DYNAMIC RESPONSE OF GRAPHITE/EPOXY PLATES SUBJECTED TO IMPACT LOADING

by

Kiernan Francis Ryan

B.S.E., Arizona State University

(1987)

Submitted to the Department of Aeronautics and Astronautics

in partial fulfillment of

the requirements for the degree of

Master of Science

in Aeronautics and Astronautics

at the

Massachusetts Institute of Technology

February 1990

© Massachusetts Institute of Technology 1989

Signature of Author _____

Department of Aeronautics and Astronautics

September 18, 1989

Certified by _____

Professor Paul A. Lagace

Thesis Supervisor

Accepted by _____

Professor Harold Y. Wachman

Chairman, Departmental Graduate Committee

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEB 26 1990

LIBRARIES

Aero

DYNAMIC RESPONSE OF GRAPHITE/EPOXY PLATES SUBJECTED TO IMPACT LOADING

by

Kiernan Francis Ryan

Submitted to the Department of Aeronautics and Astronautics on September 18, 1989 in partial fulfillment of the requirements for the Degree of Master of Science

ABSTRACT

The dynamic response of graphite/epoxy plates subjected to impact loading was investigated experimentally and analytically. The material system used was Hercules AS4/3501-6 prepreg tape. Plates of $[0/90]_{5s}$ and $[\pm 45/0]_{2s}$ layups with a test area of 89 mm by 344 mm were tested. The dynamic response was investigated on a local level and a global level. The local level was investigated experimentally with quasi-static tests and the global level was investigated with instrumented impact tests. These tests were carried out on virgin and predamaged specimens. The damage in the predamaged specimens was introduced by impacting laminates with a 1.53 kg impactor rod with a 6.35 mm radius hemispherical tup at a velocity of 4.0 m/s. The amount of damage was determined from visual and dye-penetrant-enhanced X-ray techniques to consist of small matrix cracks on the back surface and internal damage the size of the impactor diameter. The presence of damage was found to increase the contact stiffness at the local level due to the surface of a damaged laminate being more able to conform to the shape of the indentor. This dependence on the area of contact means that the contact relation will depend on the curvature of the surface. The presence of damage was found not to affect the global structural response recorded as force and displacement histories of the center of the plate. This means that the global analysis does not need to account for the presence of local damage in laminates. The analytical study at the local level produced load versus indentation data that agrees well with the experimental results. The experimental load versus indentation data was fit to a general form of the Hertzian contact law. This relation proved to be inadequate to accurately model the experimental results. The analytical force and displacement histories do not agree well with the experimental results. However, they do capture the general shape of the histories as well as the phenomenon of the impactor losing contact with the plate and then recontacting it early in the impact event. More work needs to be done to bring the global analysis into better agreement with the experimental data.

Thesis Supervisor: Paul A. Lagace

Title: Associate Professor, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

Acknowledgements

As Prof. Paul Lagace has said, it is not the expensive equipment that makes up TELAC, but the people. I am thankful that I was able to work in this laboratory because of the many friendly people here. A special thanks to Paul for making it be this way - you do an outstanding job.

I would like to thank the UROPers. Chris Park did an excellent job on the computer programs and the operation of the ACPI system. Randy Notestine and Claudia Ranniger provided much help and suggestions during the experimental phase of the research. They did an outstanding job in the laboratory for me (except for the cleaning of my laminates with acetone) and required little supervision (which was useful when I was studying for the qualifying exam). I wish them luck on their masters work. Suzanne Starnes provided much help in the data reduction of the experimental data as well as assist in some of the experiments. Thanks for your help.

Thanks to all of the graduate students for their help and friendships. Thanks to Simon Lie, Wilson Tsang, Kevin Saeger, and Pierre Minguet for taking the time to answer my many questions. A special thanks to Pierre who provided the ground work for the new time marching routine used in the global analysis.

Finally, I would like to thank the TELAC staff and professors. Thanks to Al Supple for much help in the laboratory and opening my eyes to the real world of machine shop work. Thanks to Prof. Michael Graves who participated in many of my meetings with Paul. Thanks to Prof. John Dugundji who always had an open door to answer some of my questions.

Foreword

This work was conducted at TELAC (Technology Laboratory for Advanced Composites) of the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology.

Table of Contents

CHAPTER		PAGE
1	INTRODUCTION	19
2	PREVIOUS WORK	22
	2.1 Overview	22
	2.2 Dynamic Response	22
	2.3 Impactor/Plate Contact Relation	27
3	ANALYTICAL METHODS	31
	3.1 Analysis Overview	31
	3.2 Local Model	32
	3.3 Global Model	36
	3.3.1 Overview	36
	3.3.2 Improved Time-Marching Scheme	40
	3.4 Computer Implementation	55
	3.5 Numerical Example	57
4	EXPERIMENTAL PROCEDURES	79
	4.1 Test Matrix and Specimen Description	79
	4.2 Manufacturing Procedures	85
	4.3 Test Procedures	90
	4.3.1 Static Indentation Tests	90
	4.3.2 Impact Tests	96
	4.3.3 Damage Detection	110

Table of Contents (continued)

CHAPTER		PAGE
5	EXPERIMENTAL RESULTS	112
	5.1 Impactor/Plate Contact Relation	112
	5.1.1 Undamaged Laminates	114
	5.1.2 Damaged Laminates	126
	5.2 Dynamic Impact Response	132
	5.2.1 Undamaged Laminates	132
	5.2.2 Damaged Laminates	144
6	ANALYTICAL COMPARISON AND DISCUSSION	159
	6.1 Impactor/Plate Contact Relation	159
	6.2 Dynamic Impact Response	174
7	CONCLUSIONS AND RECOMMENDATIONS	191
REFERENCES		195
APPENDIX A:	FORTRAN Source Codes	197
APPENDIX B:	Operation of ACPI (Analysis of Composite Plate Impact)	367

List of Figures

FIGURE		PAGE
3.1	Local contact problem schematic.	33
3.2	Rigid indentor contact schematic.	34
3.3	Schematic of laminated plate impact model.	38
3.4	Illustration of linearization of the forcing function at the jth time step.	44
3.5	Impact analysis flow chart.	56
3.6	Analytical impactor/plate contact relation for a $[\pm 45/0]_{2s}$ layup of AS4/3501-6.	62
3.7	Comparison of the assumed pressure distribution with the recovered pressure distribution. (Half of the axisymmetric distributions are shown.)	64
3.8	Convergence of the recovered load for the major axis analysis with an approach of 0.05 mm.	65
3.9	Analytical force history of a $[\pm 45/0]_{2s}$ laminate using 9 x 9 modes.	69
3.10	Analytical force history of a $[\pm 45/0]_{2s}$ laminate using 13 x 13 modes.	70
3.11	Analytical force history of a $[\pm 45/0]_{2s}$ laminate using 17 x 17 modes.	71
3.12	Analytical force history of a $[\pm 45/0]_{2s}$ laminate using a 2.0 μs time step.	73
3.13	Analytical force history of a $[\pm 45/0]_{2s}$ laminate using a 1.0 μs time step.	74
3.14	Analytical force history of a $[\pm 45/0]_{2s}$ laminate using a 0.5 μs time step.	75

List of Figures (continued)

FIGURE		PAGE
3.15	Analytical displacement history of a $[\pm45/0]_{2s}$ laminate using 9 x 9 modes.	76
3.16	Analytical displacement history of a $[\pm45/0]_{2s}$ laminate using 13 x 13 modes.	77
3.17	Analytical displacement history of a $[\pm45/0]_{2s}$ laminate using 15 x 15 modes.	78
4.1	Test specimen geometry.	83
4.2	Geometry of 6-32 machined nut.	84
4.3	Cure assembly cross section.	86
4.4	Curing schedule for AS4/3501-6 graphite/epoxy.	88
4.5	Location of specimen width and thickness measurements.	89
4.6	Illustration of bonding setup for machined nut.	91
4.7	Illustration of test setup for static indentation test.	93
4.8	Illustration of specimen holder for static indentation tests of $[0/90]_{5s}$ specimens.	94
4.9	Illustration of the location of the clamping rods for the static indentation tests of $[\pm45/0]_{2s}$ specimens.	95
4.10	Illustration of specimen holding jig for impact tests.	97
4.11	Illustration of impactor unit.	98
4.12	Illustration of striker unit.	100
4.13	Illustration of experimental displacement measurement.	101
4.14	Free-body diagrams of the impactor rod.	103

List of Figures (continued)

FIGURE		PAGE
4.15	Plot of force versus time for the impactor rod hit by the striker unit.	105
4.16	Results of Fast Fourier Transform to obtain experimental frequencies of the impactor rod when struck by the striker unit.	106
4.17	Typical impact force history of a $[\pm 45/0]_{2s}$ undamaged laminate with the stress wave present.	108
4.18	Fast Fourier Transform of the force history of the $[\pm 45/0]_{2s}$ undamaged laminate (of Figure 4.17).	109
4.19	Impact force history of the $[\pm 45/0]_{2s}$ undamaged laminate with the stress wave suppressed.	111
5.1	Typical plot of force versus indentation data.	113
5.2	Typical plot of log of force versus log of indentation data.	115
5.3	Plot of load versus indentation for $[0/90]_{5s}$ specimen 1S.	116
5.4	Plot of load versus indentation for $[0/90]_{5s}$ specimen 2S.	117
5.5	Plot of load versus indentation for $[0/90]_{5s}$ specimen 3S.	118
5.6	Plot of load versus indentation and curve fit for $[0/90]_{5s}$ specimen 3S.	121
5.7	Plot of load versus indentation for $[\pm 45/0]_{2s}$ undamaged specimen 1SU.	122
5.8	Plot of load versus indentation for $[\pm 45/0]_{2s}$ undamaged specimen 2SU.	123
5.9	Plot of load versus indentation for $[\pm 45/0]_{2s}$ undamaged specimen 3SU.	124

List of Figures (continued)

FIGURE		PAGE
5.10	X-ray of a $[\pm45/0]_{2s}$ specimen impacted with a 1.53 kg impactor at 4.0 m/s.	127
5.11	Plot of load versus indentation for $[\pm45/0]_{2s}$ damaged specimen 1SD.	128
5.12	Plot of load versus indentation for $[\pm45/0]_{2s}$ damaged specimen 2SD.	129
5.13	Plot of load versus indentation for $[\pm45/0]_{2s}$ damaged specimen 3SD.	130
5.14	Filtered force history of $[0/90]_{5s}$ specimen 1I impacted at 1.5 m/s.	133
5.15	Filtered force history of $[0/90]_{5s}$ specimen 2I impacted at 1.4 m/s.	134
5.16	Filtered force history of $[0/90]_{5s}$ specimen 3I impacted at 1.5 m/s.	135
5.17	Filtered force history of $[0/90]_{5s}$ specimen without the extension rod impacted at 1.6 m/s.	137
5.18	Center displacement history of $[0/90]_{5s}$ specimen 1I impacted at 1.5 m/s.	138
5.19	Center displacement history of $[0/90]_{5s}$ specimen 2I impacted at 1.4 m/s.	139
5.20	Center displacement history of $[0/90]_{5s}$ specimen 3I impacted at 1.5 m/s.	140
5.21	Filtered force history of $[\pm45/0]_{2s}$ undamaged specimen 1IU impacted at 1.5 m/s.	141

List of Figures (continued)

FIGURE		PAGE
5.22	Filtered force history of $[\pm45/0]_{2s}$ undamaged specimen 2IU impacted at 1.4 m/s.	142
5.23	Filtered force history of $[\pm45/0]_{2s}$ undamaged specimen 3IU impacted at 1.5 m/s.	143
5.24	Filtered force history of $[\pm45/0]_{2s}$ undamaged specimen without the extension rod impacted at 1.5 m/s.	145
5.25	Center displacement history of $[\pm45/0]_{2s}$ undamaged specimen 1IU impacted at 1.5 m/s.	146
5.26	Center displacement history of $[\pm45/0]_{2s}$ undamaged specimen 2IU impacted at 1.4 m/s.	147
5.27	Center displacement history of $[\pm45/0]_{2s}$ undamaged specimen 3IU impacted at 1.5 m/s.	148
5.28	Filtered force history of $[\pm45/0]_{2s}$ specimen 4ID impacted at 3.9 m/s.	149
5.29	Filtered force history of $[\pm45/0]_{2s}$ specimen 5ID impacted at 4.0 m/s.	150
5.30	Filtered force history of $[\pm45/0]_{2s}$ specimen 6ID impacted at 4.0 m/s.	151
5.31	Filtered force history of $[\pm45/0]_{2s}$ predamaged specimen 4ID impacted at 1.4 m/s.	153
5.32	Filtered force history of $[\pm45/0]_{2s}$ predamaged specimen 5ID impacted at 1.5 m/s.	154
5.33	Filtered force history of $[\pm45/0]_{2s}$ predamaged specimen 6ID impacted at 1.5 m/s.	155

List of Figures (continued)

FIGURE		PAGE
5.34	Center displacement history of $[\pm45/0]_{2s}$ predamaged specimen 4ID impacted at 1.4 m/s.	156
5.35	Center displacement history of $[\pm45/0]_{2s}$ predamaged specimen 5ID impacted at 1.5 m/s.	157
5.36	Center displacement history of $[\pm45/0]_{2s}$ predamaged specimen 6ID impacted at 1.5 m/s.	158
6.1	Convergence of the load versus indentation curves obtained from the analysis of a $[0/90]_{5s}$ laminate.	161
6.2	Analytical results and experimental data for static indentation of $[0/90]_{5s}$ laminates.	162
6.3	Analytical results and experimental data for static indentation of $[\pm45/0]_{2s}$ laminates.	165
6.4	Illustration of (a) Analytical assumption of the deformation of the plate due to a contact load; and (b) Physical deformation of the plate due to a contact load.	168
6.5	Illustration of Hertzian loading distribution.	169
6.6	The experimental impactor/plate contact relations for a $[\pm45/0]_{2s}$ laminate and a $[0/90]_{5s}$ laminate.	170
6.7	The analytical impactor/plate contact relations for a $[\pm45/0]_{2s}$ laminate and a $[0/90]_{5s}$ laminate.	172
6.8	Experimental impactor/plate contact relations for a damaged and undamaged $[\pm45/0]_{2s}$ laminate.	173
6.9	Analytical force history of a $[0/90]_{5s}$ laminate impacted at 1.52 m/s with a 1.53 kg impactor.	177
6.10	Fast Fourier Transform of the filtered experimental force data for a $[0/90]_{5s}$ laminate - specimen 3I.	178

List of Figures (continued)

FIGURE		PAGE
6.11	Fast Fourier Transform of the analytical force data for a $[0/90]_{5s}$ laminate.	179
6.12	Analytical and experimental (specimen 3I) displacement histories of a $[0/90]_{5s}$ laminate impacted at 1.52 m/s with a 1.53 kg impactor.	181
6.13	Analytical force history of a $[\pm45/0]_{2s}$ laminate impacted at 1.46 m/s with a 1.53 kg impactor.	183
6.14	Fast Fourier Transform of the filtered experimental force data for a $[\pm45/0]_{2s}$ laminate - test of specimen without the extension rod.	185
6.15	Fast Fourier Transform of the analytical force data for a $[\pm45/0]_{2s}$ laminate.	186
6.16	Analytical and experimental (test of specimen without the extension rod) displacement histories of a $[\pm45/0]_{2s}$ laminate impacted at 1.5 m/s with a 1.53 kg impactor.	187

APPENDIX FIGURE

B.1	Impact analysis flow chart.	368

List of Tables

TABLE		PAGE
3.1	AS4/3501-6 Graphite/Epoxy Ply Properties	59
3.2	$[\pm 45/0]_{2s}$ AS4/3501-6 Equivalent Engineering Properties	60
3.3	Major Axis and Minor Axis Material Property Inputs	61
3.4	Reduced Data From the Local Model Analysis	66
3.5	$[\pm 45/0]_{2s}$ AS4/3501-6 Plate Bending and Shear Properties	68
4.1	Quasi-Static Test Matrix	80
4.2	Impact Test Matrix	82
4.3	Impactor Rod Natural Frequencies	107
5.1	Reduced Data From Experimental Static Indentation Tests of $[0/90]_{5s}$ Specimens	119
5.2	Reduced Data From Experimental Static Indentation Tests of $[\pm 45/0]_{2s}$ Undamaged Specimens	125
5.3	Reduced Data From Experimental Static Indentation Tests of $[\pm 45/0]_{2s}$ Damaged Specimens	131
6.1	Inputs for Local Analysis of $[0/90]_{5s}$ AS4/3501-6 Laminates	160
6.2	Inputs for Local Analysis of $[\pm 45/0]_{2s}$ AS4/3501-6 Laminates	164
6.3	Inputs for Global Analysis of $[0/90]_{5s}$ AS4/3501-6 Laminates	175
6.4	Inputs for Global Analysis of $[\pm 45/0]_{2s}$ AS4/3501-6 Laminates	182

Nomenclature

a	acceleration of the impactor; length of the plate in x-direction
a_i	constants in Eqs. (3.18) and (3.19)
A	constant of integration; constant in the beam functions
b	length of the plate in y-direction
b_i	constants in Eq. (3.19)
A_m	constants in Eq. (3.3)
B	constant of integration; constant in the beam functions
B_m	constants in Eq. (3.3)
c_i	constants in Eq. (3.27)
$\{C\}$	vector of time-varying displacement modal amplitudes
$\{\ddot{C}\}$	vector of time-varying acceleration modal amplitudes
C_m	constants in Eq. (3.3)
CW	scalar defined by Eq. (3.59)
d_i	constants in Eq. (3.27)
D_m	constants in Eq. (3.3)
f_m	function defined by Eq. (3.3)
F_{FT}	force at the force transducer
F_{Tot}	force at tip of tup
g_m	function defined by Eq. (3.1)
K	contact stiffness

Nomenclature (continued)

$[K]$	stiffness matrix defined by Eq. (3.11)
$[K^*_{cc}]$	statically condensed stiffness matrix of plate
m	mass of the impactor; number of modes in the x-direction for the global analysis
m_k	beam function in x-direction
$[M]$	mass matrix of plate
M_{FT}	mass of the force transducer
M_{Rod}	mass of the impactor rod
M_{Tup}	mass of the the tup
n	number of modes in the y-direction for the global analysis; exponent in the contact relation of Eq. (2.1)
n_k	beam function in y-direction
r	radial coordinate
R	contact force
$\{R_c\}$	forcing function vector
R_d	radius of plate analyzed in the local model
s_1	constant found from the material properties of the plate
s_2	constant found from the material properties of the plate
t	time
t^j	time at the jth time step
u	displacement of the top surface of the plate under the indentor; displacement of the impactor

Nomenclature (continued)

$\dot{u}$	velocity of the impactor
$\ddot{u}$	acceleration of the impactor
w	displacement of the bottom surface of the plate under the indentor; displacement of the midplane of the plate under the impactor in the global analysis
$\dot{w}$	velocity of plate under the impactor
W_k	product of the kth beam function in the x-direction and the kth beam function in the y-direction
$\{Y\}$	modal amplitude defined by Eq. (3.12)
$\{\dot{Y}\}$	first time derivative of $\{Y\}$
$\{\ddot{Y}\}$	second time derivative of $\{Y\}$
α	local indentation of plate
β^j	scalar defined by Eq. (3.24)
β_k	constants in the beam functions
Δt	size of time step
ε	a small positive number
θ	constant in the beam functions
μ_m	roots of the Bessel function of the first kind
ξ_1	dimensionless coordinate of plate in x-direction
ξ_2	dimensionless coordinate of plate in y-direction

Nomenclature (continued)

$\{\phi\}$	eigenvector satisfying Eq. (3.17)
ω	eigenvalue satisfying Eq. (3.17)
ω_m	dimensionless quantity defined by Eq. (3.2)

CHAPTER 1

Introduction

The aircraft industry has been increasing its use of advanced composite materials for secondary and primary load-bearing members. The General Dynamics F-111 has a composite horizontal tail and fuselage section. The McDonnell Douglas F-15 has composite vertical and horizontal stabilizers. Recently two airframes have been designed and constructed entirely with advanced composite materials. They are the Beechcraft Starship I aircraft and the Boeing Model 360 helicopter.

One major concern in the design of composite structures is their response to impact. Their response differs from metals because of the orthotropic and laminated nature of composite materials. For example, a composite part may be impacted and show no visible damage, yet have internal damage which significantly reduces the load-carrying capacity of the part. A metal part, on the other hand, will typically have externally visible damage. Since aircraft structures are subjected to impacts such as tool drop, runway kickup, and bird strikes, it is important for engineers to have an understanding of the response of composite materials to such events. Engineers designing and analyzing these structures need analyses which would aid them in designing parts to meet any specified requirements related to impact.

A number of investigators have previously done work on developing such analyses. These analyses start by predicting the dynamic response of the composite structure to an impact event. The results of the dynamic response

analysis coupled with other appropriate analyses may be used to predict damage in the structure and, ultimately, the residual strength.

The response of the structure to an impact is assumed to occur at two levels - a global response of the structure and a local response under the point of impact. The analysis and experiments are thus broken down into these two segments. The global response is experimentally investigated by dynamic tests. Force and displacement time histories are experimentally measured during these tests. The local response is investigated by quasi-static tests.

The present research effort centers on experimentally investigating the dynamic response of an impacted composite plate. Specifically, the effect of damage on the dynamic response is investigated. One of the reasons for investigating this is because existing analyses generally assume that damage does not affect the local or global response during the event. The validity of this assumption has not been previously investigated. The experiments in the present research effort are designed such that the specimen remains undamaged. These experiments are conducted with virgin specimens and specimens which have been damaged from a previous impact (an impact with enough energy such that damage results). The results from these tests are compared to determine the effects of damage. This should provide insight to the physics of the impact event, as well as allow a check on the validity of the no damage assumption in existing analyses.

In Chapter 2, previous work done in the area of impact to composite plates is discussed. In Chapter 3, the analytical methods are described, and the differences between the present analyses and a previous analysis is highlighted. In Chapter 4, the methods used to manufacture the test specimens and conduct the experiments are described. In Chapter 5, the experimental results are presented. In Chapter 6, the analytical results are presented and

discussed in light of the experimental results. In Chapter 7, the conclusions and recommendations are given. Finally, the FORTRAN source code is contained in Appendix A and an outline on how to use the codes in Appendix B.

CHAPTER 2

Previous Work

2.1 Overview

The impact of a foreign object to a composite plate is a complex event occurring over a very short period of time (on the order of milliseconds). For this reason, it is convenient to break the event down and analyze one part at a time. The impact event may be separated into the three following areas [1]: the dynamic behavior of the composite plate during impact, the local damage induced from the impact (damage resistance), and the post-impact behavior of the plate (damage tolerance). Research has been conducted in all three of these areas. A good understanding of each is needed to fully understand the impact event.

Since the current research is solely concerned with the dynamic response of impacted composite plates, the following review is restricted to this area. Following the section on the dynamic response is a section on the impactor/plate contact relation. This is an important part of the dynamic response problem describing the force interaction between the impactor and the plate.

2.2 Dynamic Response

In order to accurately predict the stresses and strains in a composite plate subjected to impact, one must consider the dynamic response of the plate.

Many researchers have developed models to predict this response [1-13]. Some of the more prominent work, as well as recent developments, are discussed in this section.

Yang, Norris and Stavsky [2] deduced a two-dimensional linear theory of the motion of heterogeneous plates from the three-dimensional theory of elasticity. They included transverse shear deformations and rotary inertia in their formulation of the plate theory. Although they were investigating the propagation of elastic waves in a heterogeneous plate, the governing equations of the free vibration of the plate are the same as for the problem of impact. Understanding the free vibration behavior is a prerequisite to understanding the forced vibration resulting from impact loadings.

Whitney and Pagano [3] investigated the theory developed by Yang et al. as applied to laminated plates. They showed that the effect of shear deformation can be significant in laminated plates with a length-to-thickness ratio as high as 20. This influence is much greater than in isotropic plates and cannot be neglected in one's analysis of impact of composite plates.

Greszczuk [4] conducted some of the first work in predicting damage due to impact. He determined the pressure distribution on an elastic half-plane due to an indentor by analytically combining the static solution for the pressure between two bodies in contact and the dynamic solution of their impact. He used the resulting time-varying surface pressure with a standard finite-element code to predict the stresses and strains in the composite material. The results show that the largest contact stresses are under the point of impact within the composite, and that the critical stress is usually shear. Since Greszczuk models the composite plate as a half-space, it is limited to thick laminates subjected to low velocity impact. The analysis also neglects

global bending of the laminate and inertial forces. These effects need to be included to accurately model the dynamics of a thin composite plate.

Sun and Chattopadhyay [5] investigated the impact response of symmetric cross-ply laminates under initial stress. They obtained the contact force and the dynamic response of the plate by solving a nonlinear integral equation by means of a small-time increment numerical method. Their results showed that a laminate with a higher initial tensile stress had a larger maximum contact force, but smaller contact time, deflection, and stresses. Experimental tests were not done to verify their results. Such experiments need to be conducted to justify their analysis.

Caprino et al. [6] developed a simple analytical model to predict the response of a laminate under low velocity impact. Their model was based on energy considerations and used linear elastic properties of the laminate obtained from static tests. They impacted quasi-isotropic glass/polyester panels using the drop-weight method and obtained force versus time plots. They obtained good agreement between the model and experimental results for impact velocities below 2.5 m/s with a 4.9 kg impactor. The experimental data deviated from the analysis at higher impactor velocities. To allow for higher impact velocities, an analysis needs to be developed to account for the nonlinear contact behavior between the impactor and the laminate.

Shivakumar et al. [7] also developed a simple analytical model to predict the low-velocity impact response of laminates. They modeled the plate and the impactor as rigid masses and their deformations by springs. This model is very simple and gives an estimate of the force history. To obtain a better prediction of the maximum force, an energy balance model was developed. This model equates the kinetic energy of the impactor to the sum of the strain energies due to contact, bending, transverse shear, and membrane deformations of

the plate at maximum deflection, They compared the results of their analyses with reported data and obtained reasonable results for low impact energies. The nonlinear impactor/plate contact relation needs to be modeled for better agreement at higher impact energies.

Sun and Chen [8] used finite elements to investigate the impact response of initially stressed composite laminates. One of their important considerations is the contact force between the impactor and plate. Many investigators have used linear springs to calculate the contact force. However, the contact between the impactor and the plate is nonlinear. Sun and Chen chose to use the Hertzian law of contact during the initial loading, and an experimentally established contact law during unloading and reloading to account for the permanent indentation generated in the composite plates. The effects of impactor velocity, mass, and size, and the initial pre-stress of the plate were investigated with their computer program. Experimental impact tests were not conducted to verify their analysis.

Graves and Koontz [9] developed approximate closed-form solutions for orthotropic plates (for three different boundary conditions) subjected to impact loadings. A FORTRAN code was developed to allow numerous studies to be conducted. Experimental impact tests were also conducted to determine threshold values of damage initiation. The analysis was then correlated with the experimental results so that predictions of damage initiation in other laminates could be made. The analysis is capable of predicting the dynamic response, however, it was not experimentally verified during the tests.

Aggour and Sun [10] used finite elements to investigate the dynamic response of impacted composite plates. They used a two-dimensional finite element analysis which included the effects of transverse shear deformation and rotary inertia. They chose to use the Hertzian law of contact to determine

the contact force and did their time marching with Newmark's direct integration technique. They compared the results of their analysis with published experimental results. The analysis is reasonable for the low energy impact that they were investigating. However, they modified the dimensions of the plate which they input to their computer code on the basis that their experimental boundary conditions were not perfectly clamped. In addition, the analysis deviates from the experimental results as time increases. They suggest this to occur because delaminations may have initiated. They do not obtain experimental results of an impact event in which damage to the laminate does not occur. Such experimental data is needed to verify an analysis which assumes that the laminate does not fail.

Wu and Springer [11] have developed a three-dimensional finite-element analysis to predict the dynamic response of an impacted composite plate. They also developed a method based on the concept of dimensional analysis to predict the size and locations of delaminations resulting from impact. Experimental force and strain histories of impacted composite plates were obtained and compared to their numerical results. They also compared their predictions of delaminations with experimental results [12] and found reasonable agreement. Since they are using a three-dimensional finite-element analysis, their method is computationally intense. It is desirable to develop a model that may be used as an engineering tool, as well as properly model the impact event.

Cairns and Lagace [13] used a Rayleigh-Ritz energy method to develop the equations of motion of an impacted composite plate. Beam functions were used in the x and y directions to generate plate functions that satisfy both the geometric and force boundary conditions. The Hertzian law of contact was used to give the contact force, and the Newmark integration scheme was used

to march the solution through time. Cairns conducted impact experiments but did not obtain force or strain data to compare with his analysis. Experimental data needs to be obtained to verify this analysis.

Although much has been learned about the dynamic response of impacted composite plates, there is still much research which needs to be done. Composite plates have been found to behave much differently from metal plates, for which there is a vast data base. Rotary inertia, transverse shear deformation, and bending-twisting coupling are important properties of composite plates which need to be modeled. Two and three-dimensional finite element analyses have been developed to predict the dynamic response of composite plates. The three-dimensional analysis is computationally intense and expensive. The two-dimensional finite element analyses are more efficient, but do not provide adequate detail of the stress state at the point of impact. Analyses need to be developed which can be used as an engineering tool. This means that they need to be accurate and fast. Experimental tests also need to be developed which can provide data for comparison with researcher's analyses. Thus far, force and strain gage data have been used. However, more of this data is needed.

2.3 Impactor/Plate Contact Relation

One of the important considerations in the prediction of the dynamic response of an impacted composite plate is the contact relation between the impactor and plate. As previously mentioned, Sun and Chen [8] chose to use an experimentally established contact law which changes form during loading, unloading, and reloading. Most other researchers idealized the contact

relation to be linear or to follow the Hertzian law of contact for all phases. Regardless of the method, a relationship relating the force between the impactor and the plate to the amount the plate locally indents must be established.

Sun [14-15] has studied the indentation law extensively. Yang and Sun [14] conducted static indentation tests on graphite/epoxy laminates using hemispherical steel indentors. They fit their data to the equation

$$R = K\alpha^n \tag{2.1}$$

where R is the contact force, α is the local indentation, K is a constant whose value depends on the constitutive properties of the plate and the indenter, and the radius of the hemisphere, and n is a constant approximately equal to 1.5. Their data indicated that a power of n equal to 1.5 is valid. In addition, they observed permanent indentations in the laminates after the tests. This indicated that the loading and unloading curves are different. Thus, they proposed more complicated expressions to model the load versus indentation relation more accurately. The equation used to model the unloading matched the experimental results well. It requires measuring the permanent indentation sustained in the laminate, which is not an easy task. Another difficulty in their experiments was that the indentation was measured with a dial gage after small steps in load. This lead to errors because of the creep effect, i.e. the indentation slightly changes during the 10 - 20 seconds after the load was increased by a step.

Tan and Sun [15] improved the experimental measurement of the indentation and applied their static indentation laws to the low-velocity impact response of graphite/epoxy plates. The indentation was measured with a

Linear Variable Differential Transformer (LVDT) to overcome the drawbacks of a dial gage. They used their experimental results to develop an empirical contact law which they used in a finite element analysis of the impact event.

To overcome the drawbacks of an empirical contact relation, some researchers have developed analytical methods to predict the load versus indentation relation. Sankar [16] has done this for a transversely isotropic circular plate. Two integral transform methods are used to solve the problem. The first is the point matching technique and the second is the method of assumed stress distribution. The results from this analysis show that the contact stresses deviate considerably from the Hertzian solution when the contact area is large. The contact stresses in the central portion of the contact region were found to decrease and the stresses at the edge of the contact region were found to peak as the contact radius increased. They did not conduct experiments, so they could not verify their methods.

Cairns and Lagace [17] also developed an analytical method to determine the load versus indentation relation for a transversely isotropic plate. They used a stress function approach allowing the localized stresses and strains in the plate to be determined in addition to the contact relation. This is useful information which may be used in conjunction with a failure criteria to predict damage initiation in the plate. Experiments need to be conducted to verify this analysis.

An analytical method to predict the impactor/plate contact relation is desirable because of the infinite combinations of layup, material, and indentors. Tests need to be conducted to verify the analyses developed. Since an impact to a structure is likely to occur in a location once, reloading laws are generally unnecessary. An item that does need to be looked at, however, is the effect of damage on the contact relation. This is because the composite plate

may develop damage during impact. If it does, then the contact relation may change.

CHAPTER 3

Analytical Methods

3.1 Analysis Overview

As mentioned in Chapter 2, the impact event is considered to be separable into three phases: dynamic response, subsequent damage produced, and residual strength. The dynamic response analysis provides the forces, accelerations, and displacements which occur during the impact event. This information may be used to predict the resulting damage. The information on the damage state may then be used to predict the residual strength for a given model. Engineering models for each of these phases have been developed by Cairns [1].

The present analysis deals solely with the dynamic response phase. The dynamic response is broken down into two levels, a global level and a local level. The global level deals with the global structural response of the composite plate subjected to a time-varying point load. The local model deals with the small region of the plate around the impact zone. It has two uses. The first is to determine the impactor/plate contact relation needed in the global analysis. The second is to calculate the stresses and strains in the plate using the results from the global analysis. This part of the analysis gives the information necessary to predict the damage in the plate.

The development of the global and local models are presented in reference [1]. Kraft [18] made some changes to the models to make them more computationally efficient. Additional changes to the solution procedure of the

global model are adopted in this effort to further improve the computational efficiency. These changes do not significantly alter the results from the original model.

3.2 Local Model

The local model is generally used to determine both the impactor/plate contact relation and the stresses and strains in the impact region. In the current work, only the former capability is needed. The impactor/plate contact relation is needed for the global analysis. The local analysis used in this study is the same as the one developed by Cairns and Lagace [17]. The modifications made by Kraft [18] are also used. An overview of the model is presented in the subsequent paragraphs.

The model deals with a cylindrical section, around the point of impact, of the composite plate as illustrated in Figure 3.1. A number of assumptions about this cylindrical section are made to simplify the problem. The section is assumed to be transversely isotropic in the r-θ plane. This means that the mechanical properties are equal in all directions within the r-θ plane. Small strains and axisymmetric deformations are assumed, and body forces are assumed to be negligible. The analysis also assumes the indentation geometry shown in Figure 3.2 (where the indentation is greatly exaggerated). The significance of this geometric assumption is discussed in Chapter 6.

The analysis centers on the development of a stress function which satisfies equilibrium and compatibility. The stress function is assumed to be an infinite sum of the product of a function of r and a function of z. The function of r is assumed to be harmonics of a Bessel function of the first kind:

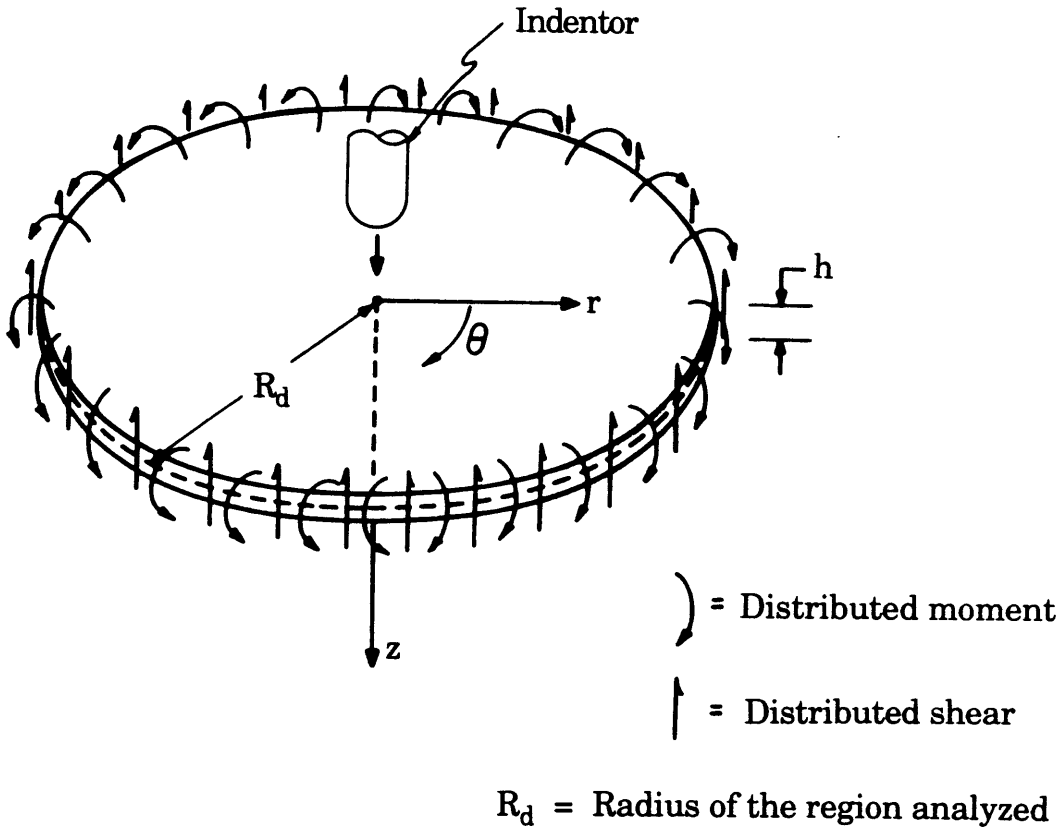


FIGURE 3.1 Local contact problem schematic.

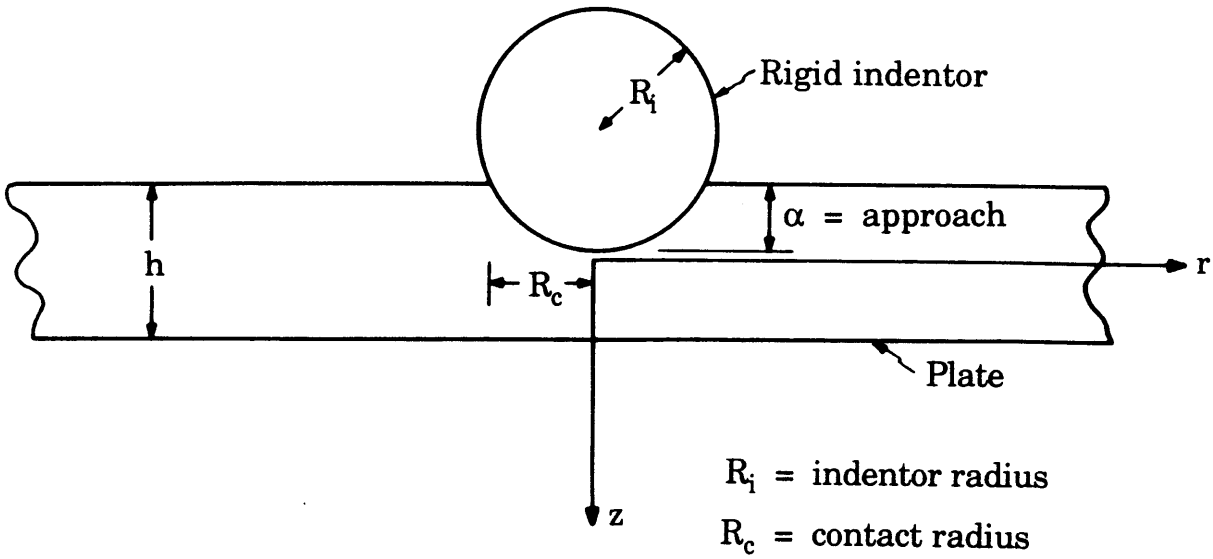


FIGURE 3.2 Rigid indentor contact schematic.

$$g_m(r) = J_0(\omega_m r) \qquad m = 1,2,3,\ldots \tag{3.1}$$

where

$$\omega_m = \frac{\mu_m}{R_d} \tag{3.2}$$

and μ_m are the roots of the Bessel function and R_d is the radius of the cylindrical section. The function of z is chosen to be a sum of hyperbolic sines and hyperbolic cosines:

$$\begin{aligned} f_m(z) = {} & A_m \sinh(s_1\omega_m z) + B_m \cosh(s_1\omega_m z) \\ & + C_m \sinh(s_2\omega_m z) + D_m \cosh(s_2\omega_m z) \\ & m = 1,2,3,\ldots \end{aligned} \tag{3.3}$$

where A_m, B_m, C_m, and D_m are constants determined from the boundary conditions, s_1 and s_2 are constants which depend on the material properties of the plate, and ω_m is defined in Eq. (3.2). The pressure distribution due to the contact between the indentor and the plate is assumed to be Hertzian. It is expanded as a Fourier-Bessel series to be compatible with the form of the stress function. This allows the unknown constants to be determined from the boundary conditions. Once the stress function is obtained, the stresses and the deformation of the plate may be determined.

The analysis has been used successfully with a failure criteria, by Cairns [1], to predict damage. The peak load and corresponding acceleration

from the global analysis of an impact event were used to define the state of the plate for the local analysis. The state of strain was determined and the failure criteria was applied. Experiments were conducted to provide data for comparison with the analysis. Specimens were impacted and the damage state determined from C-scan and X-ray techniques. The analysis compared well with the experimental results.

3.3 Global Model

3.3.1 Overview

The global model is used to predict the dynamic response of a laminated composite plate subjected to a time-varying point load. The impactor/plate contact relation discused in Section 3.2 is used to determine the values of this point load. This is done by determining the relative displacement of the impactor and the plate. The distance which the impactor penetrates the plate is the indentation which corresponds to the indentation of the local analysis. The load required to produce this indentation is determined from the local analysis and used in the global analysis. The global model used here is very similar to that of Cairns and Lagace [13]. The changes made by Kraft [18] to the model are also adopted in this study. The time-marching routine has been completely changed in the present study to improve the computational efficiency.

A number of assumptions are made in the global model to simplify the analysis. The impactor is assumed to be rigid and hemispherical. The plate is assumed to be rectangular, monoclinic, and undamaged in respect to determining the dynamic response. (The validity of this last assumption is experi-

mentally investigated.) Stretching of the plate is assumed to be negligible. The impact is assumed to be normal to the plate. Finally, the impactor/plate contact relation given by

$$R = K\alpha^n \tag{2.1}$$

is assumed to apply.

It should be noted that the analysis does take into account several of the important items discused in Section 2.2. Many of these items arise because the plate is made of plies oriented at various angles, which introduces bending-twisting coupling. It is also possible to introduce couplings such as stretching-bending and stretching-twisting. These latter type of couplings do not arise in monoclinic plates and are not considered here. Bending-twisting coupling of the plate is accounted for in the analysis. Shearing deformation is also important in laminated composites [3] and is accounted for in this analysis. Finally, the presence of in-plane loads is permitted. Although in-plane loads were not applied to any of the specimens during the present study, they were included in Cairns' development of the model and are accounted for in the computer code. They are simply set to zero in all of the present studies.

A schematic of the global model is shown in Figure 3.3. The coordinate system is chosen such that the x-y plane coincides with the midplane of the plate and the z-axis points down. The plate has length a, width b, and thickness h. The impactor has mass m and displacement u. It may strike the plate at any location in a normal direction.

An assumed modes Rayleigh-Ritz energy method is used to develop the equations of motion. The displacements are assumed to be separable in x and y. Thus, beam functions [19] are assumed for the displacement functions.

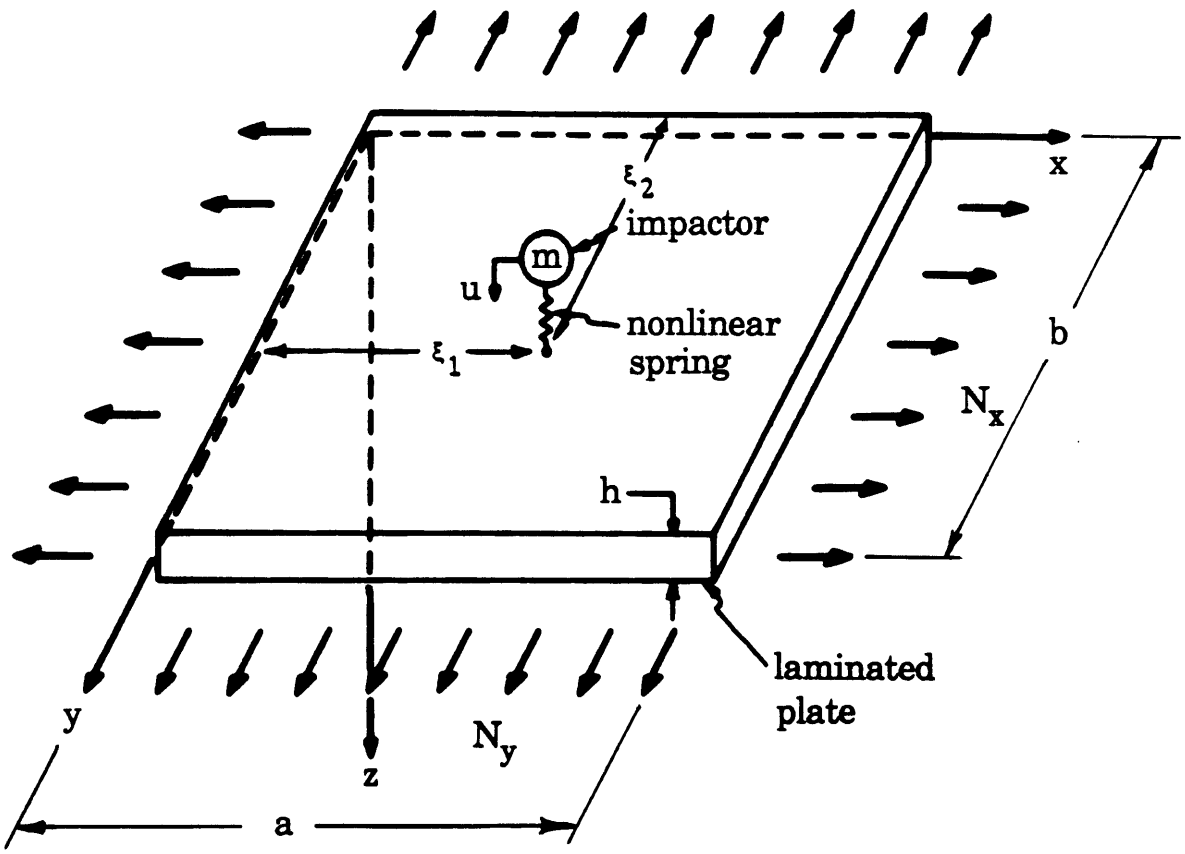


FIGURE 3.3 Schematic of laminated plate impact model.

These beam functions satisfy the force boundary conditions as well as the geometric boundary conditions and are given as:

$$m_k(x) = \sqrt{2}\sin\left(\beta_k \frac{x}{a} + \theta\right) + A\exp\left(-\beta_k \frac{x}{a}\right) + B\exp\left[-\beta_k\left(1 - \frac{x}{a}\right)\right]$$

$$k = 1, 2, 3, \ldots \qquad (3.4)$$

$$n_k(y) = \sqrt{2}\sin\left(\beta_k \frac{y}{b} + \theta\right) + A\exp\left(-\beta_k \frac{y}{b}\right) + B\exp\left[-\beta_k\left(1 - \frac{y}{b}\right)\right]$$

$$k = 1, 2, 3, \ldots \qquad (3.5)$$

where β_k, θ, A, and B are constants which depend on the boundary conditions. The displacement is constructed as the summation of the product of a time-varying modal amplitude, a beam function in the x-direction, and a beam function in the y-direction. The summation is over the product of the number of beam functions in the x-direction and the y-direction. (The variables m and n are used to denote the number of beam functions in the x-direction and the y-direction, respectively.) The functions representing the rotations are given in a similar manner. The beam functions and the first derivative of the beam functions are used so that Kirchoff plate theory solutions are recovered as the plate thickness approaches zero. The equations of motion of the composite plate are developed by forming the Lagrangian and then substituting into Lagrange's equations. The resulting equations are statically condensed into the following form:

$$[M]\{\ddot{C}\} + [K^*_{cc}]\{C\} = \{R_c\} \qquad (3.6)$$

where [M] is the mass matrix, $[K^*_{cc}]$ is the statically condensed stiffness matrix, $\{R_c\}$ is the forcing function, and {C} is the time-varying displacement modal amplitudes. The matrices are (m times n) by (m times n) in size and the vectors are (m times n) long. A dot (·) over a variable represents differentiation with respect to time (d/dt) and two dots (··) represents differentiation with respect to time twice (d^2/dt^2). The equation of motion of the impactor is given by:

$$-m\ddot{u} = K\alpha^n \tag{3.7}$$

where m is the mass of the impactor and u is its displacement in the z-direction.

3.3.2 Improved Time-Marching Scheme

A time-marching solution technique is needed to solve the above equations of motion. The Newmark implicit integration scheme was chosen by Cairns [1] because it is easy to implement and unconditionally stable. However, the time step must be chosen small enough to yield good results. Thus, analyses that have a large stiffness matrix and a large number of time steps require much computation time. For this reason, a faster time-marching solution technique is developed here. This technique is outlined subsequently.

The equations of motion of the plate given by Eq. (3.6) may be written as:

$$[M]^{1/2}[M]^{1/2}\{\ddot{C}\} + [K^*_{cc}]\{C\} = \{R_c\} \tag{3.8}$$

Premultiplying this equation by $[M]^{-1/2}$ gives:

$$[M]^{1/2}\{\ddot{C}\} + [M]^{-1/2}[K^*_{cc}]\{C\} = [M]^{-1/2}\{R_c\} \tag{3.9}$$

and multiplying by [I] in the form of $[M]^{-1/2}[M]^{1/2}$ results in:

$$[M]^{1/2}\{\ddot{C}\} + [M]^{-1/2}[K^*_{cc}][M]^{-1/2}[M]^{1/2}\{C\} = [M]^{-1/2}\{R_c\} \tag{3.10}$$

For convenience, define

$$[K] = [M]^{-1/2}[K^*_{cc}][M]^{-1/2} \tag{3.11}$$

$$\{Y\} = [M]^{1/2}\{C\} \tag{3.12}$$

$$\{\ddot{Y}\} = [M]^{1/2}\{\ddot{C}\} \tag{3.13}$$

such that Eq. (3.10) may be written as

$$\{\ddot{Y}\} + [K]\{Y\} = [M]^{-1/2}\{R_c\} \tag{3.14}$$

This is the form of the equation of motion which shall be worked with. The new variable is {Y}. After solving for {Y}, {C} may be determined from Eq. (3.12).

To solve Eq. (3.14), the solutions to the homogeneous equation and the particular equation must be determined. The homogeneous equation is given by:

$$\{\ddot{Y}\} + [K]\{Y\} = \{0\} \tag{3.15}$$

The well-known solution to this equation is

$$\{Y\} = \{\phi\}e^{j\omega t} \tag{3.16}$$

where $\{\phi\}$ is a vector consisting of constants, and ω is the eigenvalue. When substituting this into Eq. (3.15), the following eigenvalue problem results:

$$[K]\{\phi\}^{(i)} = \omega_i^2\{\phi\}^{(i)} \qquad i=1,2,\ldots,mn \tag{3.17}$$

The eigenvalues and the corresponding eigenvectors are determined from Eq. (3.17). The eigenvectors are orthogonal due to the symmetry of the matrix [K]. The eigenvectors are made to have length of unity. Since Eq. (3.15) is linear, the homogeneous solution is a superposition of each of the individual solutions:

$$\{Y\} = \sum_{i=1}^{mn} a_i\{\phi\}^{(i)} e^{j\omega_i t} \tag{3.18}$$

which may be written in terms of trigonometric functions as:

$$\{Y\} = \sum_{i=1}^{mn}\left[a_i\cos(\omega_i t) + b_i\sin(\omega_i t)\right]\{\phi\}^{(i)} \tag{3.19}$$

The particular solution of Eq. (3.14) must also be determined. The approach utilized here is to linearize the forcing function at each time step as illustrated in Figure 3.4. The actual forcing function is nonlinear and is the curved line in the figure. A tangent at the jth time step is drawn to the forcing function. This is the linearized approximation. The size of the time step is limited by how well the linear approximation models the actual forcing function. Recall that the contact force is given by

$$R = K\alpha^n \tag{2.1}$$

where the approach (α) is the distance shown in Figure 3.2. It may be calculated as the relative displacement of the top and bottom surfaces of the plate:

$$\alpha = u - w \tag{3.20}$$

where u is the displacement of the indentor measured from the point at which it just touches the plate, and w is the displacement of the back surface of the plate under the indentor. Eq. (2.1), with the expression for the approach given in Eq. (3.20), is linearized by taking the first term in the Taylor series expansion:

$$R^{j+1} = R^j + nK\left(u^j - w^j\right)^{n-1}\left(\dot{u}^j - \dot{w}^j\right)\Delta t \tag{3.21}$$

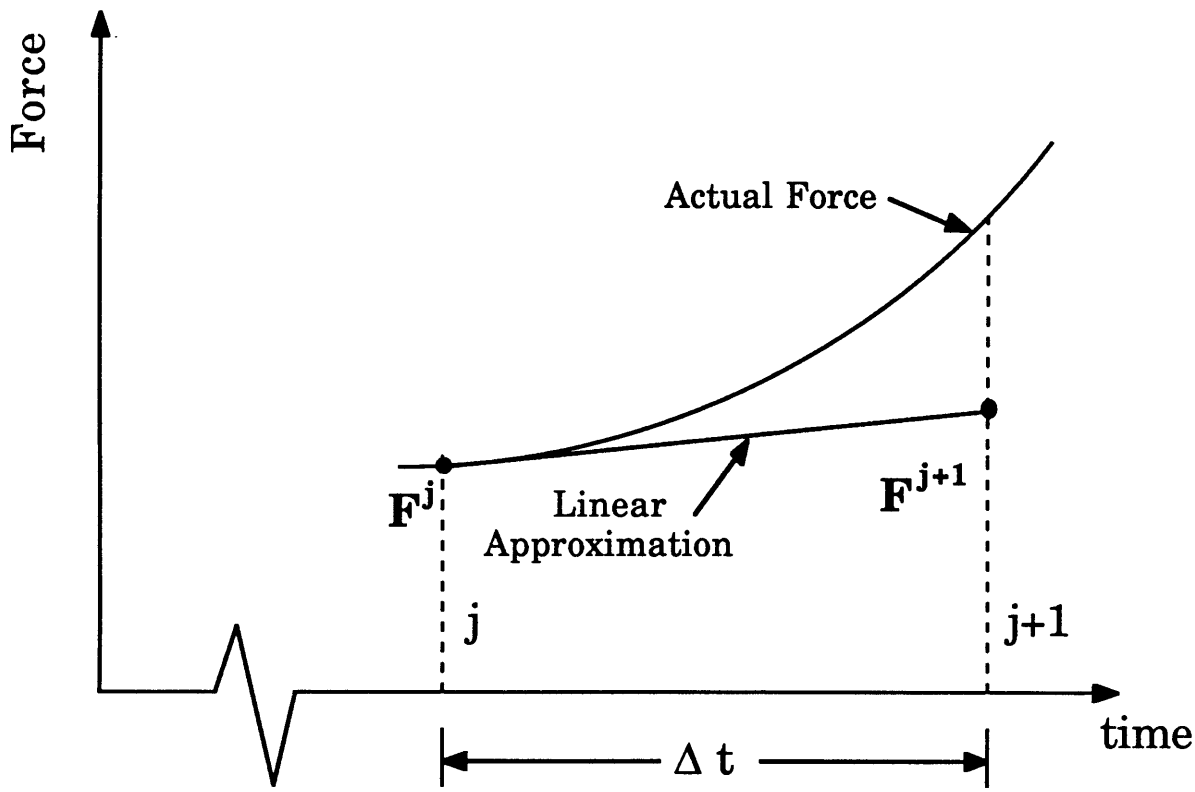


FIGURE 3.4 Illustration of linearization of the forcing function at the jth time step.

where j denotes the current time step. The forcing function in Eq. (3.6) is given by Eq. (3.3.21) in reference [1] as a time-varying amplitude times the product of the beam functions:

$$R_{c_k} = R\, m_k(\xi_1)\, n_k(\xi_2) = R\, W_k \qquad k = 1, 2, \ldots, mn \tag{3.22}$$

Substituting Eq. (3.21) into Eq. (3.22) results in:

$$\{R_c\}^{j+1} = \{R_c\}^j + \beta^j \Delta t \{W\} = \left(R^j + \beta^j \Delta t\right)\{W\} \tag{3.23}$$

where

$$\beta^j = nK\left(u^j - w^j\right)^{n-1}\left(\dot{u}^j - \dot{w}^j\right) \tag{3.24}$$

Substitution of Eq. (3.23) into Eq. (3.14) gives:

$$\{\ddot{Y}\}^{j+1} + [K]\{Y\}^{j+1} = [M]^{-1/2}\left(R^j + \beta^j \Delta t\right)\{W\} \tag{3.25}$$

$$= \left(R^j + \beta^j \Delta t\right)[M]^{-1/2}\{W\} \tag{3.26}$$

The vector $[M]^{-1/2}\{W\}$ in Eq. (3.26) is invariant with time. A typical solution to Eq. (3.26) consists of a constant and a time varying component:

$$\{Y_p\}^{j+1} = \sum_{i=1}^{mn}\left(c_i^j + d_i^j \Delta t\right)\{\phi\}^{(i)} \tag{3.27}$$

The constants c_i and d_i may be found at this point. Substitution of Eq. (3.27) into Eq. (3.26) gives:

$$[K]\{Y_p\}^{j+1} = \left(R^j + \beta^j \Delta t\right)[M]^{-1/2}\{W\} \tag{3.28}$$

Premultiplying Eq. (3.27) by [K] and using Eq. (3.17) gives:

$$[K]\{Y_p\}^{j+1} = \sum_{i=1}^{mn}\left(c_i^j + d_i^j \Delta t\right)\omega_i^2\{\phi\}^{(i)} \tag{3.29}$$

Equating Eqs. (3.28) and (3.29) gives

$$\sum_{i=1}^{mn} c_i^j \omega_i^2\{\phi\}^{(i)} = R^j[M]^{-1/2}\{W\} \tag{3.30}$$

and thus

$$c_i^j = \frac{R^j}{\omega_i^2}\{\phi\}^{(i)^T}[M]^{-1/2}\{W\} \qquad i = 1, 2, \ldots, mn \tag{3.31}$$

as well as

$$\sum_{i=1}^{mn} d_i^j \Delta t\, \omega_i^2\{\phi\}^{(i)} = \beta^j \Delta t\,[M]^{-1/2}\{W\} \tag{3.32}$$

and thus

$$d_i^j = \frac{\beta^j}{\omega_i^2}\{\phi\}^{(i)^T}[M]^{-1/2}\{W\} \qquad i = 1, 2, \ldots, mn \tag{3.33}$$

The total solution is the sum of the homogeneous solution and the particular solution:

$$\{Y\}^{j+1} = \sum_{i=1}^{mn}\left[a_i^j \cos(\omega_i \Delta t) + b_i^j \sin(\omega_i \Delta t) + c_i^j + d_i^j \Delta t\right]\{\phi\}^{(i)} \tag{3.34}$$

where eigenvalues ω_i and the eigenvectors $\{\phi\}^{(i)}$ are found from Eq. (3.17), the constants c_i and d_i are found from Eqs. (3.31) and (3.33), and a_i and b_i can be found at the current time step (j) from the information at the preceding time step.

The displacement of the plate may be constructed from the assumed shape given by Eq. (3.3.14 c) in reference [1] as:

$$w^j = \sum_{i=1}^{mn} C_i^j(t)\, W_i = \{C\}^{j^T}\{W\} \tag{3.35}$$

The modal amplitudes are found from Eq. (3.12) as

$$\{C\}^j = [M]^{-1/2}\{Y\}^j \tag{3.36}$$

Thus, the displacement of the plate may be written as:

$$w^j = \{Y\}^{j^T}[M]^{-1/2}\{W\} \tag{3.37}$$

The vector $[M]^{-1/2}\{W\}$ in this equation may be calculated beforehand and is the key reason for this method being faster than the Newmark method. This is because a vector, rather than a matrix, is multiplied with another vector at each time step. This means that much fewer multiplications must be performed.

