E C L A R A T I O N S **/
double *end_chi; /* switching parameter to indicate if plastic or
elastic*/
double *end_stress9; /* stress in each element at previous step */
double *end_stress0; /* current stress in each element */
```

158
double *end_eps9; /* total strain at previous time in each element */
double *end_eps0; /* total strain at current time in each element */
double *end_epsp0; /* plastic strain at current time */
double *end_epse9, *end_epse0; /* elastic strain at previous, current times */
double *end_deps, *end_depse; /* incremental; total and elastic strain */
double *end_bend; /* bending strain in end elements */
double *mid_chi; /* switching parameter to indicate if yielding */
double *mid_stress9; /* stress in each element at previous step */
double *mid_stress0; /* current stress in each element */
double *mid_eps9; /* total strain at previous time in each element */
double *mid_epsp0; /* total strain at current time in each element */
double *mid_epse9, *mid_epse0; /* plastic strain at current time */
double *mid_deps, *mid_depse; /* incremental total and elastic strain */
double *mid_bend; /* bending strain in MID elements */
double **yoft; /* Data matrix for y = y(time) */
double **resstress; /* data matrix for residual stress */
int i, j, k; /* variable used in loops */
int fibnum; /* number of last element (count bottom to top) */
int flag; /* indicator when convergence to final Y(mid) is achieved*/
int yoftflag; /* indicator if y = y(time) data should be printed to file*/
int straflag; /* indicator if elements' stresses vs. time should be printed to file*/
int resct=0;
int rownum, colnum=0; /* indices for data matrix */
double memstrain; /* strain due to membrane stretching */
double e_elas = 30.0e+6; /* elastic modulus of elasticity */
double e_plas = 0.0; /* plastic/tangent moduli */
double yield1 = 40000.0; /* stress at first yield */
double yield10; /* dummy variable used to remember previous yield stress read from data file */
double yield_ten; /* Tension Yield stress as a function of epsilon */
double yield_com; /* Compression Yield stress as a function of epsilon */
double y9, y00, yll; /* mid deflection; previous, current and forward time step */
double load; /* force applied to strip (pressure) */
double momsum9; /* previous time's total moment */
double momsum; /* total moment acting on half strip */
double end_mom; /* moment from elements at end hinge */
double mid_mom; /* moment from elements at mid hinge */
double p; /* stretching force */
double len; /* length of full strip */
double len9; /* remembers value of length for last iteration */
double mm; /* Mass Moment of inertia */
double tim; /* current time */
double tmax; /* max time or duration of run */
double delt; /* time step */
double mag; /* Magnitude of Applied load at time = 0( used in Loop */
double mag0; /* Magnitude of Applied Load to cause first yield */
double tau; /* characteristic time of applied load */
double vel = 0.0; /* initial velocity */
double f_mem; /* Char length Fraction for membrane strain */
double f_end; /* Char length Fraction for end strain */
double f_mid; /* Char length Fraction for mid-span strain */
double hh = .5; /* strip thickness */
double hh9; /* dummy variable used to remember previous
  thickness read from data file */
double wid = 1.0; /* strip width */
double fib_area; /* x-section area of each element */
double density = .283; /* density of steel in English units */
double v1, v2, v3; /* temp variables */

/** FUNCTION PROTOTYPES **/
/* allocate memory for a vector */
double* calocl(int);
/* allocate memory for a matrix */
double** caloc2(int, /* total number of rows in matrix */
      int); /* total number of columns in a matrix */
/* Yield function for tension */
double ften(double eps, /* strain */