The coefficients a_i and b_i in Eq. (3.34) need to be calculated. This equation may be written more generally as

$$\{Y\}\left(t - t^j\right) = \sum_{i=1}^{mn}\left\{ a_i^j \cos\left[\omega_i\left(t - t^j\right)\right] + b_i^j \sin\left[\omega_i\left(t - t^j\right)\right] + c_i^j + d_i^j\left(t - t^j\right)\right\}\{\phi\}^{(i)} \tag{3.38}$$

where t^j is the time at the jth time step and t is time. At time $t = t^{j+1}$, the equation for {Y} in Eq. (3.38) becomes:

$$\{Y\}^{j+1} = \sum_{i=1}^{mn}\left[a_i^j \cos(\omega_i \Delta t) + b_i^j \sin(\omega_i \Delta t) + c_i^j + d_i^j \Delta t\right]\{\phi\}^{(i)} \tag{3.39}$$

The equation for {Y} in the next time interval, t^{j+1} to t^{j+2}, is similar to Eq. (3.38):

$$\{Y\}\left(t - t^{j+1}\right) = \sum_{i=1}^{mn}\left\{ a_i^{j+1} \cos\left[\omega_i\left(t - t^{j+1}\right)\right] + b_i^{j+1} \sin\left[\omega_i\left(t - t^{j+1}\right)\right] + c_i^{j+1} + d_i^{j+1}\left(t - t^{j+1}\right)\right\}\{\phi\}^{(i)} \tag{3.40}$$

At time $t = t^{j+1}$ this equation reduces to:

$$\{Y\}^{j+1} = \sum_{i=1}^{mn}\left[a_i^{j+1} + c_i^{j+1}\right]\{\phi\}^{(i)} \tag{3.41}$$

Eqs. (3.39) and (3.41) must be equal since they both represent {Y} at the same point in time. Equating these two equations yields:

$$a_i^{j+1} + c_i^{j+1} = a_i^j \cos(\omega_i \Delta t) + b_i^j \sin(\omega_i \Delta t) + c_i^j + d_i^j \Delta t \tag{3.42}$$

Following the same procedure for $\{\dot{Y}\}$ gives

$$b_i^{j+1} + \frac{d_i^{j+1}}{\omega_i} = -a_i^j \sin(\omega_i \Delta t) + b_i^j \cos(\omega_i \Delta t) + \frac{d_i^j}{\omega_i} \tag{3.43}$$

The equation of motion of the impactor is linearized in a manner similar to that of the plate. The equation of motion of the impactor is given by:

$$m\ddot{u} = -R \tag{3.44}$$

Substituting Eq. (3.21) into this equation gives:

$$m\ddot{u}^{j+1} = -R^j - nK\left(u^j - w^j\right)^{n-1}\left(\dot{u}^j - \dot{w}^j\right)\Delta t \tag{3.45}$$

By integrating this equation twice, the velocity and displacement of the impactor are obtained:

$$\dot{u}\left(t-t^{j}\right) = -\frac{R^{j}}{m}\left(t-t^{j}\right) - \frac{n}{m}K\left(u^{j}-w^{j}\right)^{n-1}\left(\dot{u}^{j}-\dot{w}^{j}\right)\frac{\left(t-t^{j}\right)^{2}}{2} + A^{j} \tag{3.46}$$

$$u\left(t-t^{j}\right) = -\frac{R^{j}}{m}\frac{\left(t-t^{j}\right)^{2}}{2} - \frac{n}{m}K\left(u^{j}-w^{j}\right)^{n-1}\left(\dot{u}^{j}-\dot{w}^{j}\right)\frac{\left(t-t^{j}\right)^{3}}{6} + A^{j}\left(t-t^{j}\right) + B^{j} \tag{3.47}$$

The unknown constants of integration, A and B, may be found in a manner similar to the way in which a_i and b_i were found for Eq. (3.34). They are found to be:

$$A^{j} = \dot{u}^{j} \tag{3.48}$$

and

$$B^{j} = u^{j} \tag{3.49}$$

The initial conditions must be considered before the time-marching commences. At time zero, the plate is at rest and has zero displacement. Thus,

$$w(0) = 0 = \{Y\}^{o^{T}}[M]^{-1/2}\{W\} \tag{3.50}$$

such that

$$\{Y\}^{o^T} = \{0\} \tag{3.51}$$

Similarly, since the plate is at rest:

$$\{\dot{Y}\}^{o^T} = \{0\} \tag{3.52}$$

The initial value of a_i is found from Eq. (3.41):

$$\{Y\}^{o} = \sum_{i=1}^{mn} \{a_i^o + c_i^o\} \{\phi\}^{(i)} = \{0\} \tag{3.53}$$

This is satisfied by the following being true:

$$a_i^o = -c_i^o \tag{3.54}$$

Similarly, setting the initial value of $\{\dot{Y}\}$ to zero leads to

$$b_i^o = -\frac{d_i^o}{\omega_i} \tag{3.55}$$

Before calculating the initial values of c_i and d_i, a shortcoming of the Taylor series expansion must be pointed out. The slope of the force versus indentation curve is initially zero. Thus, the force at the next time step will be zero. To overcome this, the impactor is assumed to be initially in the plate a small amount so that a small force exists. Assume:

$$u^o = \varepsilon \Delta t \dot{u}^o \tag{3.56}$$

where $\dot{u}^o$ is the initial velocity of the impactor and ε is a small value. A value of 0.01 was used for ε in all analyses run in this study. This value was used in a comparison case with the previous method, which used the Newmark method for time-marching. Excellent agreement was found to exist between the current method and the previous method.

The initial values of c_i and d_i may now be calculated. At time zero, Eq. (3.31) becomes:

$$c_i^o = \frac{R^o}{\omega_i^2} \{\phi\}^{(i)^T} [M]^{-1/2} \{W\} \tag{3.57}$$

From Eqs. (2.1) and (3.20), R^o may be expressed as:

$$R^o = K(u^o - w^o)^n \tag{3.58}$$

Letting

$$CW_i = \{\phi\}^{(i)^T} [M]^{-1/2} \{W\} \tag{3.59}$$

and using Eq. (3.58), Eq. (3.57) becomes:

$$c_i^o = K(u^o - w^o)^n \frac{CW_i}{\omega_i^2} \tag{3.60}$$

Substituting Eq. (3.56) into Eq. (3.60), and using the fact that the plate initially has zero displacement, results in:

$$c_i^o = K\left(\varepsilon \Delta t \dot{u}^o\right)^n \frac{CW_i}{\omega_i^2} \tag{3.61}$$

From Eqs. (3.33), (3.24), and (3.59):

$$d_i^o = \frac{\beta^o}{\omega_i^2}\{\phi\}^{(i)^T}[M]^{-1/2}\{W\} \tag{3.62}$$

$$d_i^o = nK(u^o - w^o)^{n-1}\left(\dot{u}^o - \dot{w}^o\right)\frac{CW_i}{\omega_i^2} \tag{3.63}$$

Substituting Eq. (3.56) into Eq. (3.63), and using the fact that the plate initially has zero displacement and zero velocity, results in:

$$d_i^o = nK\left(\varepsilon \Delta t \dot{u}^o\right)^{n-1}\dot{u}^o\frac{CW_i}{\omega_i^2} \tag{3.64}$$

All of the information needed to start time-marching is now at hand. From Eq. (3.37), the plate displacement at the (j + 1) time step is:

$$w^{j+1} = \{Y\}^{j+1^T}[M]^{-1/2}\{W\} \tag{3.65}$$

where $\{Y\}^{j+1}$ is given by Eq. (3.34). The coefficients a_i, b_i, c_i, and d_i in Eq. (3.34) are summarized below:

$$c_i^{j+1} = K\left(u^{j+1} - w^{j+1}\right)^n \frac{CW_i}{\omega_i^2} \tag{3.66}$$

$$d_i^{j+1} = nK\left(u^{j+1} - w^{j+1}\right)^{n-1}\left(\dot{u}^{j+1} - \dot{w}^{j+1}\right)\frac{CW_i}{\omega_i^2} \tag{3.67}$$

$$a_i^{j+1} = a_i^j \cos(\omega_i \Delta t) + b_i^j \sin(\omega_i \Delta t) + c_i^j + d_i^j \Delta t - c_i^{j+1} \tag{3.68}$$

$$b_i^{j+1} = b_i^j \cos(\omega_i \Delta t) - a_i^j \sin(\omega_i \Delta t) + \frac{d_i^j - d_i^{j+1}}{\omega_i} \tag{3.69}$$

During the course of the time-marching, it is possible for the displacement of the impactor to be less than the displacement of the plate. This indicates that the impactor has lost contact with the plate. In this case, c_i and d_i are set equal to zero to allow the equations of the impactor and plate to decouple.

As was mentioned at the beginning of this section, this method was developed because it is faster than the Newmark implicit integration scheme. The Newmark method requires the multiplications of matrices, which are (m times n) by (m times n) in size, at each time step. (Recall that m and n are the number of modes in the x and y directions respectively.) The present method requires only multiplications of vectors which are (m times n) in length at each time step. As the number of modes and time steps increase, the savings in computational time becomes quite significant. There is the extra expense of

having to solve an eigenvalue problem, but this is small in comparison with the time saved during the time-marching routine. The eigenvalue problem takes about the same amount of time to solve as it takes to invert the stiffness matrix. An example case was run with 1000 time steps, and both m and n equal to 9. A Digital VAXstation II was used to run the program. The previous method took 5 minutes and 31 seconds, of which 3 minutes and 6 seconds was during the Newmark integration time-marching routine. The current method took 3 minutes and 55 seconds, of which 59 seconds was during the new time-marching method. As the number of modes and the number of time steps increase, the savings in time also increases.

3.4 Computer Implementation

The local and global models have been implemented as FORTRAN programs. A Digital VAXstation II was used to develop and run the codes. The organization of the specific programs and associated data files is shown in Figure 3.5. All of the programs may be run separately, or as a part of a system called ACPI (Analysis of Composite Plate Impact). The operation of this system is explained in Appendix B.

Program INPUT is the program that asks the user for the necessary inputs needed for all of the analyses. The information is obtained in an interactive session with the user and placed in a data file. This data file is used by program ROTATE to create a binary file for easy access by all of the programs.

Material properties of some materials are stored in the binary file LIBRARY.BIN. New materials may be added to this library with program

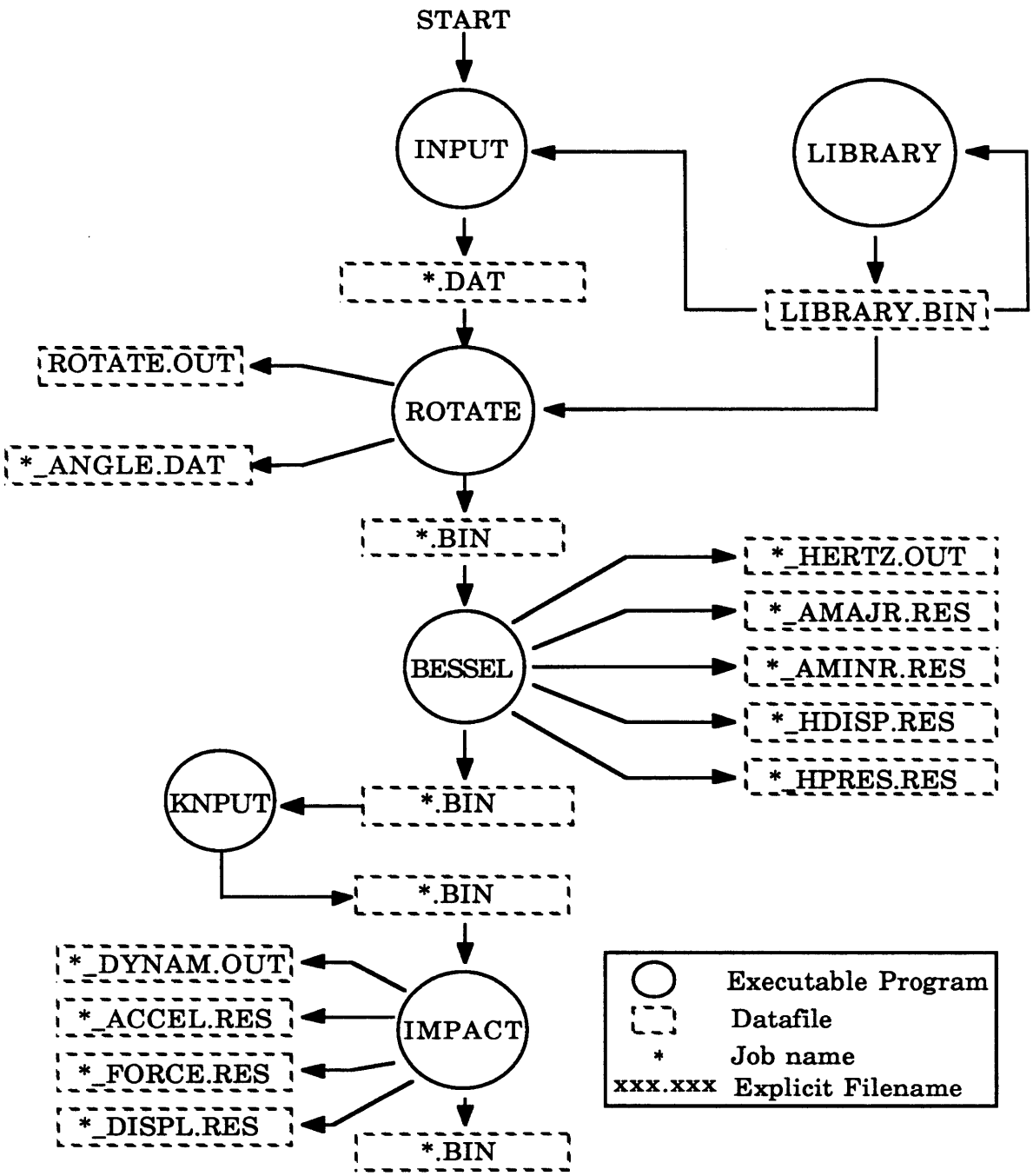


FIGURE 3.5 Impact analysis flow chart.

LIBRARY. This program also allows one to change material values, delete materials from the list, and view the current values of a material.

The average engineering properties and the D matrix of the plate are calculated by program ROTATE. As mentioned above, this program takes all of the input information for the ACPI system and places it in a binary input file. It also creates a print out of the information for the user. The run time of this program is very short. For example, a 12-ply laminate takes 8.0 seconds to run. The run times vary slightly for laminates with a different number of plies.

The local model analysis is coded in a program called BESSEL. This program calculates the load required to produce a given indentation in a composite plate. The run time of this program depends heavily on the number of modes of the loading function used, and the number of indentations for which the corresponding load is to be calculated. The calculation of the load for a single indentation, using 60 modes, takes 3 minutes and 35 seconds.

The global model is coded in a program called IMPACT. This program calculates the acceleration, force, and displacement histories of an impacted composite plate. The run time of this program depends heavily on the number of beam functions and time steps used. An 8-by-8 mode analysis with 17,100 time steps takes 15.8 minutes of computation time. A 17-by-17 mode analysis with 17,100 time steps takes 162 minutes of computation time.

3.5 Numerical Example

A $[\pm 45/0]_{2s}$ laminate of AS4/3501-6 is analyzed. The inputs used for this example are consistent with those used in the experiments as described in the

next chapter. To obtain the three-dimensional properties of the laminate, the ply properties are rotated into the laminate principal axes, summed and averaged through the thickness. The in-plane averaging is done as springs in parallel, whereas the out-of-plane averaging is done as springs in series. Program ROTATE does this averaging. The ply properties for AS4/3501-6 are listed in Table 3.1 [1]. The average engineering properties obtained from ROTATE are presented in Table 3.2.

Program BESSEL is run to obtain the impactor/plate contact relation. Since the local model assumes the plate to be transversely isotropic, material values representative of such a material, which are also representative of the monoclinic plate, must be entered. What is therefore done is to perform the analysis twice: once with the laminate properties in the x-direction, and then with the laminate properties in the y-direction. The analysis done with the x-direction properties is referred as the major axis analysis, and that with the y-direction properties as the minor axis analysis. The material property inputs for both analyses are listed in Table 3.3.

The radius of the cylindrical region is chosen such that the classical plate solution is recovered outside of this region. As was adopted by Cairns [1], a radius equal to twenty times the plate thickness is used. Thus, the radius of the cylindrical region is 32.0 mm for this example. The indentor radius is 6.35 mm.

The pressure loading is assumed to be Hertzian. Program BESSEL offers loading options other than Hertzian. This analysis is restricted to the Hertzian case. Sixty harmonics of the infinite Fourier-Bessel series are used in the analysis. The results for the major axis and minor axis are shown in Figure 3.6. Note that the two curves are very close to one another indicating that the transverse isotropy assumption is reasonable.

TABLE 3.1 AS4/3501-6 Graphite/Epoxy Ply Properties

$E_{11} = 142.0$ GPa
$E_{22} = E_{33} = 9.81$ GPa
$G_{12} = G_{13} = 6.0$ GPa
$G_{23} = 3.77$ GPa
$\nu_{12} = \nu_{13} = 0.30$
$\nu_{23} = 0.34$
$t_{ply} = 0.134$ mm
$\rho = 1540$ kg/m^3

TABLE 3.2 $[\pm 45/0]_{2s}$ AS4/3501-6 Equivalent Engineering Properties

E_{xx} = 61.8 GPa
E_{yy} = 26.7 GPa
E_{zz} = 10.7 GPa
G_{xz} = 5.01 GPa
G_{yz} = 4.30 GPa
ν_{xy} = 0.69
ν_{xz} = 0.13
ν_{yz} = 0.25
h = 1.608 mm

TABLE 3.3 Major Axis and Minor Axis Material Property Inputs for $[\pm 45/0]_{2s}$ laminates

	Major Axis	Minor Axis
E_θ	61.8 GPa	26.7 GPa
E_z	10.7 GPa	10.7 GPa
G_{rz}	5.01 GPa	4.30 GPa
$\nu_{r\theta}$	0.69	0.30
ν_{rz}	0.13	0.25

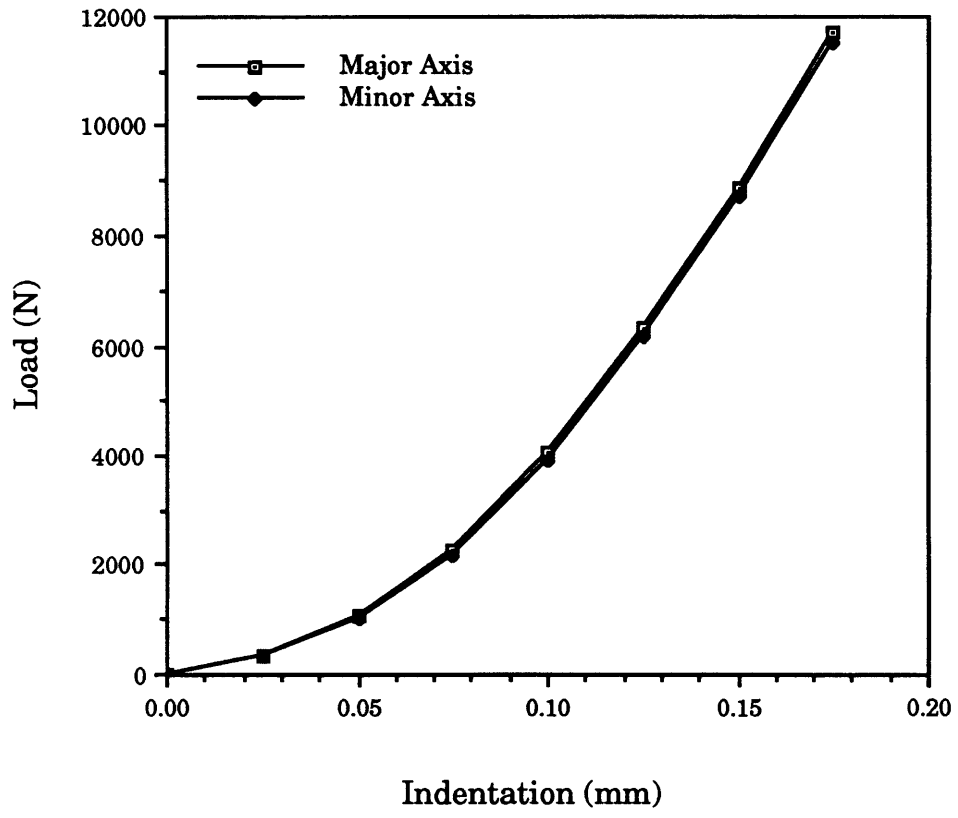


FIGURE 3.6 Analytical impactor/plate contact relation for a $[\pm 45/0]_{2s}$ layup of AS4/3501-6.

One must be sure that a sufficient number of terms are used in the Fourier-Bessel series to obtain a converged solution. A good method of checking the convergence is to observe the recovered pressure distribution from the Fourier-Bessel series in comparison with the assumed pressure distribution. An example pressure distribution obtained from the series using 60 harmonics is shown in Figure 3.7 along with the assumed pressure distribution. The recovered load is the area under the curve of the modeled pressure distribution. By calculating the recovered load each time a harmonic is included in the series, one can clearly see when the modeled pressure distribution starts to become adequate. A plot of the recovered load versus the number of harmonics used in the series is shown in Figure 3.8 for the major axis with an approach of 0.05 mm. The load oscillates with decaying amplitude about the input value of unity. The value of the recovered load has stabilized at 60 harmonics. Even though this value has stabilized, the modeled pressure distribution may still not be good in comparison with the assumed pressure distribution. Thus, one should always check the distributions as shown in Figure 3.7.

The constants n and K in Eq. (2.1) must be found for use by program IMPACT. Taking the logarithm of this equation gives

$$\log(R) = \log(K) + n\log(\alpha) \tag{3.70}$$

which is a straight line with slope n. By plotting the logarithm of the load verses the logarithm of the indentation, a line may be fit to the data by the method of least squares. The slope of this line gives the exponent n and the antilogarithm of the constant gives K. The results for the present example are summarized in Table 3.4. The curve fits were good, as indicated by the large R^2 values.

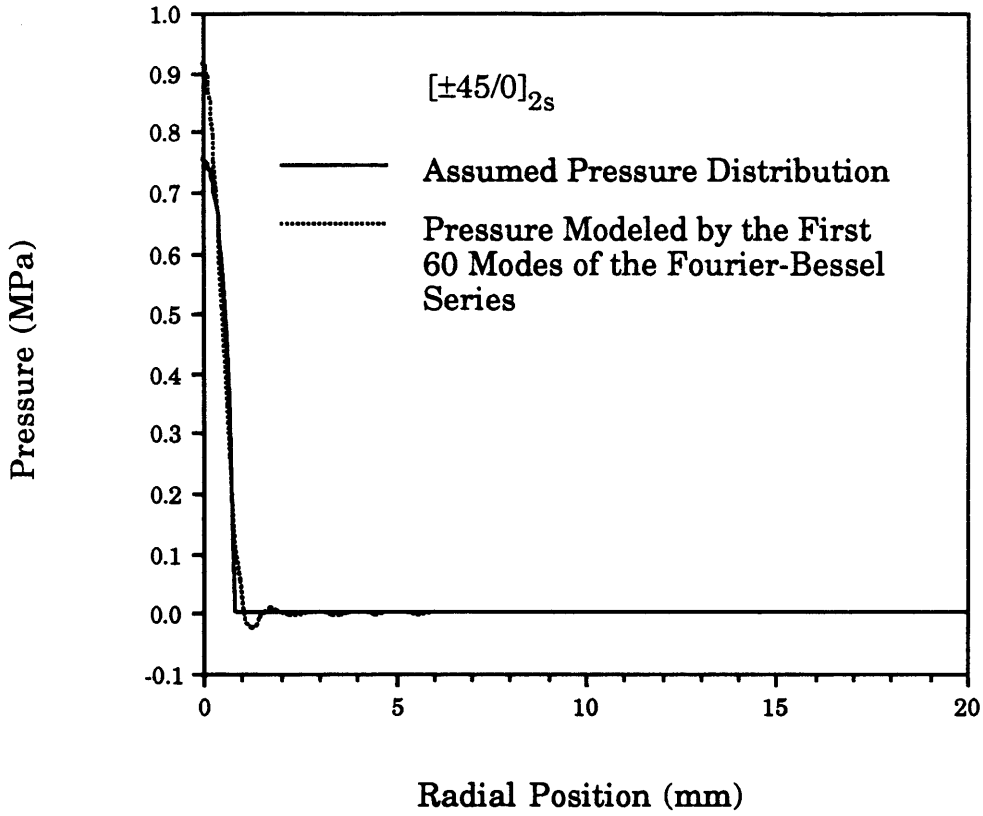


FIGURE 3.7 Comparison of the assumed pressure distribution with the recovered pressure distribution. (Half of the axisymmetric distributions are shown.)

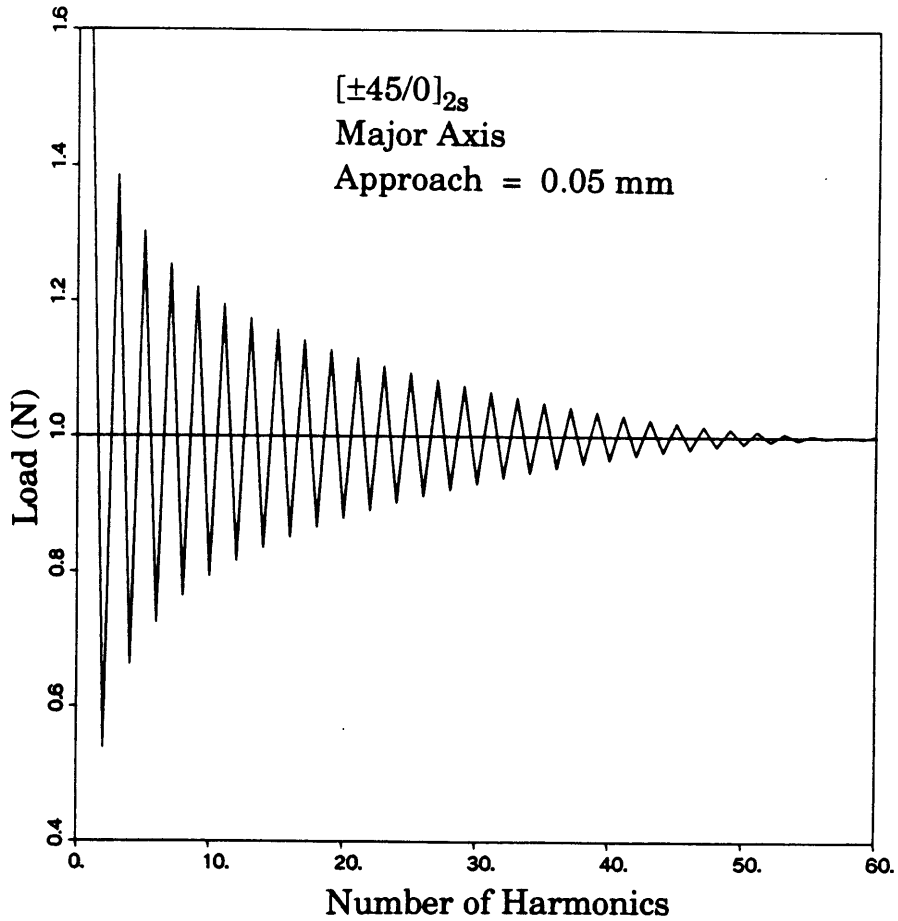


FIGURE 3.8 Convergence of the recovered load for the major axis analysis with an approach of 0.05 mm.

TABLE 3.4 Reduced Data From the Local Model Analysis

	Major Axis	Minor Axis	Average
n	1.83	1.86	1.85
K (GN/m^n)	84.6	112.6	98.6
R^2	0.998	0.998	-

Program IMPACT is run to obtain the dynamic global behavior of the impacted plate. The plate is clamped along the edges x equal to zero and x equal to a (refer to Figure 3.3), and free along the remaining two edges. The test lengths are 251 mm in the x-direction and 89 mm in the y-direction. The plate thickness is 1.608 mm and the plate density is 1540 kg/m^3. The components of the D matrix and the interlaminar shear stiffness matrix are given in Table 3.5. The impactor has a mass of 1.528 kg and a radius of 6.35 mm. It impacts the plate with a velocity of 1.46 m/s. The analytically obtained impactor/plate contact relation is used as input for this analysis. The values of n and K for Eq. (2.1) obtained from the local analysis are 1.85 and 98.6 GN/m$^{1.85}$, respectively.

A few inputs which deal with the global analysis itself must be considered. There is an option to condense the rotary inertia. This option is recommended for laminates which are reasonably thin and is chosen here. There is also an option to eliminate the even modes from the analysis. This option is useful if the D_{16} and D_{26} terms are much smaller than the D_{11} and D_{22} terms. It is not used in this example since this criteria is not satisfied for this laminate. The remaining inputs are the number of modes to use, the length of the time step, and the total number of time steps. The total number of time steps is taken care of by choosing to let the program run until the displacement of the plate returns to zero, its initial position.

The number of modes to use is determined by running the analysis several times with a different number of modes each time, and observing the trend of the force versus time plot (see Figures 3.9 through 3.11). The 9-by-9 mode analysis gives the basic shape of the force history. The force history has a half sinusoidal shape to it with large amplitude, high frequency vibrations superimposed on it. The peak force is 477 N and the length of the impact event

TABLE 3.5 $[\pm 45/0]_{2s}$ AS4/3501-6 Plate Bending and Shear Properties

D_{11} = 21.75 N·m	A_{44} = 6.92 MN/m
D_{12} = 9.79 N·m	A_{45} = 0.00 MN/m
D_{22} = 13.64 N·m	A_{55} = 8.06 MN/m
D_{66} = 10.85 N·m	
D_{16} = 2.24 N·m	
D_{26} = 2.24 N·m	

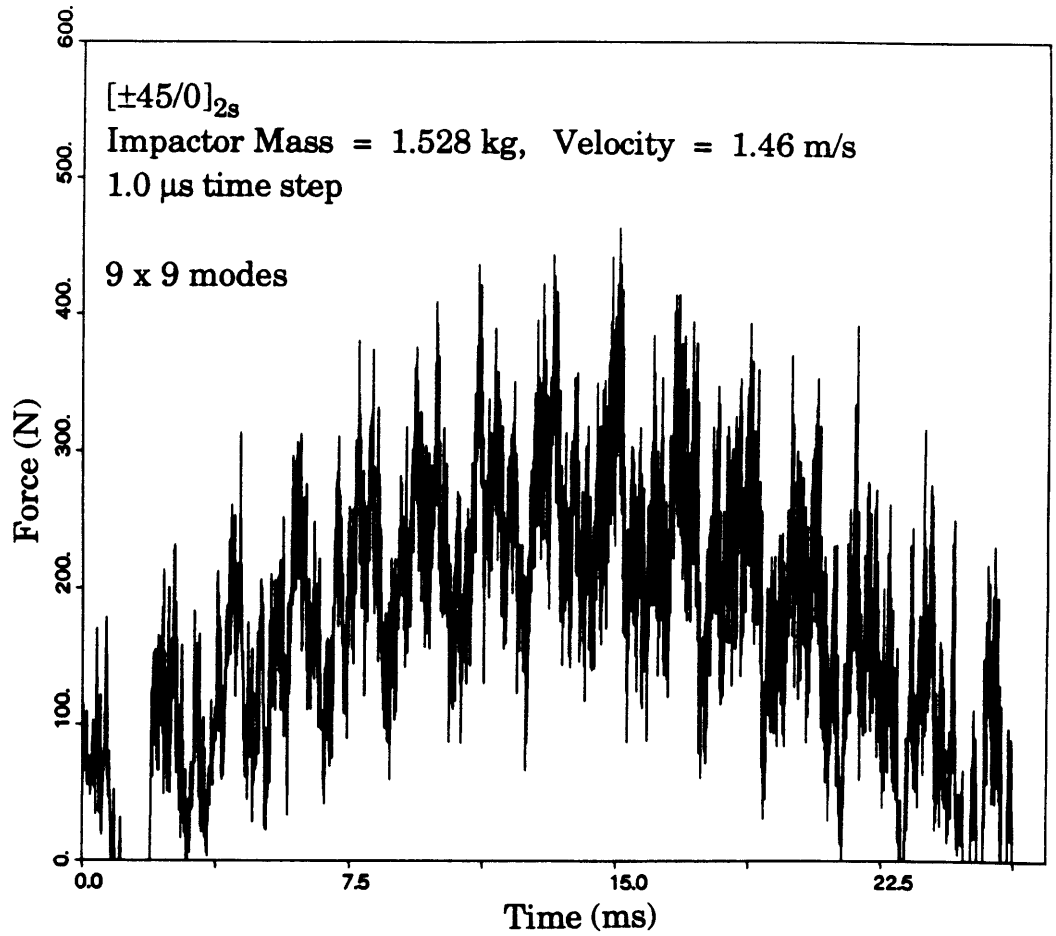


FIGURE 3.9 Analytical force history of a $[\pm 45/0]_{2s}$ laminate using 9 x 9 modes.

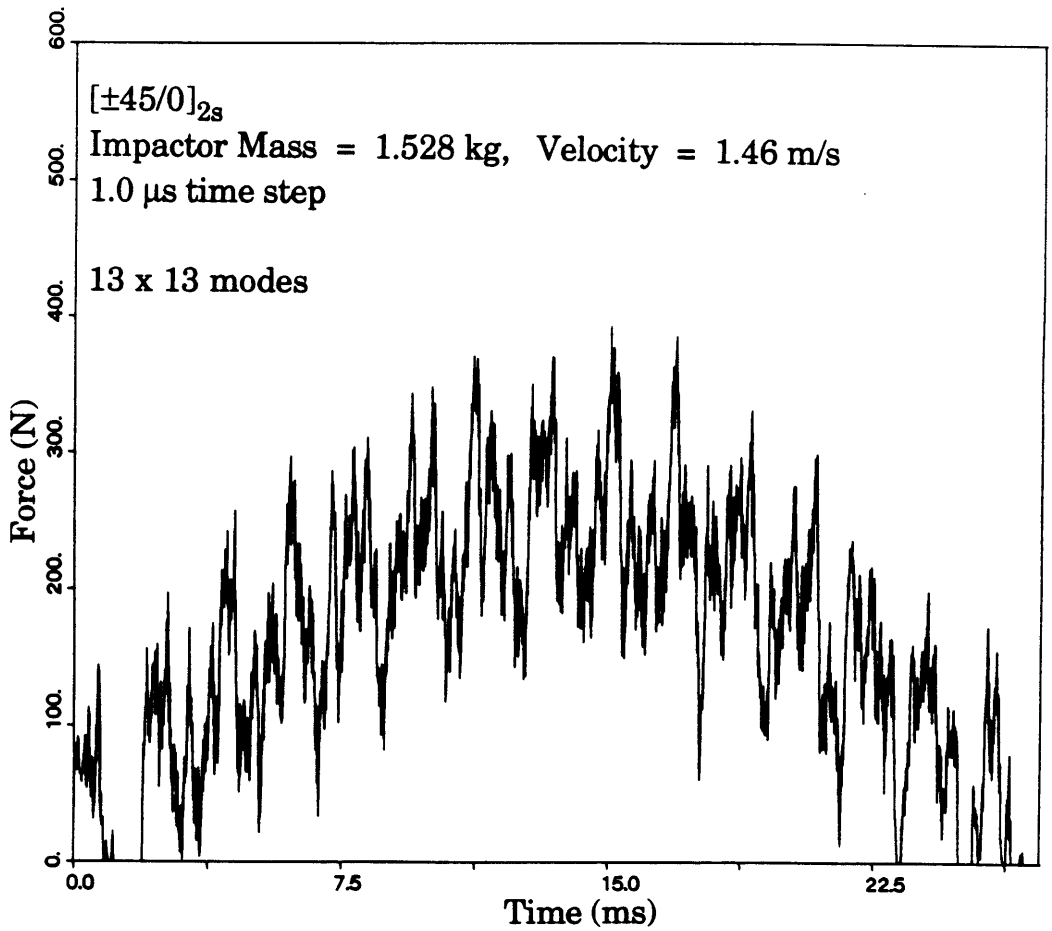


FIGURE 3.10 Analytical force history of a $[\pm 45/0]_{2s}$ laminate using 13 x 13 modes.

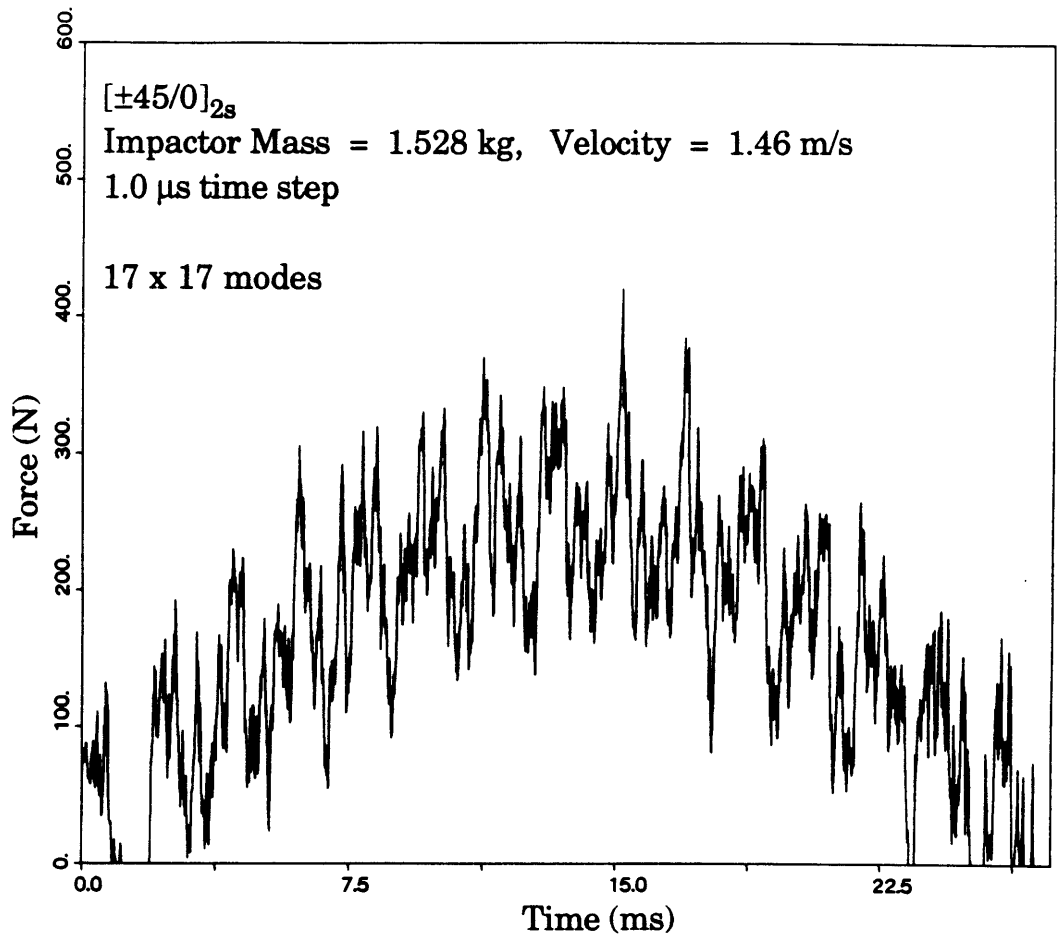


FIGURE 3.11 Analytical force history of a $[\pm 45/0]_{2s}$ laminate using 17 x 17 modes.

is 27.3 ms. There are some points in time at which the force becomes zero. This indicates that the impactor and the plate have separated. The 13-by-13 mode analysis has the same basic shape as the 9-by-9 mode analysis. Most of the large amplitude, high frequency vibrations have been eliminated. The peak force is 398 N and the length of the impact event is 27.5 ms. The 17-by-17 mode analysis is almost identical to the 13-by-13 mode analysis. This indicates that a converged solution is being approached. The peak load in the 17-by-17 mode analysis is 422 N and the length of the impact event is again 27.5 ms. The size of the time step is found in a similar manner (see Figures 3.12 through 3.14). Note that for large time steps, the solution may become unstable because of the nature of the time-marching routine. This is clearly illustrated in Figure 3.12. The force history from the analysis with a 0.5 μs time step is essentially the same as that with a 1.0 μs time step. Thus, a 1.0 μs time step is sufficient.

The analysis also gives the displacement history of the plate center (see Figure 3.15). This type of data will be compared with experimentally obtained displacement data in Chapter 6. Note that the displacement history curves converge faster than the force history curves with respect to the number of modes (see Figures 3.15 through 3.17). This is because the displacement is described by a fourth order equation whereas the force is described by a sixth order equation.

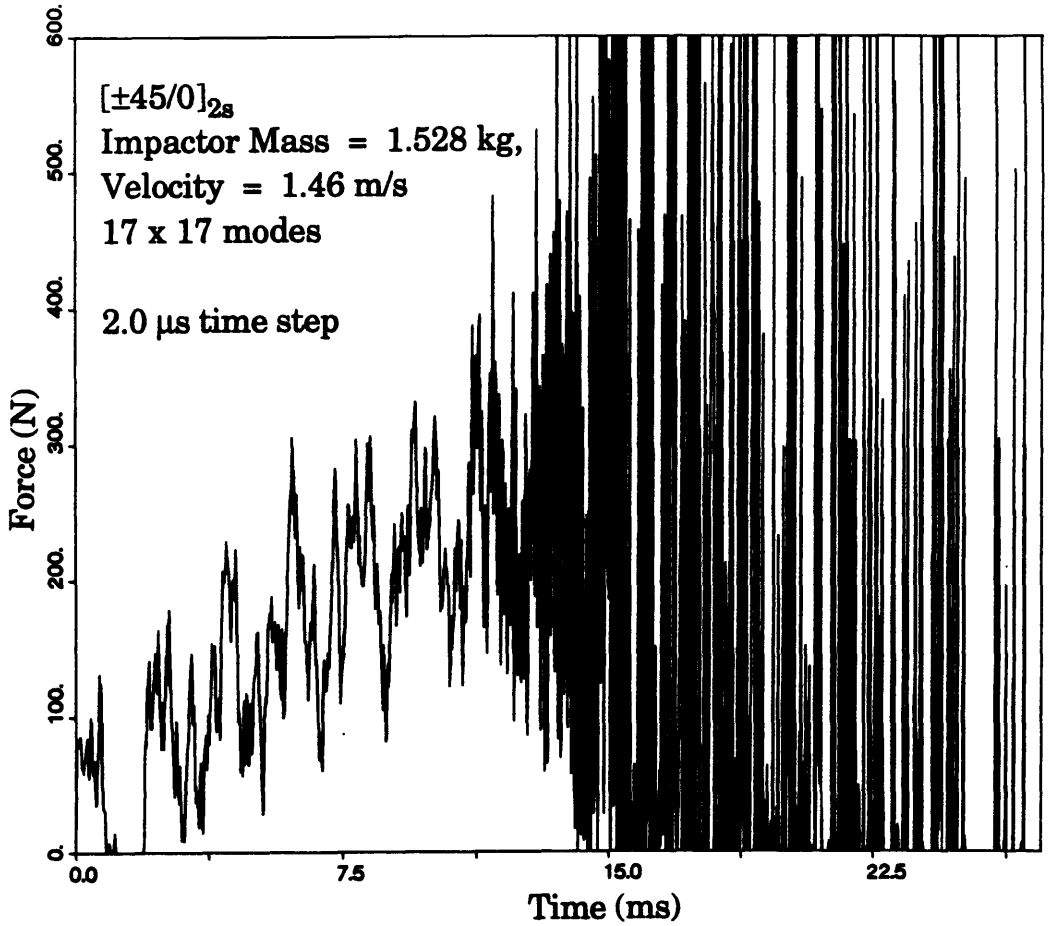


FIGURE 3.12 Analytical force history of a $[\pm 45/0]_{2s}$ laminate using a 2.0 μs time step.

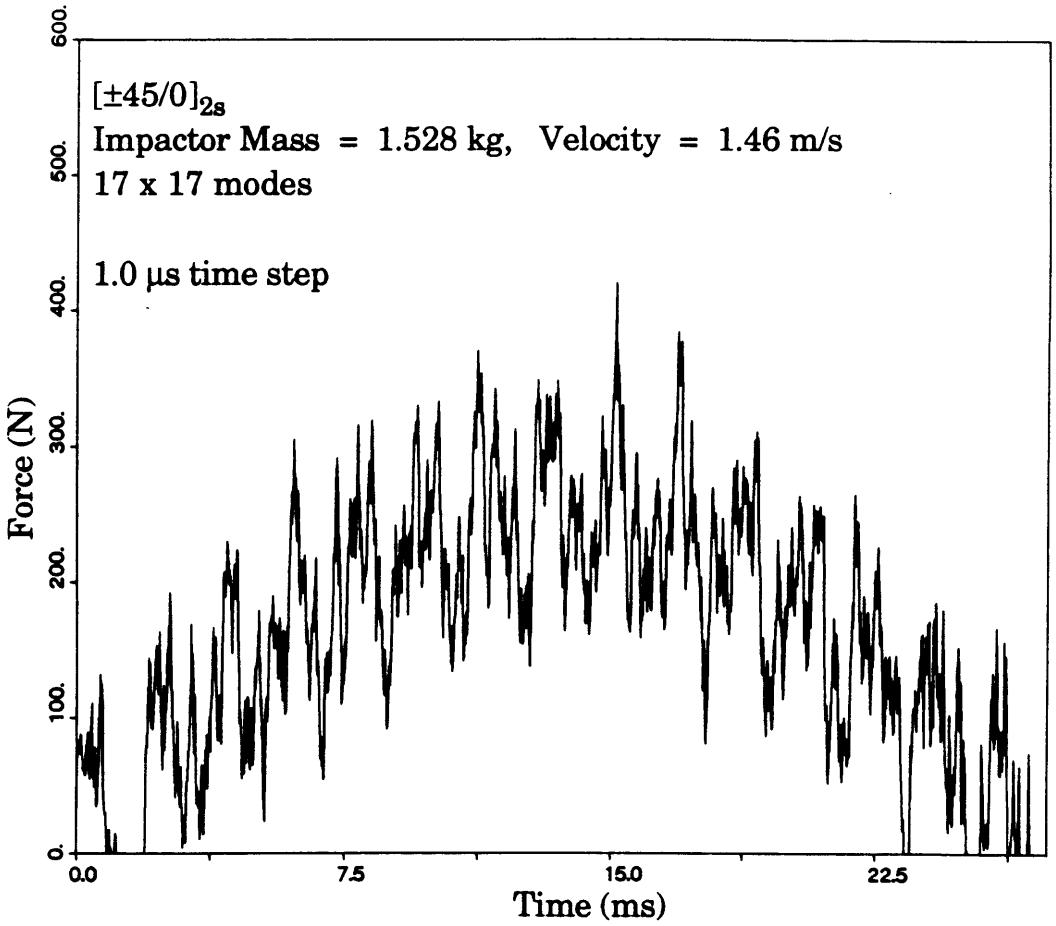


FIGURE 3.13 Analytical force history of a $[\pm 45/0]_{2s}$ laminate using a 1.0 μs time step.

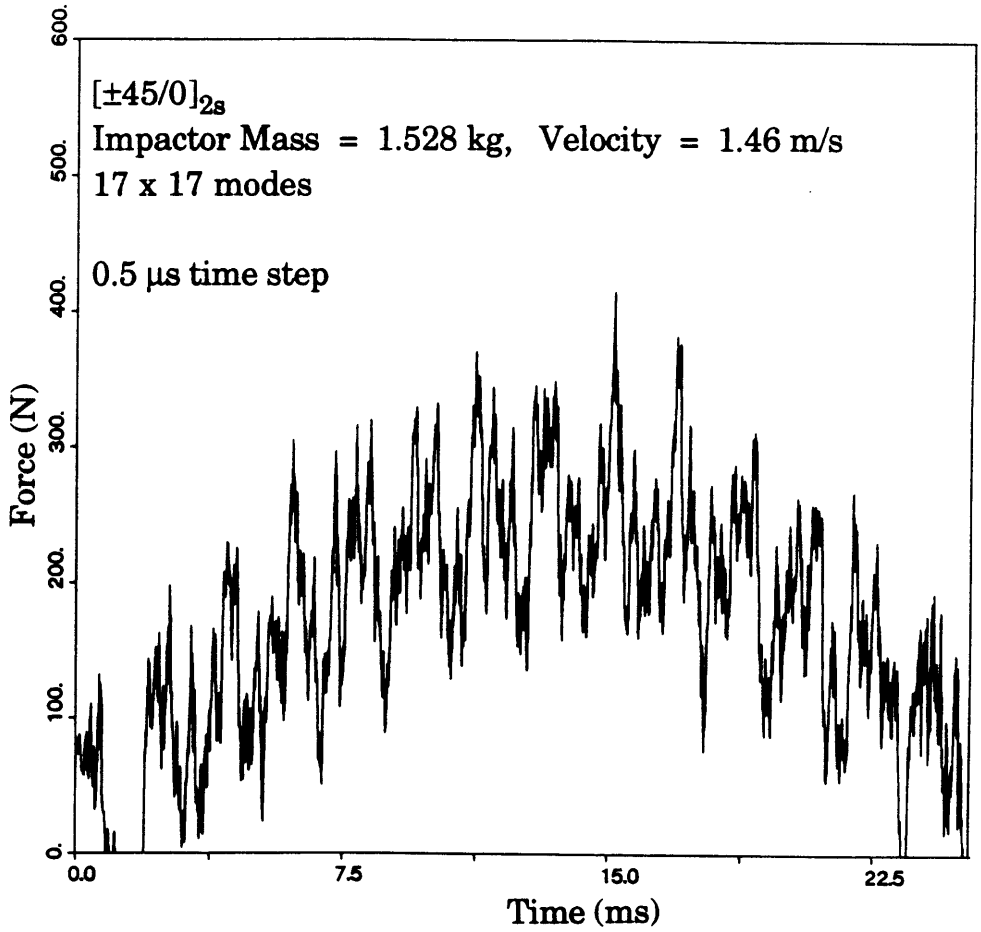


FIGURE 3.14 Analytical force history of a $[\pm 45/0]_{2s}$ laminate using a 0.5 μs time step.

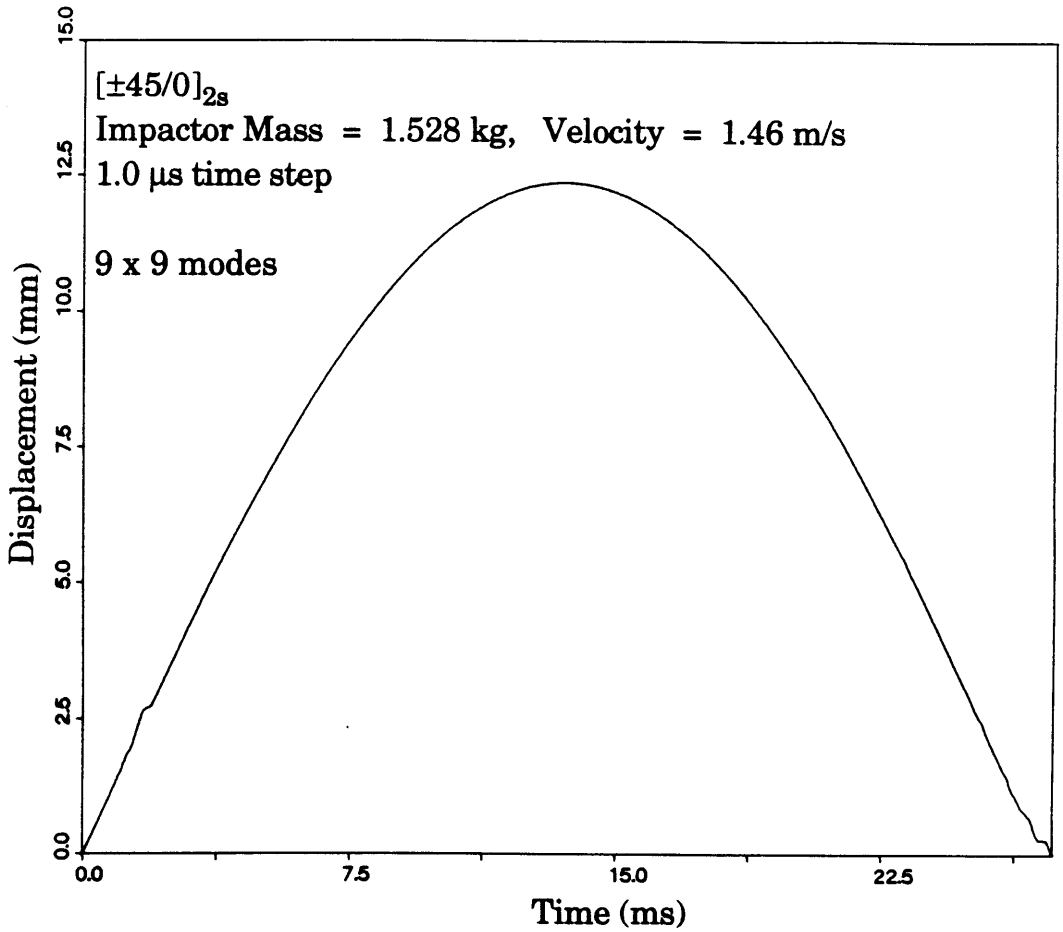


FIGURE 3.15 Analytical displacement history of a $[\pm 45/0]_{2s}$ laminate using 9 x 9 modes.

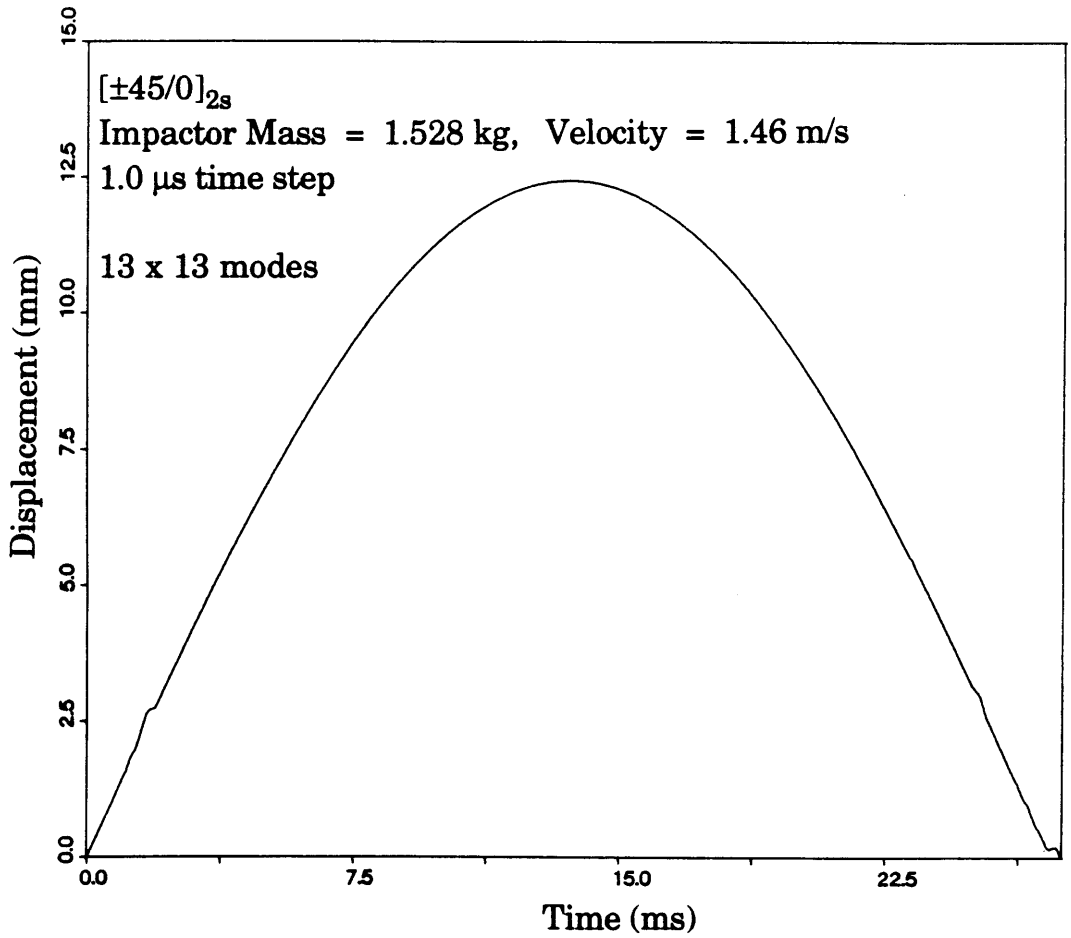


FIGURE 3.16 Analytical displacement history of a $[\pm 45/0]_{2s}$ laminate using 13 x 13 modes.

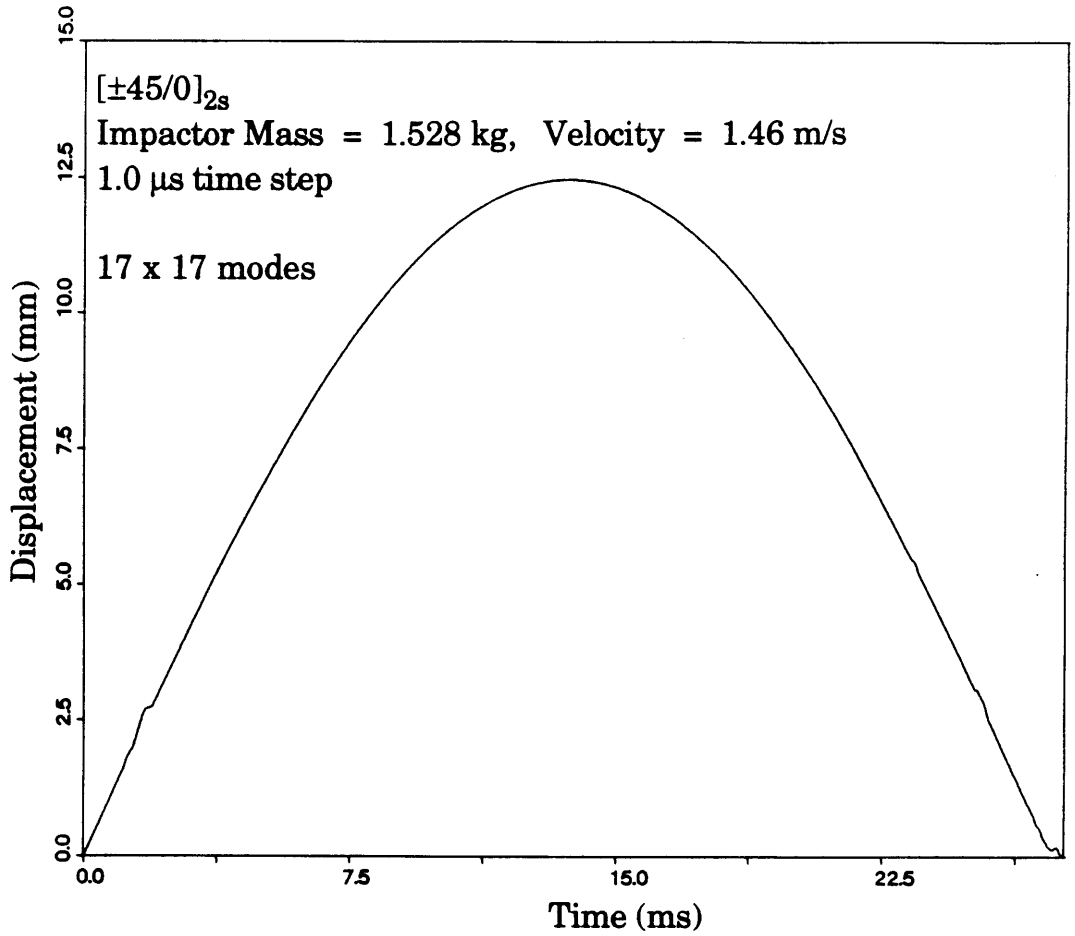


FIGURE 3.17 Analytical displacement history of a $[\pm 45/0]_{2s}$ laminate using 17 x 17 modes.

CHAPTER 4

Experimental Procedures

4.1 Test Matrix and Specimen Description

As mentioned in Chapter 1, the objectives of the experimental tests are to provide data for comparison with the impact analysis and to investigate the effect of damage. Two types of tests were thus developed to obtain the necessary data. The first test is an impact test which provides experimental force and displacement histories of the center of the impacted plate. The second test is a quasi-static test which provides experimental impactor/plate contact data.

Two types of specimens were chosen for the tests: a $[0/90]_{5s}$ layup and a $[\pm 45/0]_{2s}$ layup. The $[0/90]_{5s}$ layup was chosen because it is an orthotropic layup, free from bending-twisting coupling. The $[\pm 45/0]_{2s}$ layup was chosen because it has significant bending-twisting coupling. The material for all the laminates was AS4/3501-6 graphite/epoxy manufactured by Hercules.

The test matrix for the quasi-static tests is shown in Table 4.1. Two types of conditions are listed in this table: undamaged and damaged. The undamaged specimens were either virgin specimens, or specimens which were impacted at a low energy level such that the they were not damaged. A low energy level is defined here as an impact with a 1.53 kg impactor at a velocity of 1.5 m/s. The fact that the specimens were not damaged was verified by impacting pretest specimens at this energy level and X-raying them to detect the presence of damage. No damage was detected in these specimens. It is important that the specimens tested for comparison with the analysis remain undam-

TABLE 4.1 Quasi-Static Test Matrix

Condition	Laminate	
	$[0/90]_{5s}$	$[\pm 45/0]_{2s}$
Undamaged	3[a]	3
Damaged[b]	0	3

[a]Number indicates number of specimens tested.

[b]Damaged induced by impacting with a 1.53 kg impactor at 4.0 m/s.

aged. This is because one of the assumptions in the analysis is that no damage exists during the event. An impact with a 1.53 kg impactor at a velocity of 4.0 m/s produces internal damage approximately the size of the impactor diameter and some back face matrix splits in the $[\pm 45/0]_{2s}$ laminates. Such an impact is defined here as a high energy impact. The presence of damage was verified by X-raying one of the specimens. A total of six $[\pm 45/0]_{2s}$ specimens and three $[0/90]_{5s}$ specimens were tested quasi-statically.

The test matrix for the impact tests is shown in Table 4.2. The impactor mass for all of the tests is the same, 1.53 kg. Only the impactor velocity is varied. None of the $[0/90]_{5s}$ specimens were impacted at the high energy level (impactor velocity equal to 4.0 m/s). This is because the effect of damage was only investigated in the $[\pm 45/0]_{2s}$ specimens. Three $[\pm 45/0]_{2s}$ specimens were first impacted at high energy. These specimens were then impacted at low energy (impactor velocity equal to 1.5 m/s) with all the measurements taken. Four $[0/90]_{5s}$ specimens and four $[\pm 45/0]_{2s}$ specimens, which were not predamaged, were also impacted at low energy.

The dimensions of the test specimen were chosen such that three specimens could be obtained from one TELAC standard sized laminate of 305 mm by 350 mm. A typical specimen is shown in Figure 4.1. The specimen width is 88.9 mm and the length is 344 mm. Holes were drilled in the ends of the specimens that were to be impacted. This helped in obtaining clamped boundary conditions at these two ends. The other two ends were free in all the tests. Thus, the plate test specimens are actually wide beams. The specimens that were to be impacted also had a machined cold rolled steel nut, as shown in Figure 4.2, bonded to them. This nut was used to screw an extension rod with a magnetic core to the back of the specimen. This allowed the displacement of the specimen to be measured during impact.

TABLE 4.2 Impact Test Matrix

Impactor			Laminate	
Mass	Velocity	Predamaged[a]	$[0/90]_{5s}$	$[\pm 45/0]_{2s}$
1.53 kg	1.5 m/s	No	4[b]	4
1.53 kg	1.5 m/s	Yes	0	3

[a]Predamage caused via impact at 4.0 m/s.

[b]Number indicates number of specimens tested.

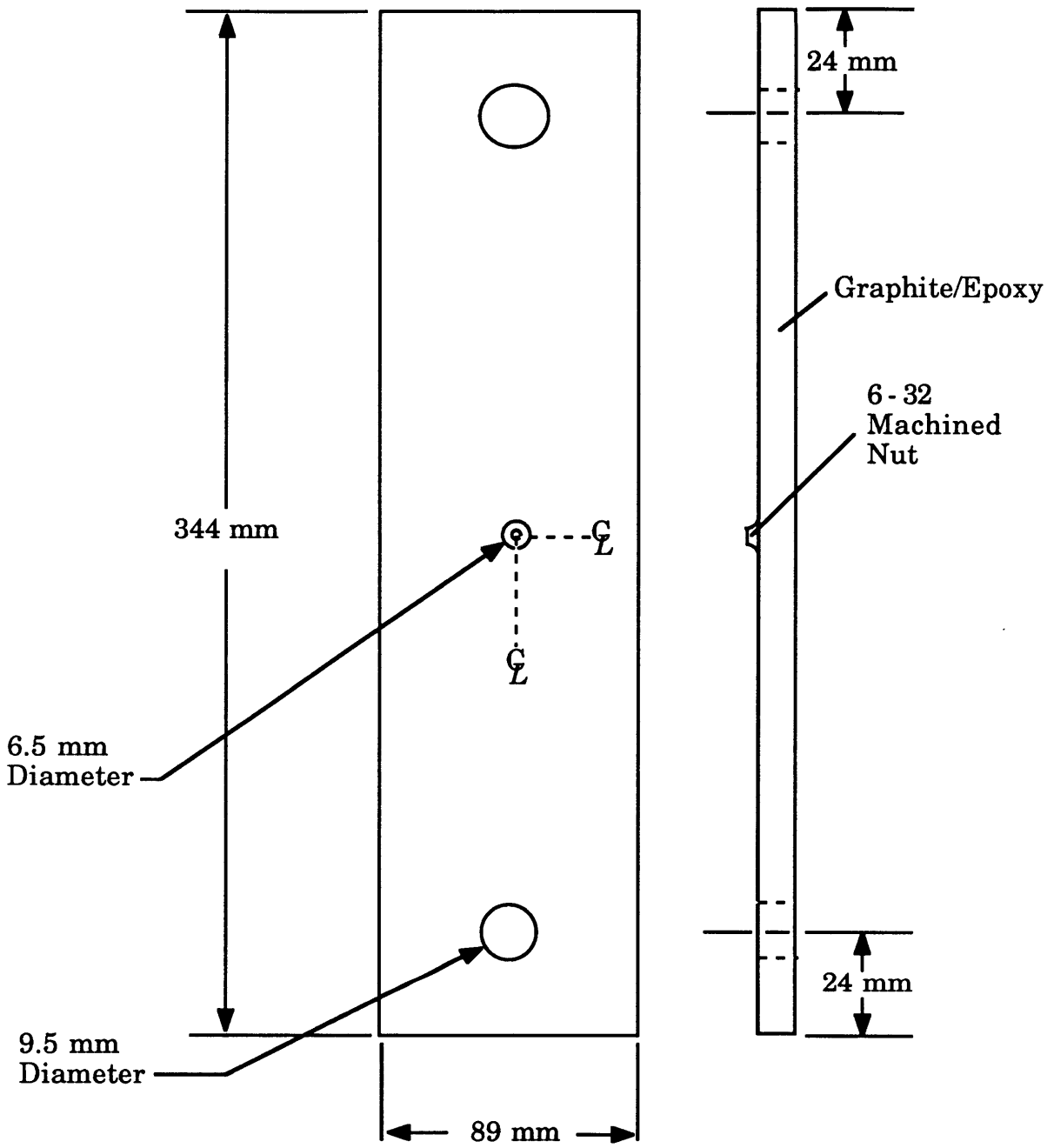


FIGURE 4.1 Test specimen geometry.

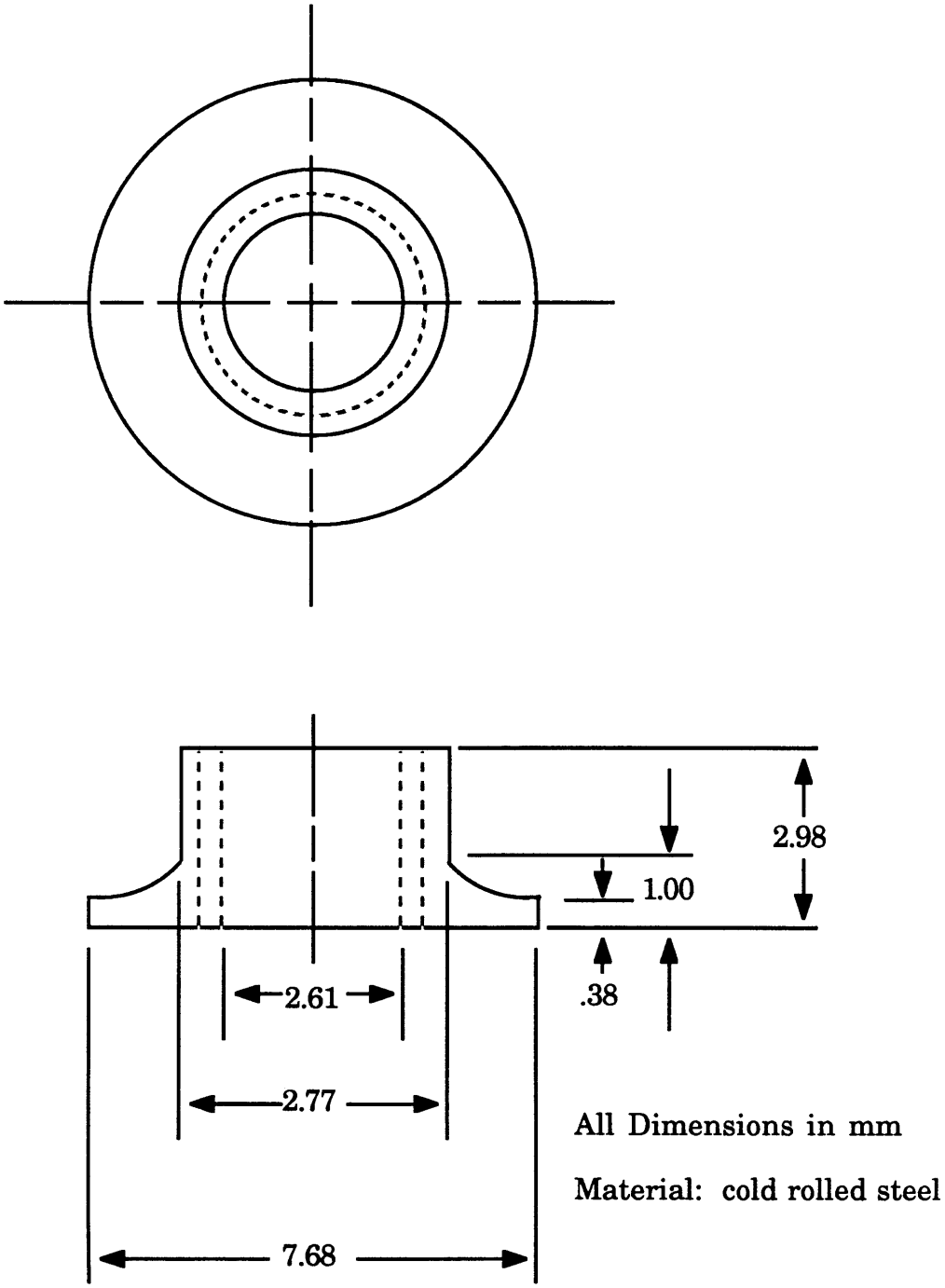


FIGURE 4.2 Geometry of 6-32 machined nut.

4.2 Manufacturing Procedures

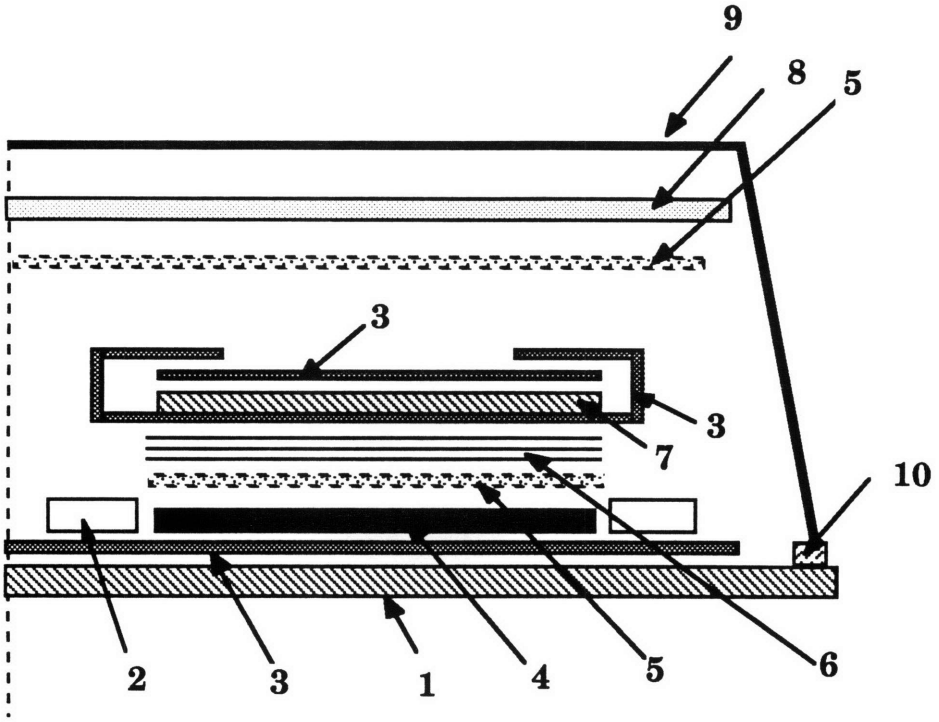
The manufacturing process consists of layup, cure, postcure, machining, drilling, and bonding. Each of these processes are discussed below.

All specimens were made with AS4/3501-6 graphite/epoxy prepreg. The prepreg is stored in a tightly sealed bag in a freezer below -18 °C. Before layup, the prepreg is removed from the freezer but left in the bag for 30 minutes. This allows the prepreg to warm up without having water vapor condense on it. Any moisture on the prepreg can result in laminates with poorer quality.

The specimens were carefully cut so that proper angles are obtained. Aluminum templates covered with teflon were used with a straight edge to obtain the proper angle. A sharp Stanley utility knife was used to make the cuts. Any sections of prepreg with visible defects were not used. Each ply was laid up with the help of a jig to keep the angles properly aligned. A sheet of peel-ply was applied to both sides of each laminate.

The setup for the cure is shown in Figure 4.3. The laminate with peel-ply rests on a piece of nonporous teflon. On top of the laminate, starting from the laminate, are: a piece of porous teflon, several sheets of paper bleeder, a piece of nonporous teflon, and an aluminum top plate. On top of the top plate is another piece of nonporous teflon. The assembly is butted up against an aluminum dam on two sides. The remaining two sides are bounded by cork dams. The entire assembly is then covered with a sheet of porous teflon, a blanket of fiberglass air breather, and the vacuum bagging.

Once the cure plate is ready, the assembly is rolled into the autoclave. A vacuum of 640 to 760 mm Hg is applied. Next, autoclave pressure is applied



1. Aluminum Caul Plate
2. Corprene (Cork) or aluminum dams
3. "Guaranteed" Nonporous Teflon
4. Laminate with Peel-Ply
5. Porous Teflon (PT)
6. Paper Bleeders
7. Aluminum Top Plate
8. Fiberglass Airbreather
9. Vacuum Bagging
10. Vacuum Tape

FIGURE 4.3 Cure assembly cross section.

up to 0.59 MPa and held. Once the desired pressure is obtained, the heater is turned on and the autoclave temperature raised at a rate of 1 - 3 °C per minute to 116 °C and held for one hour. This is the flow stage, the point of minimum viscosity of the epoxy. The temperature is again increased at 1 - 3 °C per minute to 177 °C and held for two hours. This is the cure temperature during which a chemical reaction occurs in the epoxy. The temperature is then lowered at a rate of 3 - 5 °C per minute and the pressure is decreased. The time histories of the cure temperature, pressure, and vacuum are shown in Figure 4.4.

The laminates are postcured in an oven at 177 °C for 8 hours. No vacuum or pressure is applied during the postcure. The postcure completes the chemical reaction of the epoxy.

Test specimens were cut from the laminates with a water-cooled diamond grit cutting blade on a milling machine. A 152 mm diameter blade at a spindle speed of 1100 rpm and a table feed rate of 127 mm per minute was used. Approximately 5 mm was trimmed from the edges of the laminates to remove epoxy ridges and provide uniform, straight edges.

To check for uniformity, the thickness of each specimen was measured at nine locations, and the width at three locations. These locations are shown in Figure 4.5. The average per ply thickness of all the graphite/epoxy specimens is 0.131 mm compared to the nominal ply thickness of 0.134 mm.

The density was experimentally obtained for each laminate. This was done by weighing each laminate and measuring their length and width. Using the average thickness from the nine locations shown in Figure 4.5, the volume of the laminate was obtained. The density was then obtained by dividing the mass of the laminate by its volume.

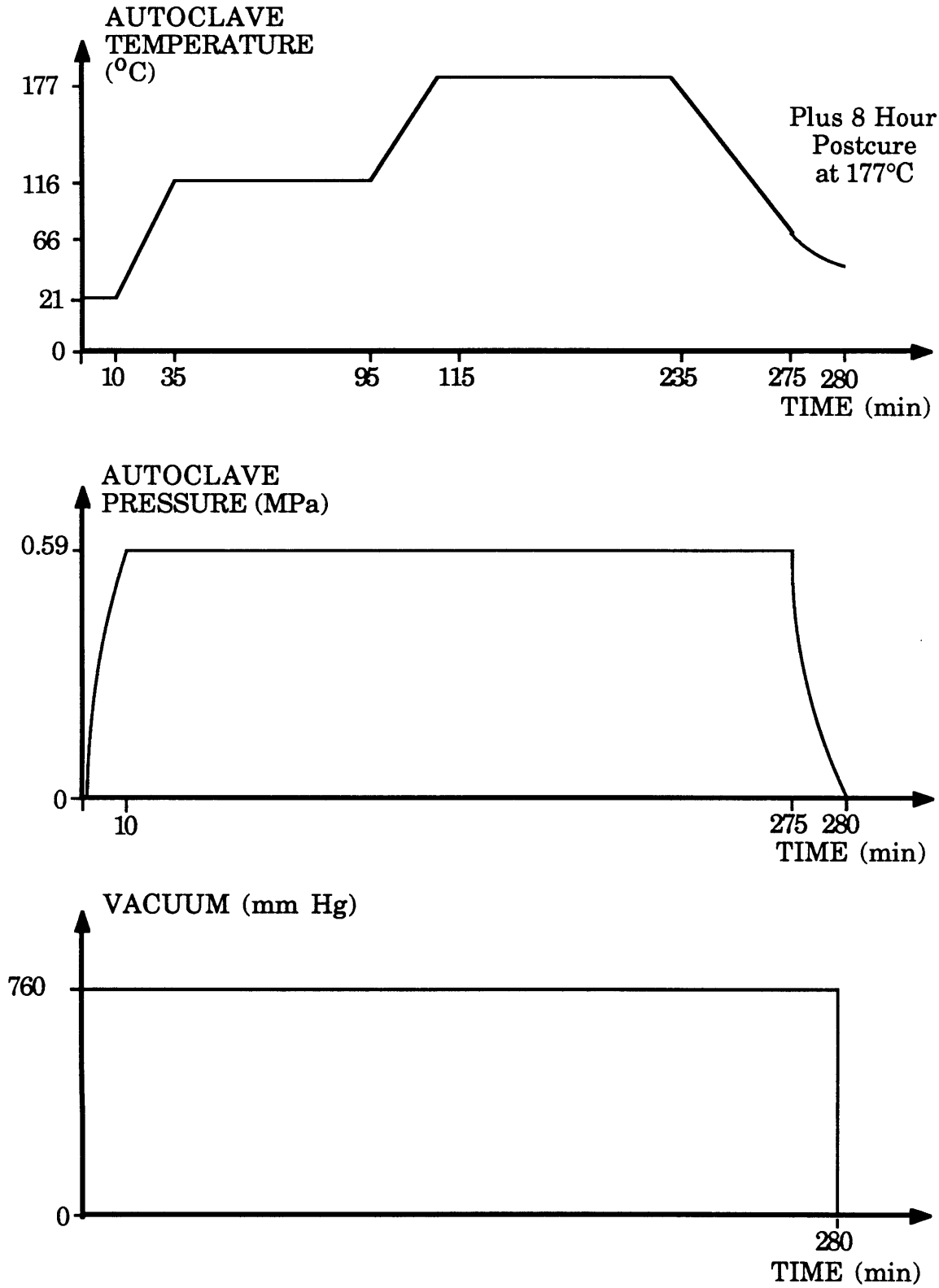


FIGURE 4.4 Curing schedule for AS4/3501-6 graphite/epoxy.

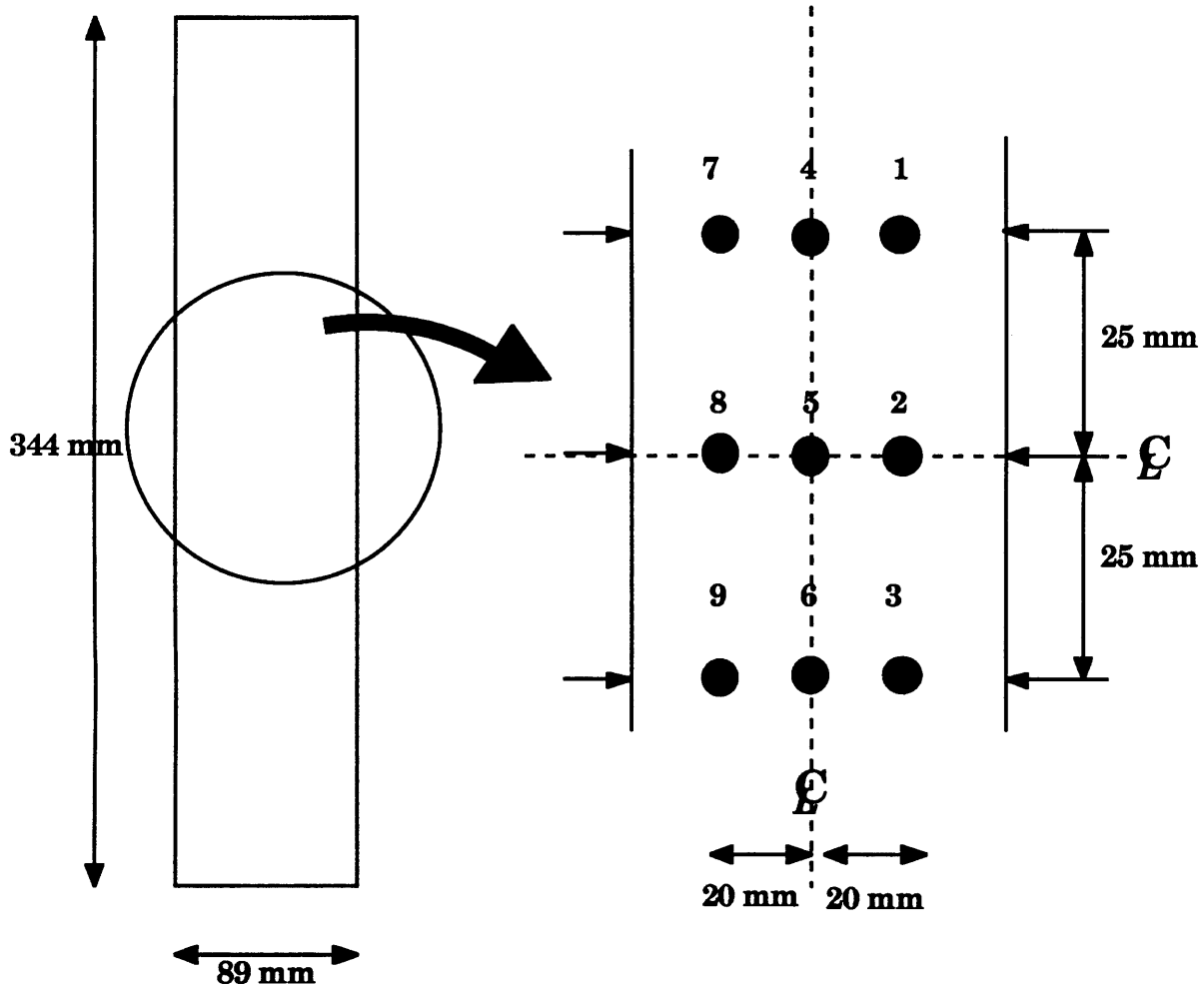


FIGURE 4.5 Location of specimen width and thickness measurements.

Test specimens that were to be impacted had holes drilled in each end to help obtain clamped boundary conditions. The holes were drilled with a 9.53 mm diameter diamond grit drill bit. A reamer was not used since the exact diameter and finish of the hole was not important.

Test specimens that were to be impacted also had a machined nut bonded to them. The bonding setup to perform this is shown in Figure 4.6. The surface of the nut to be bonded was roughed with 120 grit sandpaper. The weight of the nut was then measured. The center of the laminate was located and cleaned with acetone and cheesecloth. The aluminum positioning block was secured to the specimen with flash tape. The nut alignment rod was cleaned with cheesecloth and screwed to the nut. The surface of the nut was then placed on Cyanamid FM123-2 Adhesive Film which was cut to size with a knife. The nut alignment rod was then dropped into the positioning block and finger pressure applied. The assembly was placed in an oven and steel plates were placed on top of the nut alignment rod to keep the bond under pressure. The oven was turned on, heated to a temperature of 105°C, and held for 2 hours.

4.3 Test Procedures

Two types of tests were conducted, impact tests and quasi-static tests. These tests are explained in this section.

4.3.1 Static Indentation Tests

Static indentation tests (quasi-static tests) were performed to obtain the impactor/plate contact relation. These tests were performed with an MTS-810

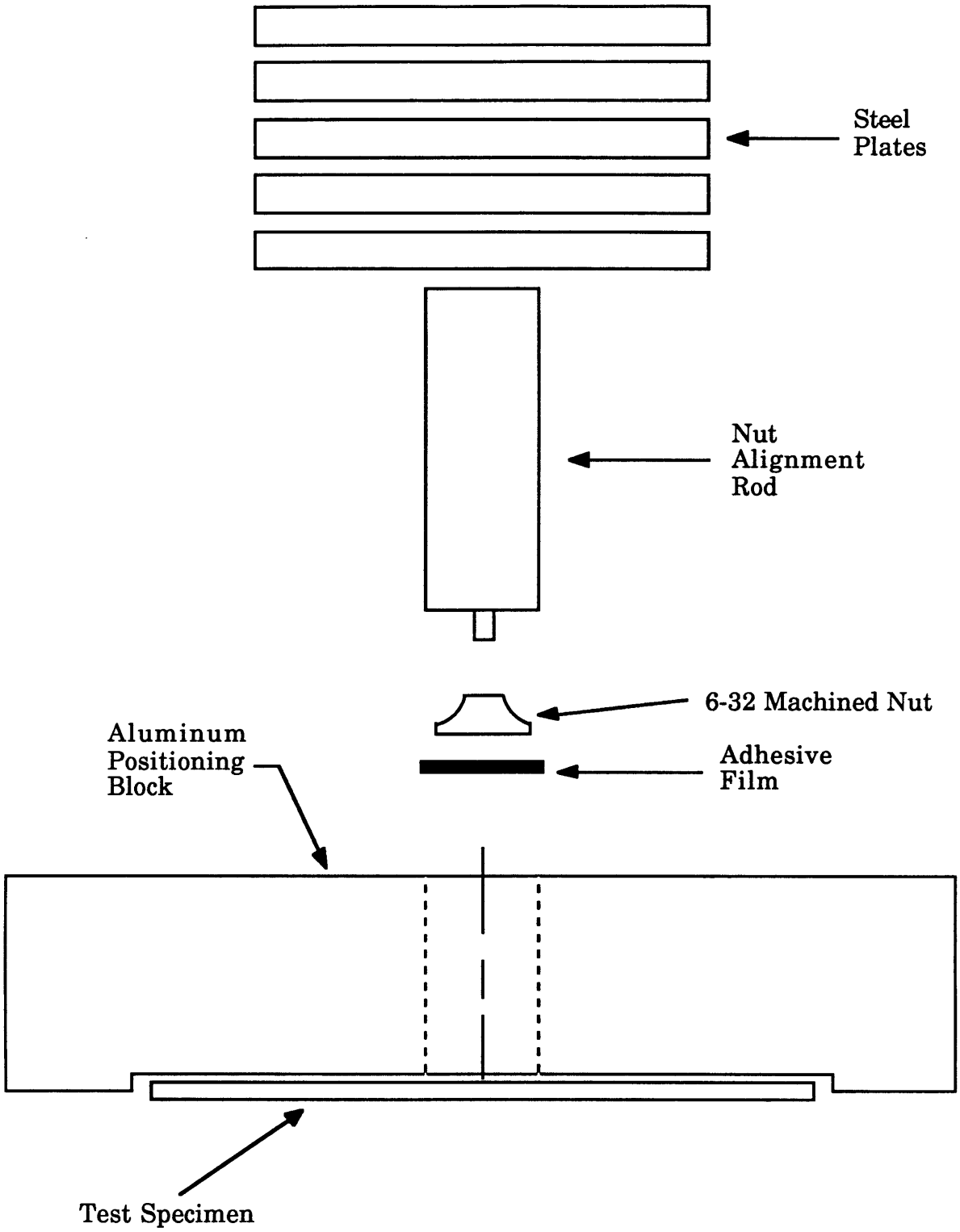


FIGURE 4.6 Illustration of bonding setup for machined nut.

uniaxial testing machine. A test jig to hold the specimen was mounted to the lower grips. A stainless steel, 12.7 mm diameter tup and a force transducer were mounted in the upper grips. The test setup is shown in Figure 4.7. The force transducer is a PCB Model 208 A05. Its signal goes to a PCB Signal Conditioner Model 484B. The Linear Variable Differential Transformer (LVDT) is a Trans-Tek Model 353.

Two different methods of clamping the specimens were used. The $[0/90]_{5s}$ specimens were mounted in the jig as shown in Figure 4.8. Square steel rods were used to clamp the specimen, giving a test length of 241 mm. A smaller test length had to be used for the $[\pm 45/0]_{2s}$ specimens to avoid damaging the laminate from excessive bending. A test length of 76.2 mm was chosen in light of the maximum load that was to be applied. The jig mounting for these specimens is shown in Figure 4.9. C-clamps were also used to help obtain clamped boundary conditions.

The tests were run as follows. The upper crosshead was lowered until the tup almost touched the test specimen. The lower crosshead was then moved up via the zeroing knob so that a slight preload of 10 to 20 N was applied. The offset for the Linear Variable Differential Transformer was recorded at this point. The test then proceeded by raising the lower crosshead at a rate of 0.102 mm/s until a peak load of 1600 N was reached. (This load is the same as the peak load which the specimens impacted at high energy experienced.) The test operation is depicted in Figure 4.7. By raising the lower crosshead, the jig moves up, which raises the clamped edges of the laminate. The Linear Variable Differential Transformer and the tup are fixed in space. Thus, indentation of the specimen is measured directly by extension of the Linear Variable Differential Transformer core. All of the data was collected at a rate of 2 Hz utilizing a DEC PDP-11/34 computer.

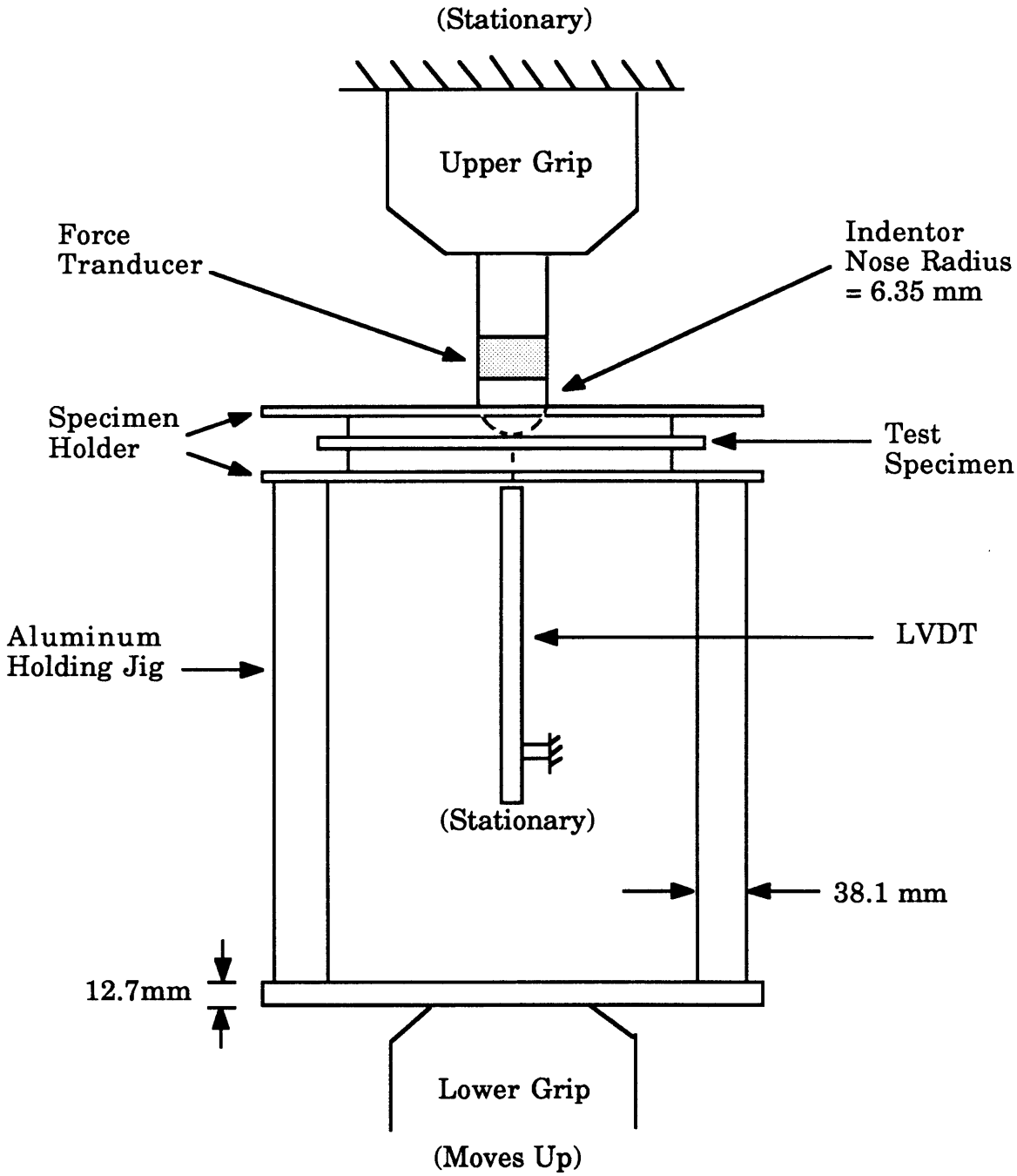


FIGURE 4.7 Illustration of test setup for static indentation test.

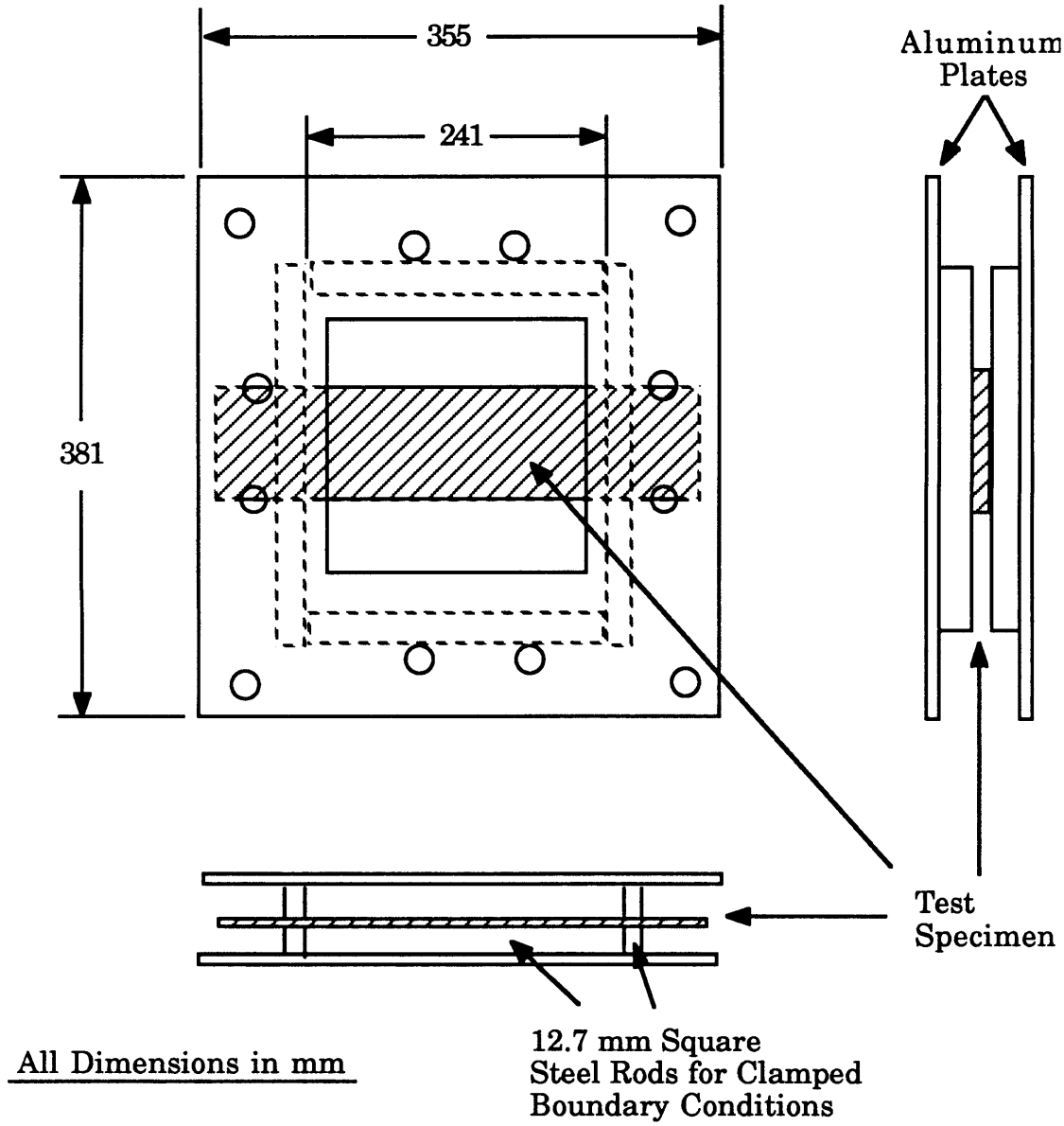


FIGURE 4.8 Illustration of specimen holder for static indentation tests of $[0/90]_{5s}$ specimens.

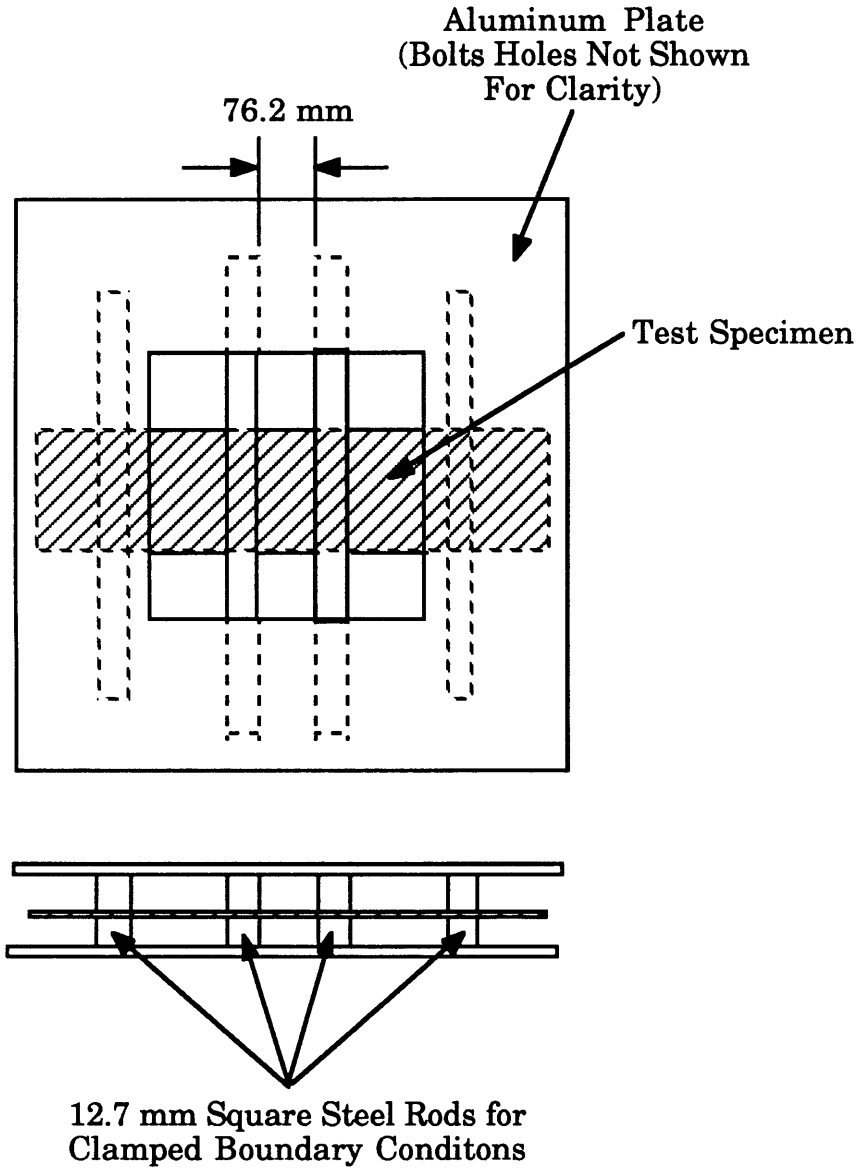


FIGURE 4.9 Illustration of the location of the clamping rods for the static indentation tests of $[\pm 45/0]_{2s}$ specimens.

4.3.2 Impact Tests

Impact test were carried out with an impact testing machine developed at TELAC [20]. The test apparatus consists of four parts: a specimen holding jig, the impactor unit, the striker unit, and the data acquisition equipment.

A schematic of the specimen holding jig is shown in Figure 4.10. This jig is mounted to a rigid steel frame [1]. Clamped boundary conditions are achieved at two ends by means of two sets of aluminum blocks. The blocks have coarse sandpaper mounted to them with double stick tape to help keep the specimen from slipping during impact. They also have a 6.35 mm diameter hole near the center. The holes at the ends of the specimens were lined up with these holes and bolts used to tighten the assembly down. This was done to provide a more uniform pressure distribution across the end of the specimen, since clamping of the four perimeter bolts alone causes the blocks to bow. All of the bolts were lubricated with "3 in 1" oil and torqued to 8.5 N·m.

A schematic of the impactor unit is shown in Figure 4.11. The 0.66 m long steel rod is free to move along its axis on linear bearings. The same PCB Model 208 A05 Force Transducer and tup used in the quasi-static tests are attached to the front end of the rod. The timing flag which is attached to the rod is a 13 mm thick plastic "doughnut." As it passes through the light gate, a light beam is interrupted. The length of time that the light beam is interrupted is measured by a CENCO Model 31707 timing system. The velocity of the rod may not be determined by simply dividing the thickness of the timing flag by the time recorded by the timing system. This is because the light beam has a finite width. The proper "thickness" to use in calculating the velocity was determined by physically moving the rod forward until the timing system was triggered. At this point, a dial gage located at the end of the rod was zeroed.

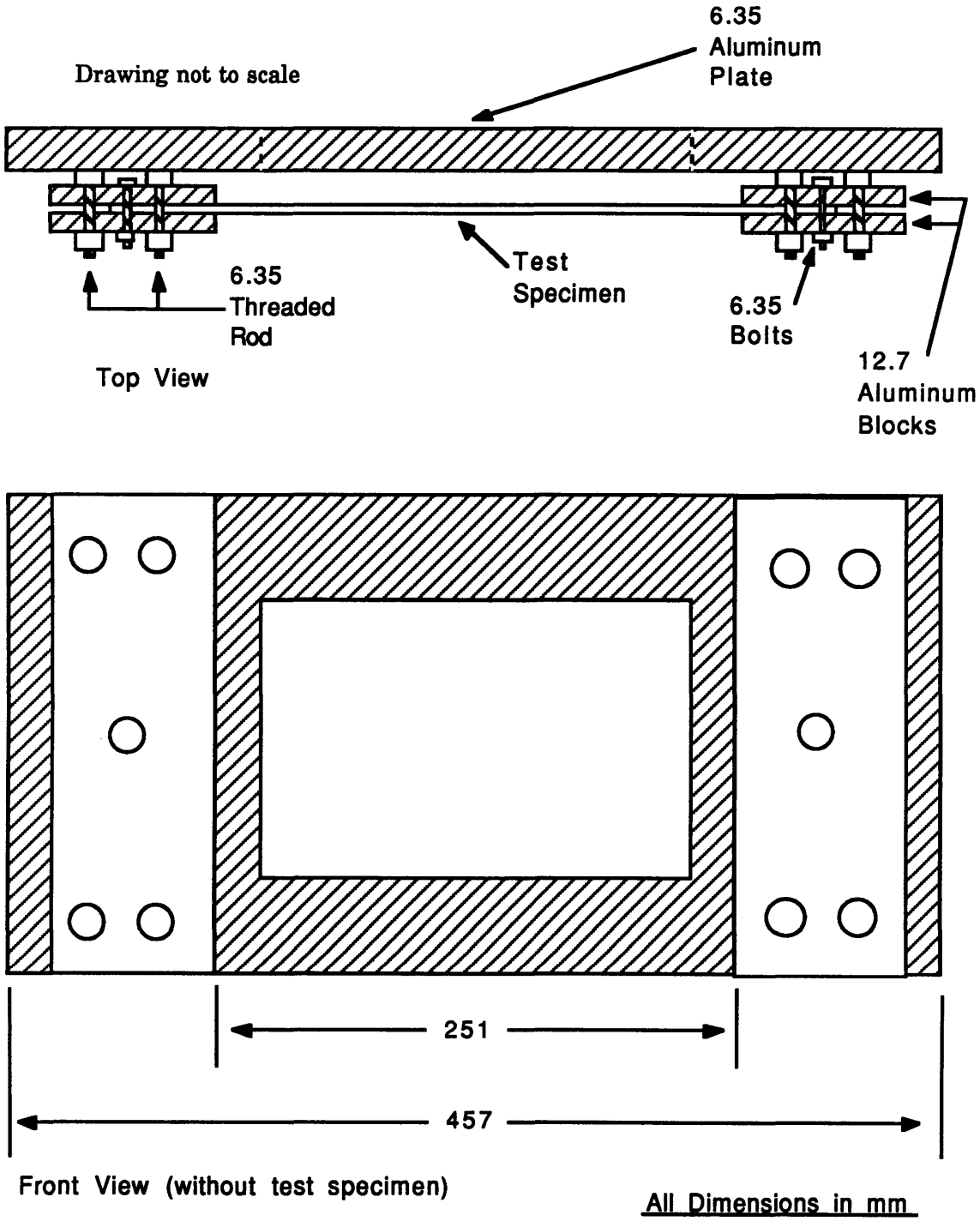


FIGURE 4.10 Illustration of specimen holding jig for impact tests.

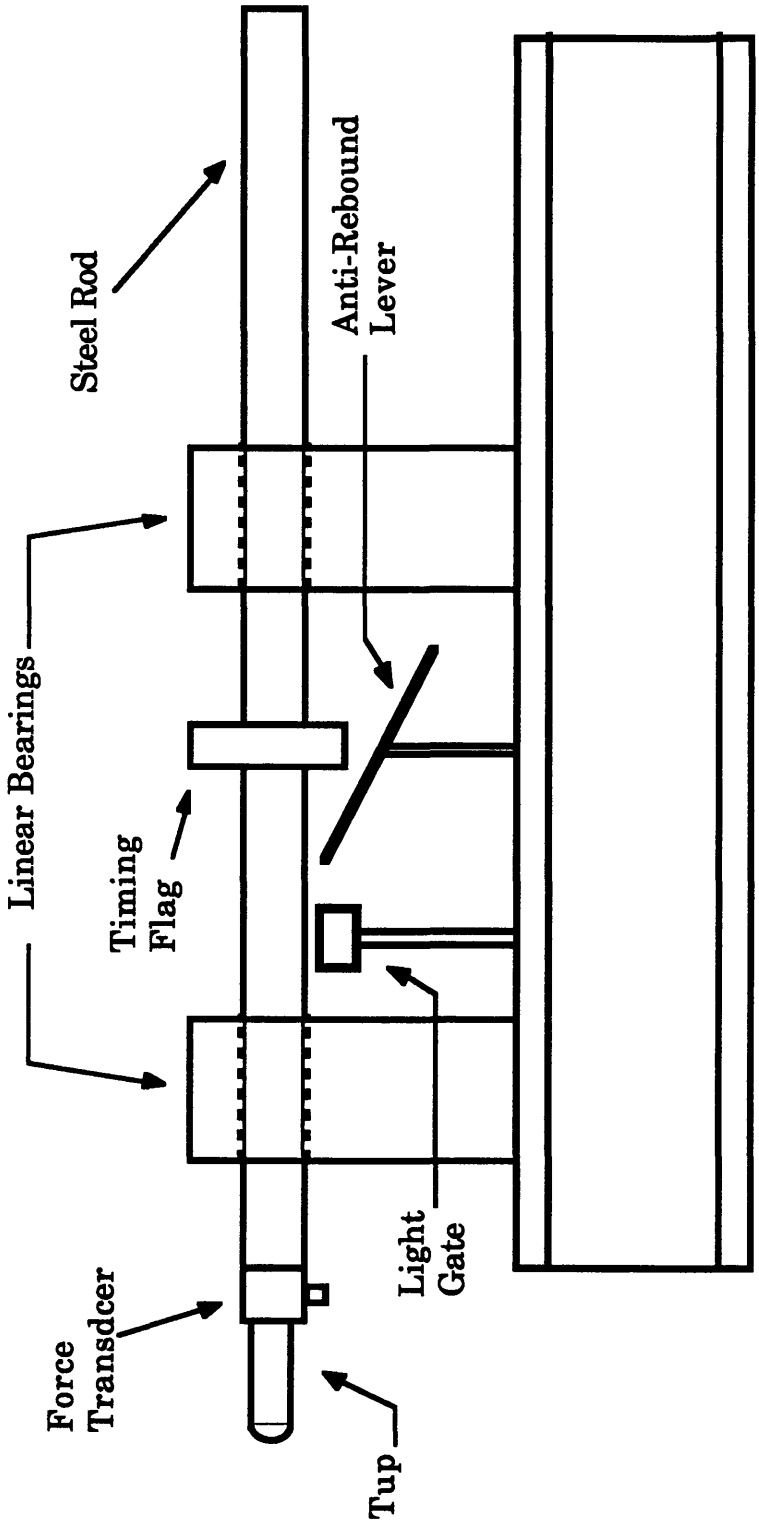


FIGURE 4.11 Illustration of impactor unit.

The rod was then moved forward until the timing system was deactivated. The distance that the rod moved was measured from the dial gage to be 12.00±0.01 mm. This is the "thickness" that was used in determining impactor velocities. A second use of the timing flag is in conjunction with the anti-rebound lever [20]. This device effectively prevents multiple impacts to the specimen.

A schematic of the striker unit is shown in Figure 4.12. Two 6700 N capacity electromagnets are energized to connect both ends of the unit. The hand winch is then used to compress the main spring. The main spring is calibrated before the tests so that it may be compressed the proper amount to obtain the desired velocity. A ruler mounted near the electromagnets is used to measure the spring compression. The electromagnets are deenergized to initiate impact. The repeatablity of velocities for the same spring compression is very good.

Two types of data are collected by a DEC Micro PDP-11/23 computer during the impact test. Force data is recorded from the force transducer and displacement data is recorded from a Trans-Tek Model 0216-0000 Linear Variable Differential Transformer with an oscillator/demodulator Model 1000-0012. The signal from the force transducer and Linear Variable Differential Transformer are sampled at a rate of 50 kHz. The signals are digitized by a Data Translation DT-3382-G-32DI analog-to-digital converter. A schematic of the Linear Variable Differential Transformer setup is shown in Figure 4.13. An aluminum extension rod (mass of 2.3 g) connects the magnetic core (mass of 5.1 g) to the back of the test specimen. The magnetic core is free to move in the Linear Variable Differential Transformer housing, i.e. it is not in contact with the walls nor is it spring-loaded.

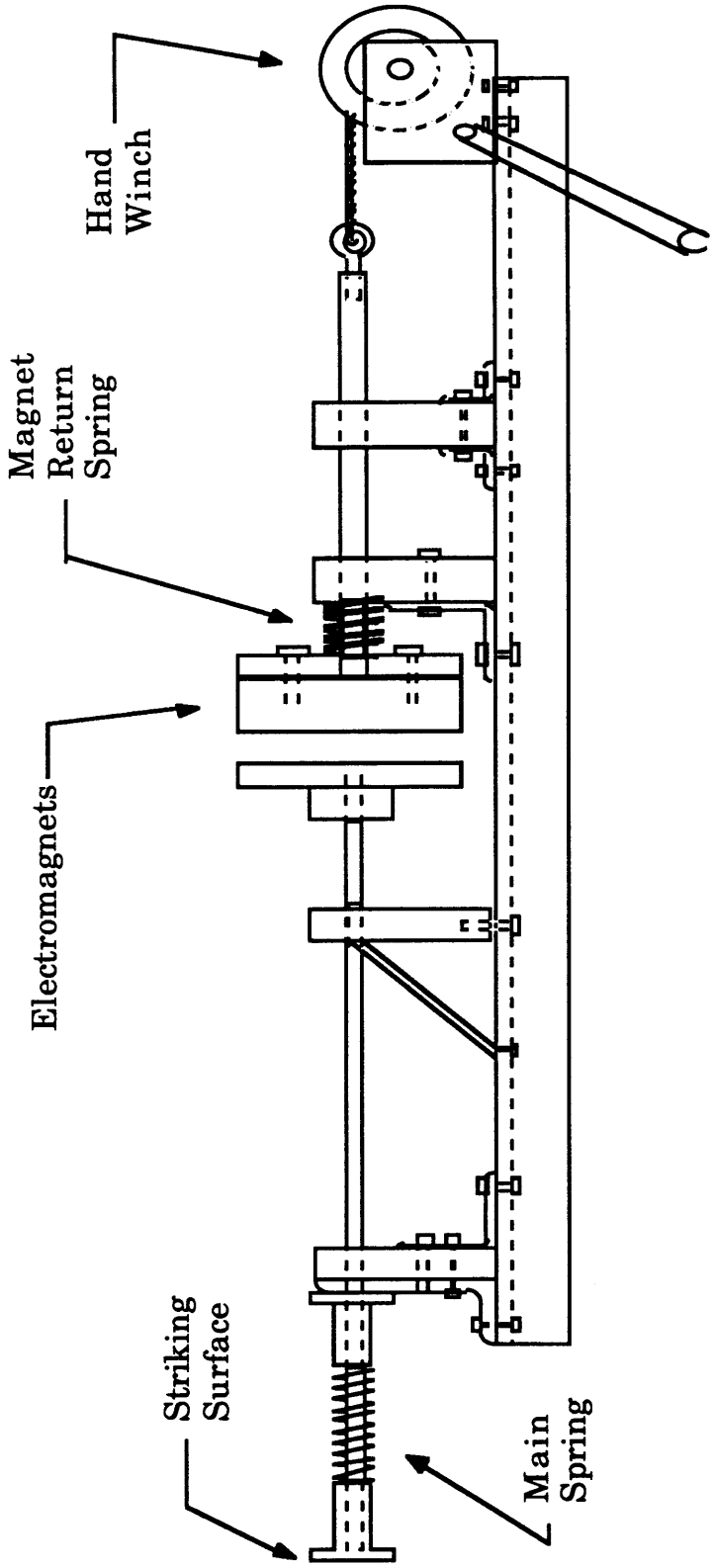


FIGURE 4.12 Illustration of striker unit.

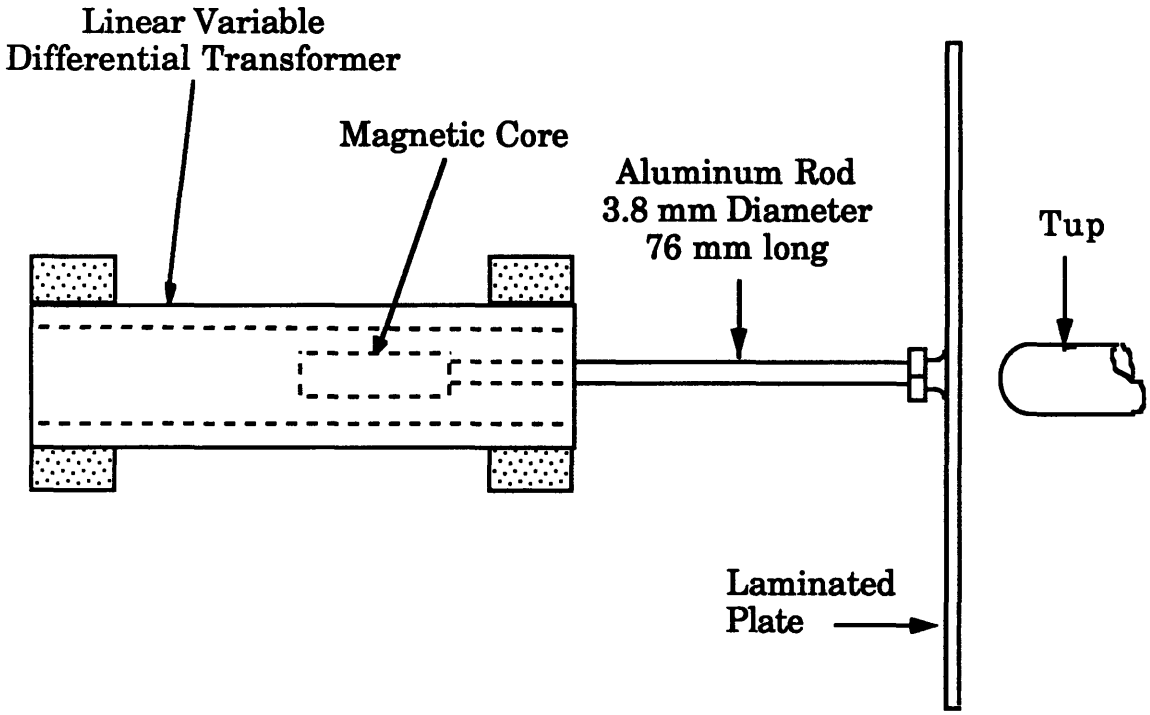


FIGURE 4.13 Illustration of experimental displacement measurement.

Since the impact event is dynamic, the force seen by the force transducer is not the same as the force at the tip of the tup. A free body diagram of the impactor rod is shown in Figure 4.14. The equation of motion of the rod is:

$$F_{Tot} = \left(M_{Tup} + M_{FT} + M_{Rod}\right) a \tag{4.1}$$

where F_{Tot} is the force at the tip of the tup, M_{Tup} is the mass of the tup (47 g), M_{FT} is the mass of the force transducer (23 g), M_{Rod} is the mass of the rod (1458 g), and a is the rigid body acceleration of the rod. The force at the tip of the tup is found by "cutting" the impactor at the force transducer and considering the resulting two free body diagrams shown in Figure 4.14. The equation of motion of the rear section is:

$$F_{FT} = \left(M_{Rod} + \frac{M_{FT}}{2}\right) a \tag{4.2}$$

where F_{FT} is the force at the force transducer. Assuming both sections to have the same acceleration, the force at the tip of the tup is found by solving for a in both Eq. (4.1) and Eq. (4.2), and equating them. The resulting equation is:

$$F_{Tot} = \left(\frac{M_{Tup} + M_{FT} + M_{Rod}}{M_{Rod} + \frac{M_{FT}}{2}}\right) F_{FT} = 1.0398\ F_{FT} \tag{4.3}$$

This equation is used to adjust the value of the force signal when the data is viewed. The raw data itself is stored in the computer along with the appropriate conversion factor.

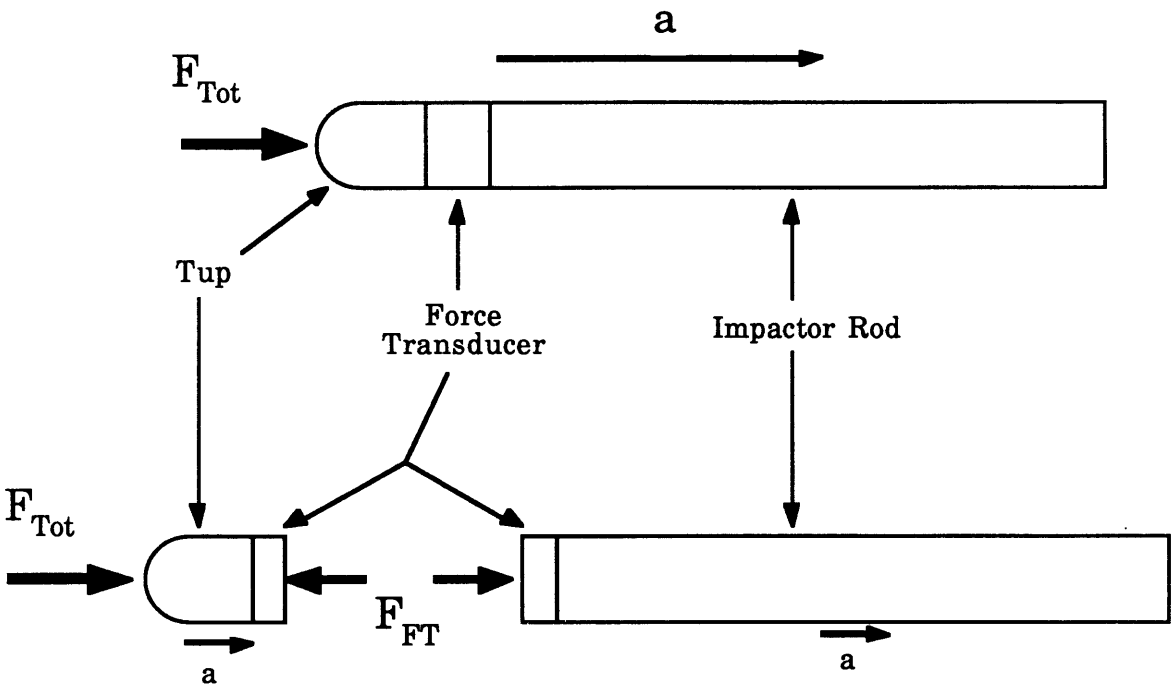


FIGURE 4.14 Free-body diagrams of the impactor rod.

In addition to measuring the force of the impactor in contact with the plate, the force transducer also detects stress waves traveling in the impactor. An example force history of the stress waves is shown in Figure 4.15. This is data of the impactor resting and then being hit by the striker unit. The rod is then traveling along the linear bearings towards the test specimen. The time constant of the decay is approximately 16 ms, which is the same order of magnitude as the length of impact to the specimen. The natural frequencies of the rod were determined by analyzing this force data with a Fast Fourier Transform (FFT). The result is shown in Figure 4.16 and summarized in Table 4.3. The fundamental frequency of the rod was also calculated from the material properties and geometry of the rod. The modulus of elasticity for steel is 199×10^3 MPa and the density is 7.83×10^3 kg/m^3. The stress wave speed is calculated as the square root of the modulus of elasticity divided by the density. Performing this calculation results in a wave speed of 5030 m/s. The distance from the force transducer to the end of the impactor rod and back to the force transducer is 1.33 m. Dividing the wave speed by the distance traveled, and multiplying by 2π, gives the frequency as 23,800 rad/s. This is close to the fundamental frequency of 23,000 - 23,600 rad/s reported in Table 4.3.

An example force history of an impact event, with the stress wave present, is shown in Figure 4.17. Sending this data through the Fast Fourier Transform results in the data shown in Figure 4.18. The rod frequencies are clearly present. Note, however, that the frequency bandwidth is not as sharp. Thus, the smallest of the amplitudes for each frequency range stated in Table 4.3 is noted. The amplitudes of the neighboring frequencies are compared to this amplitude. If the amplitude was larger, it was set to zero in the frequency domain, along with the frequencies stated in Table 4.3. This method provided a systematic way in which to eliminate the presence of the rod fre-

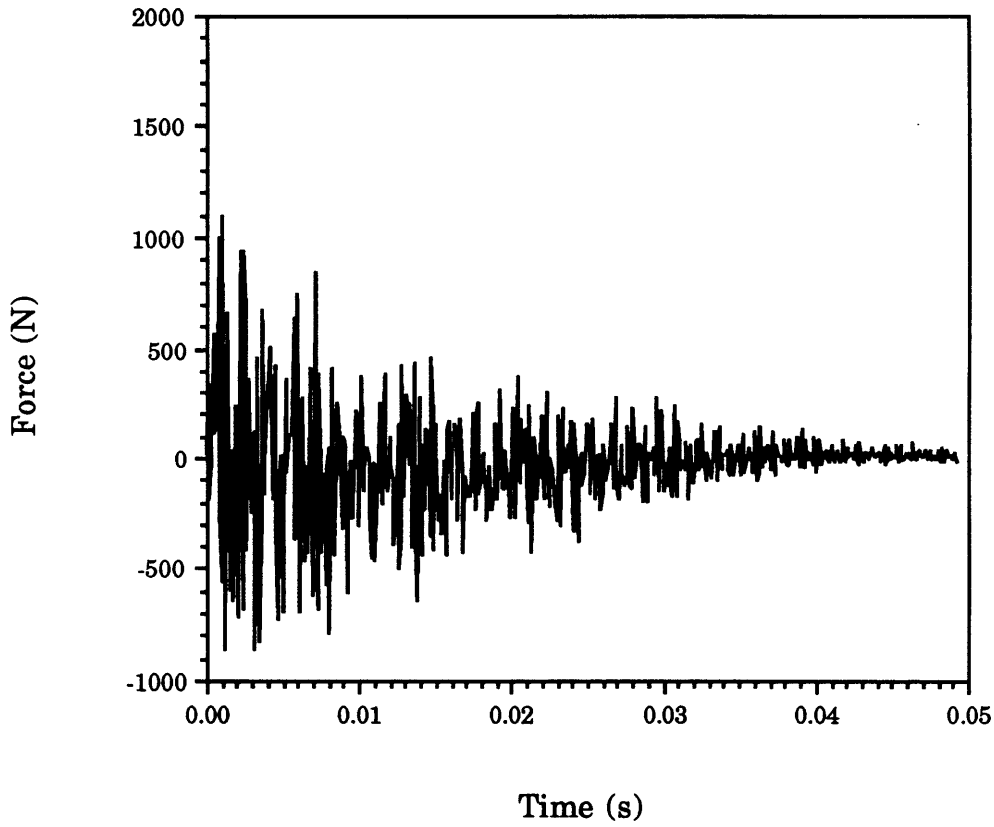


FIGURE 4.15 Plot of force versus time for the impactor rod hit by the striker unit.