      double e_elas, /* elastic modulus */
      double e_plas, /* plastic tangent modulus */
      double yield); /* yield stress */
/* Yield function for compression */
double fcom(double eps, /* strain */
      double e_elas, /* elastic modulus */
      double e_plas, /* plastic tangent modulus */
      double yield); /* yield stress */
/* Load as a function of time */
double fload(double tim, /* time */
      double mag, /* Load at t=0*/
      double tau); /* characteristic time of pressure pulse */
/* FILE POINTER FOR OUTPUT AND INPUT */
FILE *fp;
FILE *fp1;
FILE *fp2;
FILE *fp3;
FILE *fp4;
FILE *fp5;
main()
{
    fp=fopen("/mit/djhenke/pdata.mid.hinge","w"); /* data for middle hinge */
    fp1=fopen("/mit/djhenke/pdata.end.hinge","w"); /* data for end hinge*/
    fp3=fopen("/mit/djhenke/pdata.all.strs","w"); /* data for hinge stress history*/
    fp5=fopen("/mit/djhenke/yielddata","r"); /* Read data for first yield Pulse*/

    /*** INPUT DATA **/
    printf("Input the ID number of the last element to be used.\n");
    printf("Must be an even number. First element starts as zero\n");
    scanf("%d", &fibnum);
    printf("If you want yoft data printed type 1 if not type 0\n");
    scanf("%d", &yoftflag);
    printf("If you want hinge element stresses vs time printed to file, type 1 if not type 0\n");
    scanf("%d", &strsflag);

    /** THESE LINE ARE AVAILABLE SO INPUTING THE TIME STEP AND DURATION OF RUN CAN BE AN OPTION. IT IS AUTOMATICALLY DETERMINED FURTHER BELOW BASED ON THE NATURAL FREQUENCY OF THE MODEL **/
   /*** printf("Input the time STEP\n"); /***
   /*** scanf("%lf", &delt); /***/
   /*** printf("Input the time duration\n"); /***/
   /*** scanf("%lf", &tmax); /***/

    /* MEMORY ALLOCATION FOR ELEMENTS DATA */
    end_chi = calocl(fibnum+1);
    end_stress9 = calocl(fibnum+1);
    end_stress0 = calocl(fibnum+1);
    end_eps9 = calocl(fibnum+1);
    end_eps0 = calocl(fibnum+1);
    end_epsp0 = calocl(fibnum+1);
    end_epse9 = calocl(fibnum+1);
    end_epse0 = calocl(fibnum+1);
    end_deps = calocl(fibnum+1);
    end_depse = calocl(fibnum+1);
    end_bend = calocl(fibnum+1);

    mid_chi = calocl(fibnum+1);
    mid_stress9 = calocl(fibnum+1);
    mid_stress0 = calocl(fibnum+1);
    mid_eps9 = calocl(fibnum+1);
    mid_eps0 = calocl(fibnum+1);
    mid_epsp0 = calocl(fibnum+1);
    mid_epse9 = calocl(fibnum+1);
    mid_epse0 = calocl(fibnum+1);
    mid_deps = calocl(fibnum+1);
    mid_depse = calocl(fibnum+1);
    mid_bend = calocl(fibnum+1);
allocate memory for data matrix of y-y(t) and res.stress*

if (yoftflag=1)
    yoft = caloc2(402,10);
if (strsflag=1)
{
    resstress = caloc2 (fibnum+2,123);
    for (j=0;j<ibin;j++)
        resstress[j+1][0]=(double)j;
}

START OUTER LOOP FOR READING YIELD PULSE DATA
This reads data which are Length, the applied pulse Magnitude and time
decay constant, yield stress and thickness of plate. The format
of the data is Length (tab) Tau (tab) Magnitude (tab) Yield Stress
(tab) Thickness (return) where there are various lengths and
various combos of variables and yield pulses

NOTE: make values of last line input file >500 */
while (len<500.0)
{
    fscanf(fp5,"%lf\%lf\%lf\%lf\%lf
",&len,&mag0,&tau,&yieldl,&hh
    );
    printf("MAG0 = %f AND TAU(T) = %f LENGTH = %f
",mag0,tau,len);
    /* Duration of program run ('tmax') for each pulse is based on a factor
of natural Frequency which is function of (len*len/h). The
factor is sized so the strip cycles through a many oscillations
during the run.
*/
tmax = len*len/hh/2000.0;
/* Similarly this time step (delt) is based on Natural Freq and factored
so there are about 450 time steps for every cycle. */
delt = len*len/hh/6e+6;

if (yoft flag=1)
{
    /* PRINT HEADER INFO OUT FOR EACH SET OF Y-Y(Time) WHICH HAS
UNIQUE