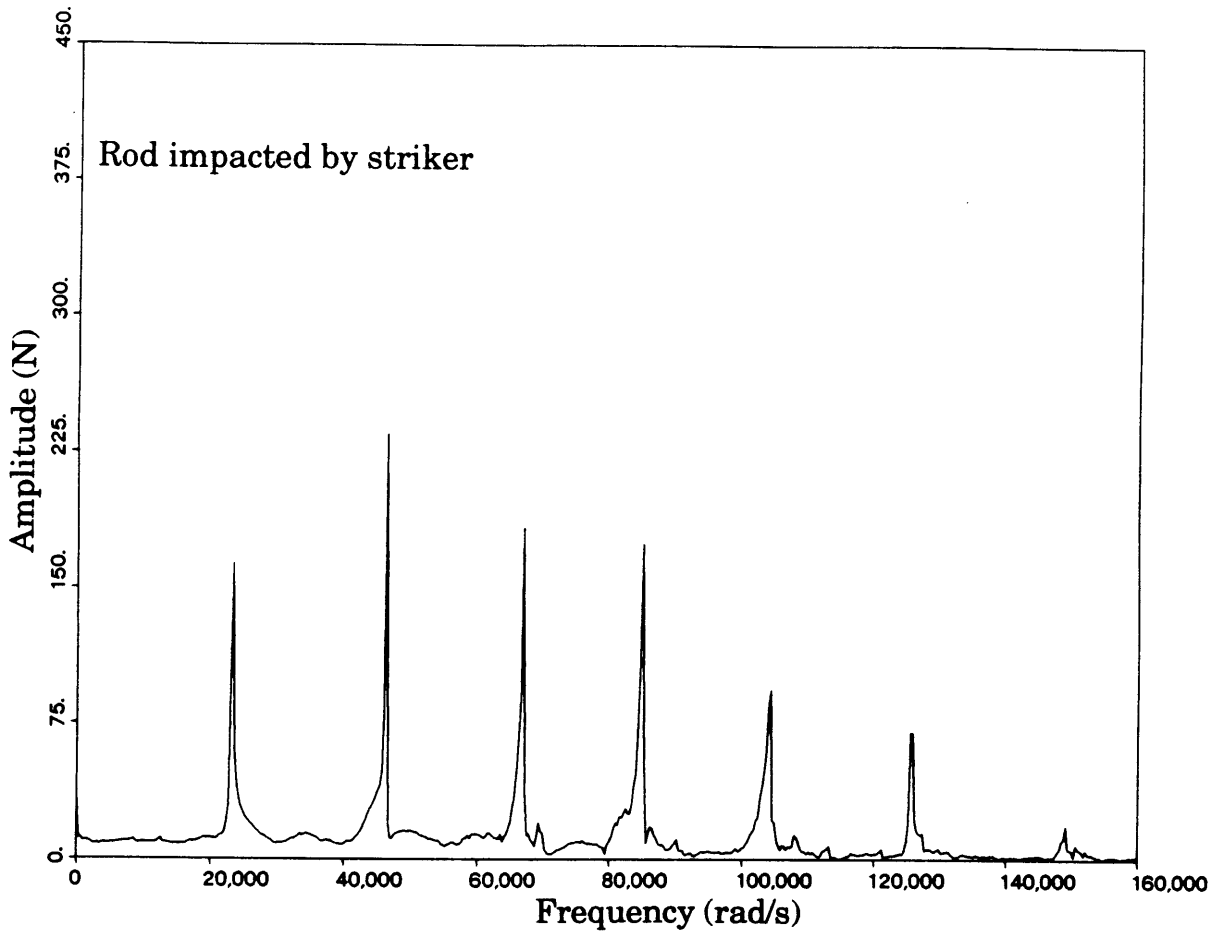


FIGURE 4.16 Results of Fast Fourier Transform to obtain experimental frequencies of the impactor rod when struck by the striker unit.

TABLE 4.3 Impactor Rod Natural Frequencies

Frequency No.	Frequency (rad/s)
1	23,000 - 23,600
2	46,000 - 46,300
3	66,300 - 66,900
4	84,400 - 85,300
5	103,700 - 104,300
6	125,500 - 125,800

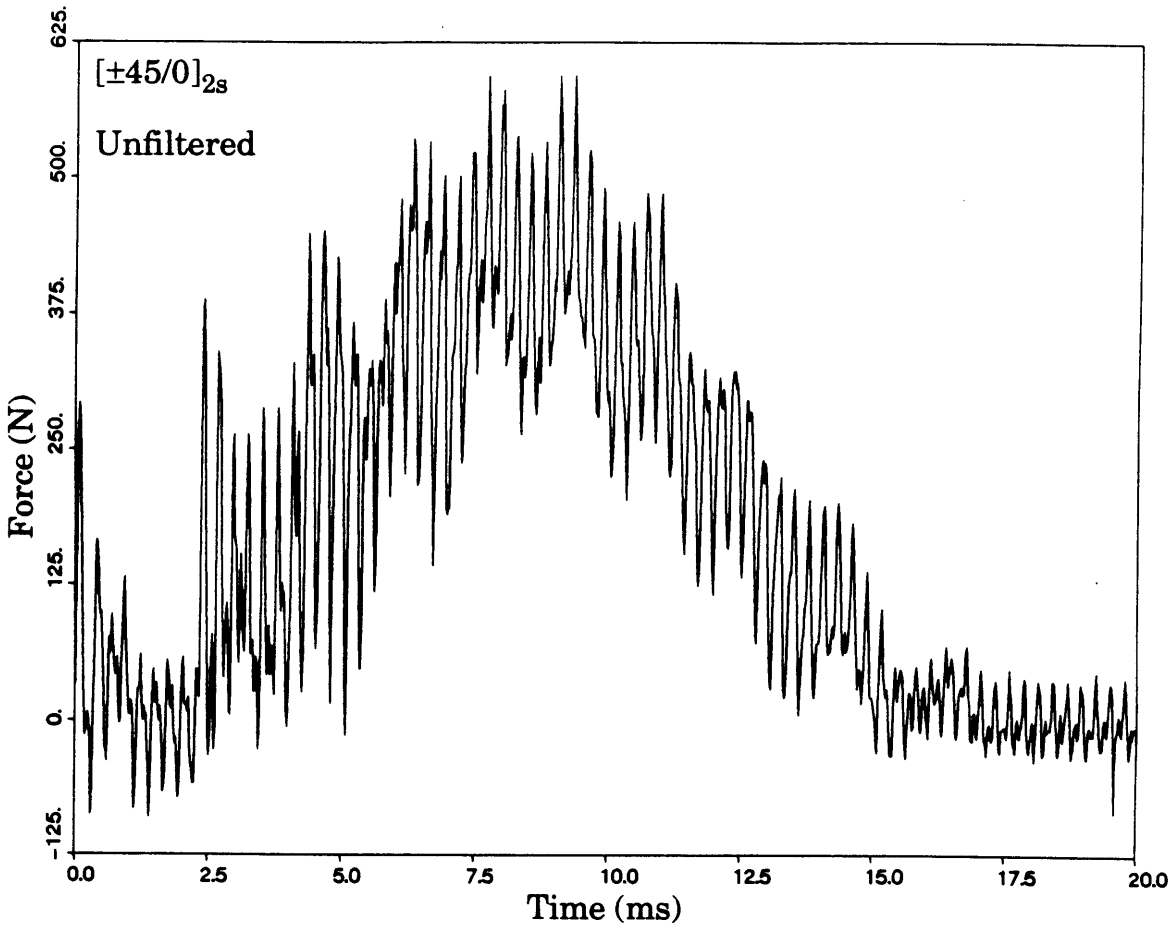


FIGURE 4.17 Typical impact force history of a $[\pm 45/0]_{2s}$ undamaged laminate with the stress wave present.

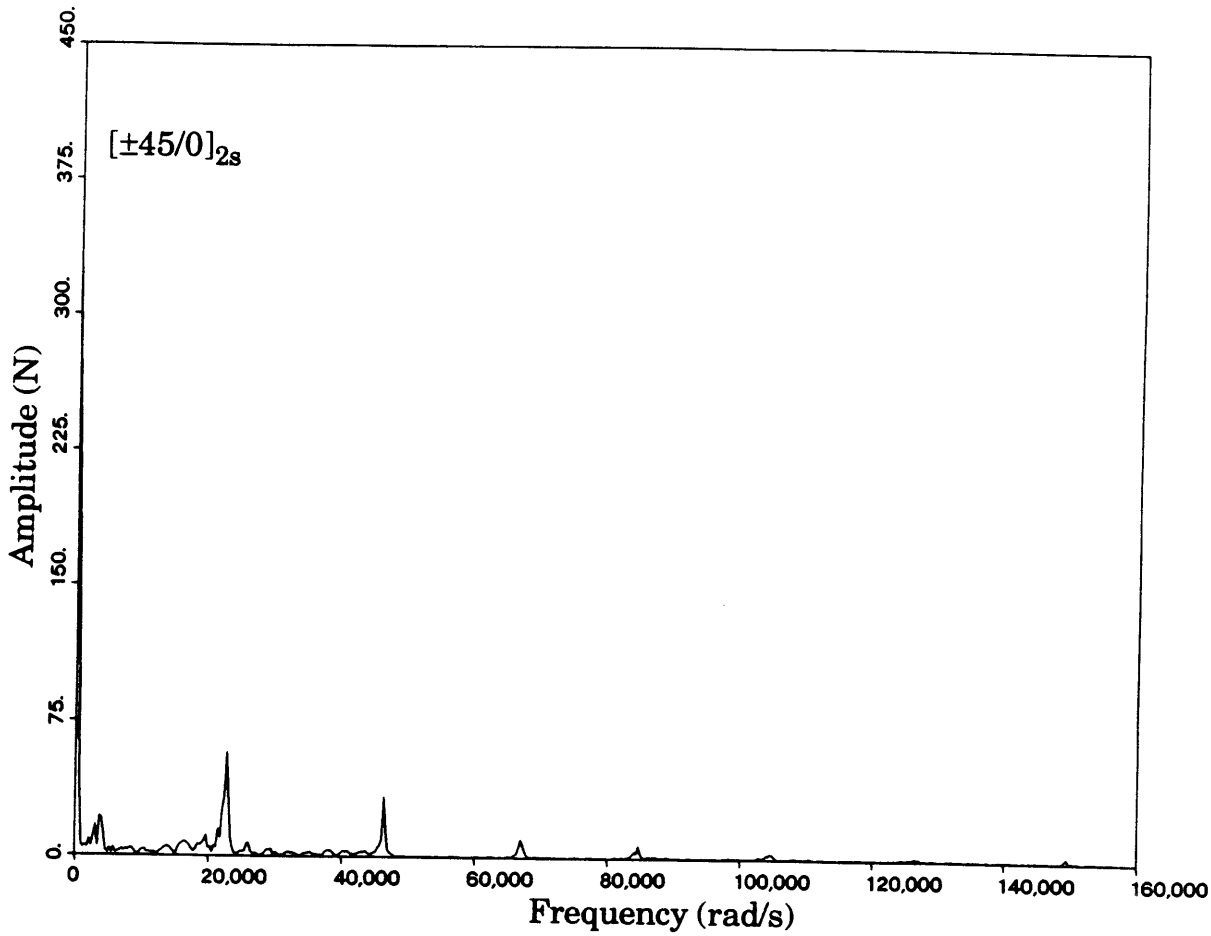


FIGURE 4.18 Fast Fourier Transform of the force history of the $[\pm 45/0]_{2s}$ undamaged laminate (of Figure 4.17).

quencies. Taking the inverse Fast Fourier Transform of the resulting data results in the force history shown in Figure 4.19. All force histories presented in Chapter 5 are found in this way.

4.3.3 Damage Detection

Some of the specimens were examined for damage. This was done by hand drilling a 0.794 mm diameter hole in the specimens with a high speed drill bit held in a pin vise, and injecting 1,4-Diiodobutane (DIB). This is a low viscosity liquid which penetrates any exposed cracks. The specimen was then placed on Polaroid Type 52 PolaPan 4x5 Instant Sheet Film and X-rayed with a Scanray Torrex 150D X-Ray Inspection System. The system was operated at a differential of 50 kVolts, and the film exposed to 240 mrad of X-rays using the "TIMERAD" control. The DIB blocks the X-rays, so damage shows up as dark areas.

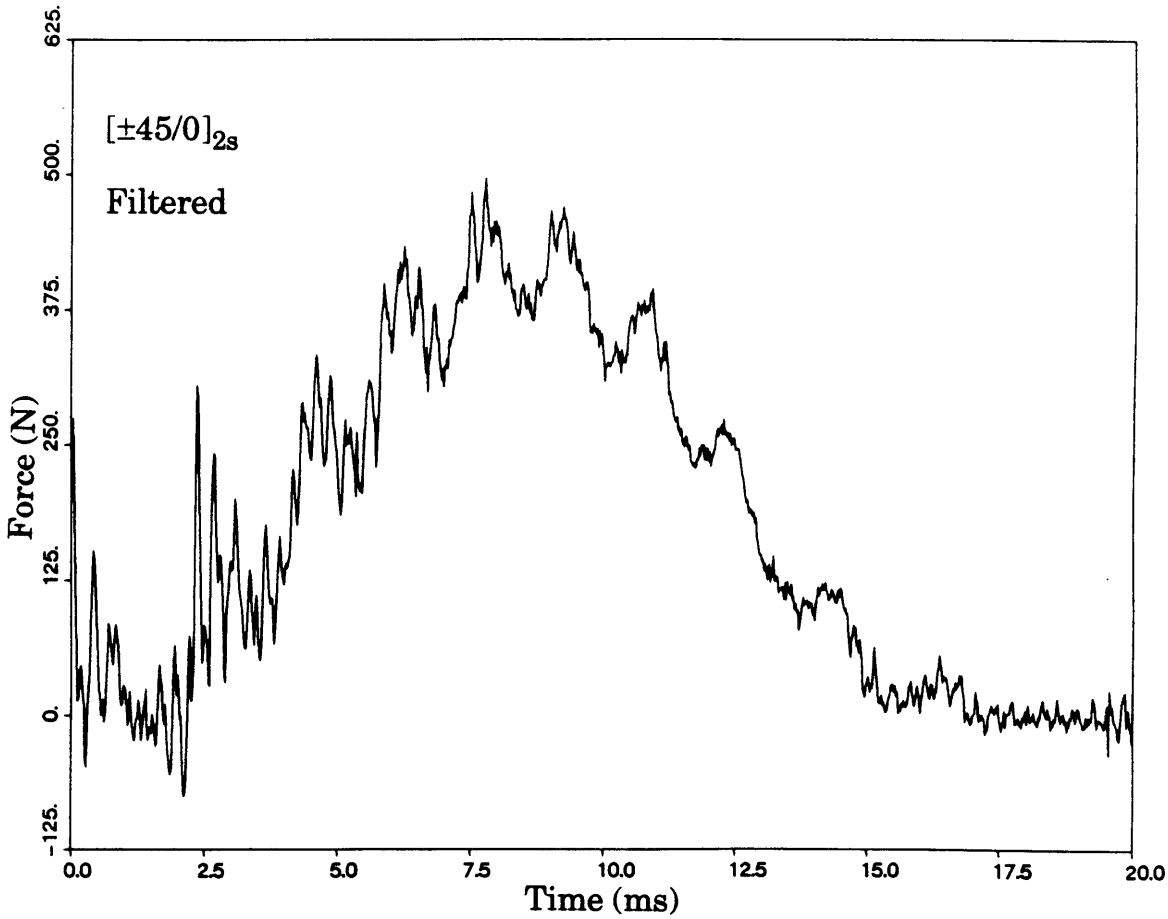


FIGURE 4.19 Impact force history of the $[\pm 45/0]_{2s}$ undamaged laminate with the stress wave suppressed.

CHAPTER 5

Experimental Results

5.1 Impactor/Plate Contact Relation

The contact results between the impactor and plate which are obtained experimentally are presented in this section. Results from both undamaged and damaged laminates are shown. The data is presented in the form of force versus indentation. The force data comes from the force transducer and the indentation data comes from the Linear Variable Differential Transformer (LVDT) shown in Figure 4.7.

An example force versus indentation plot is shown in Figure 5.1. The indentation data clearly falls on discrete values. This is because each interval is equivalent to one computer unit. This was the best resolution obtainable.

The force versus indentation data is fit to the relation

$$R = K\alpha^n \tag{2.1}$$

Taking the logarithm of this equation gives

$$\log(R) = \log(K) + n\log(\alpha) \tag{5.1}$$

which is a straight line with slope n. Thus, by plotting the logarithm of the load versus the logarithm of the indentation, a line may be fit using the method of least squares. The slope of this line gives the exponent n and the antiloga-

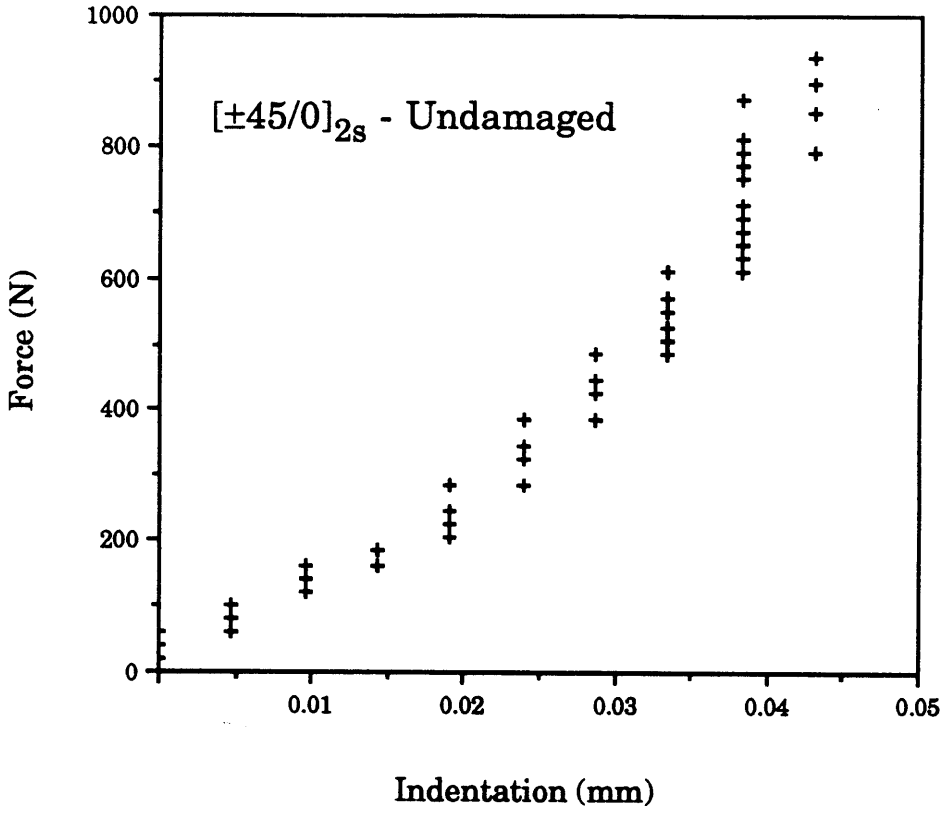


FIGURE 5.1 Typical plot of force versus indentation data.

rithm of the constant gives K. This is illustrated in Figure 5.2 where the exponent n is 1.14 and the stiffness K is antilog(4.4175) which is equal to 26.2 kN/mm^n. Note that the dimensions of K are [$Force/Length^n$]. Since n varies, the dimensions of K also vary. Hence, comparisons of K for specimens with different values of n cannot be easily made. Also note that the data starts off above the curve fit. As the indentation increases, the data falls below the line, and then eventually above it again. The slope is continuously increasing. This is an indication that Eq. (2.1) is not the best equation to describe the shape of the data. Polynomial curve fits were investigated. However, a third or forth order equation was needed to obtained a better curve fit. This introduces more variables which makes it difficult to make comparisons between different curves from the derived numbers.

5.1.1 Undamaged Laminates

The impactor/plate contact relation data for the $[0/90]_{5s}$ laminates are shown in Figures 5.3 through 5.5. The data does not go through the origin because of zeroing difficulties with the force transducer and the Linear Variable Differential Transformer. The specimen used in test 2 was previously impacted at low energy to obtain the dynamic force history. The maximum force during this impact was 660 N, which is much lower than the peak load during the static indentation test. A separate $[0/90]_{5s}$ specimen was impacted at this energy and X-rayed, and found to have no damage. Thus, it is concluded that the specimen used in test 2 was undamaged after it was impacted.

The reduced data for the three tests is summarized in Table 5.1. The average value of the exponent n is 1.09. The average value of log(K) is calculated and used with the average value of n to determine an "average" K of 21.7

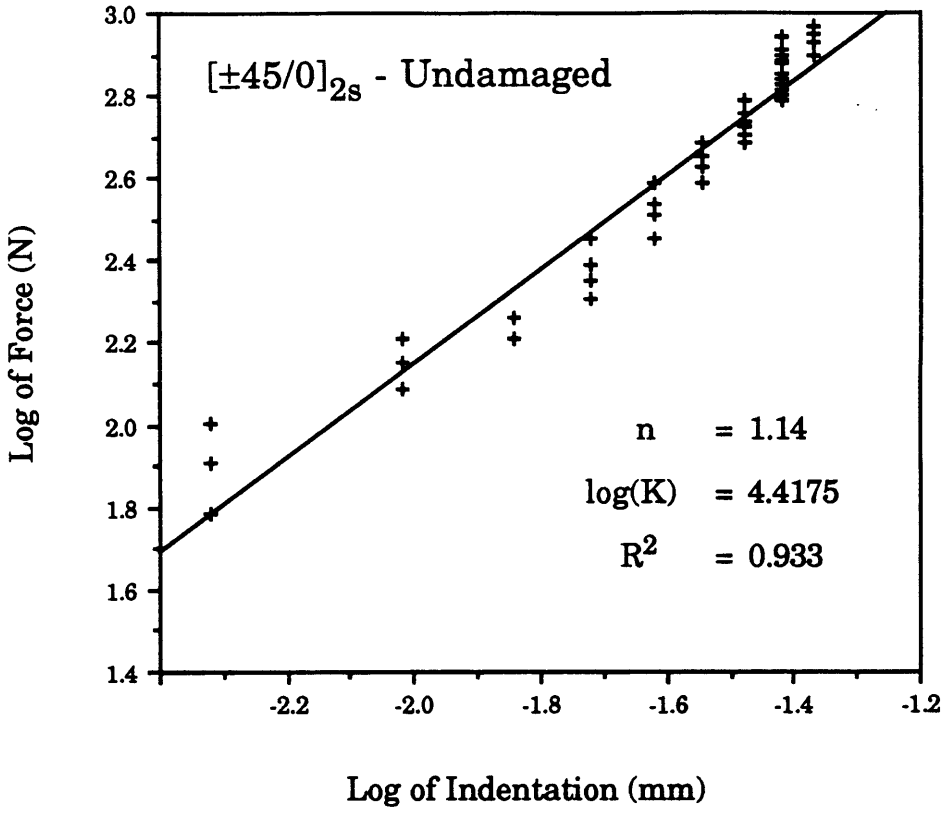


FIGURE 5.2 Typical plot of log of force versus log of indentation data.

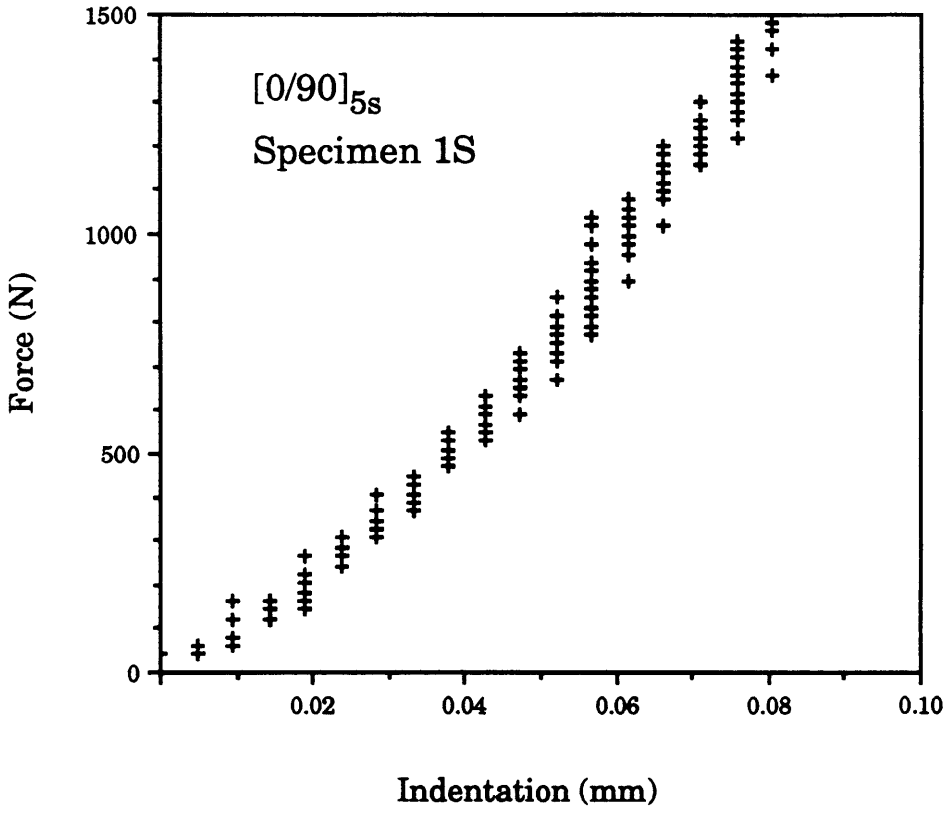


FIGURE 5.3 Plot of load versus indentation for $[0/90]_{5s}$ specimen 1S.

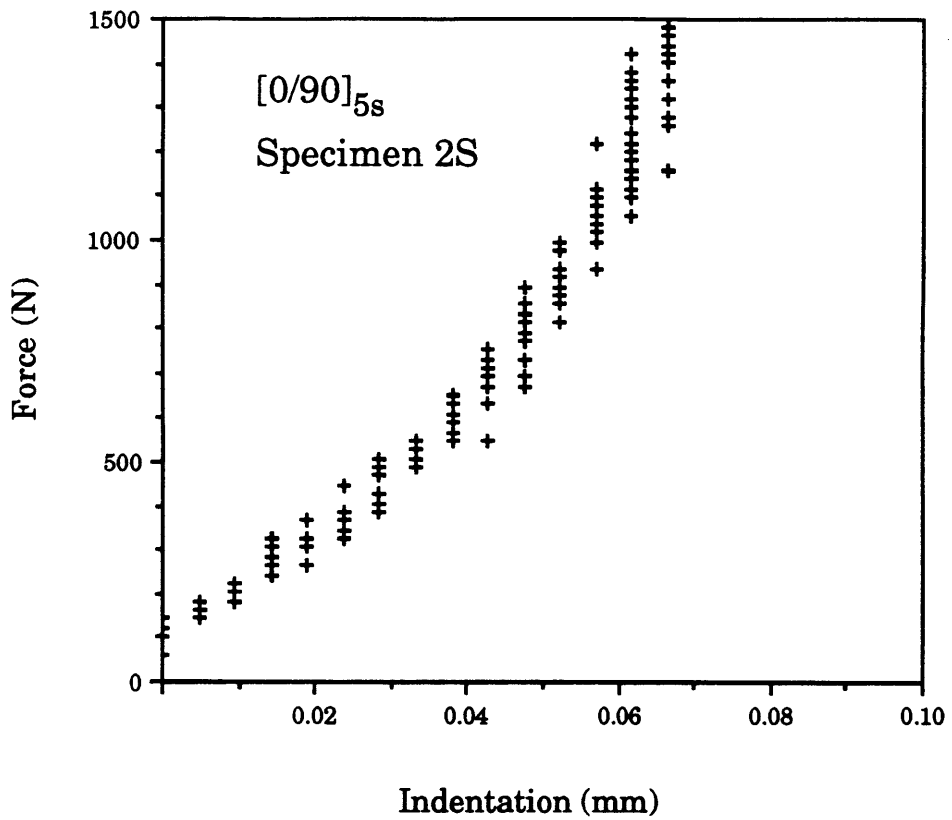


FIGURE 5.4 Plot of load versus indentation for $[0/90]_{5s}$ specimen 2S.

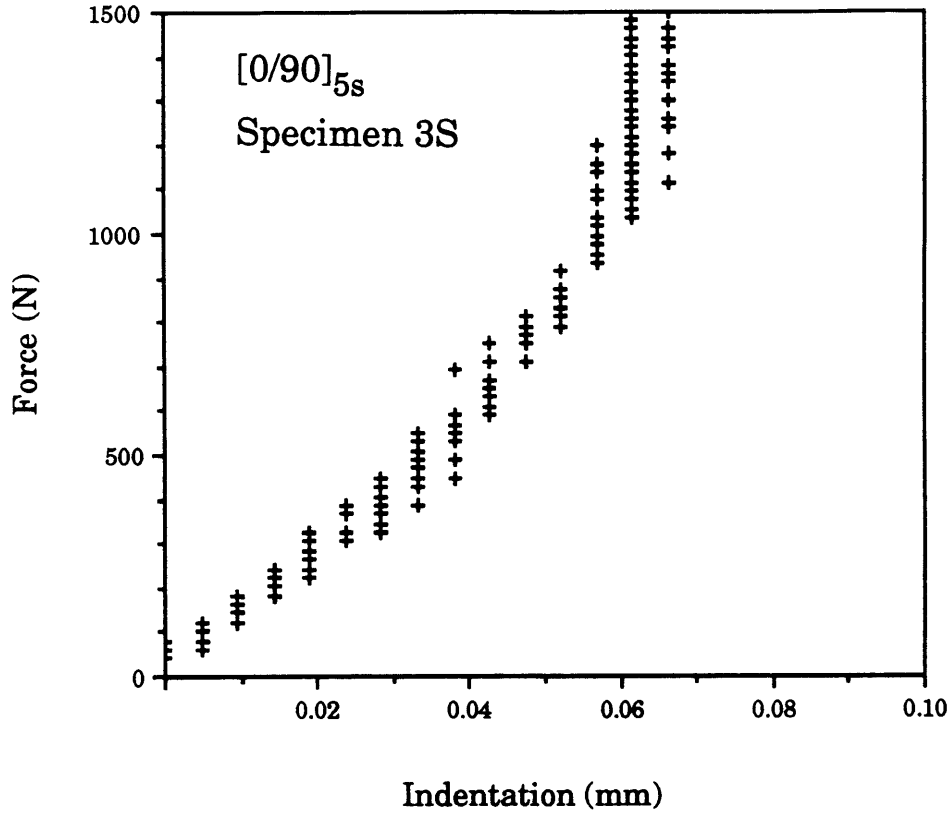


FIGURE 5.5 Plot of load versus indentation for $[0/90]_{5s}$ specimen 3S.

TABLE 5.1 Reduced Data From Experimental Static Indentation Tests of $[0/90]_{5s}$ Specimens

Test	n	log(K)	K (kN/mm^n)	R^2
1	1.29	4.55	35.3	0.978
2	0.90	4.11	13.0	0.927
3	1.08	4.35	22.2	0.946
Average	1.09 (17.9%)[a]	4.34 (5.0%)		

[a]Numbers in parentheses are coefficients of variation.

kN/mm$^{1.09}$. (This number was obtained by using more significant figures than presented in the table.) The data from test 1 shows the nonlinear contact relation well. This is also indicated by the high R^2 value of 0.978. The data from tests 2 and 3 starts off somewhat linear and then becomes nonlinear at higher indentations. The curve fits for these two specimens are not as good as for the first one with R^2 values for tests 2 and 3 of 0.927 and 0.946 respectively. The curve fit for test 3 is shown in Figure 5.6. This figure clearly illustrates the mediocracy of the $R = K\alpha^n$ curve fit to the data.

The experimental impactor/plate data for the undamaged $[\pm 45/0]_{2s}$ laminates are shown in Figures 5.7 through 5.9. The data is presented for loads up to 934 N. At this load level, a cracking sound was heard from the test specimens. This indicated that damage occurred. The test was continued up to a load of 1600 N, which is slightly less than the calculated load at which the specimens would fail due to bending. Post-test examination of the specimens revealed matrix cracks on the back surface. Thus, the data above 934 N is not considered in finding the reduced data. It should be noted that the specimen used in test 2 was accidentally cut narrow. Its width is 12% narrower, 78.4 mm as opposed to the standard width of 88.9 mm.

The reduced data is summarized in Table 5.2. The average value of the exponent n is 1.12 and the "average" value of K is 21.8 kN/mm$^{1.12}$. These numbers are very close to the numbers found for the $[0/90]_{5s}$ laminates. The force versus indentation plots have the same general trend. They appear to start off linear and then become nonlinear, i.e. the slope of the curve starts increasing for increasing indentations. The transition from linear to nonlinear occurs around an indentation of 0.02 to 0.03 mm. The curve fits have the same trend as illustrated in Figure 5.6 for a $[0/90]_{5s}$ specimen. The R^2 values

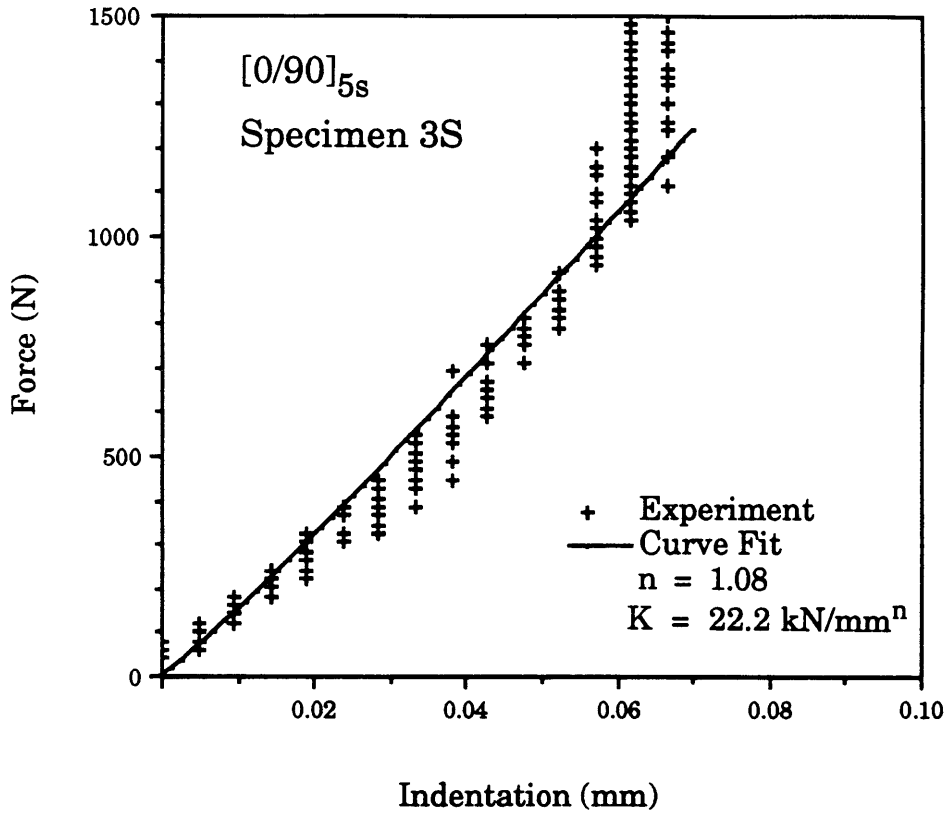


FIGURE 5.6 Plot of load versus indentation and curve fit for $[0/90]_{5s}$ specimen 3S.

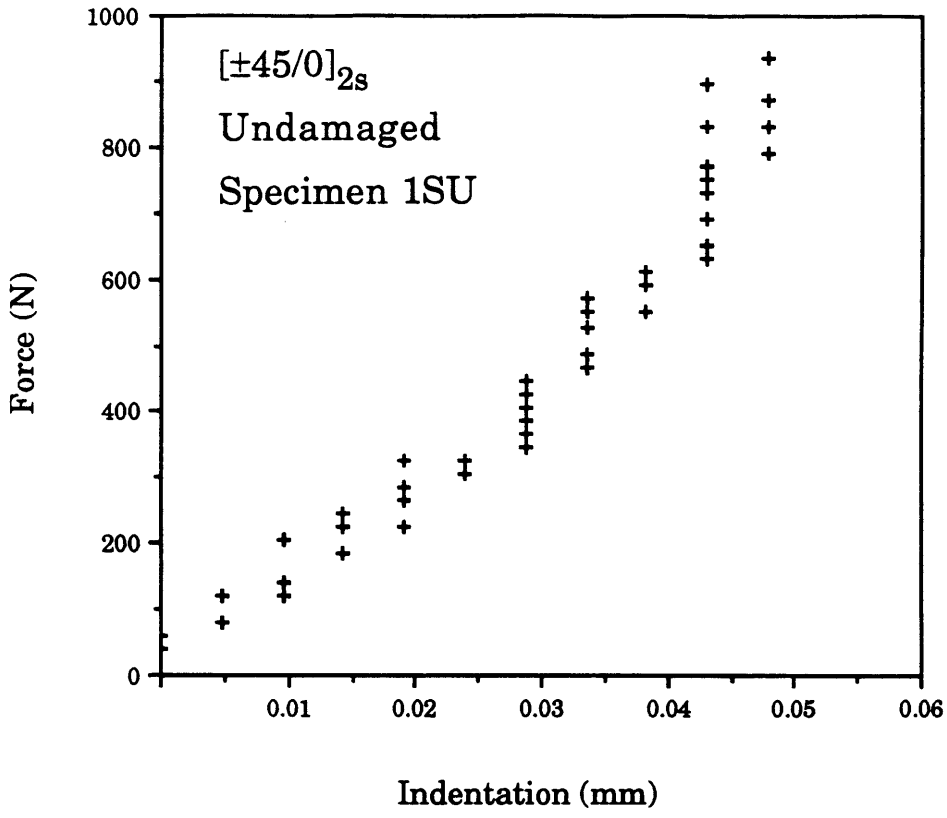


FIGURE 5.7 Plot of load versus indentation for $[\pm 45/0]_{2s}$ undamaged specimen 1SU.

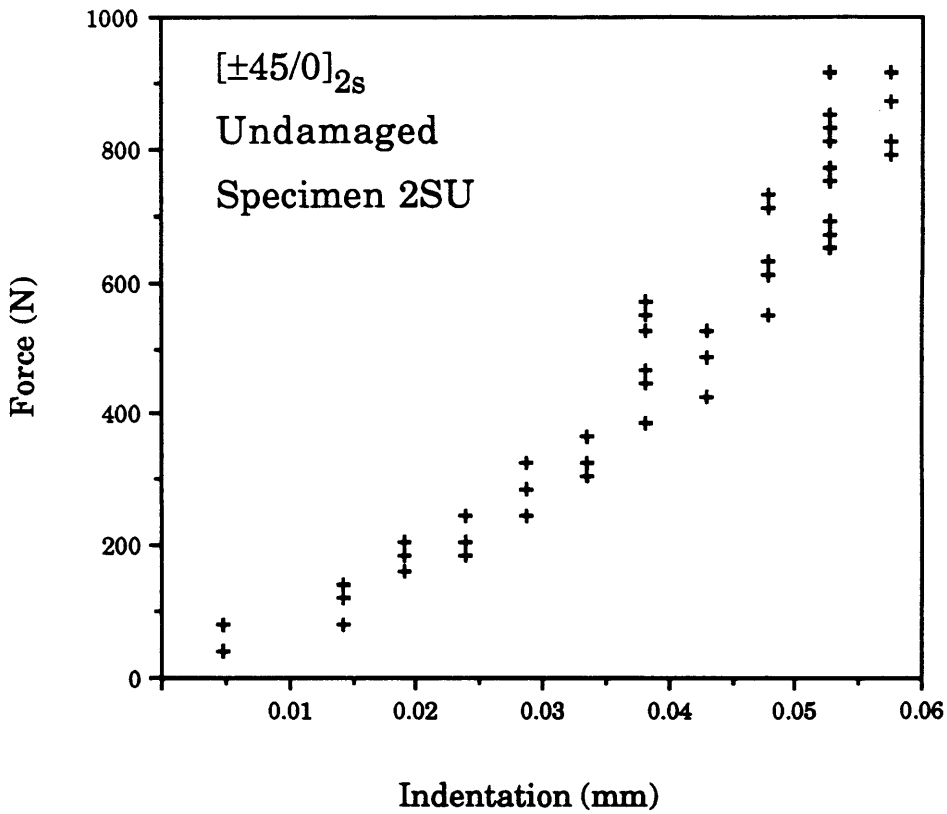


FIGURE 5.8 Plot of load versus indentation for $[\pm 45/0]_{2s}$ undamaged specimen 2SU.

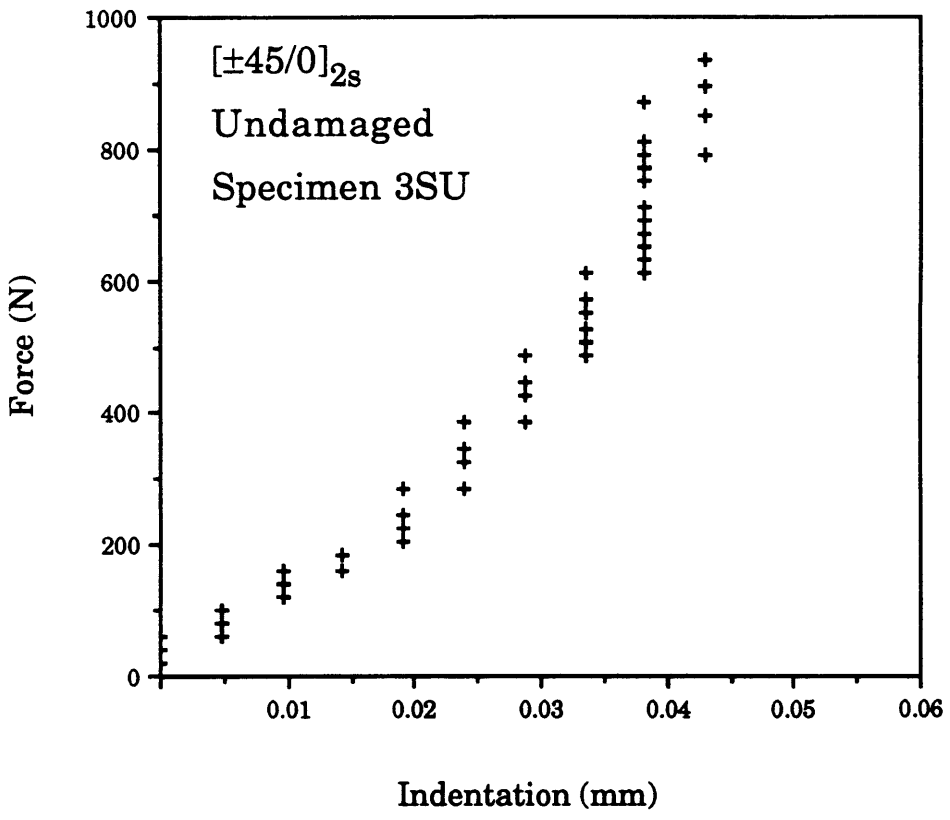


FIGURE 5.9 Plot of load versus indentation for $[\pm 45/0]_{2s}$ undamaged specimen 3SU.

TABLE 5.2 Reduced Data From Experimental Static Indentation Tests of $[\pm 45/0]_{2s}$ Undamaged Specimens

Test	n	log(K)	K (kN/mm^n)	R^2
1	0.99	4.18	15.2	0.946
2	1.24	4.42	26.0	0.921
3	1.14	4.42	26.2	0.933
Average	1.12 (11.0%)[a]	4.34 (3.1%)		

[a]Numbers in parentheses are coefficients of variation.

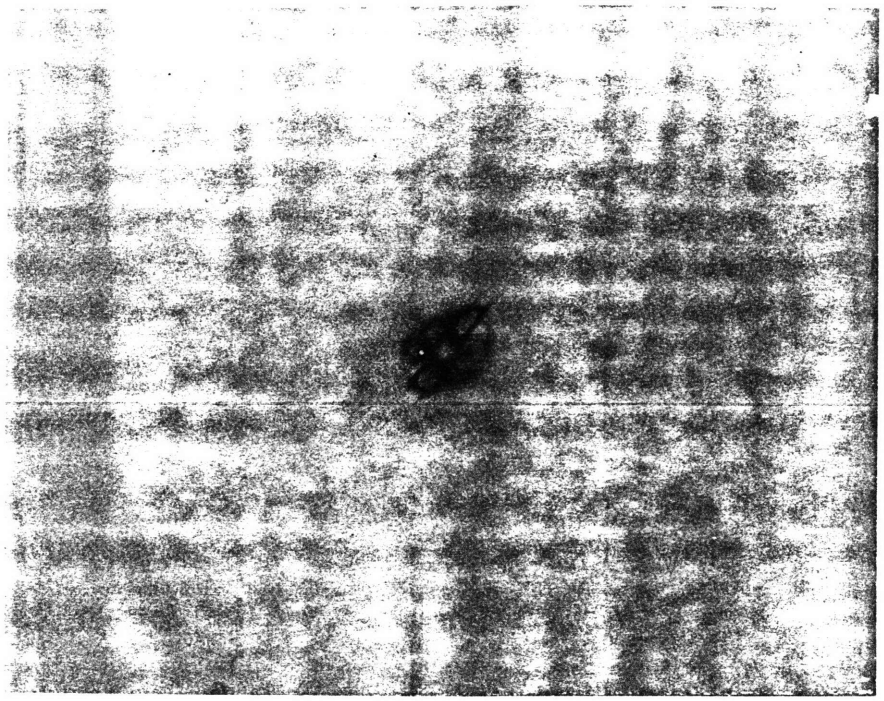
are slightly worse than the corresponding values for the $[0/90]_{5s}$ specimens. This again shows the mediocracy of the $R = K\alpha^n$ curve fit to the data.

5.1.2 Damaged Laminates

Three $[\pm 45/0]_{2s}$ laminates were impacted with the 1.53 kg impactor at 4.0 m/s (high energy) and then tested quasi-statically. A separate specimen was impacted at this energy level and X-rayed. Internal damage was present as can be seen in Figure 5.10. Matrix splits were also present on the back surface. All three of the impacted specimens tested for the impactor/plate contact relation had matrix cracks on the back surface. They were not X-rayed because the presence of the 0.794 mm diameter hole needed to inject the die may affect the test results.

The force versus indentation results are shown in Figures 5.11 through 5.13 for these predamaged specimens. For consistency with the undamaged $[\pm 45/0]_{2s}$ specimens, the data is only plotted to a load of 934 N. The reduced data is summarized in Table 5.3. The average values of the exponent n and the stiffness K are not reported. This is because each specimen may have a different amount of damage. Thus, the impactor/plate contact relation is expected to differ somewhat. This is indeed the case, as can be seen from Figures 5.11 through 5.13. The first test has a somewhat linear region up to an indentation of 0.03 mm, after which the relation becomes nonlinear. The last two tests differ from the first in that much larger loads are required to produce corresponding indentations.

Comparing the force versus indentation plots of the damaged specimens to the undamaged specimens reveals that the damaged specimens require more load for a corresponding indentation. The values of log(K) for the damaged specimens are larger than the values of log(K) for the undamaged speci-



0 1 2 cm

FIGURE 5.10 X-ray of a $[\pm 45/0]_{2s}$ specimen impacted with a 1.53 kg impactor at 4.0 m/s.

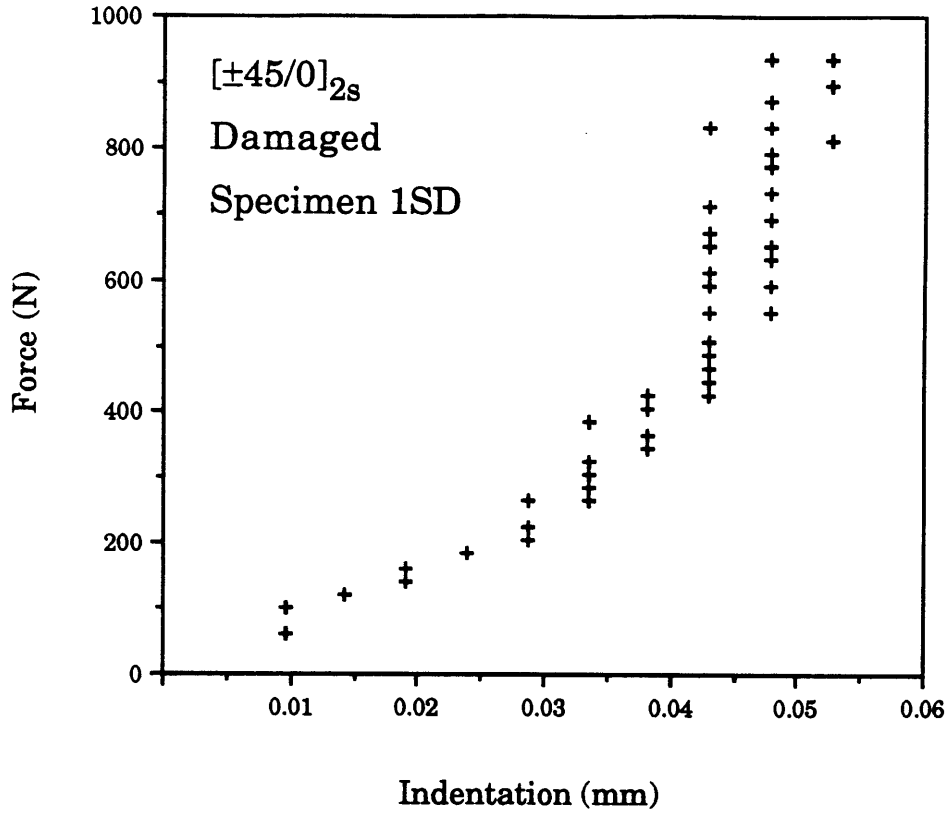


FIGURE 5.11 Plot of load versus indentation for $[\pm 45/0]_{2s}$ damaged specimen 1SD.

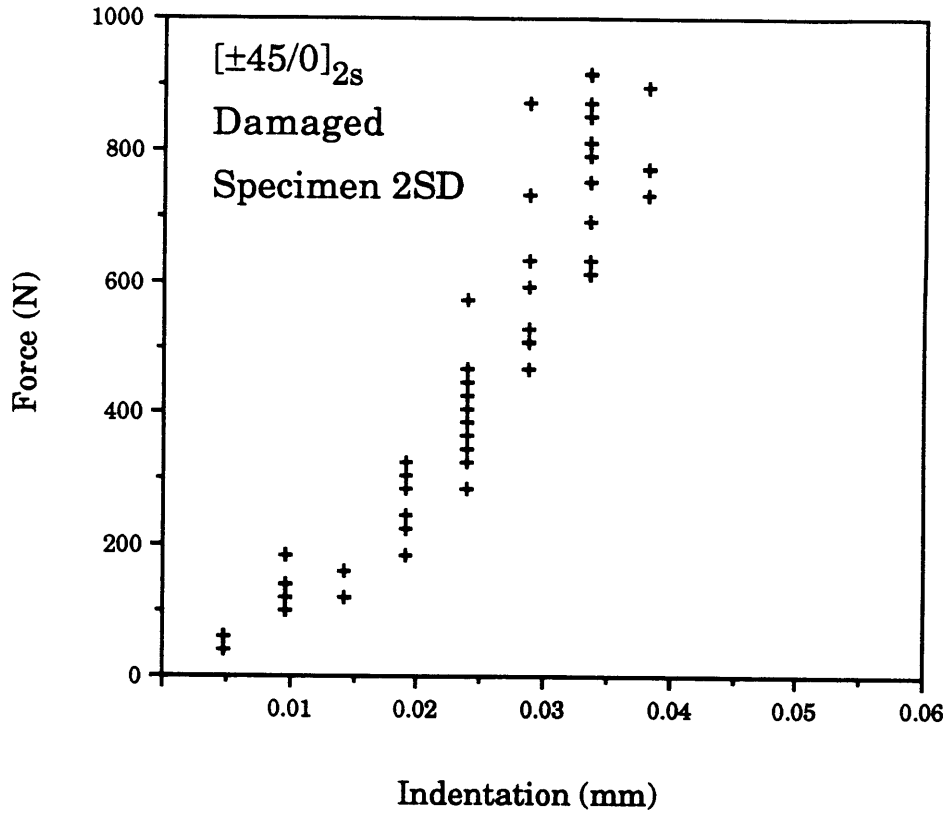


FIGURE 5.12 Plot of load versus indentation for $[\pm 45/0]_{2s}$ damaged specimen 2SD.

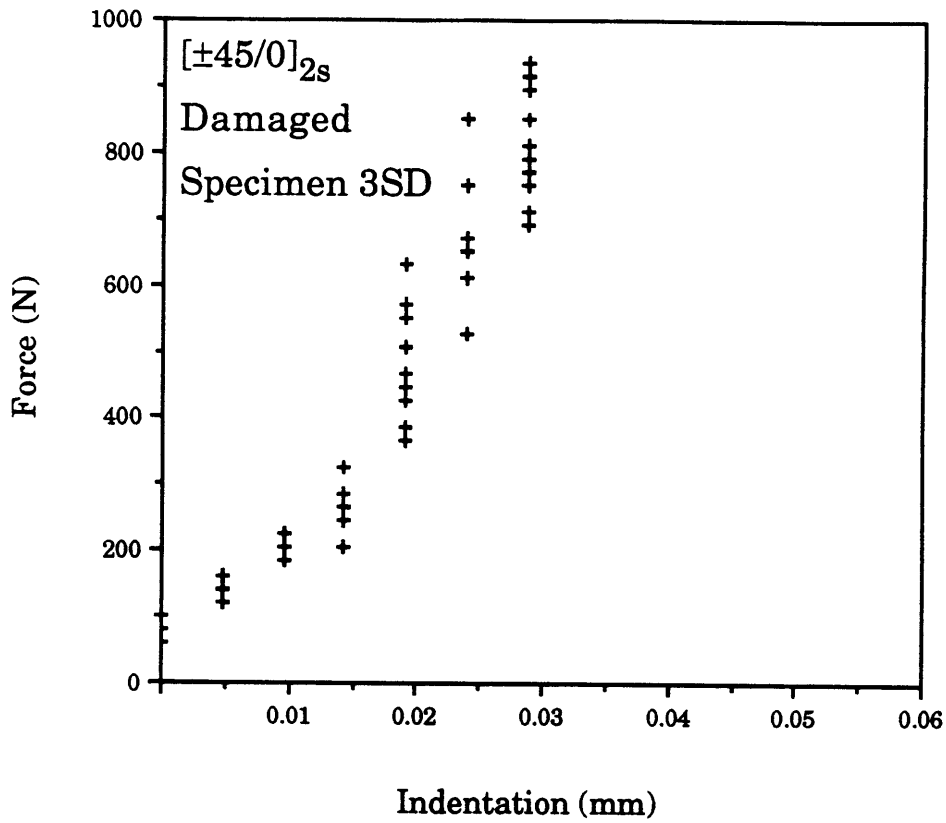


FIGURE 5.13 Plot of load versus indentation for $[\pm 45/0]_{2s}$ damaged specimen 3SD.

TABLE 5.3 Reduced Data From Experimental Static Indentation Tests of $[\pm 45/0]_{2s}$ Damaged Specimens

Test	n	log(K)	K (kN/mm^n)	R^2
1	1.54	4.84	69.5	0.886
2	1.40	4.89	78.3	0.914
3	1.06	4.49	31.0	0.881

mens. Similarly the values of n for the damaged specimens (except for test 3) are larger than the values of n for the undamaged specimens. The curve fits for the damaged specimens are not as good as the undamaged specimens. The force versus indentation plots for the damaged specimens have more scatter at higher loads than the data for the undamaged specimens.

5.2 Dynamic Impact Response

The experimentally-obtained dynamic impact response of the composite plates is presented in this section. Both force and center displacement histories of the plate are shown. The displacement data is obtained from the Linear Variable Differential Transformer shown in Figure 4.13. The force data is obtained from the force transducer shown in Figure 4.11. This data is modified, by using Equation (4.3), to give the force data at the tip of the tup. It is then "digitally" filtered, using the procedure described in Section 4.3.2, to eliminate the stress wave in the rod. The impactor mass is the same (1.53 kg) in all of the tests, only the impactor velocity is varied.

5.2.1 Undamaged Laminates

The force histories of the $[0/90]_{5s}$ laminates are shown in Figures 5.14 through 5.16. They all have the same basic shape of a half sine wave with a predominant second frequency superimposed. At the end of the impact event, the force should be zero. The nonzero readings present are a combination of noise and stress waves not completely eliminated. The peak load during the tests are 641, 588, and 641 N, respectively. The lower peak load of the second test is due to a slightly lower impact velocity (1.4 m/s versus 1.5 m/s). A $[0/90]_{5s}$

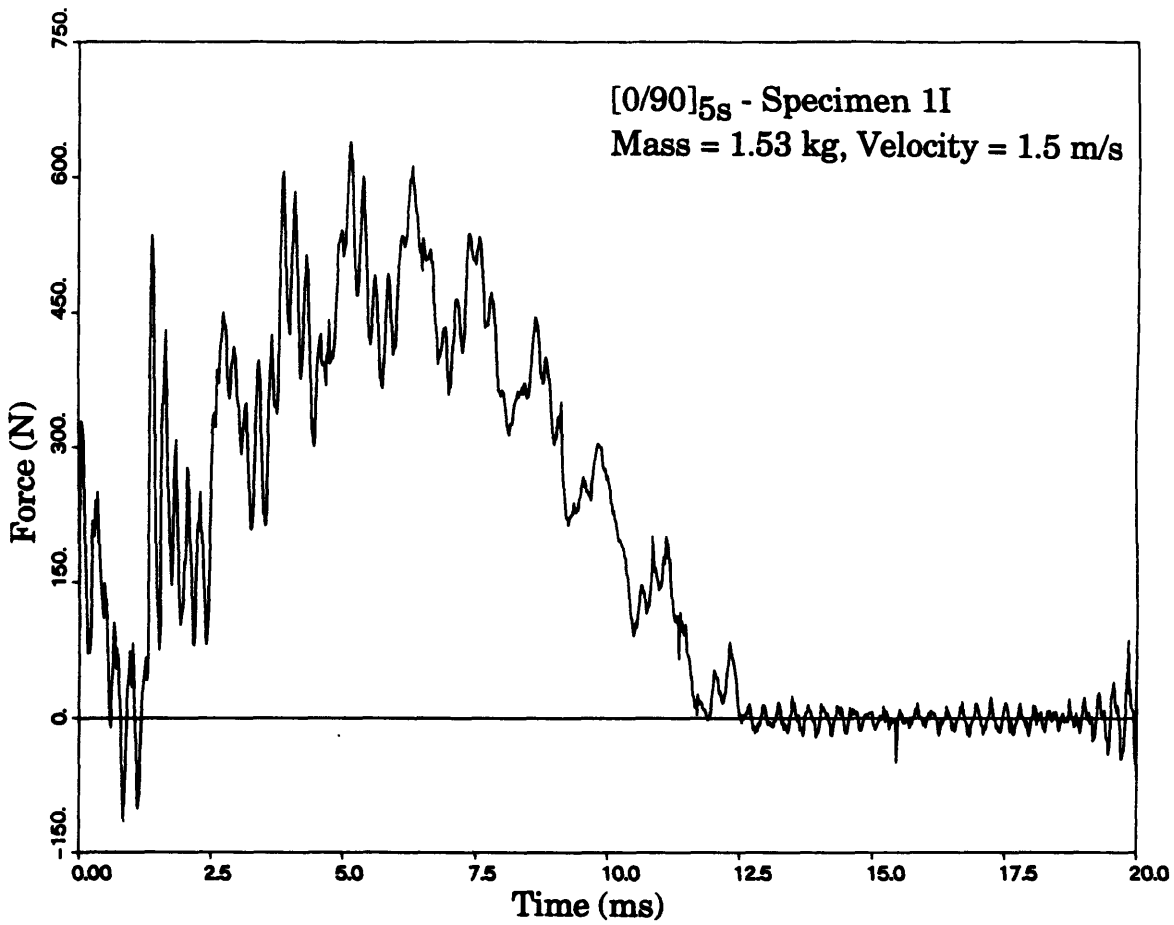


FIGURE 5.14 Filtered force history of $[0/90]_{5s}$ specimen 1I impacted at 1.5 m/s.

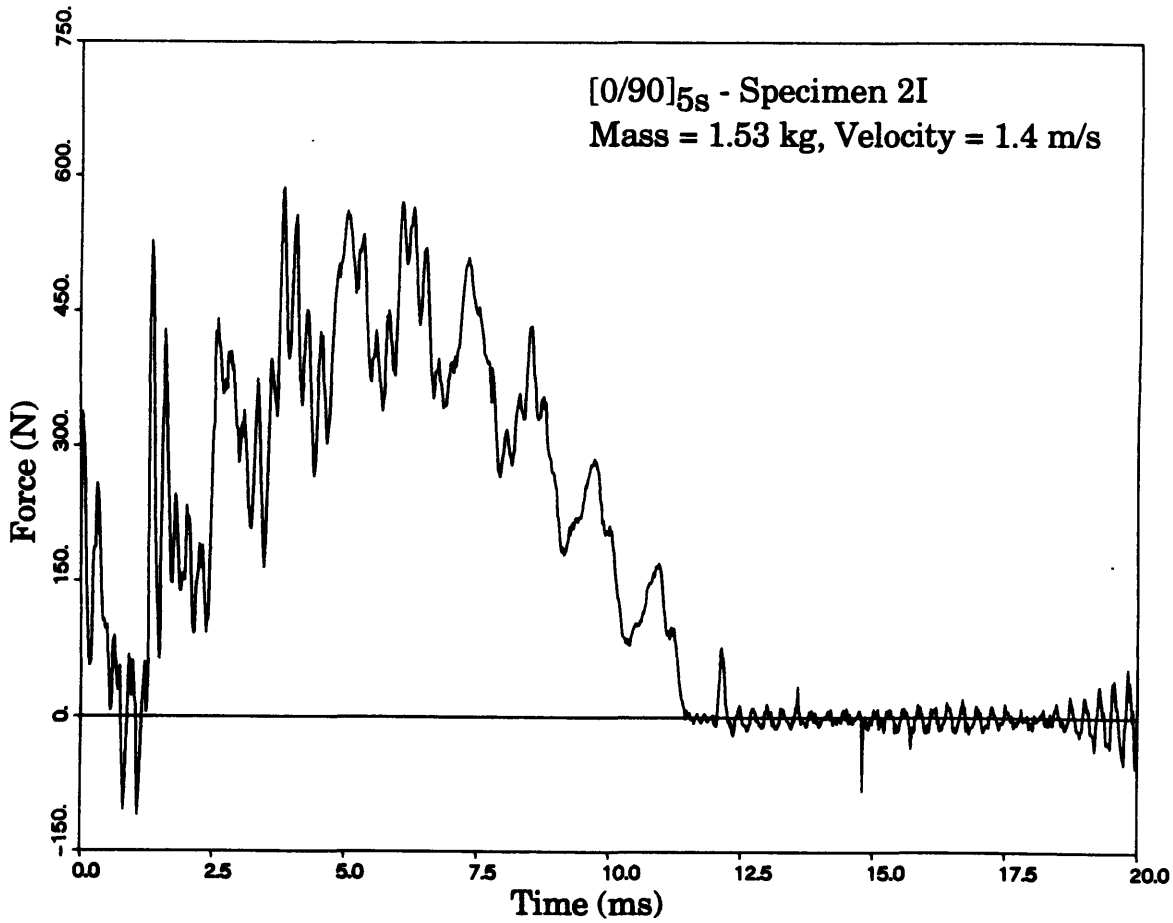


FIGURE 5.15 Filtered force history of $[0/90]_{5s}$ specimen 2I impacted at 1.4 m/s.