LENGTH AND TAU (I.E. ONLY MAGNITUDE VARIES) */
    data for y-y(time)*/
    fp4=fopen("/mit/djhenke/pdata.yoft","a");
    fprintf(fp4,"DEFLECTION VERSUS TIME\n");
    fprintf(fp4,"#Elements\%d\tE-Elastic=\%e\tE-Plastic
\%e\n",fibnum,e_elas,e_plas);
    fprintf(fp4,"Length\%f\tThickness\%f\tTdelta-T\%f\n",len,hh,delt);
    fprintf(fp4,"Yield Stress=\%f\tVelocity(0)=\%f\n",yieldl,vel);
    fprintf(fp4,"Yield Pulse\tMag(t=0)=\%f\tTAU = \%f\n",mag0,tau);
    fclose(fp4);
if(len != len9 || hh9 != hh || yieldl9 != yieldl) {
    fprintf(fp, "MIDDLE HINGE STRESS-STRAIN HISTORY
");
    fprintf(fp, "#Elements %d E-Elastic %e E-Plastic %e
", fibnum, e_elas, e_plas);
    fprintf(fp, "Length %f Thickness %f delta-T %f
", len, hh, delt);
    fprintf(fp, "YieldStress %f V(T-0) %f
", yieldl, vel);
    fprintf(fp, "END HINGE STRESS-STRAIN HISTORY
"); 
    fprintf(fp1, "#Elements %d E-Elastic %e E-Plastic %e
", fibnum, e_elas, e_plas);
    fprintf(fp1, "Length %f Thickness %f delta-T %f
", len, hh, delt);
    fprintf(fp1, "YieldStress %f V(T=0) %f
", yieldl, vel);
    fprintf(fp3, "#Elements %d E-Elastic %e E-Plastic %e
", fibnum, e_elas, e_plas);
    fprintf(fp3, "Length %f Thickness %f delta-T %f
", len, hh, delt);
    fprintf(fp3, "YieldStress %f Velocity(0) %f
", yieldl, vel);
    fprintf(fp3, "TAU = %f
", tau);
    fclose(fp2);
}

/* data for final deflection */
fp2 = fopen("/mit/djhenke/pdata.y.final", "a");
fprintf(fp2, "FINAL DEFLECTION/THICKNESS ON PLASTIC MODEL
");
fprintf(fp2, "#Elements %d E-Elastic %e E-Plastic %e
", fibnum, e_elas, e_plas);
fprintf(fp2, "Length %f Thickness %f delta-T %f
", len, hh, delt);
fprintf(fp2, "YieldStress %f V(T=0) %f Duration %f
", yieldl, vel, tmax);
fprintf(fp2, "Length Tau(t) Yield
Mag(t=0) tMult=2 tMult=3 tMult=4 tMult=5 tMult=6
");
fclose(fp2);
}

/* The following three variables are used to remember the current
values of length, thickness and yield stress. If any of these
values change the next time data is read from data file,
'yielddata', then a new header of information will be printed to
the output file so plate strip characteristics will be specified */

len9 = len;
hh9 = hh;
yield19 = yield1;

163
/* PRINT LENGTH, TAU AND MAGNITUDE OF YIELD PULSE TO FILE */
fp2=fopen("/mit/djhenke/pdata.y.final","a");
fprintf(fp2,"%lf\t%lf\t%lf\n",len,tau,mag0);
fclose(fp2);

colnum=0; /* Initialize column number of Y(t) data matrix for this set of yield pulse data*/

/* THIS IS HERE TO PREVENT A FINAL LOOP BY USING THE DUMMY DATA AT THE END OF THE YIELD PULSE DATA FILE */
if (len<500)
{
/* BEGIN LOOP FOR VARYING MAGNITUDE OF PULSE */
/* This keeps the time constant (tau) the same */
/* and runs the E-P model with multiples of the */
/* Yield Pulse Magnitude. */
/* 'mag0' is the yield pulse read in above. */

for(mag=2.0*mag0; mag<6.0*mag0+1.0; mag+=mag0)
{
if (colnum<9) /* Dont want to print out too much data */
colnum+=1; /* increment column index for y(t) data matrix */

/* PRINT TO SCREEN AND FILE */
printf("TMAX = %f Delta T = %f\n",tmax,delt);
fprintf(fp,\"nMAG\tTAU\n");
fprintf(fp,\"%f\t%f\n",mag,tau);
fprintf(fp,"Time\tTop-strain\tTop-stress\tLow-Strain\tLow-Stress\n");
fprintf(fp1,\"nMAG\tTAU\n");
fprintf(fp1,\"%f\t%f\n",mag,tau);
fprintf(fp1,"Time\tTop-strain\tTop-stress\tLow-Strain\tLow-Stress\n");

/* Initialize Significant Variables */
resct=0.0; /* counter for elements stress data matrix */
k=0; /* initialize print statement counter */
flag=0.0; /* indicator for determining final deflection */
tim=0.0; /* time at zero */
y9=0.0; /* time of previous step */
y0=0.0; /* Y at time = 0 */
rownum=1; /* reset row # of y(t) data matrix to 1 */
fib_area = hh/((double)fibnum + 1.0); /* area of element cross section */
mm=density*wid*hh/12.0*(len*len+hh*hh); /* mass moment of inertia about end hinge*/
for(i=0;i<fibnum;i++)
{
end_eps9[i]=0.0;
end_eps0[i]=0.0;
 /*  */
mid_eps9[i]=0.0;
mid_eps0[i]=0.0;
end_stress9[i]=0.0;
end_stress0[i]=0.0;
mid_stress9[i]=0.0;
mid_stress0[i]=0.0;
end_epse9[i] =0.0;
end_epse0[i]=0.0;
mid_epse9[i]=0.0;
mid_epse0[i]=0.0;

/*PRINT OUT FIRST LINE OF ELEMENT STRESS/STRAIN DATA **/
fprintf(fp,"%f\t%f\t%f\n",tim,mid_eps0[fibnum],mid_stress0[fibnum]);
fprintf(fp,"%f\t%f\n",mid_eps0[0],mid_stress0[0]);
fprintf(fpl,"%f\t%f\n",end_eps0[fibnum],end_stress0[fibnum]);
fprintf(fpl,"%f\t%f\n",end_eps0[0],end_stress0[0]);

/*PRINT OUT FIRST LINE OF Y(t) DATA **/
if (yoftflag==1)
{
yoft[0][colnum]= mag; /* Set first value in column of y(t)== to
Magnitude*/
yoft[rownum][0]=tim; /* First column of data matrix is time */
yoft[rownum][colnum]=y00;
}

/*PRINT OUT FIRST LINE OF ELEMENTS' STRESSES DATA **/
if(strsflag==1)
{
resstress[0][resct+1]=tim;
for(i=0;i<fibnum;i++)
    resstress[i+1][resct+1]=end_stress0[i];
resct+=1;
}

/*********** FIRST TIME STEP***********/
/* total Moment(no bend/stretch resistance) */
momsum=mag*len*len/8.0;
printf("tim=%f Load=%f Y=%f \n",tim,mag,y00);
/* time step to get y(delt) */
y11=momsum/mm*delt*delt/2.0+y00+vel*delt;

/*********** BEGIN REMAINING TIME STEPPING*********/
/**** BEGIN OUTER LOOP FOR TIME STEPPING ****/
while (flag!=1) {
    tim+=delt; /* INCREMENT TIME */
}

/*** increment variables with time step ***/
y9=y00;
y00=y11;
p=0.0;
mid_mom = 0.0;
end_mom = 0.0;
for(i=0;i<fibnum; i++)
{
    end_eps9[i]=end_eps0[i];
    mid_eps9[i]=mid_eps0[i];
    end_stress9[i]=end_stress0[i];
    mid_stress9[i]=mid_stress0[i];
    end_epse9[i] = end_epse0[i];
    mid_epse9[i] = mid_epse0[i];
}

/*** Calculate Characteristic Length Fractions ***/
vl=fabs(y00)/hh;
if (vl==0.0) /* '0' in the 'pow' function doesn't work **/
{
    f_mem=.74610;
    f_end=.11378;
    f_mid=.22789;
}
else /* determine fractions from curve fit function **/
{
    v2=.74610+3.0604e-3*v1-3.6241e-3*v1*v1-1.9148e-3*v1*v1*v1;
    v3=8.6681e-4*pow(v1, 4.0)-8.8117e-5*pow(v1, 5.0);
    f_mem=v2+v3;
    v2=.22789-5.5262e-3*v1+4.1685e-2*v1*v1-1.2493e-2*v1*v1*v1;
    v3=1.2651e-3*pow(v1, 4.0)-2.582e-5*pow(v1, 5.0);
    f_mid=v2+v3;
    v2=.11378+8.3223e-4*v1-3.5912e-2*v1*v1+1.7529e-2*v1*v1*v1;
    v3=-3.4504e-3*pow(v1, 4.0)+2.4961e-4*pow(v1, 5.0);
    f_end=v2+v3;
}

/*** Calculate membrane strain ***/
v1=y00/len;
memstrain = 1.0/f_mem*(2.0*v1*v1-2.0*pow(v1, 4.0)+4.0*pow(v1, 6.0)-
10.0*pow(v1, 8.0));

/*** Calculate bending and total strain for each END element ***/
for (i=0; i<fibnum; i++)
{
    v1=y00*hh*((double)fibnum/2.0-(double)i);
    v2=f_end*len/2.0*((double)fibnum+1.0)*sqrt(y00*y00+len*len/4.0);
    v3=f_mid*len/2.0*((double)fibnum+1.0)*sqrt(y00*y00+len*len/4.0);
end_bend[i] = v1/v2;
mid_bend[i] = v1/v3;
end_eps0[i] = memstrain + end_bend[i];
mid_eps0[i] = memstrain + mid_bend[i];
}

/***
Calculate delta epsilon for each element***/
for (i=0; i< fibnum; i+=1)
{ enddeps[i] = end_eps0[i] - end_eps9[i];
  mid_deps[i] = mid_eps0[i] - mid_eps9[i];
}

/***
Calculate stress at each END ELEMENT***/
for (i=0; i< fibnum; i+=1)
{ /* begin loop thru elements */
  /*tensile yield stress at current strain (eps) */
yield_ten = ften(end_eps0[i], e_elas, e_plas, yield1);
  /*Compressive yield stress at current strain (eps) */
yield_com = fcom(end_eps0[i], e_elas, e_plas, yield1);

  /* Did we exceed tension yield at this epsilon???*/
  if (end_stress9[i] + end_deps[i] * e_elas > yield_ten)
  { end_stress0[i] = yield_ten;
    endchi[i] = 1.0;
  }

  /* Did we exceed compression yield at this epsilon???*/
  else if (end_stress9[i] + end_deps[i] * e_elas < yield_com)
  { end_stress0[i] = yield_com;
    endchi[i] = -1.0;
  }

  /* Are we still in the elastic range */
  else
  { end_stress0[i] = end_stress9[i] + end_deps[i] * e_elas;
    endchi[i] = 0.0;