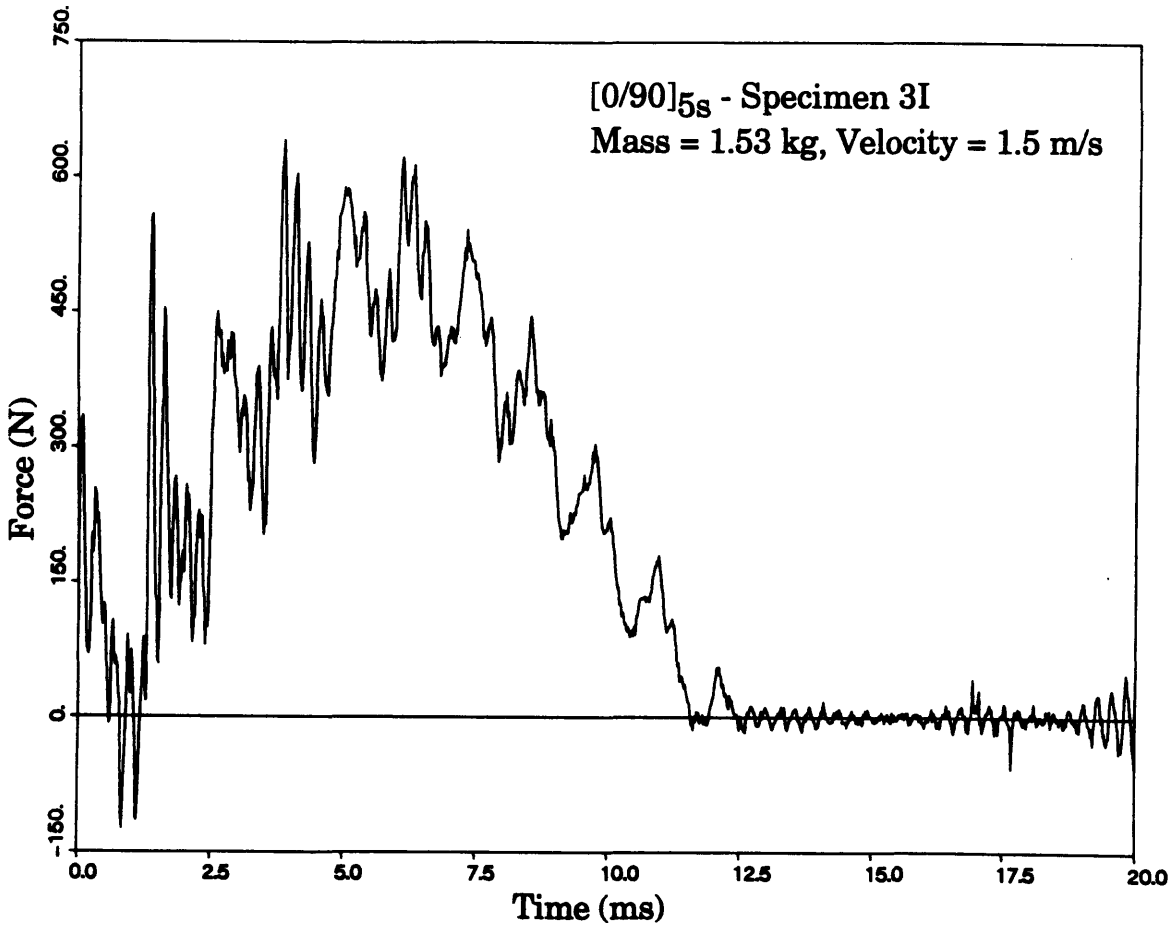


FIGURE 5.16 Filtered force history of $[0/90]_{5s}$ specimen 3I impacted at 1.5 m/s.

specimen without an extension rod was also impacted. This was done to determine the effect of the extension rod on the force history. This force history is shown in Figure 5.17. The impact velocity (1.6 m/s) is slightly higher than the other tests (1.4 - 1.5 m/s). However, the force history is very similar to the ones which had the extension rod attached. The force histories of the ones with the extension rod attached have a large load spike at the beginning of the event. They also have more force oscillations during the first part of the event. However, the peak load (638 N) and the length of the impact event are about the same as data from the specimens without the extension rod. Thus, the effect of the extension rod is small on the overall impact event. The displacement data obtained from the specimens with the extension rod is considered to be representative of a corresponding specimen without an extension rod.

The displacement histories are shown in Figures 5.18 through 5.20. All of them have a half sine wave shape during the impact event. After the impact event is over, the plate continues to vibrate. This is why the displacement history falls below zero after the impact event. The results are very similar for all the tests. The peak displacements are 5.7, 5.0, and 5.5 mm for tests 1 through 3. The corresponding lengths of the impact events are 12.6, 12.2, and 12.2 ms. The peak displacement of the second test is slightly lower than the others because of the lower impact velocity.

The force histories of the $[\pm 45/0]_{2s}$ undamaged laminates are shown in Figures 5.21 through 5.23. This data has been "filtered" using the procedure described in Section 4.3.2. The impact velocities these specimens experienced were the same as the ones the $[0/90]_{5s}$ specimens experienced (1.4 - 1.5 m/s). The force histories are similar to the force histories of the $[0/90]_{5s}$ laminates. They have the half sine wave shape with a predominant second frequency superimposed on it. The peak forces for the $[\pm 45/0]_{2s}$ laminates are 478, 462,

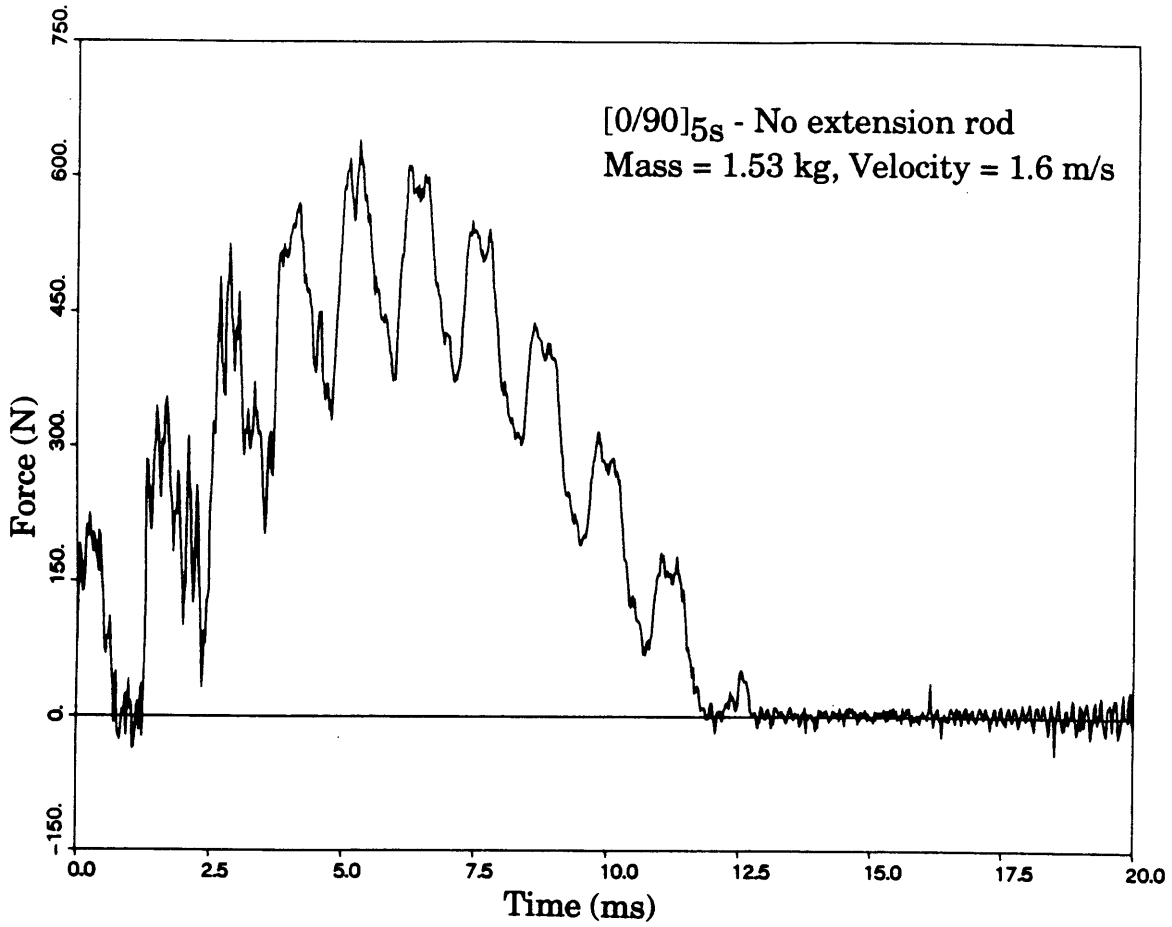


FIGURE 5.17 Filtered force history of $[0/90]_{5s}$ specimen without the extension rod impacted at 1.6 m/s.

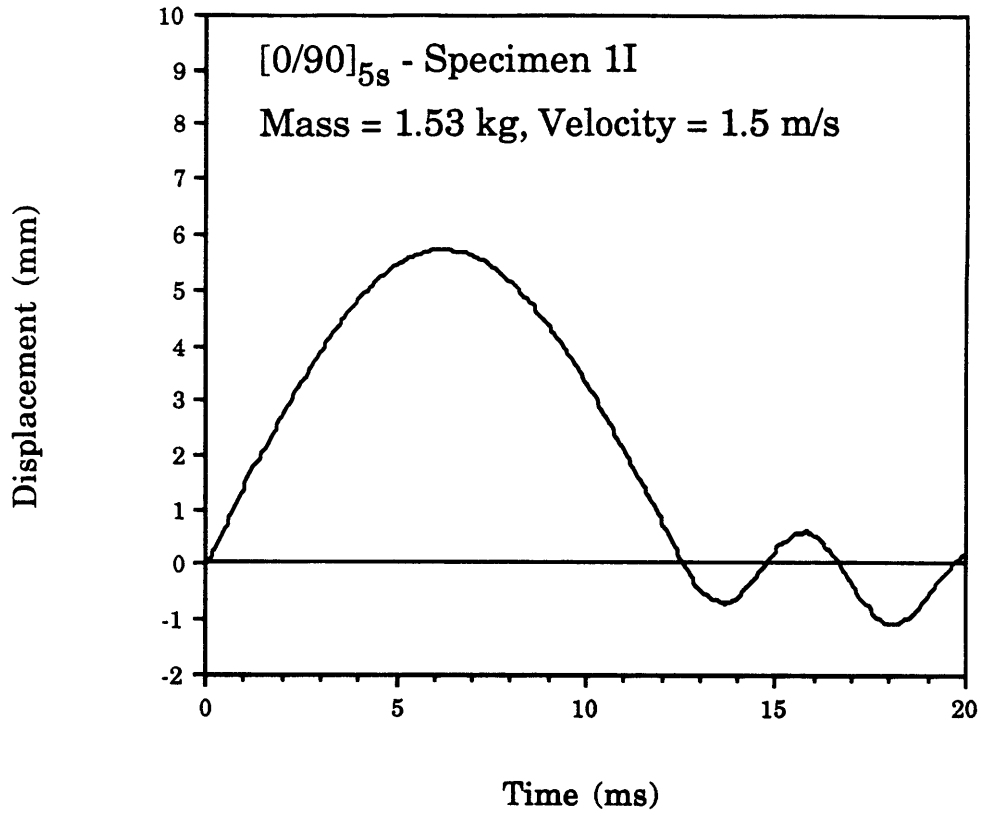


FIGURE 5.18 Center displacement history of $[0/90]_{5s}$ specimen 1I impacted at 1.5 m/s.

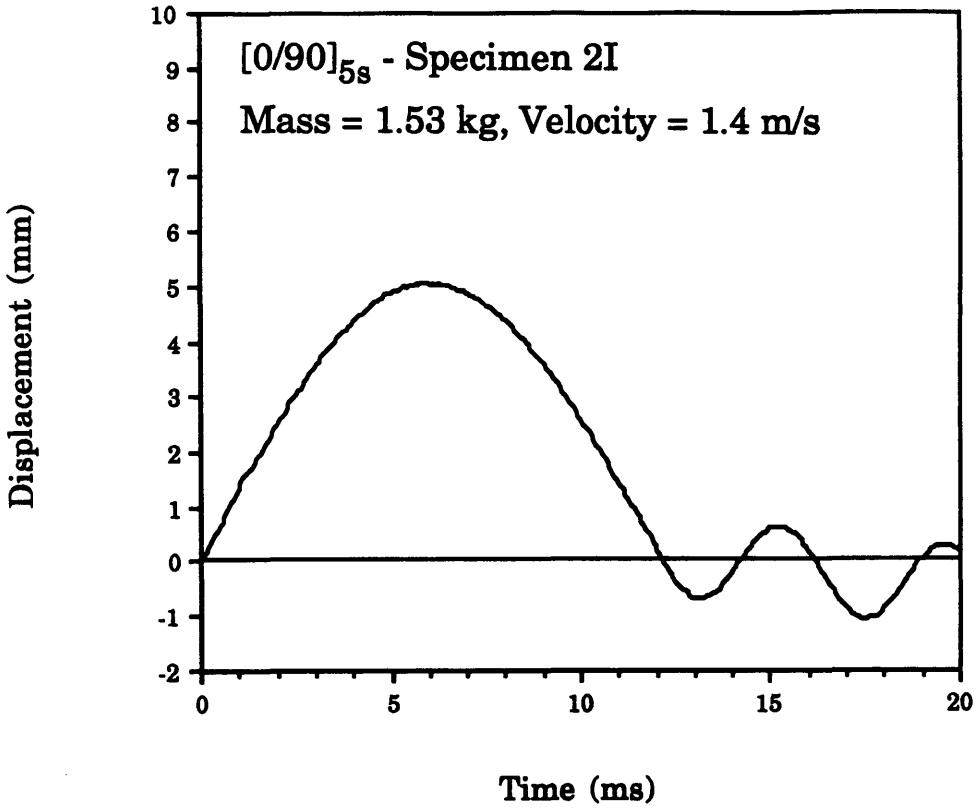


FIGURE 5.19 Center displacement history of $[0/90]_{5s}$ specimen 2I impacted at 1.4 m/s.

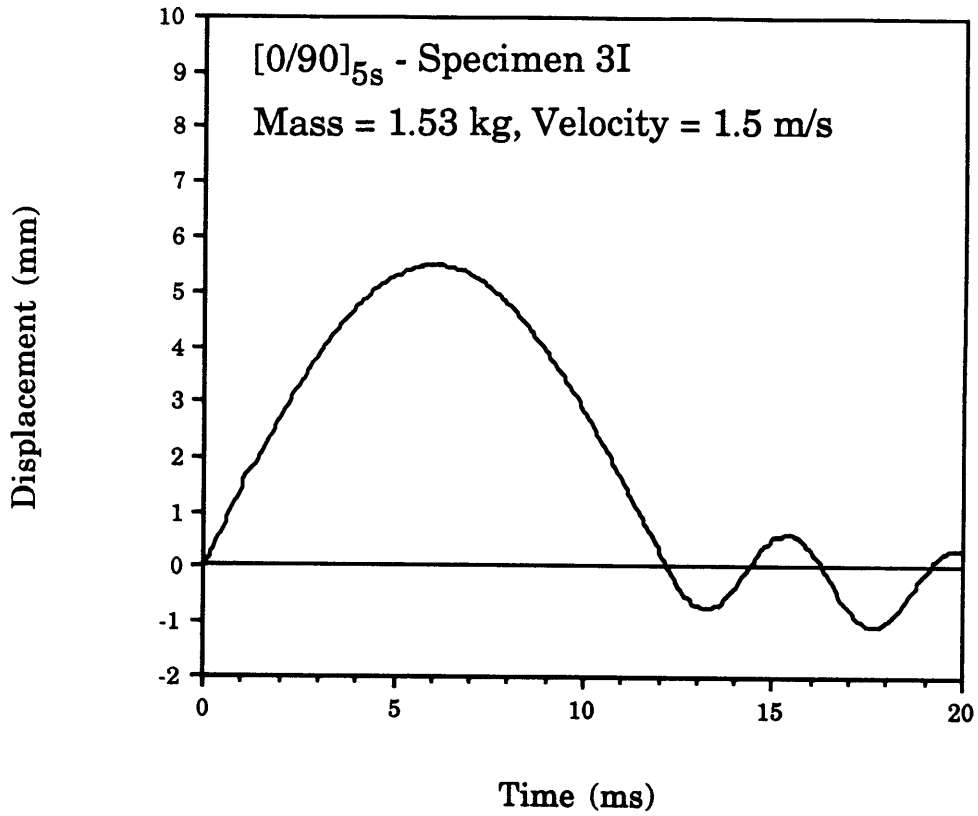


FIGURE 5.20 Center displacement history of $[0/90]_{5s}$ specimen 3I impacted at 1.5 m/s.

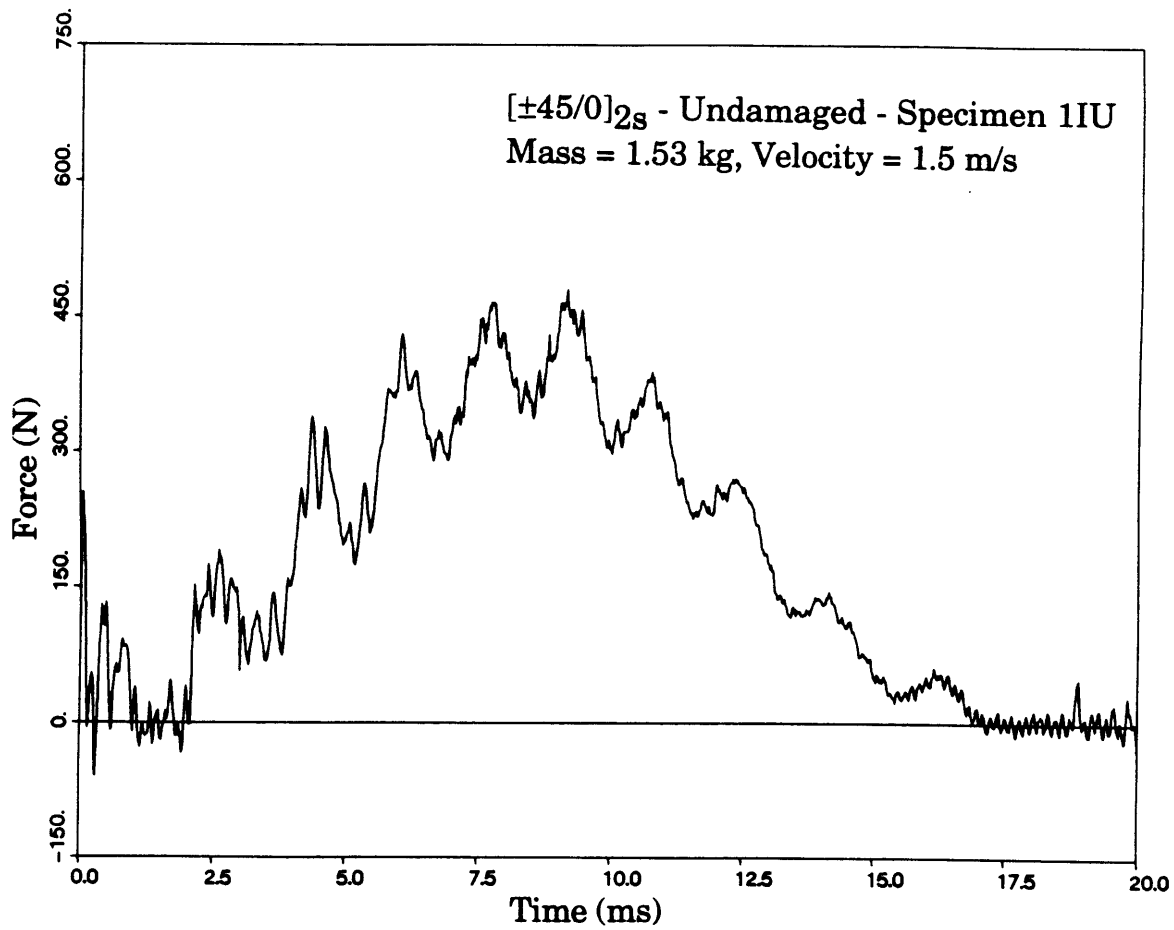


FIGURE 5.21 Filtered force history of $[\pm 45/0]_{2s}$ undamaged specimen 1IU impacted at 1.5 m/s.

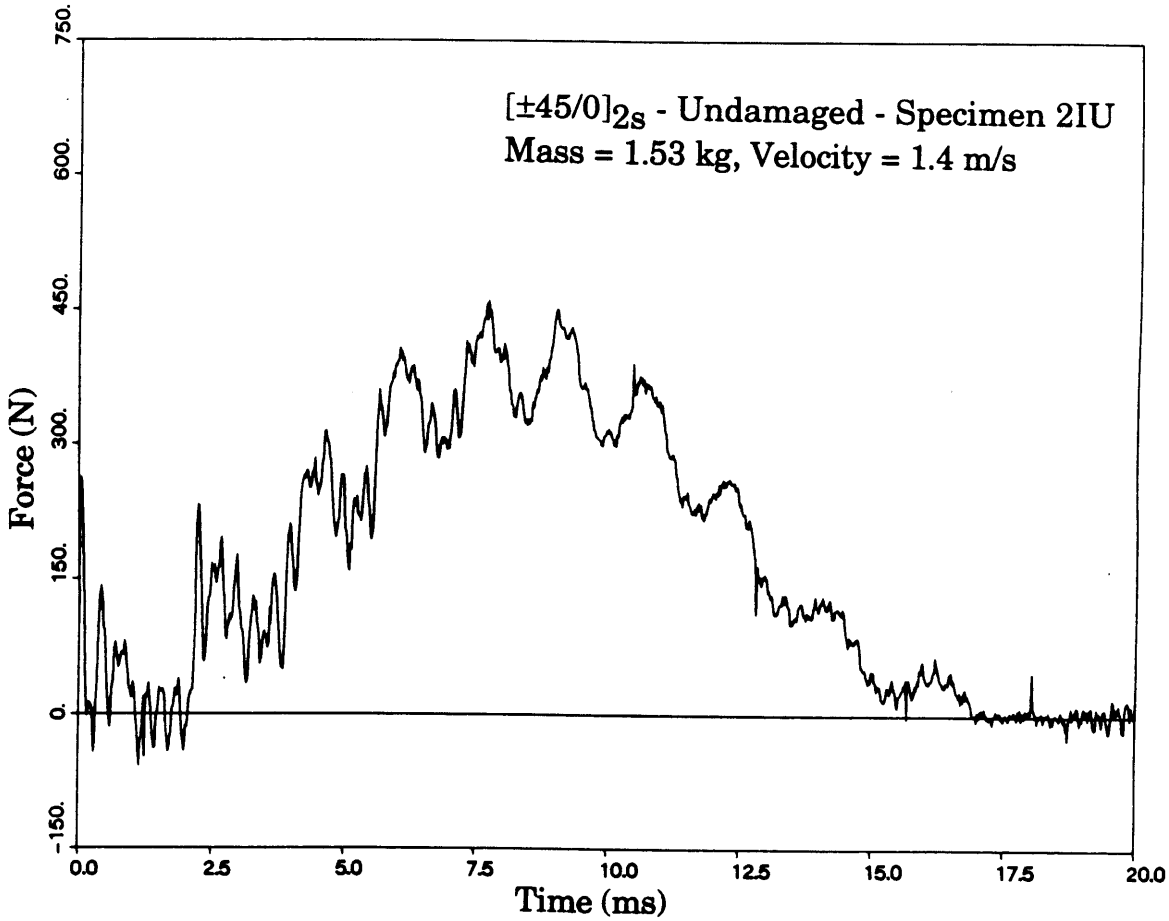


FIGURE 5.22 Filtered force history of $[\pm 45/0]_{2s}$ undamaged specimen 2IU impacted at 1.4 m/s.

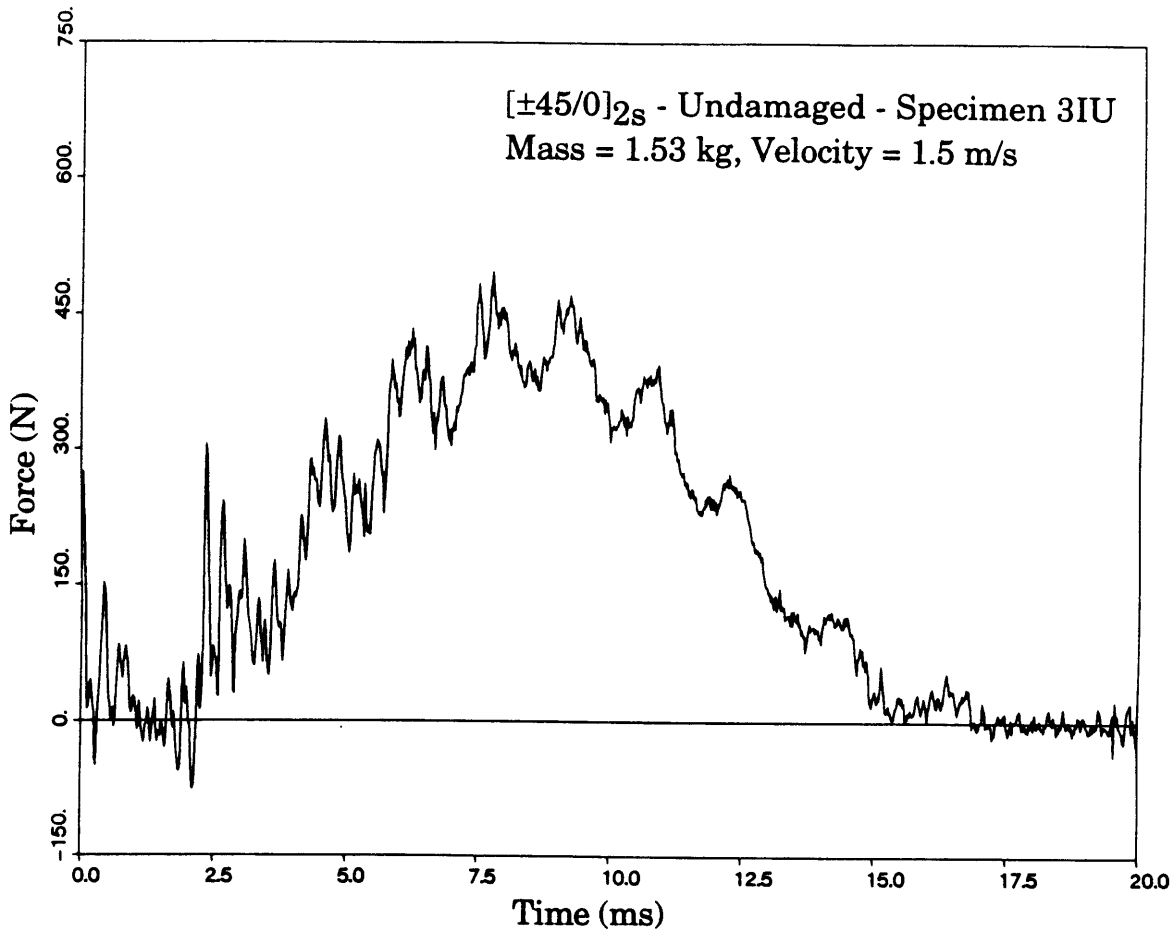


FIGURE 5.23 Filtered force history of $[\pm 45/0]_{2s}$ undamaged specimen 3IU impacted at 1.5 m/s.

and 497 N, respectively. These are smaller than the $[0/90]_{5s}$ laminates by approximately 23%. The force history of a $[\pm45/0]_{2s}$ laminate without an extension rod is shown in Figure 5.24. It is similar to the corresponding force histories with an extension rod. The peak force is 443 N. This is approximately 8% lower than the peak forces of the $[\pm45/0]_{2s}$ specimens with the extension rod, even though the impact velocity is the same (1.5 m/s). As for the $[0/90]_{5s}$ laminates, the force histories of the $[\pm45/0]_{2s}$ specimens with an extension rod have a large load spike and more force oscillations at the beginning of the event.

The displacement histories of the $[\pm45/0]_{2s}$ specimens with an extension rod attached are shown in Figures 5.25 through 5.27. They have a half sinusoidal shape to them. The peak displacement of these specimens are 8.5, 7.9, and 8.6 mm, respectively. These displacements are approximately 59% larger than the peak displacements of the $[0/90]_{5s}$ laminates. The length of the impact events are 18.6, 18.6, and 18.9 ms, respectively. These times are approximately 52% longer than the times of the $[0/90]_{5s}$ laminates. Note the small kink in the displacement history of specimen 3IU at 2.25 ms. This is the same time at which the force becomes zero after the first initial impact.

5.2.2 Damaged Laminates

Three $[\pm45/0]_{2s}$ laminates were preimpacted with a velocity of 4.0 m/s and then impacted with a velocity of 1.5 m/s. An extension rod was mounted to the back of each specimen for the high velocity impact tests. The bond between the adhesive film and the specimen failed during these impacts for each test. The force histories were obtained, and are shown in Figures 5.28 through 5.30. The peak force during the high velocity impact events is approximately 1520 N, about 3.2 times larger than the peak force of 479 N during the low velocity impacts. The duration of the high velocity impacts is 11 ms, 42% less than the

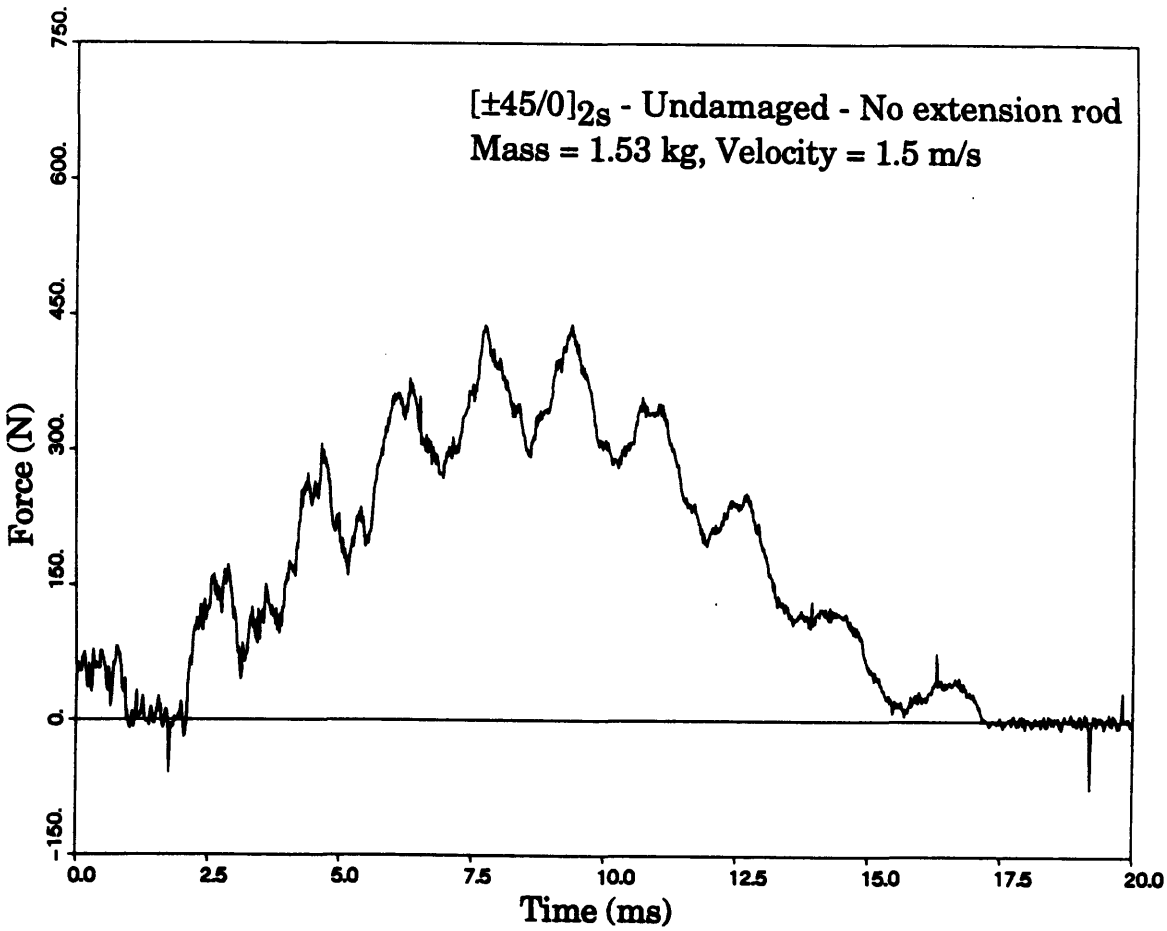


FIGURE 5.24 Filtered force history of $[\pm 45/0]_{2s}$ undamaged specimen without the extension rod impacted at 1.5 m/s.

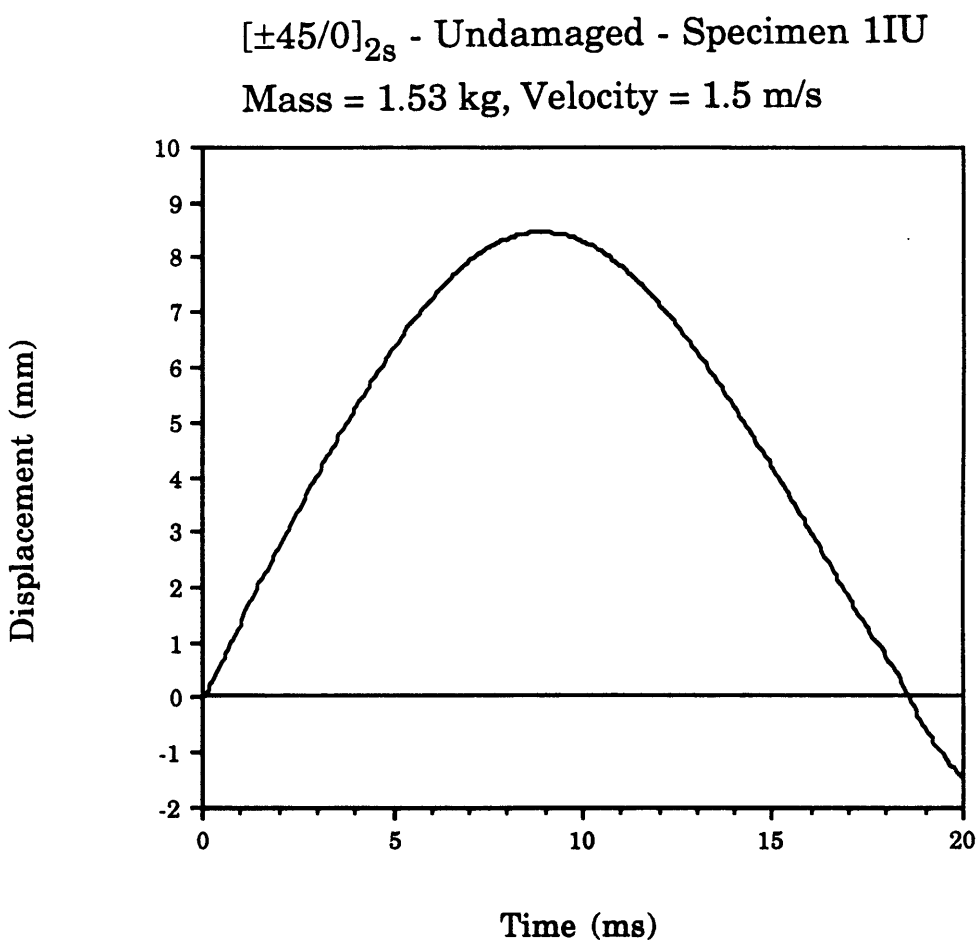


FIGURE 5.25 Center displacement history of $[\pm 45/0]_{2s}$ undamaged specimen 1IU impacted at 1.5 m/s.

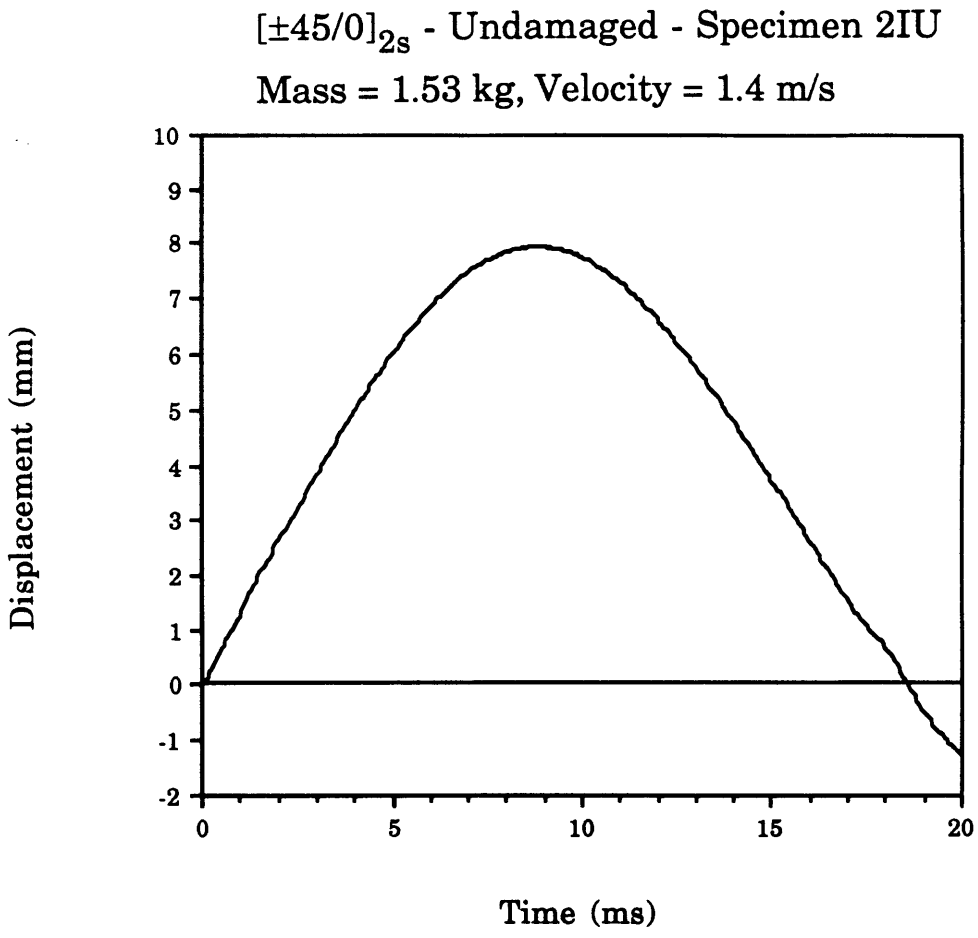


FIGURE 5.26 Center displacement history of $[\pm 45/0]_{2s}$ undamaged specimen 2IU impacted at 1.4 m/s.

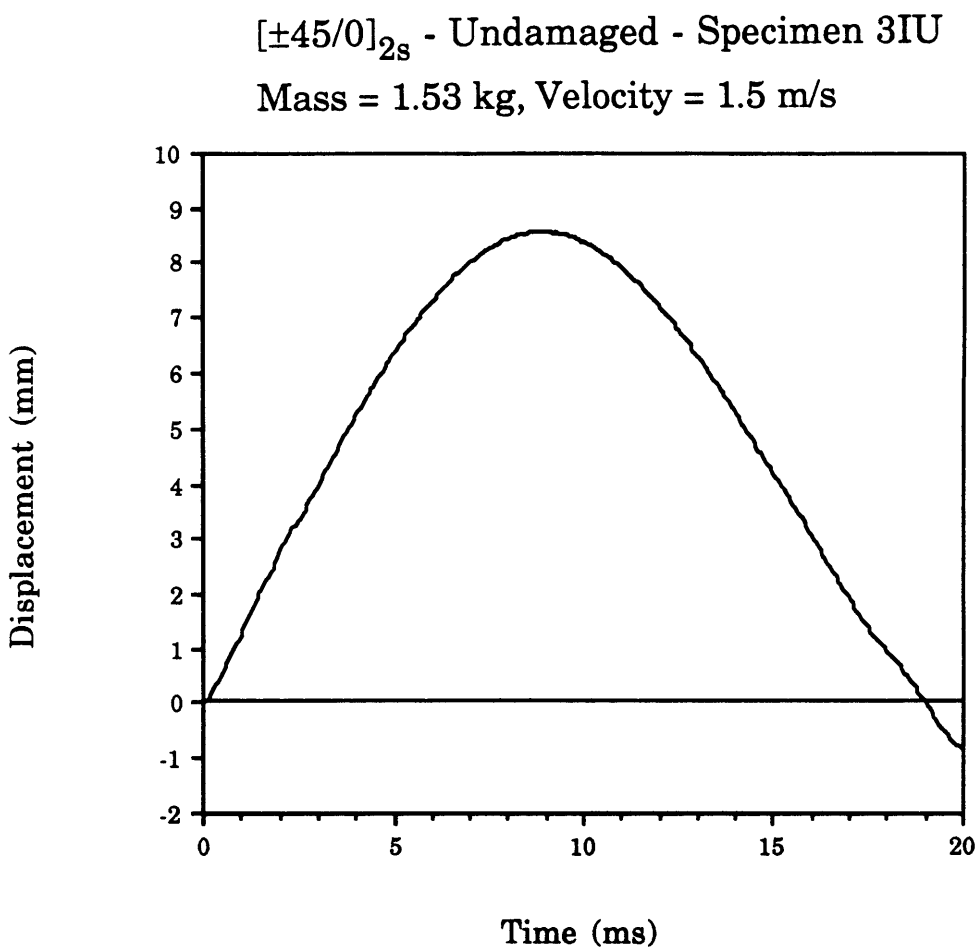


FIGURE 5.27 Center displacement history of $[\pm 45/0]_{2s}$ undamaged specimen 3IU impacted at 1.5 m/s.

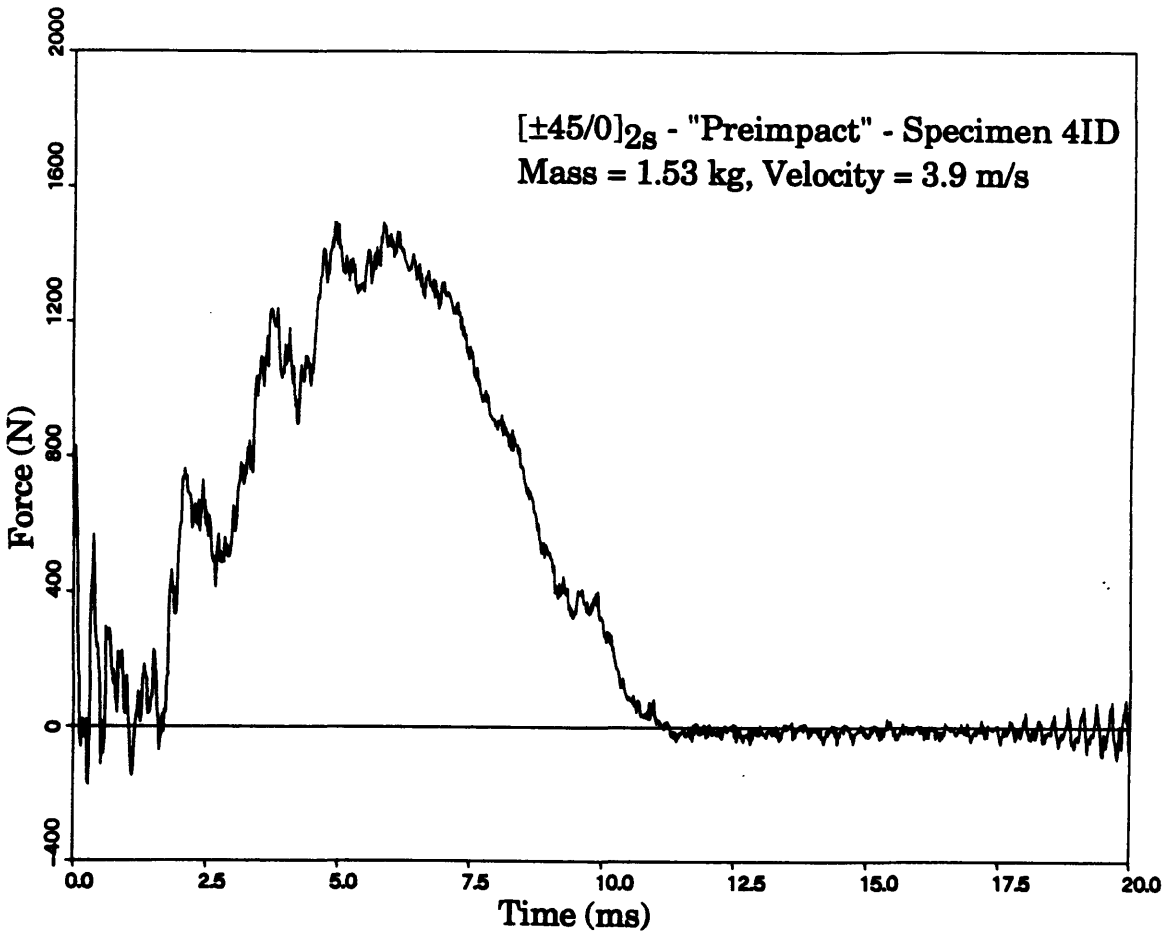


FIGURE 5.28 Filtered force history of $[\pm 45/0]_{2s}$ specimen 4ID impacted at 3.9 m/s.

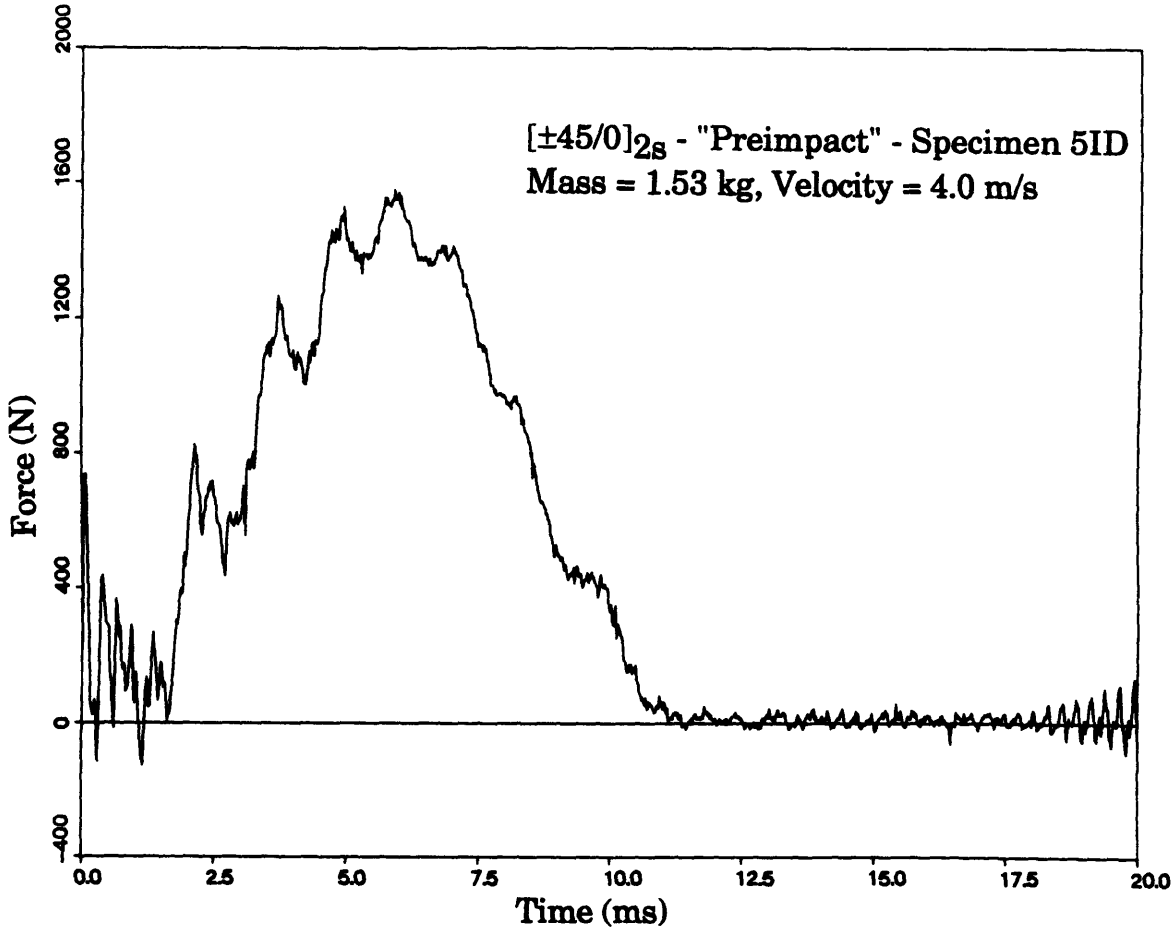


FIGURE 5.29 Filtered force history of $[\pm 45/0]_{2s}$ specimen 5ID impacted at 4.0 m/s.

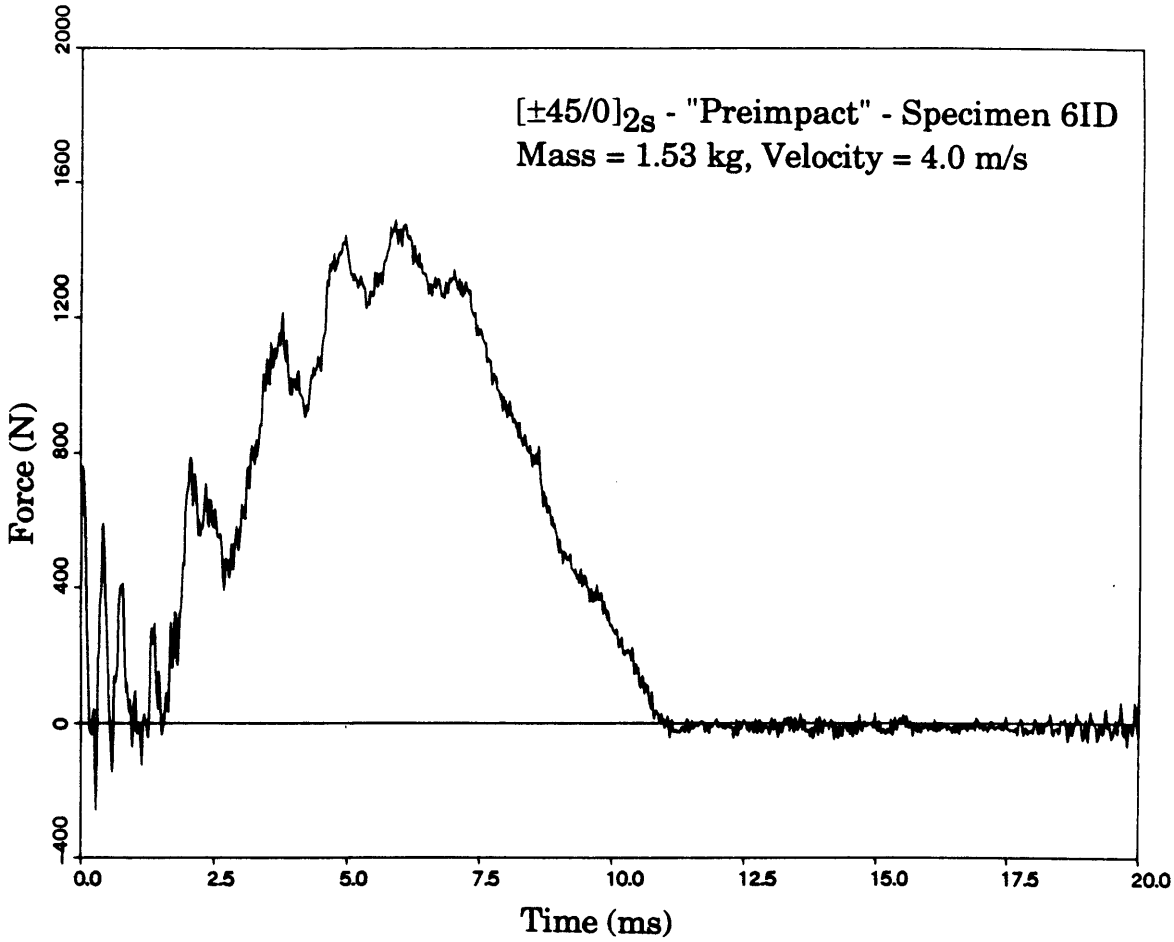


FIGURE 5.30 Filtered force history of $[\pm 45/0]_{2s}$ specimen 6ID impacted at 4.0 m/s.

duration of 19 ms for the low velocity tests. The shape of the force history is similar to the low velocity impacts during the first half of the impact event. The secondary frequency that is predominant during the first part of the test is almost completely gone in the second half. A large load spike is present right at the beginning of the event, as there was for all the force histories of laminates with an extension rod attached.

These specimens were then prepared for impacts with a velocity of 1.5 m/s. Machined nuts were rebonded to the back of the specimens so that the displacement history could be obtained. The force histories are shown in Figures 5.31 through 5.33. They are very similar to the force histories of the undamaged laminates. The peak force during these tests is about 463 N, which is 3% less than the peak force of 479 N during the impacts to the undamaged laminates. The displacement histories are shown in Figures 5.34 through 5.36. They are practically the same as the displacement histories of the undamaged laminates. The lengths of the impact events (19.1, 18.6, and 18.4 ms) and the peak displacements (8.0, 8.5, and 8.3 mm) for the damaged laminates are virtually the same as for the undamaged laminates.

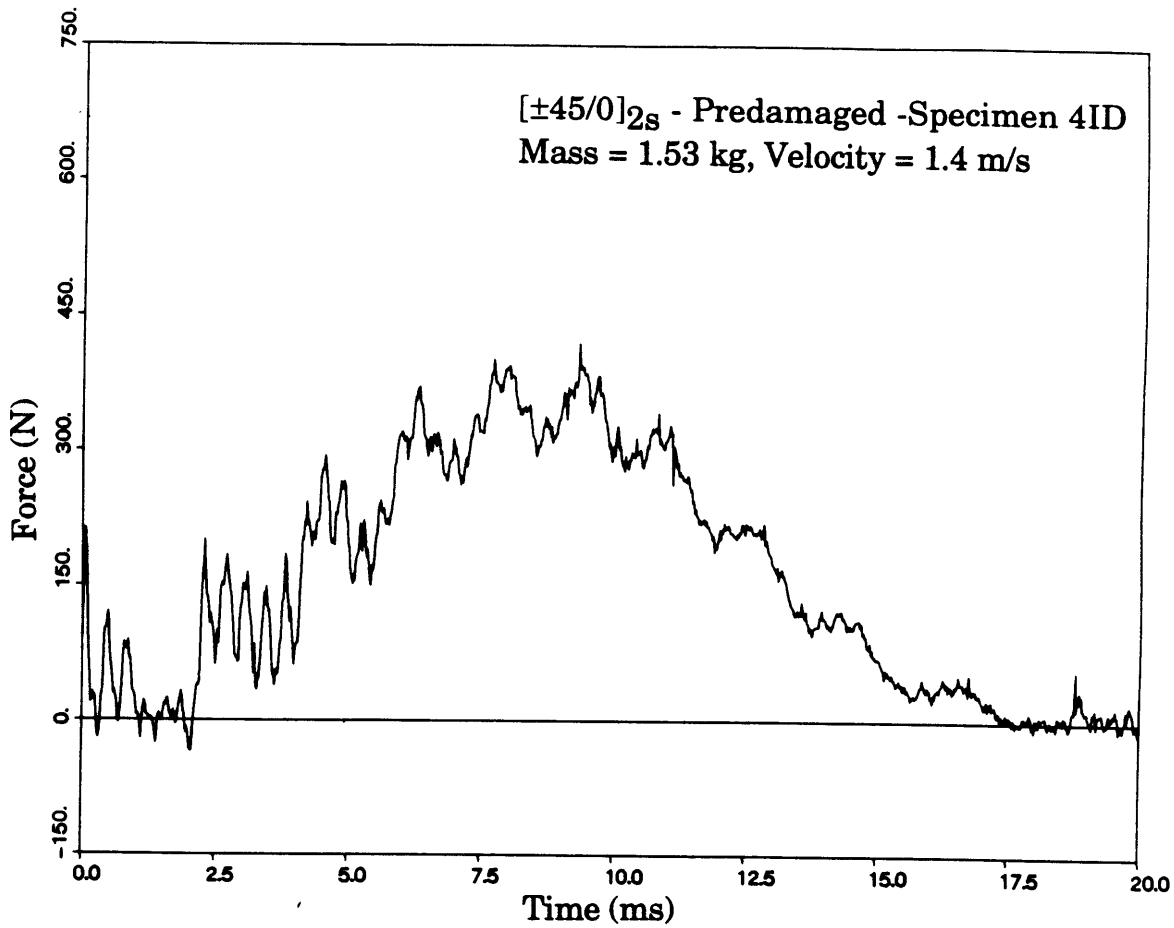


FIGURE 5.31 Filtered force history of $[\pm 45/0]_{2s}$ predamaged specimen 4ID impacted at 1.4 m/s.

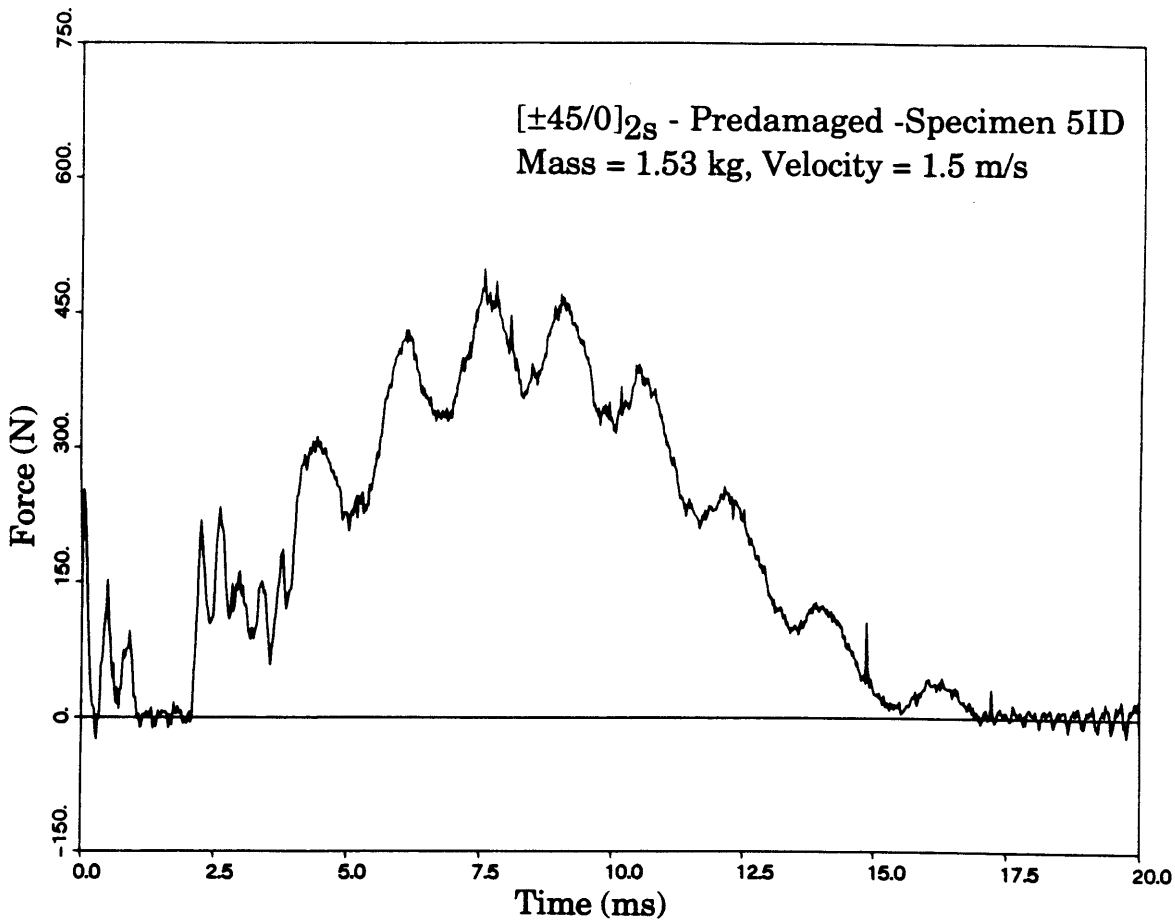


FIGURE 5.32 Filtered force history of $[\pm 45/0]_{2s}$ predamaged specimen 5ID impacted at 1.5 m/s.