  }

  /*elastic delta epsilon*/
  end_deps[i] = (end_stress0[i] - end_stress9[i]) / e_elas;

  /* current elastic strain*/
  end_epse0[i] = end_epse9[i] + end_deps[i];

  /* current plastic strain */
  end_epsp0[i] = end_eps0[i] - end_epse0[i];
} /* end loop for END ELEMENTS */
/***
Calculate stress at each MID ELEMENT***/
for (i=0; i<fibnum; i++)
{ /* begin loop thru elements */

/*tensile yield stress at current strain (eps)*/
yield_ten=ften(mid_esp0[i],e_elas,e_plas,yield1);

/*Compressive yield stress at current strain (eps)*/
yield_com=fcom(mid_esp0[i],e_elas,e_plas,yield1);

/* Did we exceed tension yield at this epsilon???*/
if(mid_stress9[i] + mid_deps[i] * e_elas > yield_ten)
{
    mid_stress0[i]=yield_ten;
    mid_chi[i]=1.0;
}
/* Did we exceed compression yield at this epsilon???*/
else if (mid_stress9[i] + mid_deps[i] * e_elas < yield_com)
{
    mid_stress0[i]=yield_com;
    mid_chi[i]=-1.0;
}
/* Are we still in the elastic range */
else
{
    mid_stress0[i] = mid_stress9[i] + middeps[i] * e_elas;
    mid_chi[i] = 0.0;
}

/*elastic delta epsilon*/
mid_depse[i]=(mid_stress0[i]-mid_stress9[i])/e_elas;

/* current elastic strain*/
mid_e0se9[i] = mid_espse9[i] + mid_depse[i];

/* current plastic strain */
mid_espse0[i] = mid_espse0[i] - mid_espse0[i];
}  /* END LOOP for MID ELEMENTS */

/*** PRINT Y(t) DATA of current time at even intervals ***/
if (yoftflag == 1)  /* if y(time) data is to be printed to file */
{
    if(tim>tmax/200.0*(double)k)
    {
        rownum+=1;
yoft[rownum][0]= tim;
yoft[rownum][colnum]=y00;
k+=1;
    }
}
/** PRINT DATA at even intervals ***/
if(tim>tmax/300.0*(double)k)
{
    vl=fload(tim,mag,tau);
    printf("tim=%f Load=%f Y=%f MOMSUM=%f
",tim,vl,y00,momsum);
    fprintf(fp,"%f	%f	%f\n",tim,mid_eps0[fibnum],mid_stress0[fibnum]);
    fprintf(fp,"%f	%f\n",mid_eps0[0],mid_stress0[0]);
    fprintf(fpl,"%f	%f
",mid_eps0[0],mid_stress0[0]);
}

/*** Calculate the stretching force from mid-span***/
for (i=0;i<=fibnum; i+=1)
{
    p=p+mid_stress0[i];
}
p*=fib_area;

/*** Calculate Moment for the end hinge and middle hinge***/
for (i=0; i<=fibnum; i+=1)
{
    end_mom+=end_stress0[i]*((double)i-(double)fibnum/2.0);
    mid_mom+=mid_stress0[i]*((double)fibnum/2.0-(double)i);
}
end_mom*=fib_area*hh/((double) fibnum+1.0);
mid_mom*=fib_area*hh/((double) fibnum+1.0);

/*** Input elements' stresses data at even intervals ***/
if (strsflag==1)
{
    if(tim>tmax/120.0*(double)resct)
    {
        resstress[0][resct+1]=tim;
        for(i=0;i<=fibnum;i+=1)
            resstress[i+1][resct+1]=end_stress0[i];
        resct+=1;
    }
}

/*** Calculate the SUM of all APPLIED MOMENTS***/
/**
if static deflection is desired, simply comment out the two lines
indicated below. Normally, in the dynamic case, the load is
removed to find the final deflection when internal residual
stresses are equilibrated. (after tmax is reached) but in the
static case, set 'delt' = 1e-6 and 'duration' = 1e-6 so maximum
time is met immediately. then the sum of moments = 0 for static
condition with the load applied; thus, we have the static
deflection under that load

**/

load = fload(tim,mag,tau);
if (tim>tmax) /* 'COMMENT OUT' FOR STATIC DEFLECTION RUN */
load = 0.0; /* 'COMMENT OUT' FOR STATIC DEFLECTION RUN */
momsum=load*len*len/8.0 - p*y00 - end_mom - mid_mom;

/**** Calculate the Y increment in time based on current stress,moments
etc.****/
yl1=momsum*delt*delt/mm+2*y00-y9;

/**** FIND FINAL DEFLECTION *****/
/* A Newton type convergence method is used to find a root of momsum =
momsum(y). i.e. find the y where sum of moments = 0 and therefore-->
final deflection with internal forces equilibrated */
if(tim>tmax)
{
k=0; /* Ensures data print to file for each loop */
/**** If y00=y9 then the slope = 0 which will be in denominator below.
Small chance with double precision but just in case...***/
if (y00==y9)
yl1=y00 + .001*hh;
else
{ /* Newton's delta y = -Y(t)/Y'(t) */
  vl=-(y00-y9)/(momsum-momsum9)*momsum;
  if (fabs(vl)>.1*hh)
    vl=.1*vl/sqrt (vl*vl)*hh;
  yl1=y00+vl;
  printf("Time=%f Y= %f Momsum=\n",tim,y00,momsum);
}
/IF CONVERGENCE TO FINAL DEFLECTION IS MET*/
if (fabs(momsum)<1e-5)
{
  fp2=fopen("/mit/djhenke/pdata.y.final","a");
  fprintf(fp2,"%lf\t",yl1/hh);
  fclose(fp2);
  printf("FINAL DEFLECTION IS \n%f\n",yl1);
  fprintf(fp,"FINAL DEFLECTION IS \n%f\n",yl1);
  fprintf(fp1,"FINAL DEFLECTION IS \n%f\n",yl1);
  flag=1;
}
} /*** END FINAL DEFLECTION DETERMINATION LOOP ***/

/* remember the last sum of moments*/
momsum9=momsum;
}
/* END OUTER LOOP FOR TIME STEPPING */
/* END MAGNITUDE MULTIPLYING LOOP */

/* Skip a line in output file: pdata.y.final */
flo2=fopen("/mit/djhenke/pdata.y.final","a");
fprintf(fp2,\"n\")
fclose(fp2);

/* PRINT Y(time) to file if yoftflag=1 */
if (yoftflag==1)
{
    fp4=fopen("/mit/djhenke/pdata.yoft","a");
    fprintf(fp4,"t");
    for (j=1;j<=colnum;j++)
        fprintf(fp4,"Magnitude t");
    fprintf(fp4,\"n t\")
    for (j=1;j<=colnum;j++)
        fprintf(fp4,"%f t",yoft0[j]);
    fprintf(fp4,\"n")
    fprintf(fp4,"Time t")
    for (j=1;j<=colnum;j++)
        fprintf(fp4,"Deflection t")
    fprintf(fp4,\"n")
    for (i=1;i<=rownum;i++)
        { 
            for (j=0;j<=colnum;j++)
                fprintf(fp4,"%f t",yoft[i][j]);
            fprintf(fp4,\"n")
        }
    fclose(fp4);
}

/* END LOOP WHICH STOPS PROGRAM BY USING DUMMY DATA */

/* END WHILE LOOP FOR READING YIELD PULSES */

/*** Print Elements' stresses history to file ***/
if (strsflag==1)
{
    fprintf(fp3,"Element number is in the left column and time is top row\n")
    for(j=0;j<=fibnum+1;j++)
    {
        fprintf(fp3,"%d t",j-1)
        for(i=1;i<=121;i++)
        {
            fprintf(fp3,"%f t",resstress[j][i])
        }
        fprintf(fp3,"\n")
    }
}
/**CLOSE THE FILES WHICH ARE STILL OPEN **/
fclose(fp);
fclose(fp1);
fclose(fp2);
fclose(fp3);
fclose(fp5);
} //********** END MAIN END MAIN **/
/****** ****************** 

/* YIELD FUNCTION FOR TENSION */
double ften(double eps, /* current total strain*/
            double e_elas, /* elastic modulus */
            double e_plas, /* plastic tangent modulus */
            double yieldl) /* yield stress */
{
    double stress;
    stress = eps*e_plas+yieldl*(1.0-e_plas/e_elas);
    return(stress);
}

/* YIELD FUNCTION FOR COMPRESSION*/
double fcom(double eps, /* strain */
            double e_elas, /* elastic modulus */
            double e_plas, /* plastic tangent modulus */
            double yieldl) /* yield stress */
{
    double stress;
    stress = eps*e_plas-yieldl* (1.0-e_plas/e_elas);
    return (stress);
}

/****** DETERMINE APPLIED LOAD*******/
double fload(double tim,
            double mag,
            double tau)
{
    double force;
    force = mag *exp(-tim/tau);
    return (force);
}

/* ALLOCATE MEMORY FOR A 1-D ARRAY */
double * calocl( int row)
{
    double * vec;
    vec = (double *)calloc(sizeof(double),row);
    return(vec);
}

/* ALLOCATE MEMORY FOR A 2-D ARRAY */
double ** caloc2(int row,int col)
{
    int i;
    double ** mat;
    mat=(double**)calloc(sizeof(double*),row);
    for (i=0;i<row-1;i++)
        mat[i]=(double*)calloc(sizeof(double),col);
    return(mat);
}
Appendix N. Static Response of E-P and Finite Difference Model to Uniform Loads
Center Deflection vs. Load.

- Elastic-Plastic Model: 60 elements per hinge
- Finite-Difference Model (300 nodes) & Elastic-Plastic Model: No Yield
- Yield = 40000 psi
- Yield = 60000 psi

Uniform Pressure (psi)

Center Deflection/Thickness
Appendix 0. E-P Model Dynamic Response with Various Numbers of Hinge Elements
Center Deflection vs. Time.
Comparison of Elastic-Plastic Model response using various number of hinge elements to demonstrate convergence.
Pulse: $360 \exp(-t/0.012)$ psi. $L=60$ in. $h=1$ in. Yield Strength = 40000 psi.
Appendix P. Finite Difference and E-P Model response to Yield Pulses.
Center Deflection vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse $37.56 \exp(-t/0.076)$ psi.