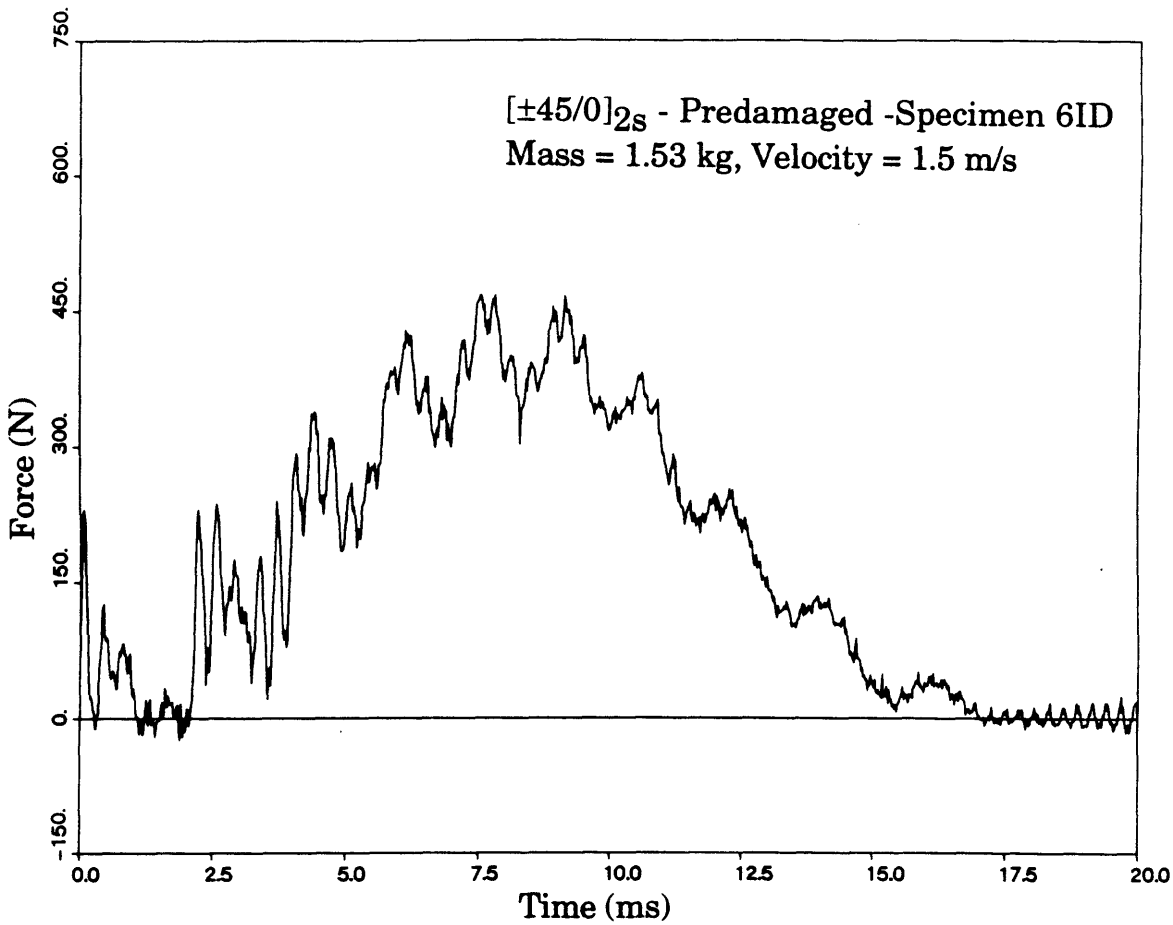


FIGURE 5.33 Filtered force history of $[\pm 45/0]_{2s}$ predamaged specimen 6ID impacted at 1.5 m/s.

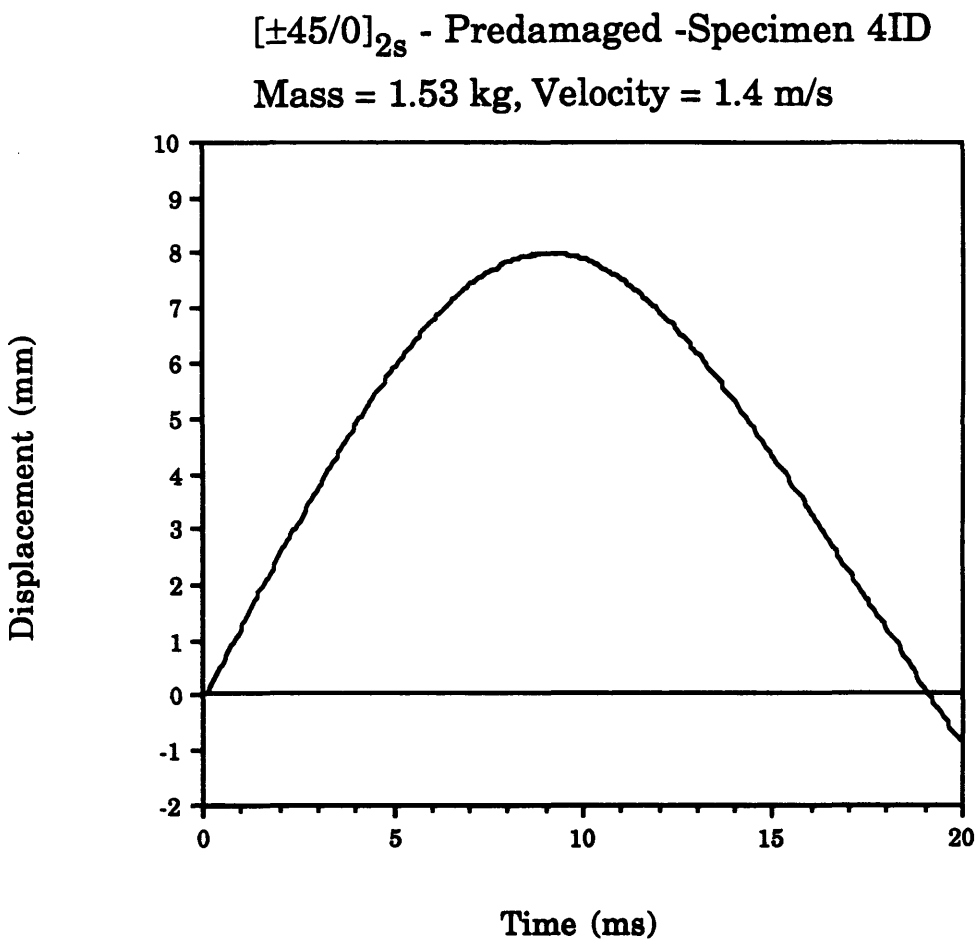


FIGURE 5.34 Center displacement history of $[\pm 45/0]_{2s}$ predamaged specimen 4ID impacted at 1.4 m/s.

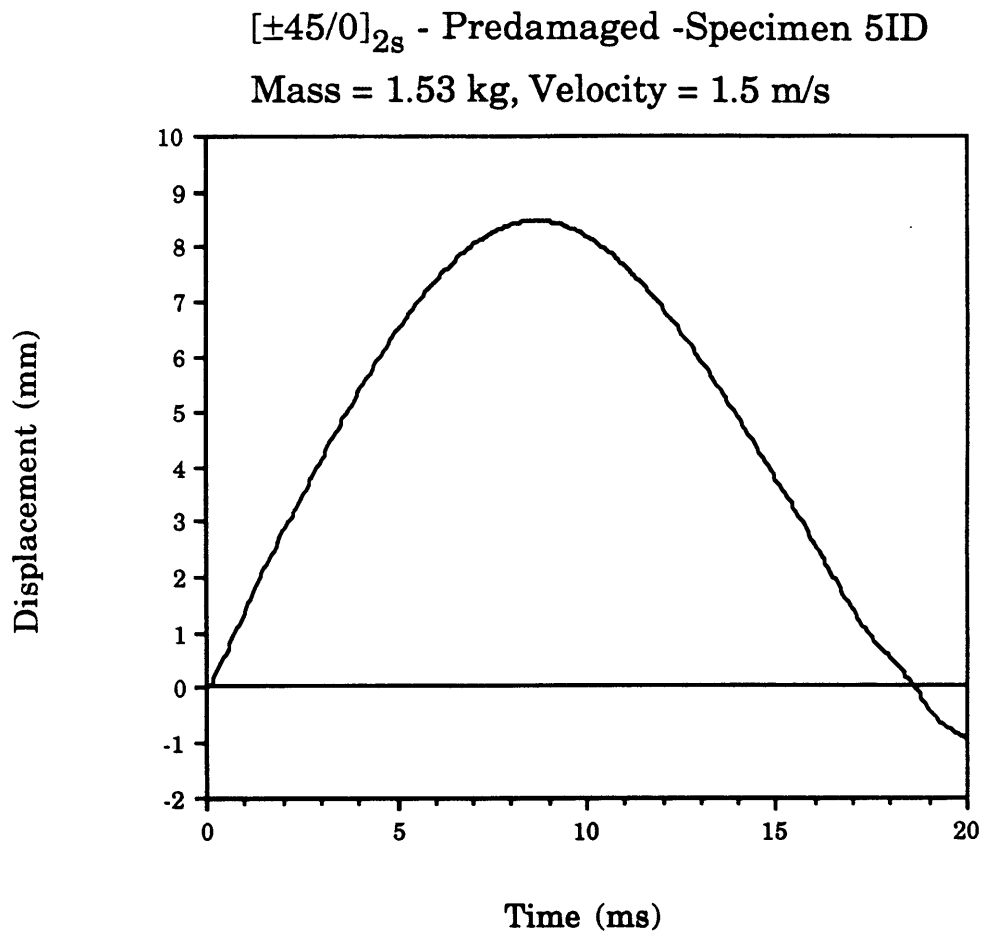


FIGURE 5.35 Center displacement history of $[\pm 45/0]_{2s}$ predamaged specimen 5ID impacted at 1.5 m/s.

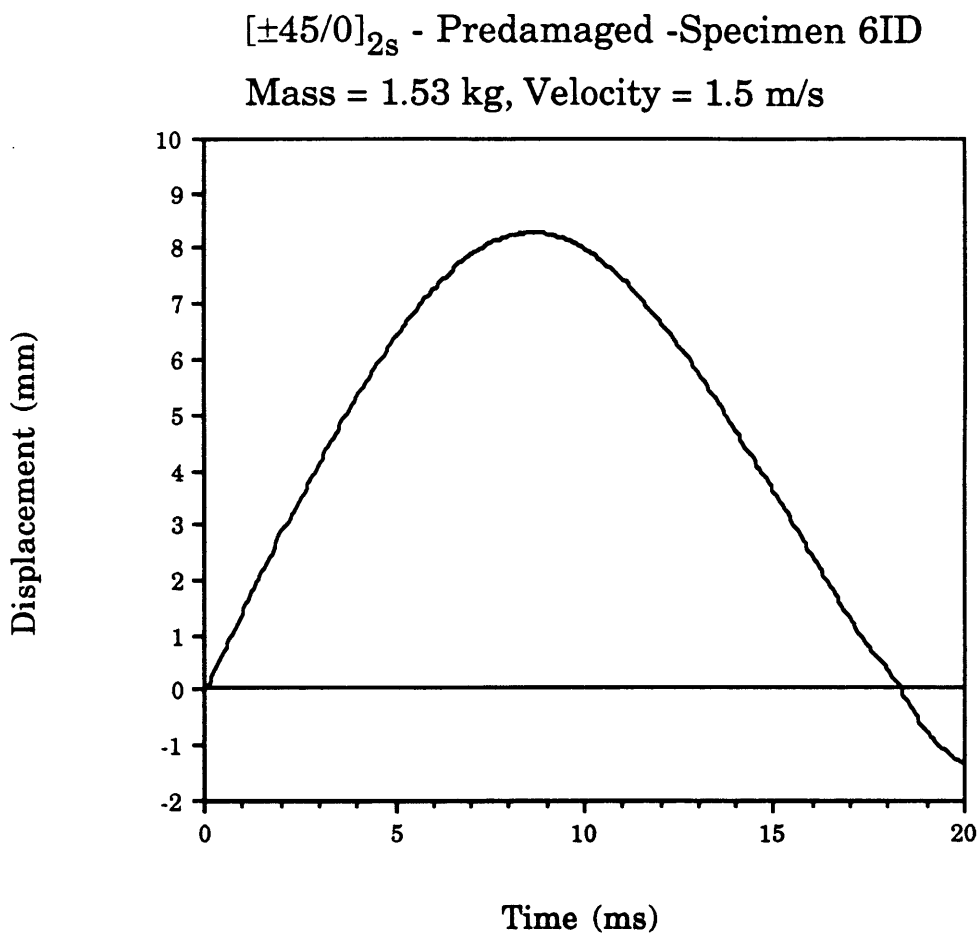


FIGURE 5.36 Center displacement history of $[\pm 45/0]_{2s}$ predamaged specimen 6ID impacted at 1.5 m/s.

CHAPTER 6

Analytical Comparison and Discussion

6.1 Impactor/Plate Contact Relation

The analysis was run for the AS4/3501-6 $[0/90]_{5s}$ laminates with an average experimental thickness of 2.600 mm. All of the important inputs are summarized in Table 6.1. The radius of the region analyzed is 52.0 mm, twenty times the plate thickness. This radius is chosen since the classical plate solution is recovered outside of the boundary [1]. The indentor radius is 6.35 mm. A total of 160 harmonics are used in the analysis. The constitutive properties for the laminate are found from program ROTATE as discussed in Section 3.5. The ply properties of AS4/3501-6 graphite/epoxy are listed in Table 3.1. The equivalent engineering properties in the x- and y-directions are found to be the same. Thus, the major axis and minor axis analyses will give the same results. The analysis was run once for indentations of 0.0133, 0.0266, 0.0399, 0.0532, and 0.0664 mm to give five evenly spaced data points in the range of the experimental data. The convergence of the analysis was verified by plotting the load versus indentation for 40, 80, 120 and 160 harmonics. This is illustrated in Figure 6.1. The points on the graph are connected with straight lines. The change in the curve from 40 harmonics to 80 harmonics is significant. The difference between the 80, 120, and 160 harmonic curves is small. Thus, a satisfactory converged solution is obtained with 160 harmonics. The curve for 160 harmonics is shown in Figure 6.2 along with the experimental results from the $[0/90]_{5s}$ test specimens. For presentation purposes, the aver-

TABLE 6.1 Inputs for Local Analysis of $[0/90]_{5s}$ AS4/3501-6 Laminates

Loading Type:	Hertzian
Plate Thickness:	2.600 mm
Radius of the Region Analyzed:	52.0 mm
Impactor Radius:	6.35 mm
Number of Harmonics:	160
Number of Indentations:	5
Increment in Indentation:	0.013288 mm
E_θ:	76.265 GPa
E_z:	10.801 GPa
$\nu_{r\theta}$:	0.038772
ν_{rz}:	0.361854
G_{rz}:	4.6305 GPa

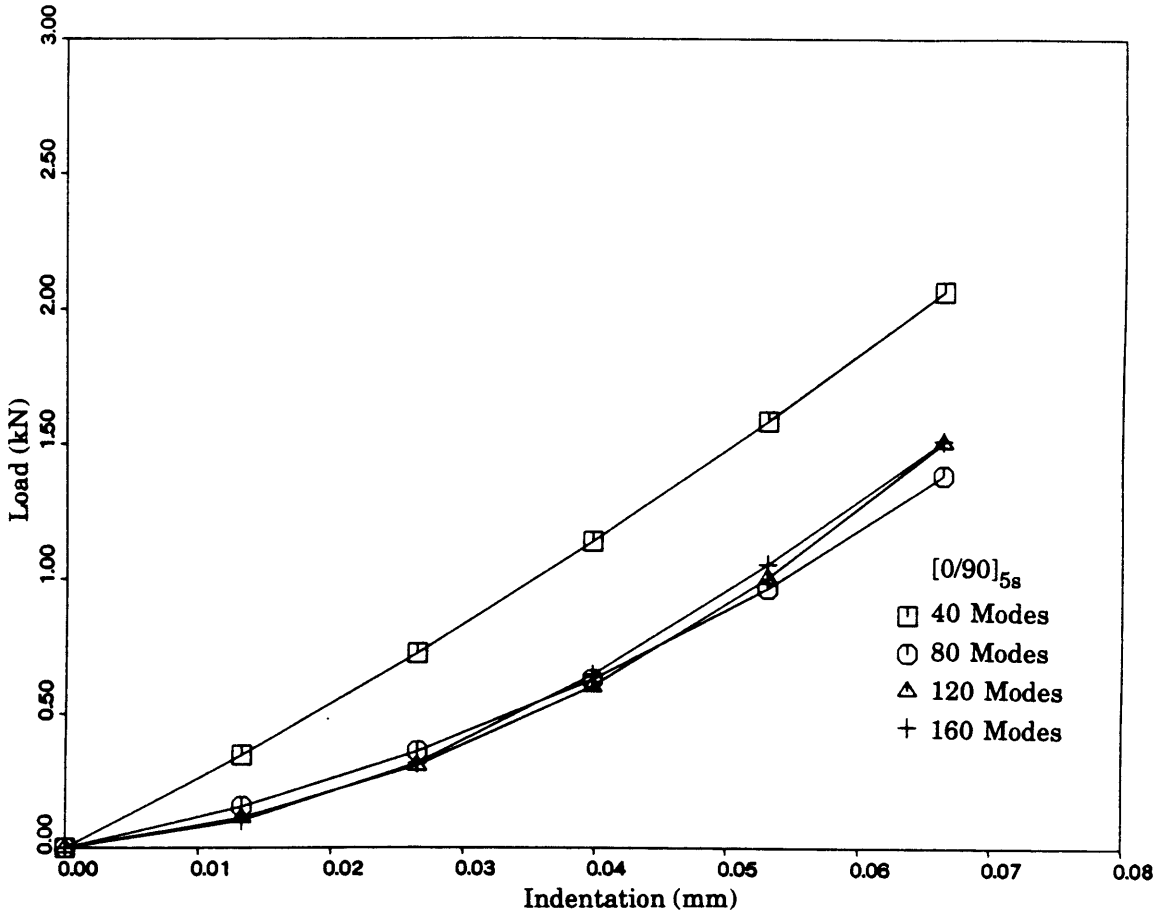


FIGURE 6.1 Convergence of the load versus indentation curves obtained from the analysis of a $[0/90]_{5s}$ laminate.

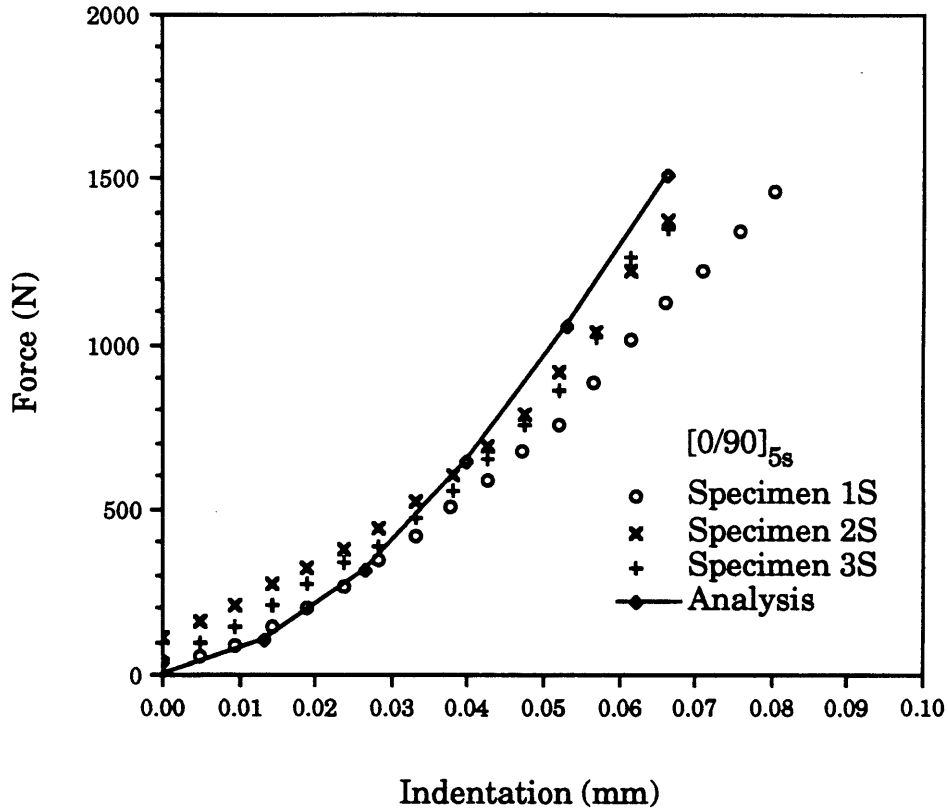


FIGURE 6.2 Analytical results and experimental data for static indentation of $[0/90]_{5s}$ laminates.

age experimental value of the load for each indentation is calculated for each specimen and plotted.

The results of the analysis match the experimental results relatively well. The experimental data does not go directly through the origin because of zeroing difficulties from noise in the force transducer and the Linear Variable Differential Transformer. Fitting the data from the analysis to the relation

$$R = K\alpha^n \tag{2.1}$$

gives K equal to 144 kN/mm^n with an exponent of 1.68 and an R^2 value of 1.00. The corresponding average values from the experimental tests are K equal to 21.7 kN/mm^n with an exponent of 1.09 and an average R^2 value of 0.95. The reduced data from the experiment is almost linear, whereas the data from the analysis is more nonlinear. As the amount of indentation increases, the load obtained from the analysis becomes larger than the experimentally-obtained load.

The inputs used for the analysis for the $[\pm 45/0]_{2s}$ laminates are summarized in Table 6.2. An average experimental thickness of 1.572 mm was used. The radius of the region analyzed is again twenty times the thickness (31.0 mm). The indentor radius is 6.35 mm and the number of harmonics used is 160. A converged solution was verified in the same manner as for the analysis of the $[0/90]_{5s}$ laminate. Program ROTATE was used to obtain the equivalent engineering properties of the $[\pm 45/0]_{2s}$ laminate. The properties in the x-direction are used for the major axis analysis and the properties in the y-direction are used for the minor axis analysis. Each analysis was run for indentations of 0.0086, 0.0172, 0.0258, 0.0344, and 0.0430 mm. The results from the analysis are shown in Figure 6.3 along with the experimental results from the $[\pm 45/0]_{2s}$

TABLE 6.2 Inputs for Local Analysis of $[\pm 45/0]_{2s}$ AS4/3501-6 Laminates

Loading Type:	Hertzian
Plate Thickness:	1.572 mm
Radius of the Region Analyzed:	31.0 mm
Impactor Radius:	6.35 mm
Number of Harmonics:	160
Number of Indentations:	5
Increment in Indentation:	0.008600 mm

MAJOR AXIS Inputs		MINOR AXIS Inputs	
E_θ:	61.799 GPa	E_θ:	26.704 GPa
E_z:	10.721 GPa	E_z:	10.721 GPa
$\nu_{r\theta}$:	0.694604	$\nu_{r\theta}$:	0.300144
ν_{rz}:	0.130820	ν_{rz}:	0.251297
G_{rz}:	5.0118 GPa	G_{rz}:	4.3031 GPa

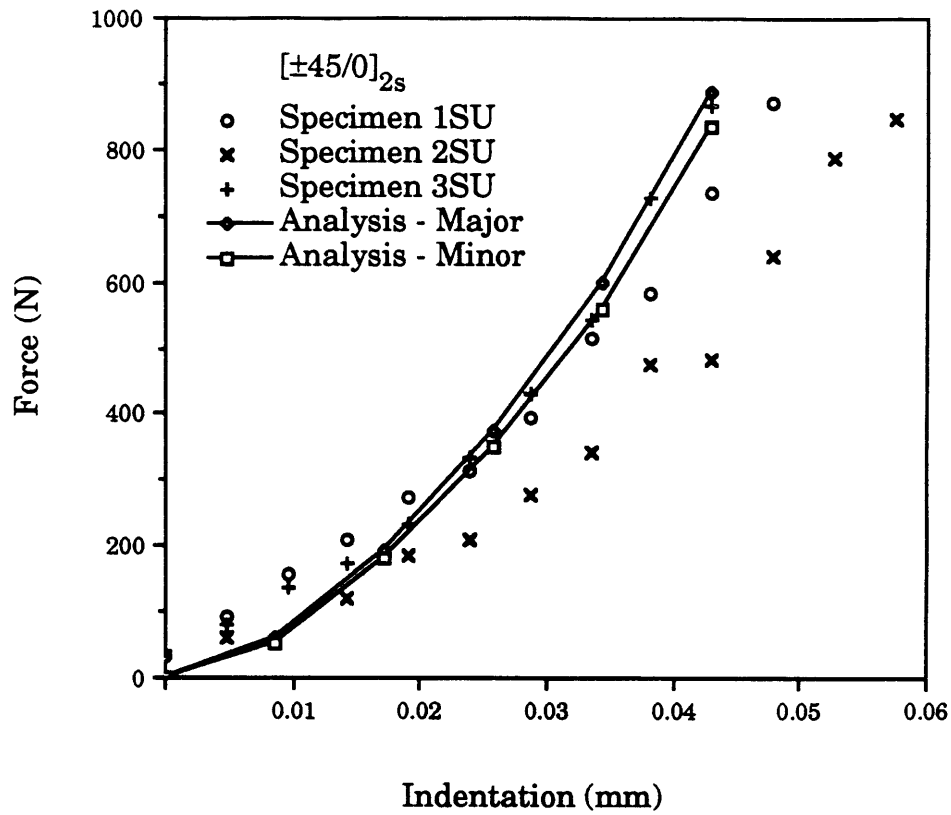


FIGURE 6.3 Analytical results and experimental data for static indentation of $[\pm 45/0]_{2s}$ laminates.

test specimens. As for the $[0/90]_{5s}$ specimens, the average value of the load for each indentation is calculated for each specimen and plotted.

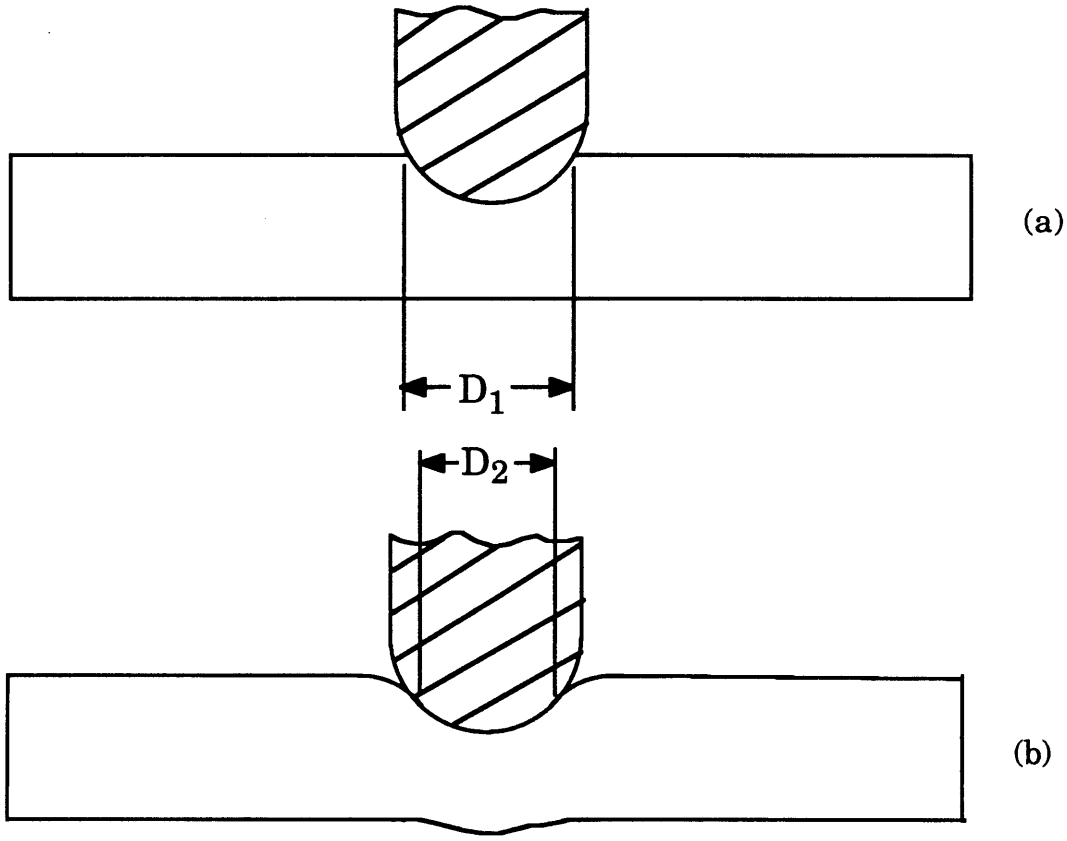
The analysis matches the experimental results from specimens 1SU and 3SU well. As for the $[0/90]_{5s}$ laminates, the experimental data does not go directly through the origin because of zeroing difficulties from noise in the force transducer and the Linear Variable Differential Transformer. The constants in Eq. (2.1) are found to be K equal to 175.1 kN/mm^n with an exponent of 1.68 for the major axis analysis, and K equal to 174.6 kN/mm^n with a value for n of 1.70 for the minor axis analysis. Both have an R^2 value of 1.00 for the curve fits. Note that these numbers are very similar even though the in-plane material properties are significantly different. This is because the contact relation is controlled by the out-of-plane properties [1]. This suggests that the assumption of transverse isotropy in the analysis is reasonable. The corresponding average values from the experimental tests are K equal to 21.8 kN/mm^n with an exponent of 1.12 and an average R^2 value of 0.93. The experimental results are close to being linear as indicated by the value of n being close to 1.0. Recall that specimen 2SU was accidentally cut narrow. One may suspect this to be the reason why its data differs from the data of the other two test specimens. However, Yang and Sun [14] conducted static indentation tests with specimens having different widths (25.4 mm and 38.1 mm) and found that the width does not appreciably affect the contact behavior. Thus, a conclusion may not be drawn here on the basis of a single test since the data is probably normal experimental variation.

As for the $[0/90]_{5s}$ specimens, the analysis predicts higher loads than experimentally measured, for large indentations. A possible explanation for this is as follows. A geometric assumption on the deformation of the plate was made in the local analysis. This assumption is illustrated in Figure 3.2. The

deformation is exaggerated, but the plate is assumed to deform around the indentor as shown. The physical deformation of the plate is shown in Figure 6.4. (The geometric assumption is also shown here for easy comparison.) The surface of the plate conforms to the shape of the indentor via a smooth transition from the free surface to the area under the impactor as shown. This reduces the area of contact between the indentor and the plate. The assumed Hertzian pressure distribution, which is illustrated in Figure 6.5, depends on the area of contact. The indentations in Figures 6.4 (a) and (b) are the same. This results in a smaller load being applied via the assumed pressure distribution to the plate with a smaller area of contact. Thus, the analysis is expected to give loads larger than they actually are since a larger area of contact is assumed than exists in reality. The difference between the analysis and the experiment increases as the physical deformation deviates more from the assumed deformation.

The experimental load versus indentation relation for a $[\pm 45/0]_{2s}$ laminate is compared with a $[0/90]_{5s}$ laminate in Figure 6.6. The curve fits are almost identical. However, the trends of the experimental data have a slight difference. The data from both laminates start off the same, but at an indentation of about 0.04 mm, the loads for the $[\pm 45/0]_{2s}$ laminate become larger than the corresponding loads for the $[0/90]_{5s}$ laminate. The important point to note is that the curves are very similar even though the two laminates have a different layup and thickness. The reason for the curves being so similar is because the contact relation is strongly dependent on the out-of-plane stiffness. The value of E_z is 10.7 GPa for the $[\pm 45/0]_{2s}$ laminate and 10.8 GPa for the $[0/90]_{5s}$ laminate. Since the modulus in the z-direction is practically the same for the two laminates, it is not surprising that the contact relation is similar. The reason for the last few data points of the $[\pm 45/0]_{2s}$ laminate having a larger

Geometric Assumption in Analysis



Physical Deformation

FIGURE 6.4 Illustration of (a) Analytical assumption of the deformation of the plate due to a contact load; and (b) Physical deformation of the plate due to a contact load.

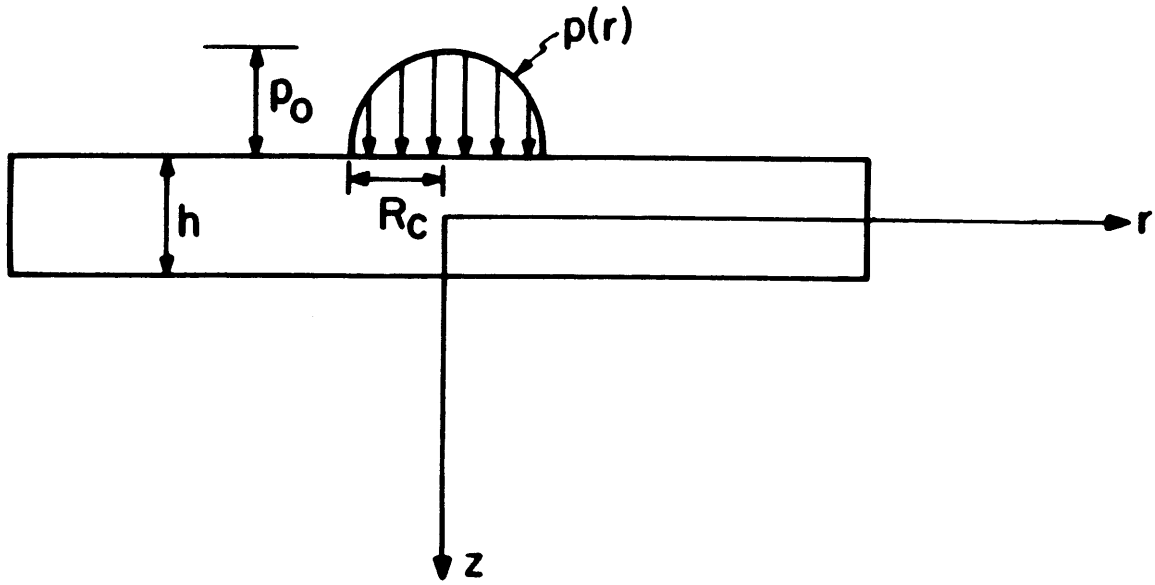


FIGURE 6.5 Illustration of Hertzian loading distribution.

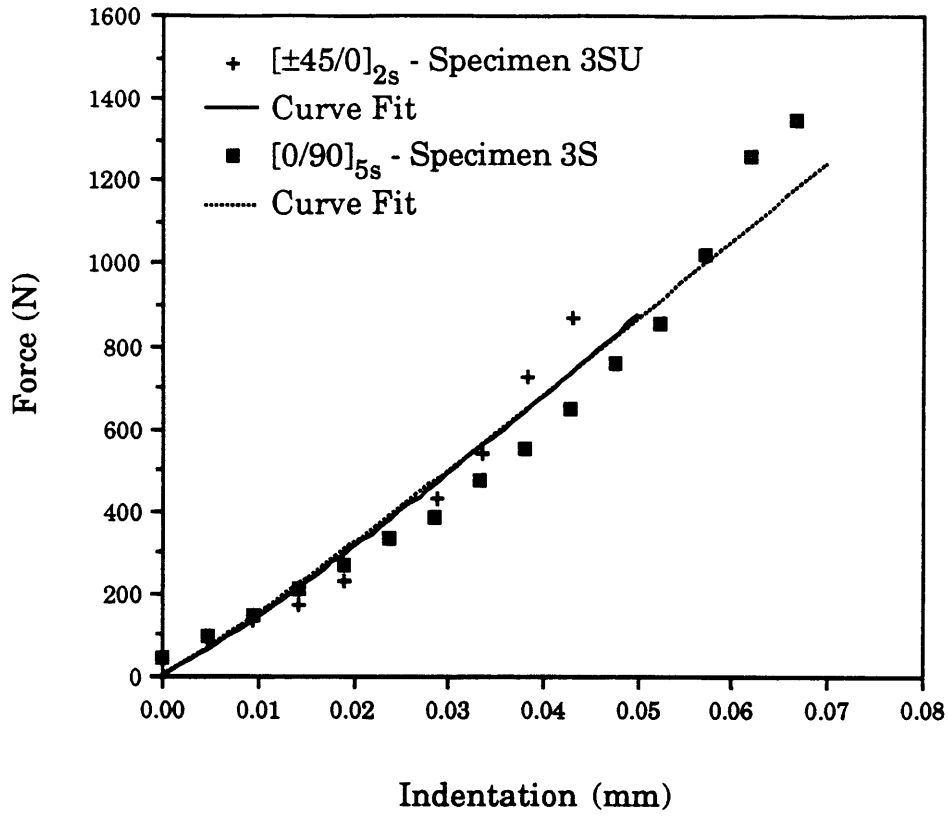


FIGURE 6.6 The experimental impactor/plate contact relations for a $[\pm 45/0]_{2s}$ laminate and a $[0/90]_{5s}$ laminate.

load than the $[0/90]_{5s}$ laminate is because it is close to the point at which first failure occurred. The laminate was probably bending excessively, causing the area of contact between the indentor and the plate to increase. This causes a larger load to be seen by the plate as discussed earlier. This means that the curvature of the plate is important in determining the contact relation. Thus, flat plates which are impacted and bend excessively see a different force history than stiffer plates which do not bend as much. This also means that cylindrical sections, as in an aircraft fuselage, will have a contact relation that differs from a flat plate.

The curve fits shown in Figure 6.6 are to the relation of Eq. (2.1). It is evident that the curve fit is not excellent. This relation was chosen because it reduces to the well-known Hertzian contact law when n is equal to 1.5, and it has been used by previous researchers [14, 15] investigating the contact relation of composite materials. Polynomials may be used to better describe the shape of the experimental data. However, this introduces more variables which makes it difficult to make comparisons between different curves from the numbers themselves. The relation of Eq. (2.1) only has two variables, yet it is inadequate in describing the contact relation. Thus, a tradeoff exits between the number of variables one wishes to deal with and the accuracy of the curve fit.

The analytical load versus indentation relation for a $[\pm 45/0]_{2s}$ laminate is compared with a $[0/90]_{5s}$ laminate in Figure 6.7. The curves are very close, indicating that the contact relation is almost the same. Again, this is because the out-of-plane properties, which dominate the contact relation, are very similar. Thus, the local analysis is able to model this phenomenon.

The load versus indentation relation for a damaged laminate is compared with an undamaged laminate in Figure 6.8. Curve fits are overlaid on

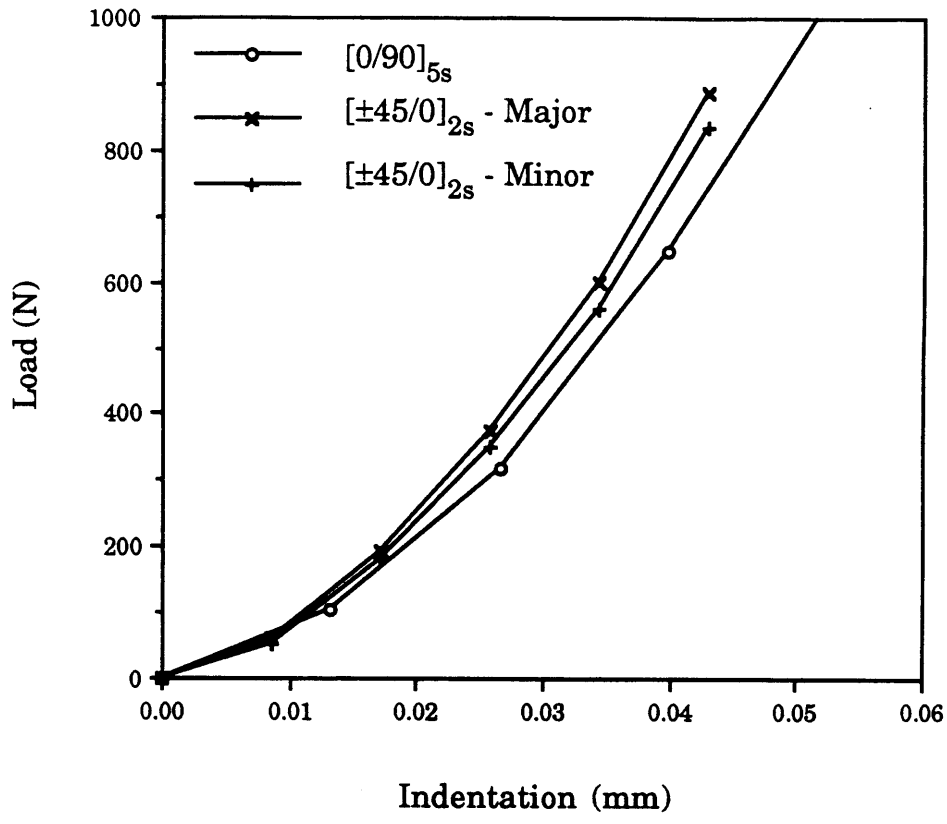


FIGURE 6.7 The analytical impactor/plate contact relations for a $[\pm 45/0]_{2s}$ laminate and a $[0/90]_{5s}$ laminate.

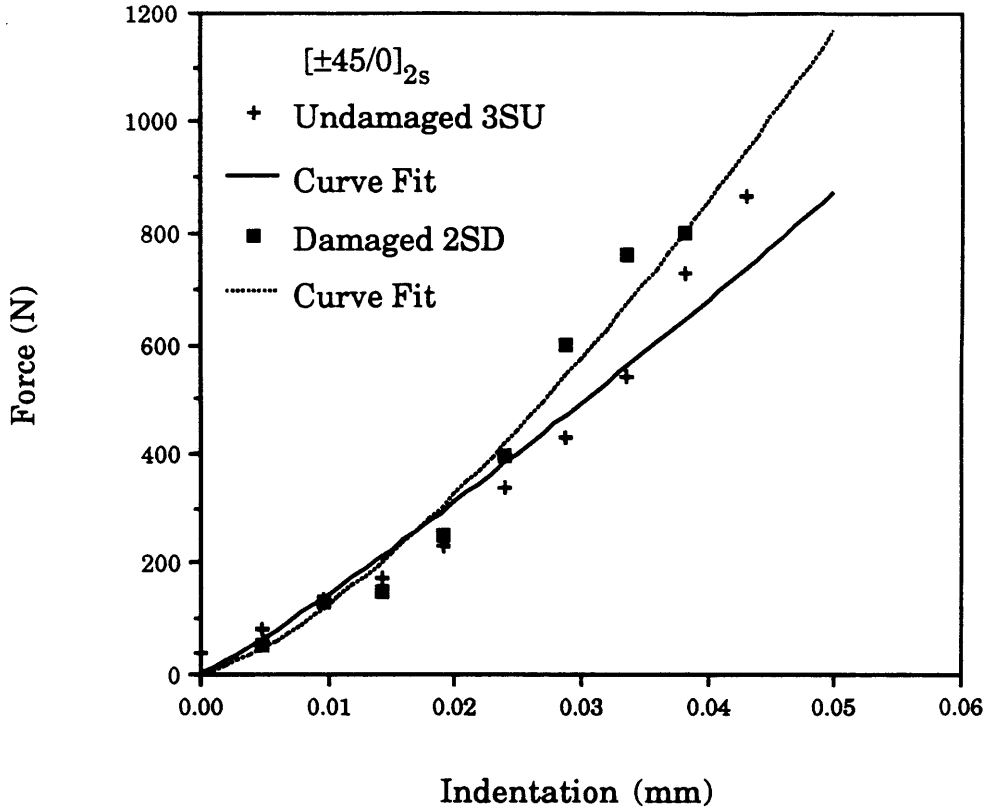


FIGURE 6.8 Experimental impactor/plate contact relations for a damaged and undamaged $[\pm 45/0]_{2s}$ laminate.

the data. The damaged laminate has a higher contact stiffness of 78.3 kN/mm^n whereas the contact stiffness of the undamaged laminate is 26.2 kN/mm^n. The exponent n has also increased. The exponent for the damaged laminate is 1.40 whereas the exponent for the undamaged laminate is 1.14. The contact stiffness for the damaged laminate is likely to have increased because the laminate is more able to conform to the shape of the indentor as illustrated in Figure 6.4 (b). If the plate conforms to the shape of the indentor more, then the area of contact increases. For a given amount of indention, a larger area of contact results in the plate carrying more load. Since the analysis assumes the laminate to be undamaged, the loads that it predicts for a given indentation will be smaller than the experimentally measured loads in damaged laminates.

6.2 Dynamic Impact Response

The analysis discussed in Section 3.3 is used here to provide data for comparison with the experimental results. The average experimental reduced data from the tests discussed in Section 6.1, rather than the data from the analysis, is used as inputs to the global analysis. Normally, one would not have the experimental data available and the data from the local analysis would have to be used. Since the experimental data is available in this case, it is used to allow a better representation of the actual contact relation for the global analysis.

The inputs for the analysis of the $[0/90]_{5s}$ laminate are summarized in Table 6.3. The plate is clamped at two ends and free at the others, like a wide beam. The impactor mass is 1.528 kg and the velocity is 1.52 m/s. This is the

TABLE 6.3 Inputs for Global Analysis of $[0/90]_{5s}$ AS4/3501-6 Laminates

x-direction B.C.:	Clamped - Clamped
y-direction B.C.:	Free - Free
Plate thickness:	2.600 mm
Plate density:	1506 kg/m^3
Plate length:	251 mm
Plate width:	88.9 mm
D_{1111}:	126.48 N·m
D_{1122}:	4.34 N·m
D_{2222}:	97.26 N·m
D_{1212}:	8.79 N·m
D_{1112}:	0.00 N·m
D_{2212}:	0.00 N·m
A_{44}:	12.04 MN/m
A_{45}:	0.00 MN/m
A_{55}:	12.04 MN/m
Shear correction factor:	0.83333
In-plane load, x-direction:	0.00 N/m
In-plane load, y-direction:	0.00 N/m
Impactor mass:	1.528 kg
Impactor velocity:	1.52 m/s
Contact stiffness:	0.040126 GN/m^n
Exponent n:	1.0891
Number of modes in x-direction:	8
Number of modes in y-direction:	8
Condense rotary inertia?:	Yes
Eliminate even modes from analysis?:	Yes
Time step length:	1.0 μs
Number of time steps:	(Program runs until the displacement of the plate returns to zero.)

average velocity at which the test specimens were impacted. The radius of the impactor is not needed here since the loading used in the analysis is assumed to be a point load. The contact stiffness of 0.040126 GN/m^n with n equal to 1.0891 is the average experimental reduced data from the quasi-static tests and is used in the dynamic analysis. The average experimental thickness and density of the test specimens were also used in the analysis.

The force history of the plate obtained from the analysis is shown in Figure 6.9. It may be compared with the experimental data shown in Figures 5.14 through 5.17. The analytical force history has a peak load of 772 N at 7.8 ms. The peak force measured during the experiments varied from 588 N to 641 N at 3.8 ms to 5.1 ms for impact velocities in the range of 1.4 m/s to 1.5 m/s. Thus, the analysis predicts a peak load approximately 23% larger than the experiments. The analysis also predicts the length of the impact event to be longer than that experimentally measured. The analysis predicts the length of contact to be 17.1 ms long, whereas the experiments have lengths of 12.2 ms to 12.6 ms. Thus, the analysis predicts the event to be approximately 39% longer than that experimentally measured.

The analytical and experimental force histories do have some points in common. Both have the same basic sinusoidal shape with predominant higher frequencies superimposed on it. The Fast Fourier Transforms of an experimental (with the impactor rod frequencies set to zero) and the analytical force histories are shown in Figures 6.10 and 6.11, respectively. The Fast Fourier Transform of the analytical force history has three well-defined higher frequencies at 7,700 rad/s, 19,800 rad/s, and 34,700 rad/s. The Fast Fourier Transform of the experimental force history, on the other hand, has one higher frequency of significant magnitude equal to 5,200 rad/s. Thus, the analysis is not properly predicting the higher vibration frequencies. Note that

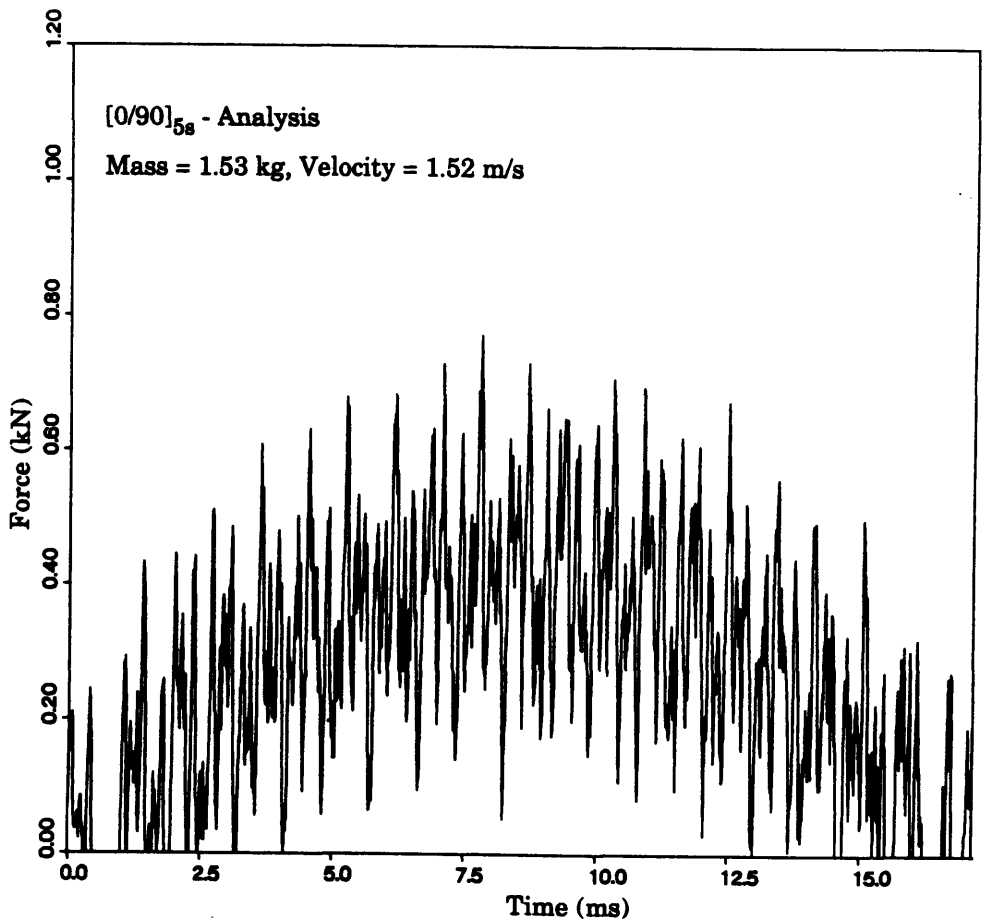


FIGURE 6.9 Analytical force history of a $[0/90]_{5s}$ laminate impacted at 1.52 m/s with a 1.53 kg impactor.

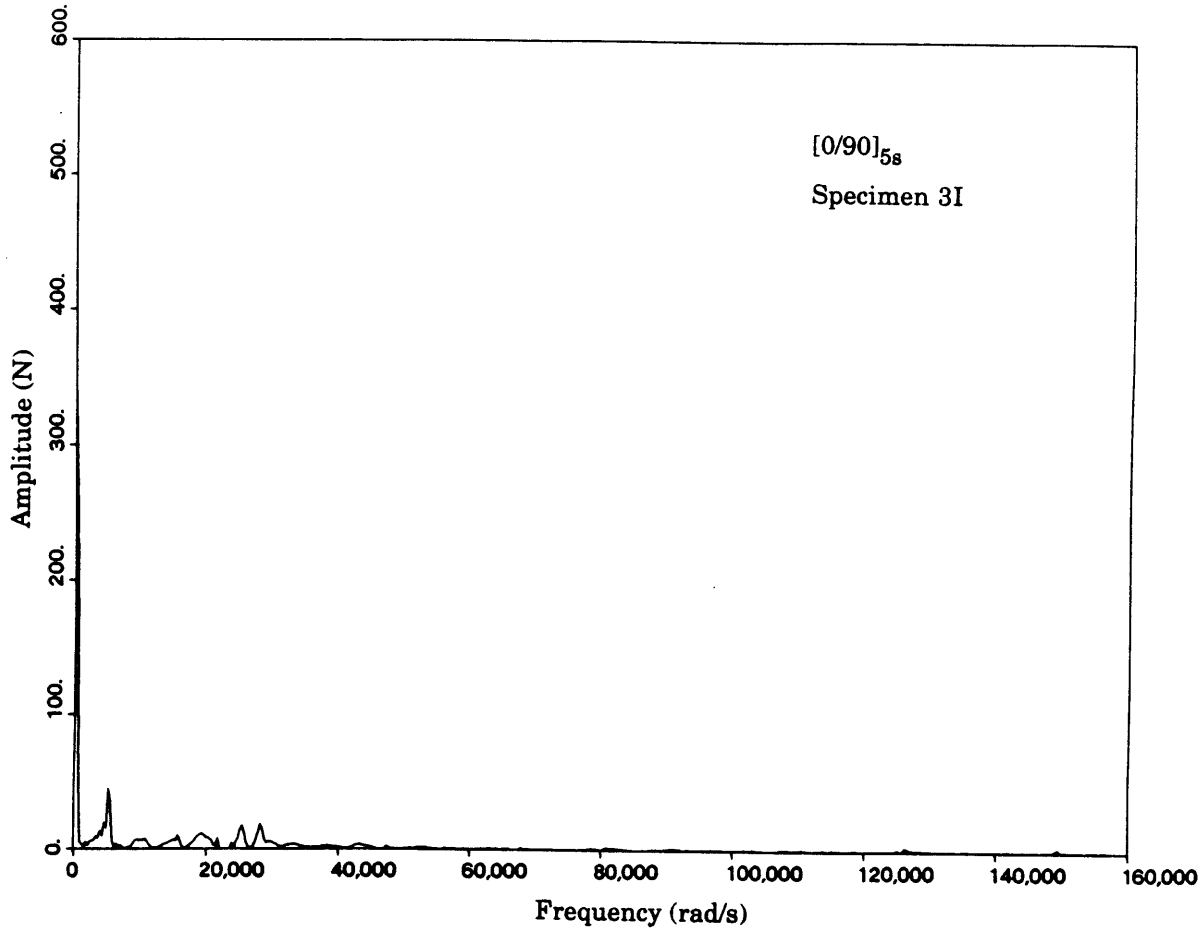


FIGURE 6.10 Fast Fourier Transform of the filtered experimental force data for a $[0/90]_{5s}$ laminate - specimen 3I.

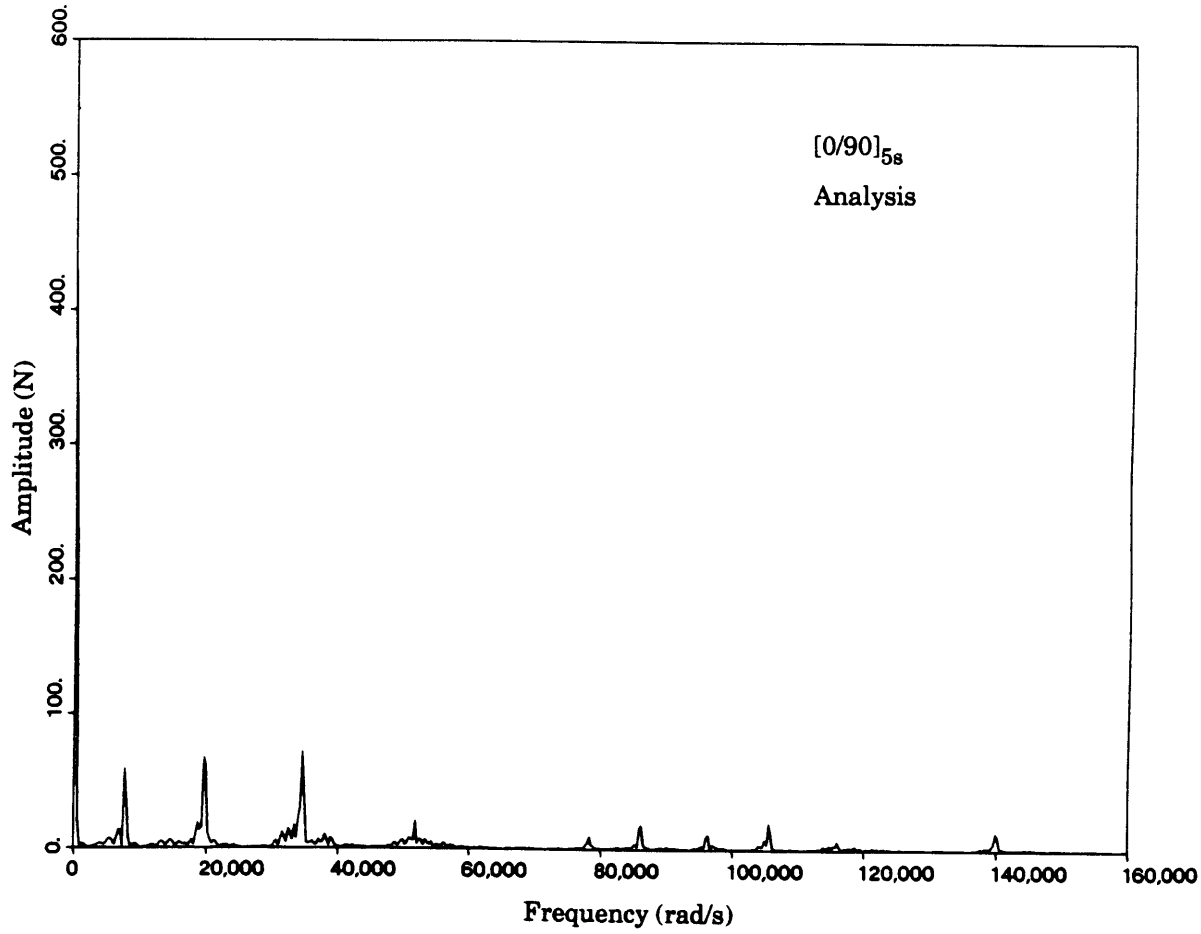


FIGURE 6.11 Fast Fourier Transform of the analytical force data for a $[0/90]_{5s}$ laminate.

these frequencies are of the impactor/plate system and not the plate itself. The fundamental plate frequency is analytically calculated, by considering the plate to be a wide beam, to be 2,020 rad/s. The analysis does predict a period of time during which there is no force. This is from 0.47 to 1.00 ms. The impactor is not in contact with the plate during this time. The experimental force histories also have this trend. The force is not exactly zero, but it hovers around zero from 0.54 to 1.22 ms. During this time, the impactor and the plate are in a state of free vibration. The impactor eventually contacts the plate again, as indicated by the force becoming nonzero.

The analytical and experimental displacement histories are shown in Figure 6.12. They both have the same half sinewave shape. However, the experiment shows a peak displacement of 5.5 mm, whereas the analysis predicts a peak displacement of 8.1 mm, a 47% difference.

The inputs used for the analysis of the $[\pm 45/0]_{2s}$ laminates are summarized in Table 6.4. The average experimental impactor velocity of 1.46 m/s was used. The contact stiffness of 0.050262 GN/m^n with n equal to 1.1210 is the average experimental reduced data from the quasi-static tests. The average experimental thickness and density of the test specimens were also used in the analysis.

The force history of the plate is shown in Figure 6.13. It may be compared with the experimental data shown in Figures 5.21 through 5.24. The analytical force history has a peak load of 386 N for an impact velocity of 1.46 m/s. The peak force measured during the experiments varied from 440 N to 496 N for impact velocities in the range of 1.4 m/s to 1.5 m/s. The analysis is predicting a peak load approximately 18% less than the experiments. This is opposite from that found for the $[0/90]_{5s}$ laminates, where the peak load was predicted by the analysis to be 23% larger. The analysis does predict the length

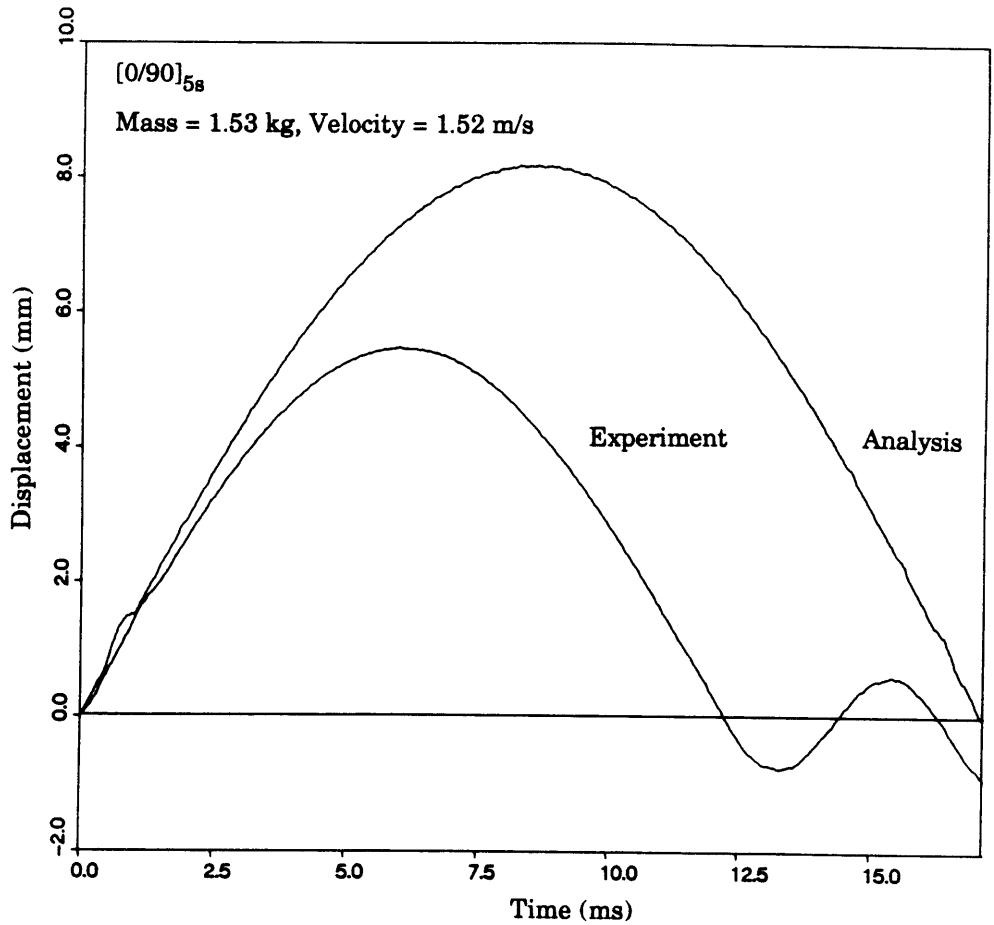


FIGURE 6.12 Analytical and experimental (specimen 3I) displacement histories of a $[0/90]_{5s}$ laminate impacted at 1.52 m/s with a 1.53 kg impactor.