(40000 psi yield pulse for $R=40$, $h=1\"$)
Clamped End Surface Stress vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse 37.5G + \exp(-t/0.76) psi. 4000 psi yield pulse for R=40, h=1"
Center Deflection vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse $214.6 \exp(-t/0.00412)$ psi. (40000 psi yield pulse for $R=40, h=1$")
Clamped End Surface Stress vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse $214.6\times\exp(-t/0.00412)$ psi. (40000 psi yield pulse for R=40, h=1")
Center Deflection vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse $22.9 \exp(-t/0.0667) \text{ psi.}$

(40000 psi yield pulse for $R=70, h=1''$)
Clamped End Surface Stress vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse $22.9^* \exp(-t/0.067) \text{ psi}$. (40000 psi yield pulse for $R=70$, $h=1''$)
Center Deflection vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse 228 exp(-t/(0.0325)) psi.
(40,000 psi yield pulse for R=70, h=1")
Clamped End Surface Stress vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse $228^* \exp(-t/0.00325)$

(40000 psi yield pulse for $R=70, h=1^\prime$)
Center Deflection vs. Time. Comparison of Elastic responses between Finite Difference Model and Elastic-Plastic Model to Pulse $21.978 \exp(-t/0.0429)$ psi.

(40000 psi yield pulse for R=100, h=1")
Clamped End Surface Stress vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse 21.978 exp(-t/0.3029) psi. (40000 psi yield pulse for R=100, h=1")
Center Deflection vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse 219.78 \( \exp(-t/0.032) \) psi. (40000 psi yield pulse for \( R=100, h=1" \))
Clamped End Surface Stress vs. Time. Comparison of responses between Finite Difference Model and Elastic-Plastic Model to Pulse 219.78 exp(-t/.0032) psi. (40000 psi yield pulse for R=100, h=1")
Appendix Q. E-P Model Response Data to Multiples of Sample Yield Pulse
Center Deflection vs. Time. Elastic-Plastic model response to multiples of yield pulse $106\times\exp(-t/0.008)$ psi. Decay constant, $\tau = 0.008$ sec., is held constant and magnitude is multiplied.
Length = 60 in. $h=1$ in. Yield Strength = 40000 psi.
Stress vs. Time of Top Element in End Hinge.
Applied Pulse = 212 exp(-t/.008) psi. L=60 in. h=1 in.
Yield Strength = 40000 psi.
Stress vs. Time of Top Element in End Hinge.
Applied Pulse = 424 \exp(-t/0.008) \text{ psi.} \quad L=60 \text{ in.} \quad h=1 \text{ in.}
Yield Strength = 40000 \text{ psi.}
Stress-Strain History of Top Element in End Hinge.
Applied Pulse = \(424 \exp(-t/0.008)\) psi. \(L=60\) in. \(h=1\) in.
Yield Strength = 40000 psi.
Stress vs. Time of Top Element in End Hinge.

Applied Pulse = 636 exp(-t/0.008) psi. L=60 in. h=1 in.
Yield Strength = 40000 psi.
Stress-Strain History of Top Element in End Hinge.
Applied Pulse = 636 exp(-t/.008) psi. L=60 in. h=1 in.
Yield Strength = 40000 psi.
Stress in End Hinge Elements vs. Time. Response to
Multiple of Yield Pulse of 106 exp(-t/.008) psi.
Yield Strength = 40000 psi. L=60 in. h=1 in.

Applied Pulse = 2 * Yield Pulse. 212 exp(-t/.008) psi.
Stress in End Hinge Elements vs. Time. Response to Multiple of Yield Pulse of 106 exp(-t/0.008) psi. Yield Strength = 40000 psi. L=60 in. h=1 in.

Applied Pulse = 4 * Yield Pulse. 424 exp(-t/0.008) psi.

Hinge Element Number. 0 at Bottom. 60 at Top.

Applied Pulse = 6 * Yield Pulse. 636 exp(-t/.008) psi.
Stress in End Hinge Elements vs. Time. Response to Multiple of Yield Pulse of 106 exp(-t/0.008) psi.

Yield Strength = 40000 psi, L=60 in., h=1 in.

Applied Pulse = 4 * Yield Pulse, 424 exp(-t/0.008) psi, Plastic Tangent Modulus = 3.0 x 10^6 psi.
Appendix R. Illustration of Added Mass Effect on E-P Model Response.
Center Deflection vs. Time.
Comparison of Responses of E-P Model with and without added mass imposed at first peak. Added mass is equivalent to 4 time plate mass.
Applied pulse = $636 \cdot \exp(-t/0.008)$ psi. $L = 60$ in. $h=1$ in. Yield Strength = 40000 psi.
Appendix S. Rigid-Plastic vs. Elastic-Plastic Response to Applied Pulse
Elastic-Plastic Model Response. Center Deflection vs. Time for varying magnitudes of the Modulus of Elasticity, $E$. Pulse = $360 \exp(-t/0.012)$ psi. $L=100$ in. $h=1$ in. Number of Hinge Elements = 60. Yield Strength = 40000 psi.