TABLE 6.4 Inputs for Global Analysis of $[\pm 45/0]_{2s}$ AS4/3501-6 Laminates

x-direction B.C.:	Clamped - Clamped
y-direction B.C.:	Free - Free
Plate thickness:	1.572 mm
Plate density:	1498 kg/m^3
Plate length:	251 mm
Plate width:	88.9 mm
D_{1111}:	20.32 N·m
D_{1122}:	9.15 N·m
D_{2222}:	12.75 N·m
D_{1212}:	10.13 N·m
D_{1112}:	2.09 N·m
D_{2212}:	2.09 N·m
A_{44}:	6.76 MN/m
A_{45}:	0.00 MN/m
A_{55}:	7.88 MN/m
Shear correction factor:	0.83333
In-plane load, x-direction:	0.00 N/m
In-plane load, y-direction:	0.00 N/m
Impactor mass:	1.528 kg
Impactor velocity:	1.46 m/s
Contact stiffness:	0.050262 GN/m^n
Exponent n:	1.1210
Number of modes in x-direction:	15
Number of modes in y-direction:	15
Condense rotary inertia?:	Yes
Eliminate even modes from analysis?:	No
Time step length:	1.0 μs
Number of time steps:	(Program runs until the displacement of the plate returns to zero.)

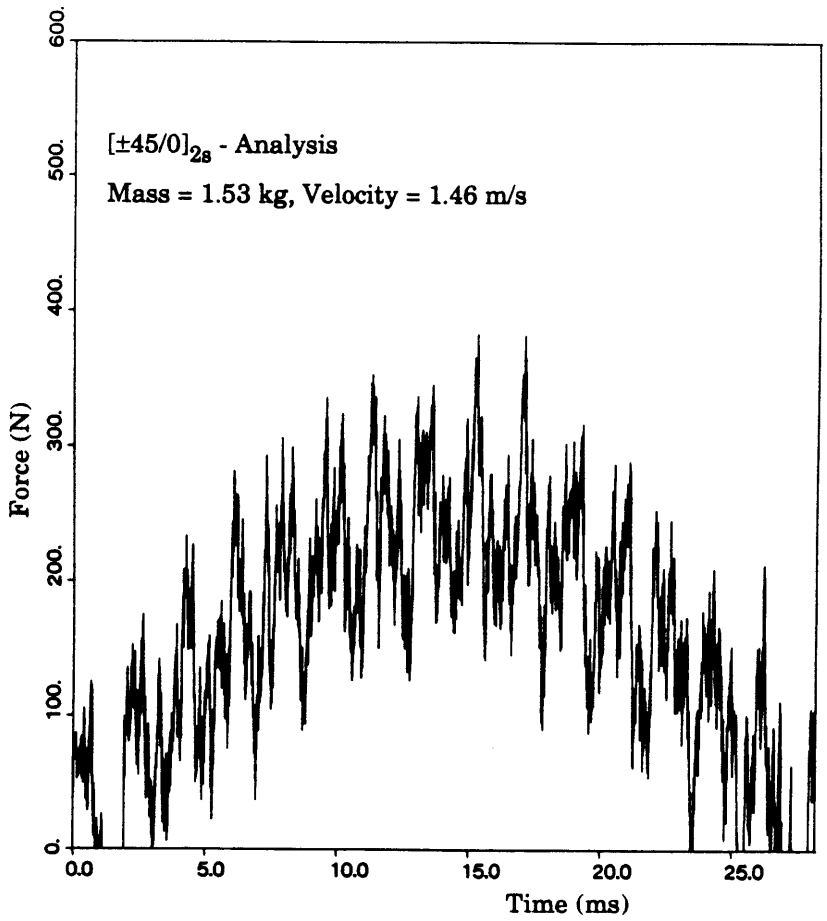


FIGURE 6.13 Analytical force history of a $[\pm 45/0]_{2s}$ laminate impacted at 1.46 m/s with a 1.53 kg impactor.

of the impact event to be longer as before. The event is predicted to be approximately 51% longer by the analysis, 28.1 ms as opposed to 18.7 ms in the experiments. This is similar to the $[0/90]_{5s}$ laminates, where the event was predicted by the analysis to be 39% longer than the experimental results.

The general shape of the force histories are similar to the force histories of the $[0/90]_{5s}$ laminates. The Fast Fourier Transforms of an experimental (with the impactor rod frequencies set to zero) and the analytical force histories are shown in Figures 6.14 and 6.15 respectively. The Fast Fourier Transform of the analytical force history shows four well-defined higher frequencies at 3,400, 6,300, 11,000, and 20,000 rad/s. The Fast Fourier Transform of the experimental force history, on the other hand, has one higher frequency of significant magnitude at 3,700 rad/s. This frequency is close to the 3,400 rad/s frequency from the analysis. The fundamental plate frequency is analytically calculated, by considering the plate as an orthotropic beam, to be 1,044 rad/s. Another item to note is the loss of contact between the impactor and the plate early on in the impact event for both experiment and analysis. The loss of contact is from 1.1 ms to 1.9 ms for the analysis and from 1.0 ms to 2.1 ms for the experiment. This was also observed for the $[0/90]_{5s}$ laminates.

The displacement histories of a $[\pm 45/0]_{2s}$ laminate from the analysis and an experiment are shown in Figure 6.16. They have the same shape, but are off both in peak displacement and length of the impact event. The experiments show a peak displacement of about 8.3 mm, whereas the analysis predicts a peak displacement of 12.9 mm, a 55% difference. Note the kink in the analytical displacement history at 1.7 ms. This is due to the lost of contact between the impactor and plate at this time. The same kink is present in the experimental data at 0.9 ms, but it is not as pronounced.

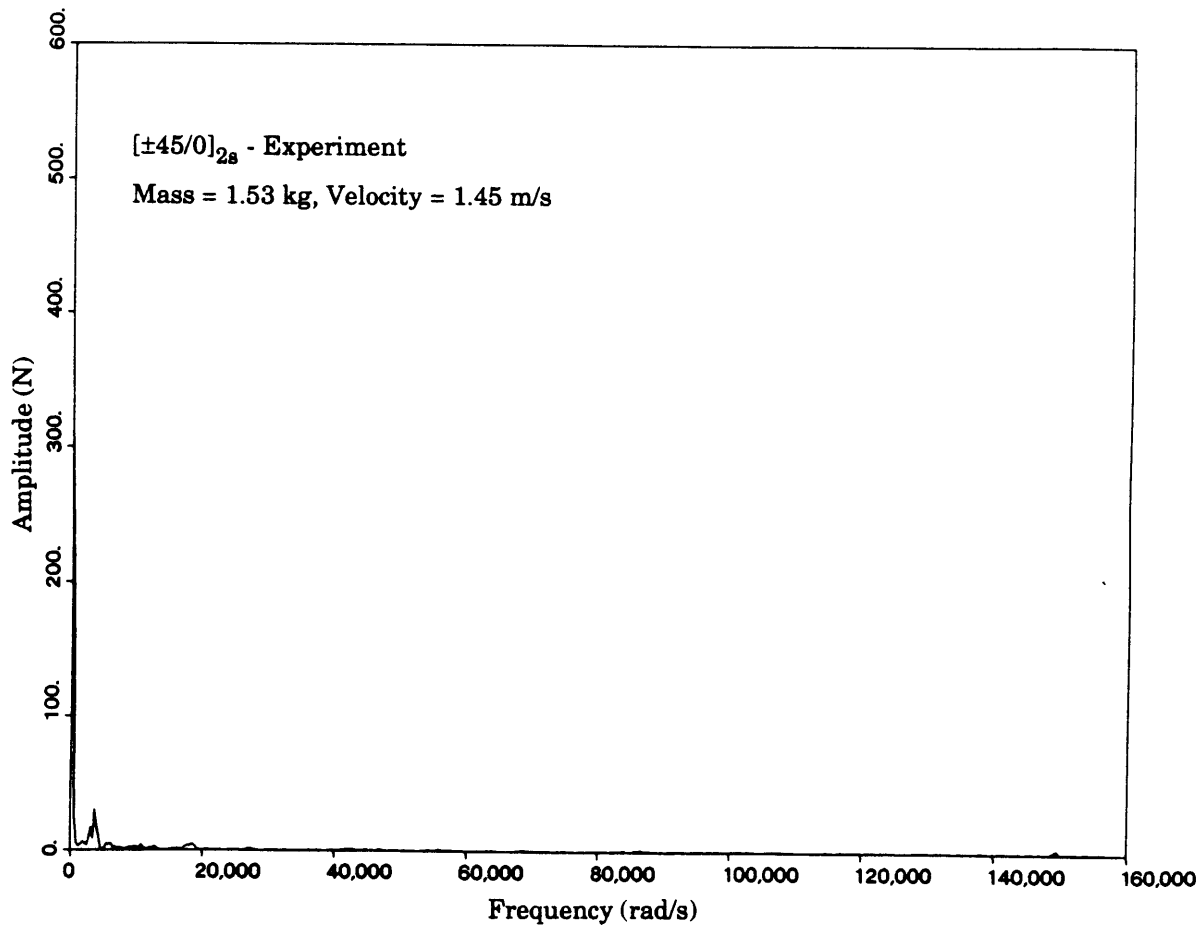


FIGURE 6.14 Fast Fourier Transform of the filtered experimental force data for a $[\pm 45/0]_{2s}$ laminate - test of specimen without the extension rod.

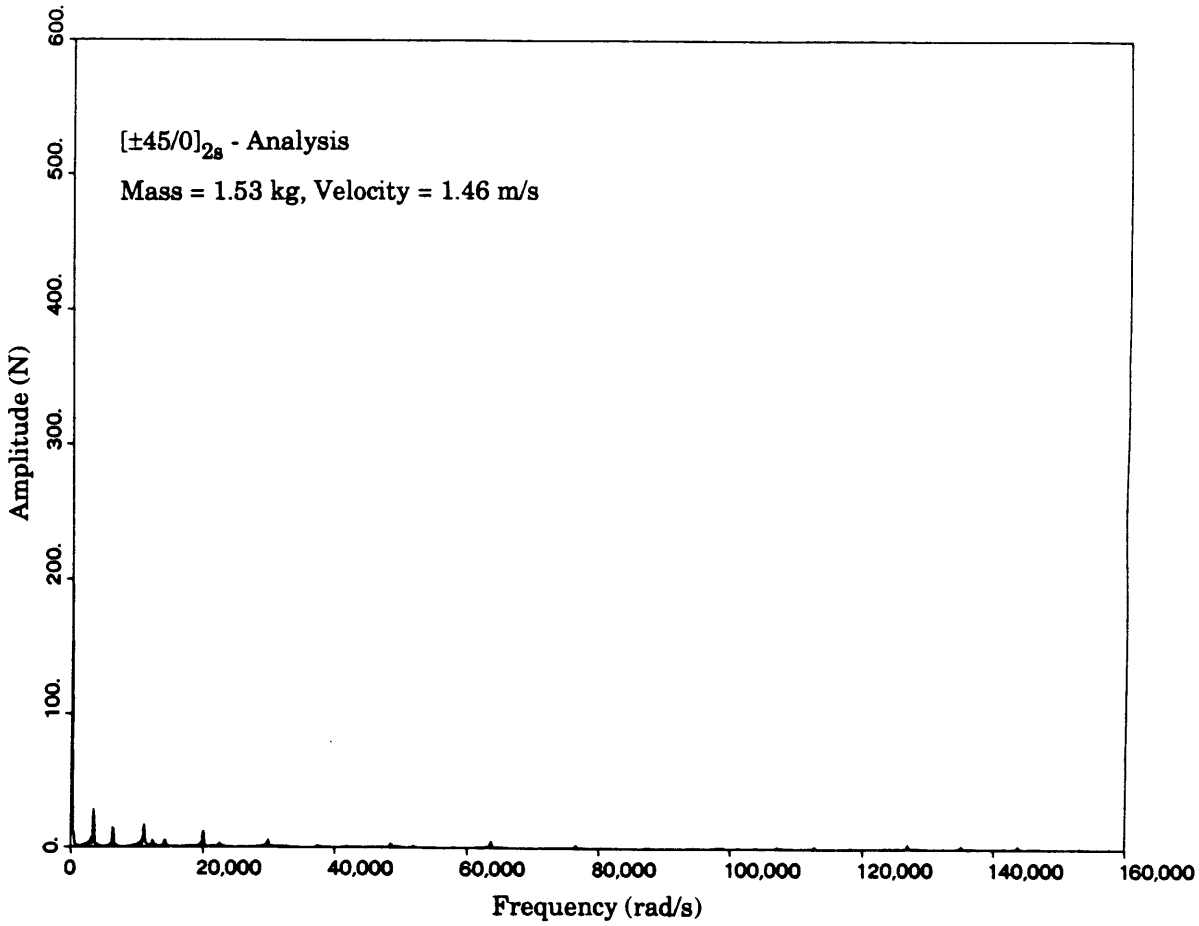


FIGURE 6.15 Fast Fourier Transform of the analytical force data for a $[\pm 45/0]_{2s}$ laminate.

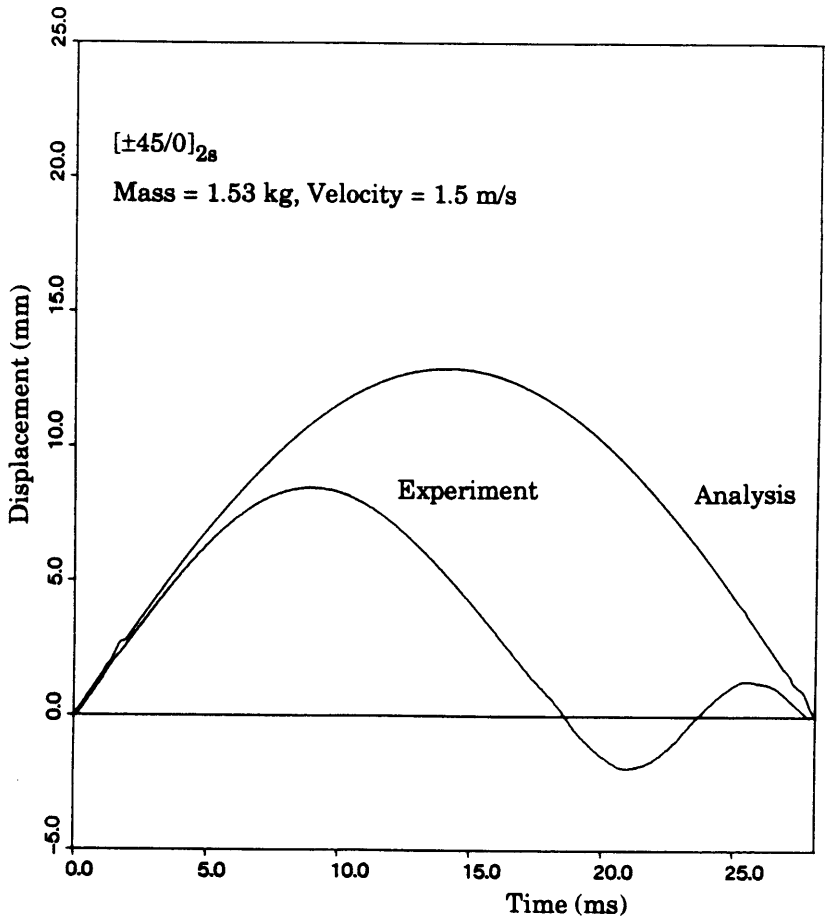


FIGURE 6.16 Analytical and experimental (test of specimen without the extension rod) displacement histories of a $[\pm 45/0]_{2s}$ laminate impacted at 1.5 m/s with a 1.53 kg impactor.

The peak force and peak displacement were both predicted to be larger than the experimentally obtained data for the $[0/90]_{5s}$ laminates. However, the data from the $[\pm45/0]_{2s}$ laminates show a different trend. The force is underpredicted, whereas the displacement is overpredicted. Note that the peak load from the experiments of the $[\pm45/0]_{2s}$ laminates (469 N) is smaller than that of the $[0/90]_{5s}$ laminates (627 N). The same trend is true for the analysis; the peak load of the $[\pm45/0]_{2s}$ laminates is 386 N, which is less than the peak load of 771 N for the $[0/90]_{5s}$ laminates. Thus, the problem seems to lie in the fact that the analysis underpredicts the force for the $[\pm45/0]_{2s}$ laminates and overpredicts the force for the $[0/90]_{5s}$ laminates. This could in part be because of the poor curve fit from Eq. (2.1) to the experimental contact relation. It is evident from Figure 6.6 that at the higher indentations, the curve fit falls below the actual load for the $[\pm45/0]_{2s}$ laminates. Since the information on the curve fit is being used in the global analysis, it may explain why the peak load from the analysis is less than the experiments for the $[\pm45/0]_{2s}$ laminates.

The force and displacement histories of the predamaged laminates are shown in Figures 5.31 through 5.36. They may be compared with the force and displacement histories of the undamaged laminates shown in Figures 5.21 through 5.27.

The general shape of the force histories of the undamaged and predamaged laminates is the same. The peak force to the undamaged laminates varies from 440 N to 496 N, whereas the peak force to the predamaged laminates varies from 418 N to 499 N. The length of the impact event is the same (18.7 ms) for both the damaged laminates and the undamaged laminates. Comparing the force history of an undamaged laminate in Figure 5.22 with the force history of a damaged laminate in Figure 5.31 reveals that the peak force seen by the damaged laminate is somewhat smaller than the peak force

seen by the undamaged laminate. The peak force of the undamaged laminate is 459 N and the peak force of the damaged laminate is 418 N, which is 9% smaller. Both of these specimens were impacted with the same impact velocity. This is a small difference, and is not experimentally significant.

The displacement histories of the predamaged laminates are almost identical to the undamaged laminates. They have the same half sinusoidal shape. The peak displacement of the damaged laminates varies from 8.0 mm to 8.5 mm. The peak displacement of the undamaged laminates varies from 7.9 mm to 8.6 mm. Comparing the displacement history in Figure 5.26 of an undamaged laminate with the displacement history in Figure 5.34 of a damaged laminate supports the conclusion that the presence of damage does not affect the displacement history. Both specimens were impacted with the same impactor velocity. The undamaged laminate has a peak displacement of 7.9 mm and the damaged laminate has a peak displacement at 8.0 mm. The lengths of the impact event are 18.6 ms and 19.1 ms, respectively.

It appears that the presence of damage does not change the force history or the displacement history. The force history was expected to change since it is dependent on the local contact relation. This relation was observed to change (from the quasi-static tests) when the laminate has damage. The displacement history, on the other hand, is not dependent on the local response. It depends on the global structural properties.

From these results, it may be concluded that the global analysis may neglect the presence of damage. This conclusion is based on tests to laminates that have damage internally (the same size of the impactor diameter) and some back-face damage. This conclusion may not apply when the impactor mass and velocity are high enough to cause penetration of the laminate. However, internal and back-face damage are a significant amount of damage.

Since the force and displacement histories of undamaged and predamaged laminates are the same, the analysis may be used in cases in which virgin composite plates are impacted hard enough to initiate damage.

CHAPTER 7

Conclusions and Recommendations

The dynamic response of graphite/epoxy plates subjected to impact loading has been investigated both experimentally and analytically in this research effort. The impact event was broken down into two levels: a global level which deals with the overall structural response, and a local level which deals with the local plate behavior at the point of impact. Both the analysis and the experiments were broken down into these levels. The experiments were conducted with virgin and predamaged specimens. The reason for this is twofold. First, experimental data from virgin specimens may be compared with the analysis. Second, the effect of damage in a specimen could be determined experimentally, and the validity of the assumption in the analysis that the specimens are undamaged could be checked.

The following conclusions may be made concerning the impactor/plate contact relation:

1. The local analysis is able to predict the impactor/plate contact relation well for the laminates studied here. The predictions for large indentations may be improved by modifying the geometric assumption of the deformation of the plate around the indentor.

2. The least squares curve fits to the relation given in Eq. (2.1) results in an exponent which is close to unity. Thus, an exponent of 1.5, which

reduces the equation to the classical Hertzian contact relation, does not appear to be appropriate for composite plates.

3. The power relation given in Eq. (2.1) results in a curve which does not fit the experimental data well.

4. The out-of-plane material properties control the local contact relation. Thus, the in-plane orientation of the plies has little effect on the contact relation.

5. A laminate with internal and limited back-face damage has an apparent larger contact stiffness than a laminate free of damage due to the surface of a damaged laminate being more able to conform to the shape of the indentor.

6. The curvature of the plate is important in determining the contact relation since the indentation that results from an applied load depends on the area of contact.

The conclusions drawn from the dynamic impact response analysis and experiments are summarized as follows:

7. The analysis does not accurately predict the force or displacement histories of the impact event, but it does capture some of the characteristic features. These include the general shape of the histories and the detection of the loss of contact and subsequent recontact of the impactor with the plate.

8. The presence of internal and limited back-face damage in a laminate does not significantly affect the force or displacement histories of the impact event. This means that the global analysis may neglect the presence of local damage.

As a result of the current research, a number of items arise which are worthy of further investigation:

1. Work should be done on the analysis to determine the actual deformation illustrated in Figure 6.4 (b). The load to produce this deformation should then be calculated and compared with experimental results.

2. Since the contact relation depends on the area of contact, the relation for a curved surface will differ from a flat surface. Thus, a method to model the contact relation in composite materials with curvature needs to be developed.

3. Since the relation given by Eq. (2.1) does not properly represent the experimental data, polynomials of various degrees should be tried. They should be incorporated into the global model to see if the dynamic response is sensitive to the improvement in the curve fit.

4. The global analysis needs to be improved to better model the dynamic response of an impacted composite plate. The following areas may be worth investigating: (a) load distributions other than a point load may

be necessary, (b) damping may need to be included, and (c) stretching of the laminate could be considered.

5. More precise indentation and force readings should be obtained in future quasi-static tests. This would not only give finer resolution, but would also reduce the scatter in the data from the noise inherent in measurement devices.

6. A circular clamping device is recommended for future quasi-static tests. This will give boundary conditions consistent with the local model analysis.

7. More quasi-static and dynamic tests need to be conducted to provide additional experimental data of the type generated here, for comparison with the analyses. The following are recommended for future tests: (a) tests should be done with laminates made from a different material system, (b) the thickness and area of the laminates should be varied, (c) the mass of the impactor should be varied, and (d) tests should be done to laminates with curvature, as well as (e) laminates under preload.

Following these recommendations should provide more insight into the the dynamic response of impacted composite plates, as well as lead to a better model. Recall that the ultimate goal is to develop models which aid engineers in the design of composite structures which meet requirements on damage resistance and damage tolerance. Thus, further research must be geared toward providing such analyses.

References

1. Cairns, D. S., "Impact and Post-Impact Response of Graphite/Epoxy and Kevlar/Epoxy Structures," TELAC Report 87-13, Massachusetts Institute of Technology, Cambridge, MA, August 1987.

2. Yang, P. C., Norris, C. H., and Stavsky, Y., "Elastic Wave Propagation in Heterogeneous Plates," International Journal of Solids and Structures, Vol. 2, No. 4, 1966, pp. 665-684.

3. Whitney, J. M. and Pagano, N. J., "Shear Deformation in Heterogeneous Anisotropic Plates," Journal of Applied Mechanics, Vol. 37, No. 4, December 1970, pp. 1031-1036.

4. Greszczuk, L. B., "Response of Isotropic and Composite Materials to Particle Impact," Foreign Object Impact Damage to Composites, ASTM STP 568, American Society for Testing and Materials, 1975, pp. 183-211.

5. Sun, C. T. and Chattopadhyay, S., "Dynamic Response of Anisotropic Laminated Plates Under Initial Stress to Impact of a Mass," Journal of Applied Mechanics, Vol. 42, No. 3, September 1975, pp. 693-698.

6. Caprino, G., Visconti, I. C., and Ilio, A. D., "Elastic Behaviour of Composite Structures Under Low Velocity Impact," Composites, Vol. 15, No. 3, July 1984, pp. 231-234.

7. Shivakumar, K. N., Elber, W., and Illg, W., "Prediction of Impact Force and Duration Due to Low-Velocity Impact on Circular Composite Laminates," Journal of Applied Mechanics, Vol. 52, No. 3, September 1985, pp. 674-680.

8. Sun, C. T. and Chen, J. K., "On the Impact of Initially Stressed Composite Laminates," Journal of Composite Materials, Vol. 19, No. 6, November 1985, pp. 490-504.

9. Graves, M. J. and Koontz, J. S., "Initiation and Extent of Impact Damage in Graphite/Epoxy and Graphite/PEEK Composites," Presented at the AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, AIAA Paper No. 88-2327, Williamsburg, VA, April 1988.

10. Aggour, H. and Sun, C. T., "Finite Element Analysis of a Laminated Composite Plate Subjected to Circularly Distributed Central Impact Loading," Computers & Structures, Vol. 28, No. 6, 1988, pp. 729-736.

11. Wu, H. T. and Springer, G. S., "Impact Induced Stresses, Strains, and Delaminations in Composite Plates," Journal of Composite Materials, Vol. 22, No. 6, June 1988, pp. 533-560.

12. Wu, H. T. and Springer, G. S., "Measurements of Matrix Cracking and Delamination Caused by Impact on Composite Plates," Journal of Composite Materials, Vol. 22, No. 6, June 1988, pp. 518-532.

13. Cairns, D. S. and Lagace, P. A., "Transient Response of Graphite/Epoxy and Kevlar/Epoxy Laminates Subjected to Impact," Presented at the AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, AIAA Paper No. 88-2328, Williamsburg, VA, April 1988.

14. Yang, S. H. and Sun, C. T., "Indentation Law for Composite Laminates," Composite Materials: Testing and Design, ASTM STP 787, American Society for Testing and Materials, 1982, pp. 425-449.

15. Tan, T. M. and Sun, C. T., "Use of Statical Indentation Laws in the Impact Analysis of Laminated Composite Plates," Journal of Applied Mechanics, Vol. 52, No. 1, March 1985, pp. 6-12.

16. Sankar, B. V., "Contact Law for Transversely Isotropic Materials," Presented at the AIAA/ASME/ASCE/AHS 26th Structures, Structural Dynamics and Materials Conference, AIAA Paper No. 85-0745, Orlando, FL, 1985.

17. Cairns, D. S. and Lagace, P. A., "Thick Composite Plates Subjected to Lateral Loading," Journal of Applied Mechanics, Vol. 54, No. 3, September 1987, pp. 611-616.

18. Kraft, M. J., "Impact Damage Response of Graphite/Epoxy Fabric Structures," TELAC Report 88-9, Massachusetts Institute of Technology, Cambridge, MA, July 1988.

19. Dugundji, J., "Simple Expressions for Higher Vibration Modes of Uniform Euler Beams," AIAA Journal, Vol. 26, No. 8, August 1988, pp. 1013-1014.

20. Lie, S. C., "Damage Resistance and Damage Tolerance of Thin Composite Facesheet Honeycomb Panels," TELAC Report 89-3, Massachusetts Institute of Technology, Cambridge, MA, March 1989.