$E=300 \times 10^6$ psi
Permanent Deflection: 1.588

$E=100 \times 10^6$ psi
Permanent Deflection: 1.531

$E=30 \times 10^6$ psi (Steel)
Permanent Deflection: 0.8525

$E=10 \times 10^6$ psi
Permanent Deflection: 0.1065

Time (sec)
Appendix T. Elastic-Plastic Analysis Results

Final Deflection vs. Multiples of Yield Pulse.
SAMPLE APPLICATION OF ELASTIC-PLASTIC MODEL RESULTS.
Final center deflection from multiplying yield pulse magnitude
while keeping associated tau constant. Elements per hinge = 60.

R = L/h = 60. Yield Strength = 60000 psi.

YIELD PULSE = 90*exp(-t/tau(t)).
Tau = .019 sec. based on sample length of 60 in. and results of elastic analysis
linking Po and tau for a given L=60 in.,
R=L/h, and yield strength.

Final deflection for
P=450*exp(-t/0.019) psi.

Final deflection for
P=360*exp(-t/0.019) psi.

Final deflection for
P=270*exp(-t/0.019) psi.

Final Deflection for
P=180*exp(-t/0.019) psi.

Yield Pulse Magnitude at time = 0. Po(tau/L, R, yield strength) psi
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

$R = \frac{L}{h} = 30$. Yield Strength = 40000 psi.
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 30. \text{ Yield Strength } = 60000 \text{ psi.} \]
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude
while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 35. \text{ Yield Strength} = 40000 \text{ psi}. \]

\[ \text{Final Center Deflection/Thickness} \]

\[ \text{Yield Pulse Magnitude at time} = 0. \text{ Po}(\tau/L, R, \text{yield strength}) \text{ psi} \]
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 35. \] Yield Strength = 60000 psi.

Graph showing final center deflection/thickness as a function of yield pulse magnitude at time 0. Po(\(\tau/L, R, \) yield strength) psi.
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated \( \tau \) constant. Elements per hinge = 60.

\( R = L/h = 40 \). Yield Strength = 40000 psi.

![Graph showing final center deflection vs yield pulse magnitude](image)

- 6 \* Yield Pulse
- 5 \* Yield Pulse
- 4 \* Yield Pulse
- 3 \* Yield Pulse
- 2 \* Yield Pulse

Yield Pulse Magnitude at time = 0. \( P_0(\tau/L, R, \text{yield strength}) \) psi
Elastic-Plastic Model Results.

Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

R = L/h = 40. Yield Strength = 6000 psi.

Yield Pulse Magnitude at time = 0. Po(tau/L, R, yield strength) psi
Elastic-Plastic Model Results.

Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

$R = L/h = 50$. Yield Strength = 40000 psi.
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

R = L/h = 50. Yield Strength = 60000 psi.

Final Center Deflection/Thickness

Yield Pulse Magnitude at time = 0. Po(\tau/L, R, yield strength) psi
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

R = L/h = 60. Yield Strength = 40000 psi.
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 60. \text{ Yield Strength} = 60000 \text{ psi}. \]

![Graph showing final center deflection against yield pulse magnitude at initial time.](image-url)
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude
while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 70. \text{ Yield Strength } = 40000 \text{ psi.} \]
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 70. \text{ Yield Strength } = 60000 \text{ psi.} \]

![Graph showing the relationship between Final Center Deflection/Thickness and Yield Pulse Magnitude at time = 0. Po(\tau/L, R, yield strength) psi.](image-url)
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 80. \text{ Yield Strength} = 40000 \text{ psi.} \]
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

$R = L/h = 80$. Yield Strength = 60000 psi.

Yield Pulse Magnitude at time = 0. $P_0(\tau/L, R, \text{yield strength})$ psi
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\( R = \frac{L}{h} = 90 \). Yield Strength = 40000 psi.
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 90. \]  Yield Strength = 60000 psi.
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[
R = L/h = 100. \text{ Yield Strength} = 40000 \text{ psi.}
\]
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated \( \tau \) constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 100 \quad \text{Yield Strength} = 60000 \text{ psi.} \]

Yield Pulse Magnitude at time = 0. \( P_0(\tau/L, R, \text{yield strength}) \text{ psi} \)
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 120. \text{ Yield Strength} = 40000 \text{ psi.} \]
Elastic-Plastic Model Results.
Final center deflection from multiplying yield pulse magnitude while keeping associated tau constant. Elements per hinge = 60.

\[ R = \frac{L}{h} = 120. \text{ Yield Strength} = 60000 \text{ psi.} \]
Appendix U. E-F Model Response Data to Peak Pulses
Appendix V. Frequency Content of Applied Pulse.
Fourier Transform Coefficients for Applied Pulse

\[ P(t) = P_0 \exp(-t/\tau) \]

where \( A = \cos(\omega t) \) coefficient \& \( B = \sin(\omega t) \) coefficient.

\[ K \cos(\omega t - \phi) = A \cos(\omega t) + B \sin(\omega t) \]

Range of model natural frequency / (1/\( \tau \)) for all peak pulses from plots of permanent deflection vs. multiples of yield pulse.