Appendix A

FORTRAN Source Codes

The following codes are contained in this Appendix: Program INPUT, Program ROTATE, Program BESSEL (local model analysis), Program KNPUT, Program IMPACT (global model analysis), Program LIBRARY, and Program VIEW. These codes are written with VAX FORTRAN Version 4.6. Some of the codes are broken down into several files. These files are to be compiled and linked to generate the executable file. For example, Program IMPACT consists of 10 files: GLOBAL1.FOR through GLOBAL8.FOR, LINPAK.FOR and EISPAK.FOR. This particular program also has an include file, IMPACT.INC, which is not compiled, but needs to be in the directory with the FORTRAN files when they are compiled. The subroutines in LINPAK.FOR and EISPAK.FOR are not shown here. They are commercially available software called Linpack and Eispack. In addition to the source code, the file ACPI.COM is listed in this Appendix. This is the main controlling file needed to run the codes in the ACPI (Analysis of Composite Plate Impact) system configuration. An outline on how to run the codes is given in Appendix B.

```
      program input
C   INPUT1.FOR                                          4/18/89
C Program written by Chris Park 1988, modifications by Kiernan F. Ryan.
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C=================================================================
C     this program generates the input file to be used by ROTATE
C
C     The possible options are:
C
C     1. Create a new input file
C     2. Edit an old input file
C     3. Editing a binary file
C     4. quit
C=================================================================
C*****************************************
C**        VARIABLE DECLARATION         **
C*****************************************
      implicit real*8 (a-h,o-z)
      integer ply
      character*80  lay, title
      character*60  title1, title2, title3
      character*40  name
      character*35  bc(6)
      character*20  ofile, filnam, level
      character*10  genfil, filtyp
      character*6   jobnam, newjob
      character*3   ld$, outyp, ldhzs, que(0:1), stat
      character*1   sure, glob, RHOYN
C
      dimension alphaa(10), tqa(10), ept(3,3),epc(3,3)
C
      logical exst,edit,default,fil2
C
C COMMON BLOCK DECLARATIONS
C
      common /block1/ mat(50),thp(50),tp(50),n,nflag,nflag2,nflag3,ply
      common /block2/ E11,E22,E33,G23,G31,G12,ENU12,ENU13,
     &                ENU23,name,thick,densty
```

```
      common /block3/ last
      common /block4/ nm,matf(50)
C
C BLOCK 5
C MODULES: INPUT, DEFAULT
C     DEFAULT = (logical) flag. If default = .true., all default
C               values are used
C
      common /block5/ default
C
C BLOCK 7
C MODULES: INPUT, MENU, DIRACC
C     MAIN   = integer (1..4) main menu option number
C     JOBNAM = analysis jobname
C     FILTYP = Flag: "binary" or "sequential" to check entrypoint
C     LEVEL  = entry/exit point descriptors
C     LENTER = analysis entry point
C     LEXIT  = analysis exit point
C     GENFIL = array containing intermedite file status
C
      common /block7/ main,jobnam,level(4),lenter,lexit,genfil(15)
C
C BLOCK 8
C MODULES: INPUT, SETFIL, ,DIRACC, EXECUTE
C     FILNAM = array containing all the filenames in the analysis
C
      common /block8/ filnam(15)
C
      data numalp,alphaa(1),alphaa(2),alphaa(3),alphaa(4),alphaa(5),
     & alphaa(6),alphaa(7),tqa(1),tqa(2),tqa(3),tqa(4),tqa(5),tqa(6),
     & tqa(7)/7,.025,.05,.075,.1,.125,.15,.175,1.,1.,1.,1.,1.,1.,1./
C
C initialize bc(n), que(n), level(n)
C
        bc(1) = 'Simply Supported - Simply Supported'
        bc(2) = 'Clamped - Free'
        bc(3) = 'Clamped - Clamped'
        bc(4) = 'Free - Free'
        bc(5) = 'Simply Supported - Clamped'
        bc(6) = 'Simply Supported - Free'
C
        que(0) = 'No'
        que(1) = 'Yes'
C
        level(1) = 'Laminate Properties'
        level(2) = 'Hertzian Spring'
        level(3) = 'Dynamic Analysis'
        level(4) = 'Damage Prediction'
C
C see if LIBRARY.BIN is in the current directory
C
      inquire(file='library.bin',exist=exst)
      if (.not. (exst)) then
```

```
        write(6,*)'Need file LIBRARY.BIN in this directory'
        stop
      end if
      open(unit=7, file='library.bin', status='old', access=
     & 'direct',recl=144)

C=========(initialization of variable LAST)==============
      last=0
      thick=0
      do while (thick .ne. -1)
        last=last+1
        call matrl(last)
      end do
C
C INITIALIZE SYSTEM CONFIGURATION VARIABLES
C
      jobnam='-------'
      filnam(1)='not.set'
      lenter=1
      lexit=4
      do i=1,15
        genfil(i)='suppress'
      end do
C=================================================================
C              Program Controller Routine
C
C  This section tells the program where to go based upon the selection
C made in the main menu.
C
   10 call menu
      if (main .eq. 1) then
        go to 100
      else if (main .eq. 3) then
        call execute
      else if (main .eq. 2) then
        call configure
      else if (main .eq. 4) then
        call quit
      end if
      go to 10
C=====&===========================================================
C                        MAIN PROGRAM
C

 100   write (6,110)
      do i=1,3
        do j=1,3
                ept(i,j)=0.
                epc(i,j)=0.
        end do
      end do
 110   format ('1',///////,t20,'Parameter Entry Menu',//
     &  ' ',20x,'1.  Create a new job file'//
```

```
     &  ' ',20X,'2.  Edit/Load an old job file'//
     &  ' ',20x,'3.  Exit to main menu'/////////)
 120   format(' ',i1,':"',60x,'"')
 130   format('+',i1,':"',$)
      edit=.false.
 140   format(////' ','Enter selection and hit <RETURN>: ',$)
      write(6,140)
 150   read(5,*)m
      if (m .eq. 3) then
        close(unit = 7, status='keep')
        go to 10
      else if (m .eq. 2) then
       if (jobnam .eq. '———————') then
 160    write(6,170)
 170    format(' ','Enter JOBNAME ("quit" to exit)[max 6 ',
     &  'characters]: ',$)
        read(5,180)newjob
 180    format(a6)
        if (newjob .eq. 'quit' .or. newjob .eq. 'QUIT') then
                go to 100
        end if
        jobnam=newjob
C
C SETFIL adds the extensions to the jobname and stores them in the array
C filnam(N) [See comments in the subroutine for the exact order
C of the filenames]
C
        call setfil(jobnam)
       end if
        inquire (file=filnam(2), exist=fil2)
        inquire (file=filnam(1), exist=exst)
       if (fil2 .and. exst) then
        write(6,185)
 185    format(/' ','Select a file type to edit:'//
     &' ','1. Initial job file (Edit to change material specs or if '/
     &' ','   you have not run the job through the analysis.  This'/
     &' ','   file is used as input only by the first entry point'/
     &' ','   in the ANALYSIS CONFIGURATION menu.)'//
     &' ','2. Intermediate job file (Edit for all non-material'/
     &' ','   property changes.  This file is created by the analysis,'/
     &' ','   so it cannot be edited unless you have run the job at'/
     &' ','   least once.  This file is the only accessed by entry'/
     &' ','   points 2-4 in the ANALYSIS CONFIGURATION menu.)'//
     &' ','Enter the file type to edit [1 or 2]: ',$)
        read(5,*) nfil
        if (nfil .lt.1 .or. nfil .gt. 2) go to 100
        go to (190,192) nfil
      else if (exst) then
        go to 190
      else if (fil2) then
        go to 192
      else
        write (6,*) 'No files exist under that jobname.'
```

```
        jobnam='------'
        go to 100
      end if
C
C       if (.not. (exst)) then
C               write(6,190)jobnam, filnam(1)
C 190            format(' ','Unable to to load initial job file for ',
C     &                  'job: ',a6,/' ','Missing file: ',a20)
C               go to 160
C       end if
 190    edit=.true.
        default=.false.
        go to 1700
 192    call diracc
        go to 100
      end if
C
C Create new initial file
C
C Get a jobname
C
 195  format(/' ','Enter a jobname ("quit" to exit) [max 6 ',
     &'characters]: ',$)
      if (jobnam .eq. '------') then
 196  write(6,195)
 197  format(a6)
      read(5,197) newjob
      if (newjob(1:1) .eq. ' ') then
        write(6,*)
        write(6,*) 'Invalid jobname.  Please reenter jobname or type ',
     &             '"quit" to exit.'
        go to 196
      else if (newjob .eq. 'quit' .or. newjob .eq. 'QUIT') then
        go to 100
      end if
      jobnam = newjob
      call setfil(jobnam)
      end if
C
C Give user option to use all default values
C
      call values
C=================================================================
C Input the title in three lines between the quotes.
C
      edit=.false.
      write(6,*)'Input 3 title lines (max 60 characters each)'
 200   format(a60)
 210     write(6,120)1
      write(6,130)1
      read(5,200)title1
      if (edit) go to 1170
 220     write(6,120)2
```

```
      write(6,130)2
      read(5,200)title2
      if (edit) go to 1170
 230     write(6,120)3
      write(6,130)3
      read(5,200)title3
      if (edit) go to 1170

C=================================================================
C    Input the material data:
C

 240   write(6,250)
 250   format(' ','Number of plies [MAX = 50]: ',$)
 260   read(5,*)np
      if (np .gt. 50 .or. np .lt. 1) go to 260
      if (edit) go to 1170
 270   format('1','****** Specimen Layup ******',///
     &' Enter the layup in the form:'/
     &' [m1 A*a/B*b m2 C m1 D*d/E*e/F*f m3 ...]Xs'//
     &' Where:'/
     &'   m1, m2, m3, ...  temporary material markers to indicate a'/
     &'                    change in materials. Explicitly type "m1"'/
     &'                    "m2"... to indicate a material change.'/
     &'   A, B, C, ...     ply angles in degrees'/
     &'   *a, *b, *c, ...  repeat preceeding angle a,b,c times'/
     &'                    (not needed for a, b, c, ... = 1)'/
     &'        /           use to separate ply angles within a given'/
     &'                    material.'/
     &'        ]           denotes end of layup'/
     &'        X           denotes # times to repeat the layup'/
     &'        s           denotes symmetric layup'//
     &' EXPLICIT EXAMPLE: [m1 45*2/-45*2 m2 0]2s   (Note the spaces)'//)
 280   write(6,270)
 290   write(6,*)'ENTER LAYUP: '
      write(6,300)
 300   format(' ','[',$)
      read(5,310)lay
 310   format(a80)
      nflag3=0
      do n=1,50
        mat(n)=0
      end do

C=================================================================
C   Disassemble the entered layup and notify user of any errors

      call layup(lay)

      if (nflag3.eq.1) then
        write(6,*)'LAYUP ERROR:'
        if (nflag2 .eq. 0) then
              write(6,*)'Missing "]"'
```

```
        else
                write(6,*)'Illegal character found'
        end if
        write(6,*)
        write(6,*)'Reenter the layup'
        go to 290
      end if
      if (ply .ne. np) then
        write(6,320)np,ply
 320     format(' ','LAYUP ERROR:'/
     &' ','Number of Plies desired = ',i2/
     &' ','Number of plies entered = ',i2//)
        write(6,*)'Reenter the number of plies AND the layup'
        go to 240
      end if

C=================================================================
C Sort the array MATF(N) which holds the negative of the
C temporary material numbers entered in the layup and
C assign GLOBAL material numbers and thicknesses.(ie if
C m1 is specified in the layup, then a -1 is stored in MATF)
C Negative numbers in MAT and MATF indicate TEMPORARY
C material numbers.
C
C VARIABLES:
C
C NFLAG     counter to indicate when all GLOBAL material
C           numbers have been assigned
C
C MREAD     temporary global material number inputted by user
C
C TRAED     temporary thickness inputted by user (0=use default)
C

      call sort
      nflag=0
 330   write(6,340)
 340   format ('1',////' ',t15,'***** MATERIAL SELECTION *****',///)
 350   if (nflag .ne. 0) then
        write(6,360)'MAT NUM','MATERIAL','THICKNESS'
        write(6,360)'-------','--------','---------'
 360     format(' ',a7,2x,a12,30x,a12)
        do 370 n= 1,nflag
            open(unit=7,file='library.bin',status='old',access='direct',
     &           recl=144)
            call matrl(matf(n))
 370        write(6,380)n,name,tp(n)
       end if

 380        format (' ',3x,i2,t15,a40,t56,g10.4)
      write(6,*)' '
      write(6,*)' '
C      nflag=0
```

```
      do while (nflag .lt. nm)

 390    format(' ','Enter the Material Number for material m',
     &  i2,'(0 for list): ')
 400    write(6,390)-matf(nflag+1)
        read(5,*)mread
        if (mread .eq. 0) then
                call list
                go to 330
        else if (mread .lt. 0 .or.
     &           mread .gt. last) then
                go to 330
        end if

        call matrl(mread)
        if (.not.(thick .gt. 0.)) then
                write(6,*)'*** Undefined Material ***'
                go to 400
        end if
        nflag=nflag+1
 410    format(' ','Enter ply thickness [default(0) = ',g12.6,'mm]: ',$)
        if (.not. (default)) then
        write(6,410)thick
        read(5,*)tread
        if (tread .eq. 0.) then
                tread=thick
        end if
        else
                tread=thick
        end if
C=============================================
C This DO LOOP assigns ply values for MAT and TP
C which are of the same material

        do nnn=1,np
                if (matf(nflag) .eq. mat(nnn)) then
                        mat(nnn)=mread
                        tp(nnn)=tread
                end if
        end do
        write(6,*)' '
        matf(nflag)=mread
      end do
      if (edit) go to 1170
C*********************************************************************
C*                    TEST SPECIMEN SPECIFICATIONS                   *
C*********************************************************************
      write(6,*)
      write(6,*)
      write(6,*)'Enter Specimen Parameters:'
      write(6,*)
 420  write(6,430)
 430  format(' ','X-direction length [mm]: ',$)
```

```
      read(5,*)a$
      if (.not.(a$ .gt. 0.)) go to 420
      if (edit) go to 1170
 440  write(6,450)
 450  format(' ','Y-direction length [mm]: ',$)
      read(5,*)b
      if (.not.(b .gt. 0.)) go to 440
      if (edit) go to 1170
C
C  Change to be automatically calculated
C
C 460  write(6,470)
C 470  format(' ','Density [kg/m↑3]: ',$)
C      read(5,*)rho
  460 rho=0.
      totthk=0.
      do ii=1,np
        call matrl(mat(ii))
        rho = rho + tp(ii) * densty
        totthk=totthk+tp(ii)
      end do
      rho = rho / totthk
      WRITE(6,465) RHO
  465 FORMAT(' Do you wish to use the default density of ',F8.1,
     &   ' kg/m↑3? (Y/N): ',$)
      READ(5,466) RHOYN
  466 FORMAT(A)
      IF(RHOYN.EQ.'N' .OR. RHOYN.EQ.'n') GOTO 471
  470 IF (EDIT) THEN
  471    WRITE(6,475)
  475    FORMAT(' Density [kg/m↑3]: ',$)
         READ(5,*) RHO
      if (edit) go to 1170
      END IF
C*********************************************************************
C*                        LOADING PARAMETERS                         *
C*********************************************************************
      write(6,*)
      write(6,*)
      write(6,*)'Loading Parameters'
      write(6,*)
 480  write(6,490)
 490  format(' ','In-plane load (X-direction) [N/m]: ',$)
      read(5,*)pnx
      if (edit) go to 1170
 500  write(6,510)
 510  format(' ','In-plane load (Y-direction) [N/m]: ',$)
      read(5,*)pny
C
C Do not allow all B.C.'s to be free
C
      if (edit) go to 1170
 520  write(6,*)
```

```
      write(6,*)'Select Boundary Conditions'
      write(6,*)'1) Simply Supported - Simply Supported'
      write(6,*)'2) Clamped - Free'
      write(6,*)'3) Clamped - Clamped'
      write(6,*)'4) Free - Free'
      write(6,*)'5) Simply Supported - Clamped'
      write(6,*)'6) Simply Supported - Free'
      write(6,*)
      if (.not.(edit) .or. nselec .eq. 3) then
      write(6,530)
530   format(' ','B.C. for X-direction [1-6]: ',$)
540   format(' ','B.C. for Y-direction [1-6]: ',$)
      read(5,*)nx
      if ((edit) .and. (nx .eq. 4) .and. (ny .eq.4)) then
        write(6,*) 'Illegal B.C."s - all boundaries are free'
        write(6,*) 'Reenter B.C.'
        go to 520
      end if
      if (nx .lt. 1 .or. nx .gt. 6) then
        write(6,*) '*** INVALID B.C. ***'
        go to 520
      end if
      end if
      if (.not.(edit) .or. nselec .eq. 4) then
550   write(6,540)
      read(5,*)ny
      if (ny .lt. 1 .or. ny .gt. 6) then
        write(6,*) '*** INVALID B.C. ***'
        go to 550
      end if
      if (ny .eq.4 .and. nx .eq. ny) then
        write(6,*) 'Illegal B.C."s - all boundaries are free'
       if (.not.(edit)) then
        write(6,*) 'Reenter X and Y direction B.C.s'
        go to 520
       else
        write(6,*) 'Reenter B.C.'
        go to 550
       end if
      end if
      end if

      if (edit) go to 1170
560   format(' ','Shear Correction Factor'/
     &       ' ','[default(0) = 5/6]: ',$)
      if (.not.(default)) then
570   write(6,560)
      read(5,*)shear
      if (shear .eq. 0.0) then
        shear=(5.0D0/6.0D0)
      end if
      else
        shear=(5.D0/6.D0)
```

```
      end if
      if (edit) go to 1170
C*********************************************************************
C*                        IMPACTOR PARAMETERS                        *
C*********************************************************************
      write(6,*)
      write(6,*)
      write(6,*)'Impactor Parameters'
      write(6,*)
 580  write(6,590)
 590  format(' ','Mass [kg]: ',$)
      read(5,*)bm
      if (.not.(bm .gt. 0.)) go to 580
      if (edit) go to 1170
 600  format(' ','Velocity [m/s]: ',$)
 610  format(' ','Diameter [mm]: ',$)
 620  write(6,600)
      read(5,*)vi
      if (edit) go to 1170
 630  write(6,610)
      read(5,*)di
      if (.not.(di .gt. 0.)) go to 630
      ri=di/2
      if (edit) go to 1170
C*********************************************************************
C*                  Other Miscellaneous Values                       *
C*********************************************************************
      write(6,*)
      write(6,*)
      write(6,*)'Other Parameters'
      write(6,*)
      if (.not.(default)) then
 670  write(6,680)
 680  format(/' # of harmonics for calculating hertzian spring ',
     &'constant'/
C     &' constant in program BESSEL. (Be sure the approach'/
C     &' converges.)
     &' ','[(0)default = 60]: ',$)
      read(5,*)nohark
      if (nohark.eq.0) then
        nohark=60
      else if (nohark .lt. o) then
        go to 670
      end if
      else
        nohark=60
      end if
      if (edit) go to 1170
      write(6,*)
      if (.not.(default)) then
 690  write(6,700)
 700  format(' ','# of harmonics for calculating stresses and strains '/
C     &' ','in program BESSEL. (Be sure the pressure distribution'/
```

```
C     &' ','converges.)
     &' ','[(0)default = 200]: ',$)
      read(5,*)nohars
      if (nohars.eq.0) then
        nohars=200
      else if (nohars .lt. 0) then
        go to 690
      end if
      else
        nohars=200
      end if
      if (edit) go to 1170
      write(6,*)
C
C Alphas and target loads are aoutomatically assigned
C
C 265  write(6,270)
C 270  format(' ','# of alphas [max 10]: ',$)
C      read(5,*)numalp
C      if (numalp .gt. 10 .or. numalp .lt. 1) go to 265
C      if (edit) go to 1170
C      write(6,275)
C 275  format(' ','Enter the alpha values [mm] and ',
C     &  'total load target values [N]:')
C      do 290 n= 1,numalp
C        write(6,280)n
C        read(5,*)alphaa(n)
C 280  format(' ','Alpha #',i2,' [mm]: ',$)
C      write(6,285)n
C 285  format(' ','Target Load #',i2,' [N]: ',$)
C      read(5,*)tqa(n)
C      write(6,*)
C 290  continue

 710  write(6,720)
C
C The possible values of LD$ are  PT, HZ, and HT but the variable
C LOAD is written to the output file due to file access considerations
C The possible values of LOAD are 1, 2, and 3
C (1 = HZ, 2 = PT, 3=HT)
C
 720  format(' ','Input loading type for stress-strain calculation:'
     &/' ','PT = Point Load'/
     & ' ','HZ = Hertzian Distribution'/' ','HT = Hat Load'/
     & ' ','[HZ suggested]: ',$)
 730  format(a2)
      read(5,730)ld$
      if (ld$ .eq .' ' .or. ld$.eq.'hz' .or. ld$.eq.'HZ') then
        ld$='HZ'
        load=1
      else if (ld$.eq.'pt' .or. ld$.eq.'PT') then
        ld$='PT'
        load=2
```

```
      else if (ld$.eq.'ht' .or. ld$.eq.'HT') then
        ld$='HT'
        load=3
      else
        go to 710
      end if
      if (edit) go to 1170
      if (ld$.eq.'HT') then
740     write(6,750)
750     format(' ','Loading radius [mm]: ',$)
        read(5,*)xep
        if (edit) go to 1170
      end if
      write(6,*)
760   write(6,770)
770   format(' ','# Radius divisions in the strain field [integer]: ',$)
      read(5,*)nrs
      if (.not.(nrs .gt. 0)) go to 760
      if (edit) go to 1170
780   tthick = 0.
      do n= 1,np
        tthick=tthick+tp(n)
790   end do
      if (.not.(int(20.*tthick/nrs) .gt. 0)) then
        tthick = nrs/20.
      end if
      if (.not.(default)) then
      write(6,800)int(20.*tthick/nrs)
800   format(' ','Length of each radius division [mm].'/
     & ' ','(Default chosen such that 20 * Thickness = Radius Analyzed)'
     &/' ','[(0) default = ',i4,' mm]: ',$)
      read(5,*)alrd
      lrd=int(alrd)
      if (lrd .lt. 0) go to 780
      if (lrd .eq. 0) then
        lrd=int(20.*tthick/nrs)
      end if
      else
        lrd=int(20.*tthick/nrs)
      end if
      if (lrd .lt. 1) lrd = 1
      r=nrs*lrd
      if (edit) go to 1170
      ldhzs='HZS'
      write(6,*)
810   format(' ','Do you want to calculate strain at each theta'/
     &' ','division, each radius step, and each ply? [y/n]: ',$)
820   write(6,810)
      read(5,830)outyp
C
C If OUTYP = 'Y' then '1' is written to the output file
C          = 'N' then '0' is written to the output file
C
```

```
830  format(a1)
     if (outyp.ne.'y' .and. outyp.ne.'Y' .and. outyp.ne.'n' .and.
    & outyp.ne.'N') then
       go to 820
     end if
     if (outyp.eq.'y' .or. outyp.eq.'Y') then
       outyp='Y'
       noutyp=1
     else
       outyp = 'N'
       noutyp=0
     end if
       if (edit) go to 1170
     if (outyp .eq. 'Y') then
840    write(6,850)
850    format(' ','# theta divisions in strain field [integer]: ',$)
       read(5,*)nth
       if (edit) go to 1170
       write(6,860)
860    format(' ','Input the matrix of maximum allowable tensile'/
    &   ' ','strains [Tensor Notation] [microstrain]:')
       do 900 n1= 1,3
               do 890 n2= n1,3
                       write(6,870)n1,n2
870                    format(' ','[',i1,',',i1,']: ',$)
                       read(5,880)ept(n1,n2)
880                    format(g12.6)
890            continue
900    continue
       write(6,*)
       write(6,910)
910    format(' ','Input matrix of maximum allowable compressive'/
    &   ' ','strains [Tensor Notation] [microstrain]:')
       do 930 n1= 1,3
               do 920 n2= n1,3
                       write(6,870)n1,n2
                       read(5,880)epc(n1,n2)
920            continue
930    continue
     else
       outyp='N'
       noutyp=0
     end if
     write(6,*)
940  write(6,950)
950  format(' ','# X - axis modes to be used in dynamic analysis',
    &' [integer]: ',$)
     read(5,*)ix
     if (.not. (ix .gt. 0)) go to 940
     if (edit) go to 1170
960  write(6,970)
970  format(' ','# Y - axis modes to be used in dynamic analysis',
    &' [integer]: ',$)
```

```
      read(5,*)iy
      if (.not.(iy .gt. 0)) go to 960
      if (edit) go to 1170
      write(6,*)
      go to 1010
980   write(6,990)
990   format(/' ','Condense rotary inertia? ("Y" suggested for ',
     &'laminates) [y/n]: ',$)
      read(5,1000)ifc$
1000  format(a1)
      if (ifc$.eq.'y' .or. ifc$.eq.'Y') then
        ifc=1
      else if (ifc$.eq.'n' .or. ifc$.eq.'N') then
        ifc=0
      else
        go to 980
      end if
      if (edit) go to 1170
      go to 1030
1010  write(6,1020)
1020  format(/' ','Do you want to eliminate the even modes from ',
     & 'the global'/' analysis? [y/n]: ',$)
      read(5,1000)glob
      if (glob .eq. 'y' .or. glob .eq. 'Y') then
        glob='Y'
        melim=1
      else if (glob .eq. 'n' .or. glob .eq. 'N') then
        glob='N'
        melim=0
      else
        go to 1010
      end if
      if (edit) go to 1170
      go to 980
1030  write(6,1040)
1040  format(/' ','Input time step length [micro sec]: ',$)
      read(5,*) delt
      if (.not.(delt .gt. 0.)) go to 1030
      if (edit) go to 1170
1050  write(6,1060)
1060 format(/' ','# of time steps in dynamic analysis [integer]:'/
     &' ',' (0 = stop when impactor velocity = 0.0 m/s)'/
     &' ',' (1 = stop when plate displacement = 0.0 m)'/
     &' ',' (# greater than 1 = # of time steps used): ',$)
      read(5,*)nt
      if (nt .lt. 0) go to 1050
      if (edit) go to 1170
      write(6,*)
      write(6,*)

1070  write(6,1080) filnam(1)
1080  format(/' ','Saving data in file: ',a20/)
```

```
 1100  format(a20)
c 515  inquire(file=filnam,exist=exst)
c      if (exst) then
c       write(6,517)filnam
c 517    format(' ','*** File [',a,'] already exists ***')
c       write(6,520)
c 520    format(' ','Filename [(same) to overwrite existing]: ',$)
c       read(5,1100)fil2
c        if (.not.(fil2 .eq. 'same')) then
c               filnam=fil2
c               go to 515
c        end if
c      end if
      if (edit) go to 1170
      stat='new'
C=================================================================
C                    ROUTINE savfil
C
C    This routine saves the current parameters in the file specified
C  by the variable FILNAM.
C=================================================================


 1110  open(unit=15, file=filnam(1),status=stat)
      write(15,1120)title1
      write(15,1120)title2
      write(15,1120)title3
 1120  format(' ',a60)
      write(15,*)np
      do n=1,np
        write(15,*)mat(n),tp(n),thp(n)
      end do
      write(15,*)a$*1.E-3,b*1.E-3
      write(15,*)rho
      write(15,*)pnx,pny
      write(15,*)nx,ny
      write(15,*)shear
      write(15,*)bm,vi
      write(15,*)ri
      ldhzs='HZS'
      write(15,*)LDHZS
      write(15,*)r
      write(15,*)nohark,nohars
      write(15,*)numalp
      write(15,*)(alphaa(n), n=1,numalp)
      write(15,*)(tqa(n), n=1,numalp)
C     write(15,1140)ld$ ------> CHANGED TO: WRITE(15,*)LOAD
      write(15,*)load
 1140  format(' ',a2)
      if (ld$(1:2).eq.'HT') then
        write(15,*)xep
      end if
      write(15,*)nrs
```

```
C     write(15,1140)outyp ----> CHANGED: Y=1, N=0
        write(15,*)noutyp
      if (outyp .eq. 'Y') then
        write(15,*)nth
        write(15,*)(ept(1,n), n=1,3)
        write(15,*)(ept(2,n), n=2,3)
        write(15,*)ept(3,3)
        write(15,*)(epc(1,n), n=1,3)
        write(15,*)(epc(2,n), n=2,3)
        write(15,*)epc(3,3)
      end if
      write(15,*)ix,iy
      write(15,*)ifc
      write(15,*)melim
      write(15,*)delt,nt
      close(unit=15,status='keep')
      go to 100
C=================================================================
C                    EDIT FILE ROUTINE
C
 1160  npage = 1
 1170 go to (1180,1340,1360,1420)npage
C=================================================================
C   Print the first page of the parameters
C
 1180 write(6,1190)jobnam,title1,title2,title3,np,a$,b,rho
 1190 format('1',///' ',t20,'(1) Edit Job: ',a6//
     &' ','Title:  (2): ',a60/
     &' ','        (3): ',a60/
     &' ','        (4): ',a60//
     &' ','(5) # plies: ',i2//
     &' ','(6) Change/Display Material Specifications'//
     &' ','Specimen Parameters:'/
     &5x,'(7) X - direction length: ',g10.4,' [mm]'/
     &5x,'(8) Y - direction length: ',g10.4,' [mm]'/
     &5x,'(9) Density: ',g10.4,' [kg/mt3]'////////
     &' ','[Enter: -1 = exit   0=next page   #=change ',
     &'item]: ',$)
      read(5,*)choice
      if (choice .lt. -1 .or. choice .gt. 9) then
        go to 1180
      else if (choice .eq. 0) then
        npage=2
        go to 1170
      else if (choice .eq. -1) then
        go to 1480
      end if
      go to(1191,210,220,230,240,1200,420,440,470)choice
C
C Enter new jobname
C
 1191 write(6,1192)
 1192 format(/' ','Enter new jobname ("quit" to exit) [max 6 ',
```

```
     &'characters]: ',$)
 1193 format(a6)
      read (5,1193) newjob
      if (.not.(newjob .eq. 'quit' .or. newjob .eq. 'QUIT')) then
        jobnam = newjob
        call setfil(jobnam)
      end if
      go to 1170
C
C        EDIT material selection
 1200 ncycle=1
      n=0
 1210 write(6,1220)
      in=n
 1220 format(//15x,'Material Specifications'//)
      write(6,1230)'Ply','Angle','Mat #','Thickness',
     &            'Ply','Angle','Mat #','Thickness'
      write(6,1230)'---','-----','-----','---------',
     &            '---','-----','-----','---------'
 1230 format(' ',a3,2x,a5,2x,a5,2x,a9,t40,a3,2x,a5,2x,a5,2x,a9)
      do ncycle=1,13
        n=n+1
        if (.not. (n+13 .gt. np)) then
                write(6,1240)n,thp(n),mat(n),tp(n),n+13,thp(n+13),
     &                       mat(n+13),tp(n+13)
        else if (.not. (n .gt. np)) then
                write(6,1250)n,thp(n),mat(n),tp(n)
        else
                write(6,*)' '
        end if
      end do
 1240 format(' ',1x,i2,2x,f5.1,4x,i2,3x,f8.4,' mm',t41,i2,2x,f5.1,4x,i2,
     &      3x,f8.4,' mm')
 1250 format(' ',1x,i2,2x,f5.1,4x,i2,3x,f8.4,' mm')
      write(6,*)' '
 1260 format(//' ','[Enter: 0=done  1=change ply info]:',$)
 1270 format(//' ','[Enter: 0=done  1=change ply info  2=Next page]:',
     & $)
        if (np .lt. 27) then
                write(6,1260)
        else
                write(6,1270)
        end if
      read(5,*)nselec
      if (nselec .lt. 0 .or. nselec .gt. 2) then
        n=in
        go to 1210
      else if (nselec .eq. 0) then
        go to 1170
      else if (nselec .eq. 2) then
        if (np .lt. 27) then
          n=in
          go to 1210
```

```
        else
          n=26
          if (in .eq. 26) n=0
          go to 1210
        end if
      end if

 1280   write(6,1290)

 1290   format(' ','Ply # [0=exit]: ',$)
 1300 read(5,*)nply
      if (nply .lt. 0 .or. ((nply .gt. np) .and.
     & (.not.(nply .eq. 51)))) then
        go to 1300
      else if (nply .eq. 0) then
        n=in
        go to 1210
      end if
      if (nply .ne. 51) then
 1310 format(' ','Material# [(0)show list]: ',$)
 1320   write(6,1310)
        read(5,*)mread
        if (mread .lt. 0 .or. mread .gt. last) go to 1320
        if (mread .eq. 0) then
                call list
                go to 1320
        end if
        mat(nply)=mread
 1330    format(' ','Thickness [(0)default=',f8.4,'mm]: ',$)
        call matrl(mat(nply))
        write(6,1330) thick
        read(5,*) temp
        if (temp .eq. 0.) then
          tp(nply)=thick
        else
          tp(nply)=temp
        end if
        go to 1280
      end if
C
C write the page 2
C
 1340 write(6,1350) jobnam,pnx,pny,nx,bc(nx),ny,bc(ny),shear,
     & bm,vi,ri*2.0
 1350  format('1'///' ',t20,'Edit Job: ',a6//
     &' ','Loading parameters:'/
     &' ',5x,'(1) In-plane load (X-direction): ',f8.4,' N/m'/
     &' ',5x,'(2) In-plane load (Y-direction): ',f8.4,' N/m'//
     &' ','Boundary conditions:'/
     &' ',5x,'(3) X-direction: [',i1,'] ',a35/
     &' ',5x,'(4) Y-direction: [',i1,'] ',a35//
     &' ','(5) Shear correction factor: ',f12.8//
     &' ','Impactor parameters:'/
```

```
     &' ',5x,'(6) Mass: ',f12.8,' kg'/
     &' ',5x,'(7) Velocity: ',f8.4,' m/s'/
     &' ',5x,'(8) Diameter: ',f8.4,' mm'//////
     &' ','[Enter: -1 = exit   0 = next page  # = ',
     &'change item]:',$)
      read(5,*)nselec
      if (nselec .lt. -1 .or. nselec .gt. 8) go to 1340
      if (nselec .eq. 0) then
        npage=3
        go to 1170
      else if (nselec .eq. -1) then
        go to 1480
      end if
      go to (480,500,520,520,570,580,620,630)nselec
C
C write page 3
C
 1360  write(6,1370) jobnam,nohark,nohars
 1370  format('1'///' ',t20,'Edit Job: ',a6//
     &' ','(1) # harmonics to calculate Hertzian spring constant: ',i4/
     &' ','(2) # harmonics to calculate stress and strains: ',i4/)
C     &' ','(3) # alphas: ',i2/
C     &' ','    Alpha [mm]',t35,'Target load [N]'/
C     &' ','   ------------',t35,' ---------------')
C 820  format(' ','(',i2,') ',f8.4,t35,'(',i2,') ',f8.4)
C      do ii=1,numalp
C       j=ii+numalp+3
C       write(6,820)ii+3,alphaa(ii),j,tqa(ii)
C      end do
C      write(6,*)
      j=3
      write(6,1380) 3, ld$
 1380  format(' ','(',i1,') Loading type: ',a2)
      if (ld$ .eq. 'HT') then
        j=j+1
        write(6,1390) j, xep
      end if
 1390  format(' ','(',i1,') Loading radius: ',f8.4,' mm')
      write(6,1400) j+1,nrs,j+2,lrd
 1400  format(' ','(',i1,') # radius divisions: ',i4/
     &' ','(',i1,') Length of each radius division: ',i4,' mm')
C      if (numalp .lt. 10) then
        do i=1,10
        write(6,*)
        end do
C      end if
      write(6,1410)
 1410  format(//' ','[Enter: -1 = exit   0 = next page   ',
     &'# = change item]: ',$)
      read(5,*) nselec
      if (nselec .lt. -1 .or. nselec .gt. j+3) go to 1170
      if (nselec .eq. 0) then
         npage = 4
```

```
         go to 1170
      else if (nselec .eq. -1) then
         go to 1480
      end if
      if (nselec .eq. 1) go to 670
      if (nselec .eq. 2) go to 690
C      if (nselec .eq. 3) go to 265
      if (nselec .eq. 3) go to 710
      if (nselec .eq. j+1) go to 760
      if (nselec .eq. j+2) go to 780
C      if (nselec .gt.3 .and. nselec .lt. 4+numalp) then
C 860    format(/' ','Enter Alpha #',i2,' [mm]: ',$)
C        write(6,860) nselec-3
C        read(5,*) alphaa(nselec-3)
C      else if (nselec .lt. 4+2*numalp) then
C 870    format(/' ','Enter target load for alpha #',i2,' [N]: ',$)
C        write(6,870) nselec-3-numalp
C        read(5,*) tqa(nselec-3-numalp)
C      else
        go to 740
C      end if
C      go to 1170
C
C write page 4
C
 1420  write(6,1430) jobnam,outyp
 1430  format('1'///' ',t20,'Edit Job: ',a6//
     &' ','(1) Calculate strain at each theta division, radius ',
     & 'step, and ply: ',a1)
      if (outyp .eq. 'Y') then
        write(6,1440) nth,ept(1,1),ept(1,2),ept(1,3),ept(2,2),
     &          ept(2,3),ept(3,3)
        write(6,1450) epc(1,1),epc(1,2),epc(1,3),epc(2,2),
     &          epc(2,3),epc(3,3)
 1440   format(' ','(2) #theta divisions: ',i2//
     &  ' ','Matrix of maximum tensile strains [tensor ',
     &  'notation] [microstrain]:'/
     &  ' ',t5,'(3) ',f8.4,t20,'(4) ',f8.4,t35,'(5) ',f8.4/
     &  ' ',t20,'(6) ',f8.4,t35,'(7) ',f8.4/
     &  ' ',t35,'(8) ',f8.4)
 1450    format(' ',
     &  'Matrix of maximum compressive strains [tensor ',
     &  'notation] [microstrain]:'/
     &  ' ',t5,'(9) ',f8.4,t19,'(10) ',f8.4,t34,'(11) ',f8.4/
     &  ' ',t19,'(12) ',f8.4,t34,'(13) ',f8.4/
     &  ' ',t34,'(14) ',f8.4/)
        write (6,1460) 15,ix,16,iy,17,que(ifc),18,que(melim),19,delt,
     &  20,nt
        write (6,1470)
 1460    format(' ','('i2') # X-axis modes to be used by IMPACT: ',i4/
     &  ' ','(',i2,') # Y-axis modes to be used by IMPACT: ',i4//
     &  ' ','(',i2,') Condense rotary inertia: ',a3/
     &  ' ','(',i2,') Eliminate even modes from global analysis: ',a3//
```

```
    &  ' ','(',i2,') Time step length: ',f8.4,' micro_s'/
    &  ' ','(',i2,') # time steps: ',i4/)
1470  format(' ','[Enter: -1 = exit/save  0 = next page  # = change]',
    &   ': ',$)
      read(5,*) nselec
      if (nselec .lt. -1 .or. nselec .gt. 20) then
              go to 1420
      else if (nselec .eq. 0) then
              npage = 1
              go to 1170
      else if (nselec .eq. -1) then
1480          write (6,1490) jobnam
1490          format(//' Exit editing mode:'//
    &         ' ','     (1) Exit without saving changes'/
    &         ' ','     (2) Exit and save changes under jobname'/
    &         ' ','     (3) Change Jobname [',a6,']'/
    &         ' ','     (4) Continue edit mode'/)
              read(5,*) nexit
              if (nexit .lt. 1 .or. nexit .gt. 3) then
                      go to 1170
              else if (nexit .eq. 1) then
                      write(6,1500)
1500                  format(' ','Are you sure? (y/n): ',$)
                      read(5,1720) sure
                      if (sure .eq. 'y' .or. sure .eq. 'Y') then
                              go to 100
                      end if
                      go to 1480
              else if (nexit .eq. 3) then
1505                  format(' ','Enter new jobname [max 6 ',
    &                 'characters]("quit" to exit): ',$)
                      write(6,1505)
                      read(5,197) newjob
                      if (newjob .ne. 'quit' .or. newjob .eq.
    &                         'QUIT') then
                              jobnam = newjob
                              call setfil(jobnam)
                      end if
                      go to 1480
              end if
              inquire(file=filnam(1),exist=exst)
              if (exst) then
                stat='new'
              else
                stat='new'
              end if
              write(6,*) 'Saving Parameters'
              go to 1110
      end if
      go to (820,840,1520,1530,1540,1550,1560,1570,1580,1590,1600,
    &  1610,1620,1630,940,960,980,1010,1030,1050)nselec
1510  format(' ','Enter EP',a1,'_MAX [',i1,',',i1,']: ',$)
1520  write(6,1510)'T',1,1
```

```
        read(5,*) ept(1,1)
        go to 1170
1530    write(6,1510)'T',1,2
        read(5,*) ept(1,2)
        go to 1170
1540    write(6,1510) 'T',1,3
        read(5,*) ept(1,3)
        go to 1170
1550    write(6,1510) 'T',2,2
        read(5,*) ept(2,2)
        go to 1170
1560    write(6,1510) 'T',2,3
        read(5,*) ept(2,3)
        go to 1170
1570    write(6,1510) 'T',3,3
        read(5,*) ept(3,3)
        goto 1170
1580    write(6,1510) 'C',1,1
        read(5,*) epc(1,1)
        go to 1170
1590    write(6,1510) 'C',1,2
        read(5,*) epc(1,2)
        go to 1170
1600    write(6,1510) 'C',1,3
        read(5,*) epc(1,3)
        go to 1170
1610    write(6,1510) 'C',2,2
        read(5,*) epc(2,2)
        go to 1170
1620    write(6,1510) 'C',2,3
        read(5,*) epc(2,3)
        go to 1170
1630    write(6,1510) 'C',3,3
        read(5,*) epc(3,3)
        go to 1170
      end if
      write(6,*)
      write(6,1460) 2,ix,3,iy,4,que(ifc),5,que(melim),6,delt,7,nt
      do iii=1,8
        write(6,*)
      end do
      write (6,1470)
      read(5,*) nselec
      if (nselec .lt. -1 .or. nselec .gt. 7) then
        go to 1420
      else if (nselec .eq. -1) then
        go to 1480
      else if (nselec .eq. 0) then
        npage =1
        go to 1170
      end if
      go to (820,940,960,980,1010,1030,1050)nselec
```

```
C=================================================================
C                    Routine GETFILE
C
C This subroutine inputs the information from a sequential file to be
C edited by the  edit mode.
C=================================================================
 1700   open(unit=15, file=filnam(1), status='old')
        write(6,1705) filnam(1)
 1705   format(/' ','Loading data from ',a20/)
 1710   format(a80)
 1720   format(a)
        read(15,1710) title
        ntitl=1
 1721   if (title(ntitl:ntitl) .eq. ' ') then
                ntitl=ntitl+1
                if (ntitl .gt. 71) go to 1722
                go to 1721
        end if
 1722   if (ntitl .lt. 71) then
        do nxfer= ntitl,ntitl+59
                n=nxfer-ntitl+1
                title1(n:n)=title(nxfer:nxfer)
        end do
        else
                title1=' '
        end if
        read(15,1710) title
        ntitl=1
 1723   if (title(ntitl:ntitl) .eq. ' ') then
                ntitl=ntitl+1
                if (ntitl .gt. 71) go to 1724
                go to 1723
        end if
 1724   if (ntitl .lt. 71) then
        do nxfer=ntitl,ntitl+59
                n=nxfer-ntitl+1
                title2(n:n)=title(nxfer:nxfer)
        end do
        else
                title2=' '
        end if
        read(15,1710) title
        ntitl=1
 1725   if (title(ntitl:ntitl) .eq. ' ') then
                ntitl=ntitl+1
                if (ntitl .gt. 71) go to 1726
                go to 1725
        end if
 1726   if (ntitl .lt. 71) then
        do nxfer=ntitl,ntitl+59
                n=nxfer-ntitl+1
                title3(n:n)=title(nxfer:nxfer)
        end do
```

```
      else
              title3=' '
      end if
      read(15,*) np
      do i=1,np
              read(15,*) mat(i), tp(i), thp(i)
      end do
      read(15,*) a$, b
C     convert m to mm
      a$=a$*1.E3
      b=b*1.E3
      read(15,*) rho
      read(15,*) pnx, pny
      read(15,*) nx, ny
      read(15,*) shear
      read(15,*) bm, vi
      read(15,*) ri
      read(15,1720) idhzs
      read(15,*) r
      read(15,*) nohark, nohars
      read(15,*) numalp
      read(15,*) (alphaa(i), i=1,numalp)
      read(15,*) (tqa(i), i=1,numalp)
      read(15,*) load
      if (load .eq. 1) then
              ld$='HZ'
      else if (load .eq. 2) then
              ld$='PT'
      else
              ld$='HT'
              read(15,*) xep
      end if
      read(15,*) nrs
      ird=int(r/nrs)
      read(15,*) noutyp
      if (noutyp .eq. 0) then
              outyp = 'N'
      else
              outyp = 'Y'
              read(15,*) nth
              read(15,*) (ept(1,n),n=1,3)
              read(15,*) (ept(2,n),n=2,3)
              read(15,*) ept(3,3)
              read(15,*) (epc(1,n),n=1,3)
              read(15,*) (epc(2,n),n=2,3)
              read(15,*) epc(3,3)
      end if
      read(15,*) ix, iy
      read(15,*) ifc
      read(15,*) melim
      read(15,*) delt, nt
      close(unit=15)
      go to 1160
```

```
C****&*****************************************************************

      end
C=====================================================================
C                          END of main code
C=====================================================================
```

```
C*********************************************************************
C*                        SUBROUTINE MATRL                           *
C*                                                                   *
C*  This routine gets the nth material from the file LIBRARY.BIN     *
C*********************************************************************
      subroutine matrl(l)

      implicit real*8 (a-h,o-z)
      integer l
      character name*40

      common /block2/ E11,E22,E33,G23,G31,G12,ENU12,ENU13,
     &                ENU23,name,thick,densty
      read(7,rec=l) E11,E22,E33,G23,G31,G12,ENU12,ENU13,ENU23,name,
     &              thick,densty
      end


C*********************************************************************
C*                       SUBROUTINE LIST                             *
C*                                                                   *
C*  This subroutine prints the material number, material name, and   *
C*  default thickness, and material properties of each material stored *
C*  in LIBRARY.BIN .                                                 *
C*********************************************************************
      subroutine list

      implicit real*8 (a-h,o-z)
      integer matn, count
      character qq,name*40,nn
      common /block2/ E11,E22,E33,G31,G32,G12,ENU12,ENU23,
     &                ENU13,name,thick,densty
      common /block3/ last

      n=1
 560  format('1',///,' ',t15,'MATERIAL LIBRARY'///)
 565  write(6,560)
      ncycle=1
      nstart=n
      write(6,566)
 566  format(' ',' #   Material Name',29x,'Thickness [mm]')
      write(6,567)
 567  format(' ','---  ',40(1h-),'  -------------')
      do while (ncycle .lt. 16)
 568    read(7,rec=n)e11,e22,e33,g31,g23,g12,enu12,enu13,enu23,
     &               name,thick
        if (thick .gt. 0.) then
                write(6,569)n,name,thick
 569            format(' ',' ',i2,'  ',a40,'     ',g10.4)
                ncycle=ncycle+1
        else if (thick .eq. -1.) then
                do n2=ncycle,15
                        write(6,*)' '
```

```
                end do
                go to 573
          end if
          n=n+1
        end do
 573    write(6,*)' '
        write(6,575)
 575    format(' ','[Enter "0" to Exit or material # to View ',
       &'Properties]: ',$)
        read(5,*)matn
        if (matn.eq.0) then
          return
        else if (.not.(matn .gt. 0 .or. matn .lt. last)) then
          n=nstart
          go to 565
        end if
*********************************************************************
*         This section displays the material properties stored in   *
*  LIBRARY.BIN                                                      *
*                                                                   *
*********************************************************************
        n=nstart
        read(7,rec=matn)e11,e22,e33,g31,g23,g12,enu12,enu13,enu23,
       &                name,thick
        write(6,580)matn,name,thick,e11,g23,e22,g31,e33,g12,
       & enu12,enu13,enu23
 580    format(1h1,///' ',15x,'Material #',i2,//
       &'   Name: ',a40,/
       &'   Thickness: ',g10.4,'[mm]'//
       &'   E11: ',g12.6,'[GPa]',5x,'G23: ',g12.6,'[GPa]'/
       &'   E22: ',g12.6,'[GPa]',5x,'G31: ',g12.6,'[Gpa]'/
       &'   E33: ',g12.6,'[GPa]',5x,'G12: ',g12.6,'[GPa]'//
       &'   NU12: ',g12.6/
       &'   NU13: ',g12.6/
       &'   NU23: ',g12.6////)
        write(6,590)
 585    format(a1)
 590    format(' ','[Hit <RETURN> when done]: ',$)
        read(5,585)nn
        go to 565
        end
C************************* END OF LIST ****************************

C*********************************************************************
C*                          SUBROUTINE LAYUP                         *
C*                                                                   *
C*  This routine takes the layup string argument and containing      *
C*  ply orientation and sorts it into the angle array THP(NP).       *
C*                                                                   *
C*  The character string LAY is in the form:                         *
C*     [m1 A*a/B*b m2 C*c m1 D*d/E*e/F*f ...]s                       *
C*  where:                                                           *
C*     m1,m2,m3,...  temporary mateterial number (m1 = material 1)   *
```

```
C*      A,B,C,...      ply angles in degrees                          *
C*      *a,*b,*c,...   number of times to repeat a given angle        *
C*                     (not needed for a,b,c,... = 1)                 *
C*          /          separates ply angles for a given material      *
C*          ]          denotes end of layup                           *
C*          s          indicates a symmetric layup (optional)         *
C*                                                                    *
C*  example:  [m1 45*2/-45*2 m2 0]s                                   *
C*  result:                                                           *
C*  ply #      material   angle                                       *
C*   1           m1         45 \___ 45*2                              *
C*   2           m1         45 /                                      *
C*   3           m1        -45 \___ -45*2                             *
C*   4           m1        -45 /                                      *
c*   5           m2          0 ________ line of symmetry (s option)   *
C*   6           m2          0                                        *
C*   7           m1        -45                                        *
C*   8           m1        -45                                        *
C*   9           m1         45                                        *
C*  10           m1         45                                        *
C*********************************************************************
      subroutine layup(setup)

      implicit real*8 (a-h,o-z)
      character setup*80,symb,symb2,symb3,name*40
      integer nsymb,count1,count2,ply,nlt,ngt
      logical single

      common /block1/ mat(50),thp(50),tp(50),n,nflag,nflag2,nflag3,ply
      common /block2/ E11,E22,E33,G23,G31,G12,ENU12,ENU13,
     &                ENU23,name,thick,densty
      common /block4/ nm,matf(50)


C
C     VARIABLE DECLARATIONS
C
C     SETUP                  layup string which is being searched
C     SYMB, SYMB2, SYMB3     contain single characters from SETUP string
C     NSYMB                  length of SETUP
C     COUNT1,COUNT2          generic counters
C     PLY                    ply # being operated on
C     NLT/NGT                (number less than)/(number greater than)
C                            used in inequalities to see if character
C                            in question is a number
C     NFLAG2                 Indicator of the end of the layup; 1=true
C     NFLAG3                 Error indicator in layup; 1=true


      nlt=58
      ngt=47
      count1=1
      ply=0
```

```
      nsymb=len(setup)
      nm=0
      single = .false.
      nflag2=0
      nflag3=0
C*********************************************************************
C*         Do until we reach the end of the string "setup"           *
C*********************************************************************

      do while (count1.lt.nsymb+1)
           symb=setup(count1:count1)

C********** Check for change of material # ****************************

           if (symb.eq.'m' .or. symb.eq.'M') then
                ply=ply+1
                if (ichar(setup(count1+2:count1+2)).gt.ngt .and.
     &             ichar(setup(count1+2:count1+2)).lt.nlt) then
                   mat(ply)=-(10*(ichar(setup(count1+1:count1+1))-48)+
     &                 ichar(setup(count1+2:count1+2))-48)
                   add = 3
                else
                   mat(ply)=-ichar(setup(count1+1:count1+1))+48
                   add = 2
                end if
C=================================================
C See if we have a new temporary material number:
C If so, increment NM (Number of Materials).

                        ncnt=1
                        do while (ncnt .lt. ply .and.
     &                            mat(ncnt) .ne. mat(ply))
                        ncnt=ncnt+1
                        end do
                        if (ncnt .eq. ply) then
                                nm=nm+1
                                matf(nm)=mat(ply)
                        end if
C*********** Only 1 material in layup *********************************
            else if (symb .ne. 'm' .and. symb .ne. 'M' .and. count1
     &                .eq. 1 .and. (.not. (single))) then
                ply = ply +1
                single = .true.
                matf(1) = -1
                mat(1) = -1
                nm = 1
                go to 999
C*********** Check for a blankspace ***********************************

            else if (symb.eq.' ') then
                add=1
C*********** Check for a plus sign ************************************
            else if (symb .eq. '+') then
```

```
                add = 1
C*********** Check for a change of angle in a given material ***********

            else if (symb.eq.'/') then
                ply=ply+1
                mat(ply)=mat(ply-1)
                add = 1

C*********** Check for a new ply angle (+/-) ***************************

            else if (symb.eq.'-' .or. (ichar(symb).gt.ngt .and.
     &              ichar(symb).lt.nlt)) then
                if (ichar(setup(count1+1:count1+1)).gt.ngt .and.
     &              ichar(setup(count1+1:count1+1)).lt.nlt) then
                    if (ichar(setup(count1+2:count1+2)).gt.ngt .and.
     &                  ichar(setup(count1+2:count1+2)).lt.nlt) then

C*********** Ply angle > 99 ********************************************

                                if (symb.ne.'-') then
                                   thp(ply)=100*(ichar(symb)-48) +
     &                             10*(ichar(setup(count1+1:count1+1))-
     &                                                          48)+
     &                             ichar(setup(count1+2:count1+2))-48
                                   add = 3

C*********** Ply angle < -99 *******************************************

                                else if (ichar(setup(count1+3:count1+3))
     &                             .gt.ngt .and. ichar(setup(count1+3:
     &                             count1+3)).lt.nlt) then
                                   thp(ply)=-(100*(ichar(setup(count1+1:
     &                             count1+1))-48) + 10*(ichar(setup(
     &                             count1+2:count1+2))-48) + ichar(
     &                             setup(count1+3:count1+3)) - 48)
                                   add = 4

C*********** -9 > ply angle > -100 *************************************

                                else
                                   thp(ply)=-(10*(ichar(setup(count1+1:
     &                             count1+1))-48) + ichar(setup(
     &                             count1+2:count1+2)) - 48)
                                   add = 3
                                end if

C*********** 100> ply angle > 9 ****************************************

                        else if (symb.ne.'-') then
                                thp(ply)=10*(ichar(symb)-48) + ichar(
     &                          setup(count1+1:count1+1)) -48
                                add = 2
```

```
C*********** 0 > ply angle > -10 ***************************************

                        else
                                thp(ply)=-(ichar(setup(count1+1:
     &                          count1+1))-48)
                                add = 2
                        end if

C*********** 10 > ply angle > -1 ***************************************

                else
                        thp(ply)=ichar(symb)-48
                        add = 1
                end if

C*********** Check for repetition of ply angle and material ************

            else if (symb.eq.'*') then
                symb2=setup(count1+1:count1+1)
                symb3=setup(count1+2:count1+2)
                if (ichar(symb3).gt.ngt .and. ichar(symb3).lt.nlt) then
                        rep=10*(ichar(symb2)-48) + ichar(symb3) - 48
                        add = 3
                else
                        rep=ichar(symb2)-48
                        add = 2
                end if
                count2=1
                do while (count2.lt.rep)
                        count2=count2+1
                        ply=ply+1
                        mat(ply)=mat(ply-1)
                        thp(ply)=thp(ply-1)
                end do

C*********** Check for end of layup ********************************

            else if (symb.eq.']') then
                if (ichar(setup(count1+1:count1+1)) .gt. ngt .and.
     &              ichar(setup(count1+1:count1+1)) .lt. nlt) then
                        nrept=ichar(setup(count1+1:count1+1))-48
        if (ichar(setup(count1+2:count1+2)) .gt. ngt .and.
     &      ichar(setup(count1+2:count1+2)) .lt. nlt) then
                        nrept=nrept*10+ichar(setup(count1+2:count1+2))
     &                          -48
        end if
                        do nreptd=1,nrept-1
                        do kount=1,ply
                                mat(nreptd*ply+kount)=mat(kount)
                                thp(nreptd*ply+kount)=thp(kount)
                        end do
                        end do
                        add =2
```

```
                ply=ply*nrept
            else
                add = 1
            end if
            nflag2=1

C*********** Check for symmetry option ******************************

        else if ((symb.eq.'s' .or. symb.eq.'S') .and. nflag2.eq.1)
     &    then
            count1 = 2*ply
            do while (ply.lt.count1)
                ply=ply+1
                mat(ply)=mat(count1-ply+1)
                thp(ply)=thp(count1-ply+1)
            end do
            count1=nsymb+1

C*************** ILLEGAL CHARACTER ************************************
        else
            nflag3=1
            nflag2=1
            count1=nsymb+1
        end if

C******************** Increment to get next symbol *******************
        count1=count1+add
 999  end do
      if (nflag2 .eq. 0) then
        nflag3=1
      end if
C*********************************************************************
C*                          END OF DO LOOP                           *
C*********************************************************************

      end

C********************** END OF SUBROUTINE LAYUP ************************

C*********************************************************************
C                          SUBROUTINE SORT                            *
C
C  This routine sorts the string of material numbers, MATF(N), so the
C  Global material numbers can be inputted in sequence.  A traditional
C  BUBBLE SORT is used due to the short list.
C

      subroutine sort

      common /block4/ n,nums(50)

C       VARIABLES:
C
```

```
C     N      = number of materials
C   NUMS(N) = array of material numbers to sort
C     L      = sort completion indicator (0=true)
C     M      = temporary storage for swap

      l=1

 20   if (l .eq. 1) then
        l=0
        do j=2,n
                if (-nums(j) .lt. -nums(j-1)) then
                        m=nums(j)
                        nums(j)=nums(j-1)
                        nums(j-1)=m
                        l=1
                end if
        end do
        go to 20
      end if
      end

C=================================================================
C            Subroutine VALUES
C
C This routine prints all the default values to the screen and
C asks if they are to be used.
C
      subroutine values

      implicit real*8 (a-h,o-z)
      logical default
      character input

      common /block5/ default

    1 format(a)
    5 write (6,10)
   10 format('1',////20x'Default Values'//
     &' ','- # harmonics to calculate hertzian spring constant = 60'/
     &' ','- # harmonics to calculate stresses and strains = 200'/
     &' ','- Length of each radius division in the strain field is'/
     &' ','  an integer number of millimeters chosen such that:'/
     &' ','        20 * layup thickness = radius analyzed'/
     &' ','- Shear correction factor = 5/6'/
     &' ','- Material default thicknesses are used'//////
     &' ','Use these defaults? (y/n): ',$)
      read(5,1) input
      if (input .eq. 'y' .or. input .eq. 'Y') then
        default = .true.
      else if (input .eq. 'n' .or. input .eq. 'N') then
        default = .false.
        write(6,15)
   15 format(/' ','*** Enter 0 to select default values if one is ',
```

```
     &'given  ***'/)
      else
        go to 5
      end if
      end
C-----&------------------------------------------------------------
```

```
C=====================================================================
C            Subroutine  DIRACC                                  3/31/89
C
C  This routine allows the user to edit selected parts of the binary
C  file from ROTATE.  The editing is done to allow for minor changes in
C  the input to the program GLOBAL without having to create a new file
C  and push it through the programs ROTATE and BESSEL.
C
C  The following parameters can be changed via this subroutine:
C      PARAMETER                          VARIABLE    RECORD #
C      ___________________________________ __________  ________
C      Title Blocks                        title1        1
C                                          title2        2
C                                          title3        3
C      Specimen dimensions                   a$         10
C                                            b          10
C      In-plane material loads              pnx         12
C                                           pny         12
C      Boundary conditions                   nx         13
C                                            ny         13
C      Shear correction factor             shear        14
C      Impactor mass                         bm         15
C      Impactor velocity                     vi         15
C             <not used>                                 18
C      # harmonics to calculate strains    nohars       20
C      Loading type (stress-strain)         ld$         23
C      Loading radius (for ld$=HT)          xep        (23)
C      # radius divisions in strain field   nrs         24
C      Calculate strain field at each
C        ply, theta, and radius            outyp        25
C      # theta divisions                    nth        (25)
C      Max tensile strains (matrix)       ept_max      (25)
C      Max compressive strins (matrix)    epc_max      (25)
C      # of modes used by IMPACT             ix         26
C                                            iy         26
C      Option to condense rotary inertia    ifc         27
C      Option to eliminate even modes      melim        27
C      Time step length                     delt        28
C      Number of time steps                  nt         28
C
C  OTHER VARIABLES:
C
C      SAVED        Flag.  SAVED = .TRUE. if the current parameters
C                   have been saved in BINFIL
C      STATE        Flag.  If STATE=EXIT then the program will go to
C                   the main menu after saving changes in BINFIL.
C      QUE          QUE(1) = Yes; QUE(0) = No
C      BC(N)        Contains the boundary condition types
C      BINFIL       Binary filename being edited
C      LOAD(N)      Contains loading type description
C
      subroutine diracc
```

```
      implicit real*8 (a-h,o-z)
      character*60  title1,title2,title3
      character*35  bc(6)
      character*20  filnam, temfil
      character*14  load(3)
      character*6  jobnam, newjob
      character*3   que(0:1),temp$,ldhzs,stat
      character*2   ld$
      character*4   state
      character*1   outyp,choice

      dimension     ept_max(3,3), epc_max(3,3), alphaa(10), tqa(10),
     &                  a(2,2)

      logical exst,saved

      common /block7/ null, jobnam
      common /block8/ filnam(15)

C------- Get filename and see if it exists in the directory -------

 20   format(/' ','Enter jobname ("quit" to exit) [max 6 characters]',
     &': ',$)
      npg=1
 30   format(a6)
      if (jobnam .eq. '------') then
      write(6,20)
      read(5,30) newjob
      if (newjob .eq. 'quit' .or. newjob .eq. 'QUIT') return
      call setfil(newjob)
      jobnam=newjob
      end if
 45   inquire(file=filnam(2), err=50, exist=exst)
      if (.not.(exst)) then
 50      write(6,*)'Unable to load intermediate job file.'
        write(6,*)'Missing file: ',filnam(2)
        write(6,55)
 55     format(' Enter the complete name of the intermediate file[',
     &'Max 20 characters]'/' ','[OR enter "quit" to exit]: ',$)
 57     format(a20)
        read(5,57) temfil
        if (temfil .eq. 'quit' .or. temfil .eq. 'QUIT') return
        filnam(2)=temfil
        go to 45
      end if


      open(unit=16,file=filnam(2),status='old',access='direct',
     & recl=110)
C------- Initialize arrays ----------------------------------------
      bc(1)='Simply Supported - Simply Supported'
      bc(2)='Clamped - Free'
      bc(3)='Clamped - Clamped'
```

```
      bc(4)='Free - Free'
      bc(5)='Simply Supported - Clamped'
      bc(6)='Simply Supported - Free'

      que(0)='No'
      que(1)='Yes'

      load(1)='[PT] Point'
      load(2)='[HZ] Hertzian'
      load(3)='[HT] Hat'

      saved=.true.
C==========Get all the necessary parameters==========================
   60 read(16,rec=1)title1
      read(16,rec=2)title2
      read(16,rec=3)title3
      read(16,rec=4) np
      read(16,rec=5) thick
      read(16,rec=6)a(1,1),a(1,2),a(2,2)
      read(16,rec=7) e11,e1z,enu12,enu13,g13
      read(16,rec=8) e22,e2z,enu21,enu23,g23
      read(16,rec=9) d11,d12,d22,d66,d16,d26
      read(16,rec=10)a$,b
C    Convert from m to mm
        a$=a$*1.E3
        b=b*1.E3
      read(16,rec=11) rho
      read(16,rec=12)pnx,pny
      read(16,rec=13)nx,ny
      read(16,rec=14)shear
      read(16,rec=15)bm,vi
      read(16,rec=16) ri
      read(16,rec=17) ldhzs
      read(16,rec=19) r
      read(16,rec=20)nohars
      read(16,rec=21) numalp,(alphaa(i),i=1,numalp)
      read(16,rec=22) (tqa(i),i=1,numalp)
      read(16,rec=23)temp$,xep
      ld$=temp$(1:2)
      if (temp$(1:1) .eq. ' ') ld$=temp$(2:3)
      if (ld$ .eq. 'PT') then
        ld=1
      else if (ld$ .eq. 'HZ') then
        ld =2
      else
        ld =3
      end if
      read(16,rec=24)nrs
      read(16,rec=25)outyp
      noutyp=0
      if (outyp .eq. 'Y') then
        noutyp=1
        read(16,rec=25) outyp, nth, ((ept_max(i,j),j=i,3),i=1,3),
```

```
  &      ((epc_max(i,j),j=1,3),i=1,3)
   end if
   read(16,rec=26)ix,iy
   read(16,rec=27)ifc,melim
   read(16,rec=28)delt,nt
   read(16,rec=29)hk
   read(16,rec=30) ldd,acc
   close(unit=16)
   npg=1
   go to 90
 70 saved=.false.

 80 go to(90,210,400) npg
 90 write(6,100)jobnam,title1,title2,title3,a$,b,pnx,pny,nx,bc(nx),ny,
  & bc(ny),shear
100 format('1',///,t20,'Edit ',a6///
  &  ' ','Title Block:  (1) ',a60/
  &  ' ','              (2) ',a60/
  &  ' ','              (3) ',a60//
  &  ' ','Specimen:  (4) X-length: ',f8.4,' mm  (5) Y-length: ',
  &  f8.4,' mm'//
  &  ' ','Loads:        (6) X-dir: ',f8.4,'N/m    (7) Y-dir: ',
  &  f8.4,'N/m'//
  &  ' ','Boundary Conditions:  (8) X-dir: [',i1,'] ',a35/
  &  ' ','                      (9) Y-dir: [',i1,'] ',a35//
  &  ' ','(10) Shear correction factor: ',f8.4////)
110 format(' ','[Enter: -1=exit  0=nextpage  #=change]:'$)
   write(6,110)
   read(5,*)nselec
   if (nselec .lt. -1 .or. nselec .gt. 10) then
     go to 90
   else if (nselec .eq. -1) then
     go to 360
   else if (nselec .eq. 0) then
     npg=2
     go to 80
   else if (nselec .eq. 1) then
     write(6,120)nselec
     write(6,130)nselec
120  format(' ','Enter new title line'/' ',i1,':"',60x,'"')
130  format('+',i1,':"',$)
     read(5,150)title1
     go to 70
140 format(a2)
150 format(a60)
   else if (nselec .eq. 2) then
     write(6,120)nselec
     write(6,130)nselec
     read(5,150)title2
     go to 70
   else if (nselec .eq. 3) then
     write(6,120)nselec
     write(6,130)nselec
```

```
      read(5,150)title3
      go to 70
    else if (nselec .eq. 4) then
      write(6,160)'X'
160   format(' ',a1,'-direction length [mm]: ',$)
      read(5,*)a$
      go to 70
    else if (nselec .eq. 5) then
      write(6,160)'Y'
      read(5,*)b
      go to 70
    else if(nselec .eq. 6) then
170   format(' ',a1,'-direction load [N/m]: ',$)
      write(6,170)'X'
      read(5,*)pnx
      go to 70
    else if (nselec .eq. 7) then
      write(6,170)'Y'
      read(5,*)pny
      go to 70
    else if (nselec .eq. 8 .or. nselec .eq. 9) then
      write(6,*)'Select new boundary condition:'
      do i=1,6
              write(6,180)i,bc(i)
      end do
180 format(' ',5x,'[',i1,'] ',a35)
      if (nselec .eq. 8) then
              write(6,190)'X'
              read(5,*)nx
      else
              write(6,190)'Y'
              read(5,*)ny
      end if
      go to 70
190 format(' ',a1'-direction B.C. : ',$)
    else if (nselec .eq. 10) then
200   format(' ','Shear correction factor: ',$)
      write(6,200)
      read(5,*)shear
      go to 70
    end if
210 write(6,220) jobnam,bm,vi,nohars,load(ld)
    if (ld$ .eq. 'HT') then
      write(6,230) xep
      write(6,240) 6,nrs,7,que(noutyp),8,ix,9,iy,10,que(ifc),
   &            11,que(melim),12,delt,13,nt
      ns=6
    else
      ns=5
      write(6,240) 5,nrs,6,que(noutyp),7,ix,8,iy,9,que(ifc),
   &            10,que(melim),11,delt,12,nt
    end if
220 format('1',///,t20,'Edit ',a6///
```

```
   & ' ','Impactor: ( 1) mass: ',f8.4,' kg   ( 2) velocity: ',f8.4,
   & ' m/s'//
   & ' ','( 3) # harmonics to calculate strain field: ',i4/
   & ' ','( 4) Loading type for stress-strain calculation: ',a14)
230 format(' ','( 5) Loading radius: ',f8.4,' mm')
240 format(' ','(',i2,') # radius divisions in strain field: ',i3/
   & ' ','(',i2,') Calculate strain at each radius, theta, and ply: ',
   & a3//
   & ' ','# modes in dynamics analysis:  (',i2,') X: ',i3,
   & '     (',i2,') Y: ',i3//
   & ' ','(',i2,') Condense rotary inertia: ',a3//
   & ' ','(',i2,') Eliminate even modes from dynamic analysis: ',a3//
   & ' ','(',i2,') Time step length: ',f8.4,' micro_s'/
   & ' ','(',i2,') Number of time steps: ',i6////)
    write(6,110)
     read(5,*)nselec
250 if (nselec .eq. 0 .and. noutyp .eq. 1) then
      npg=3
      go to 80
    else if(nselec .eq. 0) then
      npg=1
      go to 80
    else if (nselec .eq. -1) then
      go to 360
    else if (nselec .lt. 1 .or. nselec .gt. ns+7) then
      go to 80
    else if (nselec .eq. 1) then
260 format(' ','Impactor mass [kg]: ',$)
      write(6,260)
      read(5,*)bm
      go to 70
    else if (nselec .eq. 2) then
270 format(' ','Impactor velocity [m/s]: ',$)
      write(6,270)
      read(5,*)vi
      go to 70
    else if (nselec .eq. 3) then
280 format(' ','Enter # of harmonics to calculate ',
   &'strain field: ',$)
      write(6,280)
      read(5,*) nohars
      go to 70
    else if(nselec .eq. 4) then
290   format(' ','Select loading type:'//
   &' ','PT - Point Load'/
   &' ','HZ - Hertzian Load'/
   &' ','HT - Hat Load'//
   &' ','[Type HZ suggested]: ',$)
      write(6,290)
      read(5,140)ld$
      if (ld$ .eq. 'pt' .or. ld$ .eq. 'PT') then
              ld$='PT'
              ld=1
```

```
      else if (ld$ .eq. 'hz' .or. ld$ .eq. 'HZ') then
              ld$='HZ'
              ld=2
      else if (ld$ .eq. 'ht' .or. ld$ .eq. 'HT') then
              ld$='HT'
              ld=3
      else
              go to 250
      end if
      go to 70
    else if (nselec .eq. ns) then
300 format(' ','Enter # radius divisions in the strain',
   &' field: ',$)
      write(6,300)
      read(5,*)nrs
      go to 70
    else if (nselec .eq. ns+1) then
      noutyp=abs(noutyp-1)
      if (noutyp .eq. 1) then
              outyp = 'Y'
              npg=3
      else
              outyp = 'N'
              npg=2
      end if
      go to 70
    else if (nselec .eq. 5 .and. ns .eq. 6) then
310   format(' ','Enter loading radius [mm]: ',$)
      write(6,310)
      read(5,*)xep
      go to 70
    else if (nselec .eq. ns+2) then
320 format(' ','# ',a1,' modes to be used in dynamic ',
   &'analysis: ',$)
      write(6,320)'X'
      read(5,*)ix
      go to 70
    else if (nselec .eq. ns+3) then
      write(6,320)'Y'
      read(5,*)iy
      go to 70
    else if (nselec .eq. ns+4) then
      ifc=abs(ifc-1)
      go to 70
    else if (nselec .eq. ns+5) then
      melim=abs(melim-1)
      go to 70
    else if (nselec .eq. ns+6) then
330 format(' ','Time step length [micro_s]: ',$)
      write(6,330)
      read(5,*)delt
      go to 70
    else if (nselec .eq. ns+7) then
```

```
  340 format(' ','# of time steps:'/
     &' ',' (0 = stop when impactor velocity = 0.0 m/s)'/
     &' ',' (1 = stop when plate displacement = 0.0 m)'/
     &' ',' (# greater than 1 = # of time steps used): ',$)
        write(6,340)
        read(5,*)nt
        go to 70
      else if (nselec .eq. -1) then
  350 format(/' ','Enter selection and hit <RETURN>'//
     &' ','1.  Exit without saving changes'/
     &' ','2.  Save changes and exit'/
     &' ','3.  Change jobname [',a6,']'/
     &' ','4.  Continue edit mode'/$)
  360 write(6,350) jobnam
      read(5,*)nselec
      if (nselec .eq. 1) then
        write(6,370)
  370 format(' ','Are you sure? (y/n): ',$)
        read (5,380) save
  380 format(a1)
        if (save .eq. 'y' .or. save .eq. 'Y') then
                return
        else
                go to 360
        end if
      else if (nselec .eq. 2) then
        state='exit'
        go to 570
      else if (nselec .eq. 3) then
 390    format(/' ','Enter jobname ["quit" to exit; Max 6 ',
     &'characters]: ',$)
 395    write(6,390)
        read(5,30) newjob
        if (newjob .ne. 'quit' .and. newjob .ne. 'QUIT') then
                jobnam = newjob
                call setfil(jobnam)
        end if
        go to 360
      else
        go to 80
      end if
      end if
C
C Write page 3
C
  400 write(6,410) jobnam,nth,((ept_max(i,j),j=i,3),i=1,3),
     & ((epc_max(i,j),j=i,3),i=1,3)
  410 format(///t20,'Edit: ',a6//
     &' ','(1) # theta divisions in strain field: ',i2//
     &' ','Matrix of maximum tensile strains (tensor notation):'//
     &' ',' (2) ',f8.4,t20,' (3) ',f8.4,t40,' (4) ',f8.4/
     &' ',t20,' (5) ',f8.4,t40,' (6) ',f8.4/
     &' ',t40,' (7) ',f8.4//
```

```
   &' ','Matrix of maximum compressive strains (tensor notation):'//
   &' ',' (8) ',f8.4,t20,' (9) ',f8.4,t40,'(10) ',f8.4/
   &' ',t20,'(11) ',f8.4,t40,'(12) ',f8.4/
   &' ',t40,'(13) ',f8.4///////)
    write(6,110)
    read(5,*)nselec
420 format(' ','Enter EP',a1,'_MAX(',i1,',',i1,'): ',$)
    if (nselec .lt. -1 .or. nselec .gt. 13) then
      go to 80
    else if (nselec .eq. -1) then
      go to 360
    else if (nselec .eq. 0) then
      npg=1
      go to 80
    end if
    go to (550,430,440,450,460,470,480,490,500,510,520,
   &       530,540)nselec
430 write(6,420)'T',1,1
    read(5,*) ept_max(1,1)
    go to 70
440 write(6,420)'T',1,2
    read(5,*) ept_max(1,2)
    go to 70
450 write(6,420) 'T',1,3
    read(5,*) ept_max(1,3)
    go to 70
460 write(6,420) 'T',2,2
    read (5,*) ept_max(2,2)
    go to 70
470 write(6,420) 'T',2,3
    read(5,*) ept_max(2,3)
    go to 70
480 write(6,420) 'T',3,3
    read(5,*) ept_max(3,3)
    go to 70
490 write(6,420) 'C',1,1
    read(5,*) epc_max(1,1)
    go to 70
500 write(6,420) 'C',1,2
    read(5,*) epc_max(1,2)
    go to 70
510 write(6,420) 'C',1,3
    read(5,*) epc_max(1,3)
    go to 70
520 write(6,420) 'C',2,2
    read(5,*) epc_max(2,2)
    go to 70
530 write(6,420) 'C',2,3
    read(5,*) epc_max(2,3)
    go to 70
540 write(6,420) 'C',3,3
    read(5,*) epc_max(3,3)
    go to 70
```

```
  550 write(6,560)
  560 format(' ','Enter # theta divisions in strain field [max 90]: ',$)
      read (5,*) nth
      go to 70
C=================================================================
C       Save changes in the binary file

  570 saved=.true.
        inquire(file=filnam(2),exist=exst)
        if (exst) then
                write(6,580) jobnam
  580           format(/' ','Intermediate file already exists. ',
     & ' Select an option and hit RETURN:'//
     & ' ',' C = Change jobname [',a6,']'/
     & ' ',' O = Overwrite existing file'/
     & ' ',' V = Save with same name, but different version #'//
     & ' ',' [Or just hit RETURN to exit]: ',$)
                read(5,590) choice
  590 format(a1)
                if (choice .eq. 'C' .or. choice .eq. 'c') then
                        go to 395
                else if (choice .eq. 'O' .or. choice .eq. 'o') then
                        stat='old'
                else if (choice .eq. 'V' .or. choice .eq. 'v') then
                        stat='new'
                else
                        go to 80
                end if
        else
                stat='new'
        end if
        open(unit=16,file=filnam(2),status=stat,access='direct',
     &       recl=110)
        write(16,rec=1)title1
        write(16,rec=2)title2
        write(16,rec=3)title3
        write(16,rec=4) np
        write(16,rec=5) thick
        write(16,rec=6) a(1,1),a(1,2),a(2,2)
        write(16,rec=7) e11,e1z,enu12,enu13,g13
        write(16,rec=8) e22,e2z,enu21,enu23,g23
        write(16,rec=9) d11,d12,d22,d66,d16,d26
        write(16,rec=10)a$*1.E-3,b*1.E-3
        write(16,rec=11) rho
        write(16,rec=12)pnx,pny
        write(16,rec=13)nx,ny
        write(16,rec=14)shear
        write(16,rec=15)bm,vi
        write(16,rec=16) ri
        write(16,rec=17) idhzs
        write(16,rec=19) r
        write(16,rec=20)nohars
        write(16,rec=21) numalp,(alphaa(i),i=1,numalp)
```

```
      write(16,rec=22) (tqa(j),j=1,numalp)
      write(16,rec=23)ld$,xep
      write(16,rec=24)nrs
      write(16,rec=25)outyp,nth,((ept_max(i,j),j=i,3),i=1,3),
     &                 ((epc_max(i,j),j=i,3),i=1,3)
      write(16,rec=26)ix,iy
      write(16,rec=27)ifc,melim
      write(16,rec=28)delt,nt
      write(16,rec=29) hk
      write(16,rec=30) ldd,acc
      close(unit=16)
      if (state .eq. 'save') go to 80
     end
C=================================================================
C     Main Menu
C
      subroutine menu

      character jobnam*6, level*20, genfil*10
      integer lenter,lexit
      common /block7/ main,jobnam,level(4),lenter,lexit,genfil(10)
   10 format('1',///t15,'Analysis of Composite Plate Impact'/
     &' ',t15,'            (ACPI)'
     &' ',t20,'      Main Menu'  //
     &' ',15x,'(1)  Enter Analysis Parameters'/
     &' ',15x,'(2)  Select Intermediate Analysis Output [OPTIONAL]'/
     &' ',15x,'(3)  Run Analysis'/
     &' ',15x,'(4)  Quit Analysis'/////////
     &' ','[Enter selection and hit <RETURN>]: ',$)
   20 write(6,10)
      read(5,*) main
      if (main .lt. 1 .or. main .gt. 4) go to 20
      end
C=================================================================
C    System Configuration Routine
C
C       This routine allows the more experienced user to select various
C  options determining the analysis.  The following can be set by the
C  user:
C              1.  Suppress/Generate intermediate result files
C              2.  Set analysis entry location
C              3.  Set analysis exit location
C              4.  Set the analysis jobname
C
C  Preset default values :
C              1.  ALL intermediate files are SUPPRESSED
C              2.  Analysis ENTRY = LAMINANT PROPERTIES
C              3.  Analysis EXIT  = DAMAGE PREDICTION
C              4.  Jobname = "------"
      subroutine configure

      character*20  level, binfil
      character*10  genfil
```

```
      character*6   jobnam, newjob

      logical exst

      common /block7/ main, jobnam, level(4), lenter,
     &                lexit, genfil(15)

   10 format('1'///' ',t20,'System Configuration'/
     &' ',t20              '(1) Job: ',a6/
     &' ','(2) Analysis entry location: ',a20/
     &' ','(3) Analysis exit location:  ',a20//
     &' ',t6,'FILE EXT',t19,'STATUS',t30,'CONTENTS'/
     &' ',t6,'————————',t19,'————————',t30,'————————'/
     &' ','Output from level: Hertzian Spring'/
     &' ',' (4) _HERTZ.OUT',t19,a10,t30,'Hertzian spring determination'/
     &' ',' (5) _HPRES.RES',t19,a10,t30,'Hertzian pressure convergence',
     &' data'/
     &' ',' (6) _HDISP.RES',t19,a10,t30,'Hertzian displacement ',
     &'convergence data'/
     &' ',' (7) _AMAJR.RES',t19,a10,t30,'Major axis alphas VS load'/
     &' ',' (8) _AMINR.RES',t19,a10,t30,'Minor axis alphas VS load'//
     &' ','Output from level: Dynamic Analysis'/
     &' ',' (9) _DYNAM.OUT',t19,a10,t30,'Screen output for dynamic ',
     &' ','analysis'/
     &' ','(10) _FORCE.RES',t19,a10,t30,'Dynamic loading history of ',
     &'laminate')
   15 format(
     &' ','(11) _DISPL.RES',t19,a10,t30,'Dynamic displacement history'/
     &' ','(12) _ACCEL.RES',t19,a10,t30,'Dynamic acceleration history'//
     &' ','Output from level: Damage Prediction'/
     &' ','(13) _DPRES.RES',t19,a10,t30,'Stress-strain pressure '
     &'convergence data'//
     &' ','[Enter: 0 = exit; 1-13 = change parameter]: ',$)
   20 write(6,10) jobnam,level(lenter),level(lexit),(genfil(i),i=5,11)
      write(6,15) (genfil(i),i=12,14)
      read(5,*) nconf
      if (nconf .eq. 0) return
      if (nconf .lt. 1 .or. nconf .gt. 13) go to 20
      if (nconf .eq. 2) then
   30   format(/' ','Select analysis ',a5,' location:'//
     &' ','1.  Layup properties  - determine the overall material ',
     &'properties'/
     &' ','2.  Hertzian spring   - determination of Hertzian spring ',
     &'constant'/
     &' ','3.  Dynamic analysis  - determine peak force and its ',
     &'corresponding'/' ',t26,'acceleration via a dynamic model'/
     &' ','4.  Damage prediction - obtain stress-strain results from ',
     &'the impact'/
     &' ','5.  Exit this selection'//
     &'[Enter selection and hit <RETURN>]: ',$)
   40   write(6,30) 'entry'
        read(5,*) ichng
        if (ichng .eq. 5) go to 20
```

```
      if (jobnam .eq. 'undeclared') then
              write(6,*)'* Cannot change analysis entry of an ',
  &                   'undeclared job'
              write(6,*)'[hit <RETURN> to continue]'
 50   format(a)
              read(5,50) temp
              go to 20
      end if

      if (ichng .lt. 1 .or. ichng .gt. 4) go to 40
      if (ichng .eq. 1) then
              ienter = 1
              go to 20
      end if
      do i=1,6
              if (.not.(jobnam(i:i) .eq. ' ')) then
              binfil(i:i)=jobnam(i:i)
              else
              go to 60
              end if
      end do
 60   binfil(i:i+3)='.bin'
      inquire(file=binfil, exist=exst)
      if (.not. (exst)) then
 70 format(' ','Insufficient results in job to enter at this point'/
  &' ','Entry point request denied. (Hit <RETURN> to continue.)',$)
 80           write(6,70)
              read(5,50)
              go to 20
      end if
    open(unit=16,file=binfil,status='old',access='direct',recl=110)
      go to (90,100,110) ichng-1
 90   close(unit=16)
      ienter=ichng
      go to 20
100   read(16,rec=29) temp2
      if (temp2 .eq. 0.) go to 80
      go to 90

110   read(16,rec=30) temp2,temp3
      if (temp2 .eq. 0. .or. temp3 .eq. 0.) go to 80
      go to 90


    else if (nconf .eq. 3) then
120   write(6,30)'exit'
      read(5,*) ichng
      if (ichng .eq. 5) go to 20
      if (ichng .lt. 1 .or. ichng .gt. 4) go to 120
      if (ichng .lt. ienter) then
130   format(' ','Exit point cannot preceed entry point.  Exit point',
  &' change denied.'/' ','[Hit <return> to continue.]',$)
              read(5,50) temp
```

```
                go to 120
        end if
        lexit=lchng
        go to 20
      else if (nconf .eq. 1) then
        write(6,140)
 140    format(' ','Enter new jobname [max 6 characters]: ',$)
        read(5,150) newjob
        if (.not.(newjob(1:1) .eq. ' ')) then
                jobnam = newjob
                call setfil(jobnam)
        end if
 150    format(a)
        go to 20
      else
        if (genfil(nconf+1) .eq. 'create') then
                genfil(nconf+1) = 'suppress'
        else
                genfil(nconf+1) = 'create'
        end if
      end if
      go to 20
      end
C=================================================================
C               Subroutine SETFIL
C
C     This routine adds the file extensions to the jobname and stores
C  the complete filenames in the array FILNAM in the order:
C array #    Extension   Unit#   Use/Contents
C ------    ----------  -----   ------------------------------------
C    1         .DAT       15    Beginning file created by this program.
C                               Used as input to ROTATE.
C    2         .BIN       16    Master binary file used by BESSEL and
C                               IMPACT.  Created by ROTATE.
C    3       _ANGLE.DAT    8    Angle information used by BESSEL in the
C                               strain analysis. Created by ROTATE.
C    4       _DAMGE.OUT         Strain output from BESSEL. Text format.
C    5*      _HERTZ.OUT   12    Text results from calculating the
C                               Hertzian spring constant in BESSEL.
C    6*      _HPRES.RES         Pressure convergence data for
C                               calculating the Hertzian constant.
C    7*      _HDISP.RES         Displacement convergence data for
C                               calculating the Hertzian constant.
C    8*      _AMAJR.RES    9    Major axis alpha VS load data.
C    9*      _AMINR.RES   10    Minor axis alpha VS load data.
C   10*      _DYNAM.OUT    6    Monitor output from dynamic analysis.
C   11*      _FORCE.RES         Loading history in dynamic analysis.
C   12*      _DISPL.RES         Displacement history in dyn analysis.
C   13*      _ACCEL.RES         Acceleration history in dyn analysis.
C   14*      _DPRES.RES         Pressure convergence data for strain
C                               calculations in BESSEL.
C   15       _SS.BIN            Stress-strain output file. RECL=72.
C                               RECORD 1
```

```
C                                       DATA: NP, NRS
C                                 RECORDS 2+
C                                       DATA: SRR, STT, SZZ, SRZ, ESS,
C                                             ETT, EZZ, ERZ
C                                 S = sigma, E = epsilon, R,T,Z = direction/face
C                                 NP=# plies, NRS=# radis divisions
C   * indicates optionally generated files determined by the system
C     configuration.  Not generated by dfault.
C
      subroutine setfil(job)

      character*20  filnam
      character*6   job

      common /block8/ filnam(15)

      do i=1,15
        filnam(i)=' '
      end do

      do i = 1,7
        if (i .eq. 7) goto 10
        if (job(i:i) .eq. ' ') go to 10
      end do
   10 n=i-1
      filnam(1)=job(1:n)//'.dat'
      filnam(2)=job(1:n)//'.bin'
      filnam(3)=job(1:n)//'_angle.dat'
      filnam(4)=job(1:n)//'_damge.out'
      filnam(5)=job(1:n)//'_hertz.out'
      filnam(6)=job(1:n)//'_hpres.res'
      filnam(7)=job(1:n)//'_hdisp.res'
      filnam(8)=job(1:n)//'_amajr.res'
      filnam(9)=job(1:n)//'_aminr.res'
      filnam(10)=job(1:n)//'_dynam.out'
      filnam(11)=job(1:n)//'_force.res'
      filnam(12)=job(1:n)//'_displ.res'
      filnam(13)=job(1:n)//'_accel.res'
      filnam(14)=job(1:n)//'_dpres.res'
      filnam(15)=job(1:n)//'_ss.bin'
      end
C*****************************************************************************
```

```
      PROGRAM ROTATE
C
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C
C ==================================================================
C ==================================================================
C
C  THIS IS THE FIRST PROGRAM OF ACPI (ANALYSIS OF COMPOSITE PLATE
C  IMPACT).
C
C ==================================================================
C
C  THIS PROGRAM ROTATES PLY PROPERTIES AND GENERATES THE AVERAGE
C
C  THREE-DIMENSIONAL CONSTITUTIVE PROPERTIES TO USE IN ANISOTROPIC
C
C  ANALYSES.
C
C  ORIGINAL PROGRAM WRITTEN 5/86 BY D.S. CAIRNS
C  ORTHOTROPIC PLATE AND I/O MODIFICATIONS 1/88 BY M.J. KRAFT
C  ADDITIONAL I/O MODIFICATIONS 4/88 BY K.F. RYAN
C
C ==================================================================
C
C The data for this program can be entered in one of three ways:
C
C     1)  The computer keyboard.
C     2)  An input file that only contains ply information.
C         - These two forms are used if you are only running this
C           program.
C     3)  A master input file that contains all of the information
C         needed for ACPI.
C         - This form is used if you plan to run the remaining programs
C           in ACPI.  The data needed to run the second program of ACPI
C           is transfered to a direct access file named 'DIRACC.BIN'.
C
C ==================================================================
C ==================================================================
```

```
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C  THE FOLLOWING FILES ARE OPENED IN THIS PROGRAM:
C
C    UNIT     FILE              COMMENTS
C    ----     ----              --------
C
C     7       'LIBRARY.BIN'*    CONTAINS THE MATERIAL PROPERTIES IN
C                                     BINARY FORM
C    13       'ANGLE.DAT'       THE ANGLE (IN DEGREES) OF EACH PLY IS
C                                     SENT TO THIS FILE IF YOU USE AN
C                                     INPUT FILE.
C    30       'ROTATE.OUT'      THE DATA IN 'DIRACC.BIN' IS DISPLAYED IN
C                                     THIS FILE FOR EASY REFERENCE.  THIS
C                                     FILE IS ONLY CREATED IF YOUR INPUT
C                                     FILE IS THE MASTER INPUT FILE FOR
C                                     ACPI.
C    11       INPUT FILE        YOU NAME THIS FILE.  IT IS THE INPUT
C                                     FILE FOR THIS PROGRAM.
C    12       OUTPUT FILE*      THE DATA NEEDED TO RUN THE REMAINING
C                                     PROGRAMS IN ACPI IS SENT TO THIS
C                                     DIRECT ACCESS FILE IF YOUR INPUT
C                                     FILE IS THE MASTER INPUT FILE FOR
C                                     ACPI.
C    * THESE ARE DIRECT ACCESS FILES.  HENCE, THE DATA IS IN BINARY
C      FORM.
C
      CALL STARTUP
      CALL INPUT
      CALL QBAR
      CALL OUTPUT
      END
C
C ==================================================================
C
      SUBROUTINE INPUT
C
C ==================================================================
C
C  THIS SUBROUTINE READS INPUT TO DETERMINE THE LAYUP AND
C  MATERIAL INFORMATION
C
C ------------------------------------------------------------------
C
C  DEFINITION OF INPUT VARIABLES:
C
C  VARIABLE    DESCRIPTION                                  UNITS
C
C   NP         Number of plies in the laminate (integer)    MAXIMUM = 50
C   MAT        Input material number (integer)
C                   The material numbers used in this
C                   program are the same as those used in
```

```
C                    LASP.
C   TP          Ply thickness                                  [mm]
C   THP         Ply angle                                      [deg]
C
C ------------------------------------------------------------------------
C
      IMPLICIT REAL*8 (A-H,O-Z)

      character*60 title1, title2, title3
      character*20 filnam
      character*6  jobnam
      character*1  again

      logical local,genfil,exst

      common /system/ jobnam,local,filnam(15), genfil(15)
      COMMON/PLYDAT/THP(50),MAT(50),TP(50),NP,ZUPPER(51),D1111,D1122,
     &     D2222,D1212,D1112,D2212
      COMMON/INOUT/THICK,NYN,KFR,TITLE1,TITLE2,TITLE3
      DATA D1111,D1122,D2222,D1212,D1112,D2212/6*0.0/
C
      if (.not.(local)) go to 26
      WRITE(6,1)
    1 FORMAT(1H1,79(1H=)//5X,'THIS IS THE FIRST PROGRAM OF ACPI ',
     *  '(ANALYSIS OF COMPOSITE PLATE IMPACT).'/1H0,79(1H=)//
     *  14X,'THIS PROGRAM ROTATES PLY PROPERTIES AND GENERATES THE'/
     *  13X,'AVERAGE 3-D CONSTITUTIVE PROPERTIES GIVEN A PRESCRIBED'/
     *  20X,'LAMINATE FOR USE IN ANISOTROPIC ANALYSES')
      WRITE(6,2)
    2 FORMAT(1H0,79(1H=))
    3 WRITE(6,4)
    4 FORMAT(1H0,'IS THIS PROGRAM USING AN INPUT FILE',
     *  ' (1=YES OR 0=NO)?')
      READ(5,*)NYN
      WRITE(6,2)
C
      IF (NYN .EQ. 0) THEN
C
    5   WRITE(6,*)' '
        WRITE(6,*)'INPUT THE NUMBER OF PLIES IN THE LAMINATE'
        READ(5,*)NP
        THICK=0.
        DO 10 I=1,NP
          WRITE(6,*)'INPUT MAT. #, THICKNESS[mm], PLY ANGLE[deg] ',I
          READ(5,*)MAT(I),TP(I),THP(I)
          THP(I)=THP(I)*3.141592654/180.
          THICK=THICK+TP(I)
   10   CONTINUE
C
      ELSE IF (NYN .EQ. 1) THEN
C
C  READ IN THE PLY INFORMATION FROM THE INPUT FILE
C
```

```
      inquire(file=filnam(1),exist=exst)
      if (exst) then
        kfr = 2
        nyn = 1
        if (local) write(6,*) 'Reading data from file: ',filnam(1)
      end if
      if (.not.(exst)) then
   15 WRITE(6,*)
      WRITE(6,*)'IF YOUR FILE ONLY CONTAINS PLY INFORMATION, ENTER "1".'
      WRITE(6,*)' '
      WRITE(6,*)'IF YOUR FILE IS THE MASTER FILE FOR ACPI, ENTER "2".'
      WRITE(6,*)' '
      READ(5,*)KFR
      go to 26
C   20 WRITE(6,*)'WHAT IS THE FULL NAME OF THE INPUT FILE?'
C       READ(5,21) filnam(1)
   21 FORMAT(A12)
   17 format(a20)
      end if
C       WRITE(6,*)'OPENNING ',DATA
   26 OPEN(UNIT=11,FILE=filnam(1),STATUS='OLD',IOSTAT=IERR,ERR=30)
      if (.not.(local)) then
        kfr=2
        nyn = 1
      end if
      READ(11,22)TITLE1
      READ(11,22)TITLE2
      READ(11,22)TITLE3
   22 FORMAT (A)
      READ(11,*)NP
   24 FORMAT(I5,4X,I5,9X,F7.4,7X,F5.1)
      if (local) then
      WRITE(6,*)
      WRITE(6,*)'   PLY    ','MATERIAL  ','THICKNESS[mm]  ','ANGLE[deg]'
      WRITE(6,*)'  ———————','——————————','————————————————','——————————'
      end if
      THICK=0.
   28 OPEN(UNIT=13,IOSTAT=KOPEN8,ERR=50,FILE=filnam(3),STATUS='NEW')
      DO 25 J=1,NP
        READ(11,*)MAT(J),TP(J),THP(J)
        if (local) then
                WRITE(6,24)J,MAT(J),TP(J),THP(J)
        end if
                WRITE(13,*)THP(J)
        THP(J)=THP(J)*3.141592654/180.
        THICK=THICK+TP(J)
   25 CONTINUE
      CLOSE(UNIT=13)
      NPI = NP+1
      ZUPPER(1) = THICK/2.
      DO 27 K=2,NPI
   27    ZUPPER(K) = ZUPPER(K-1) - TP(K-1)
```

```
C
      ELSE
C
        GOTO 3
C
      END IF
C
      RETURN
C   30 WRITE(6,*)'ERROR IN OPENING INPUT FILE.  STATUS = ',IERR
C      IF (IERR .EQ. 29) THEN
C        WRITE(6,*)' '
C        WRITE(6,*)'YOUR INPUT FILE WAS NOT FOUND.'
C        WRITE(6,*)'DO YOU WISH TO TRY AGAIN? (Y/N)'
C        READ(5,40)AGAIN
C   40   FORMAT(A)
C        IF (AGAIN .EQ. 'Y') GOTO 20
C      END IF
C
 30   write(6,35)
 35   format(' ','Enter the name of the input file: ',$)
      read(5,17) filnam(1)
      go to 26
 50   write(6,60)
 60   format(' ','Enter the name of the file containing ply angles: ',
     & $)
      read(5,17) filnam(3)
      go to 28
C   50 WRITE(6,*)'ERROR IN OPENING "ANGLE.DAT".  STATUS = ',KOPEN8
C      STOP
      END
C
C ==================================================================
C
      SUBROUTINE QBAR
C
C ==================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 M,N
C
C  THIS SUBROUTINE DETERMINES THE REDUCED STIFFNESS MATRIX OF EACH PLY
C  SUMS THEM IN AN AVERAGE SENSE AND DETERMINES THE AVERAGE PROPERTIES
C
      COMMON/PLYDAT/THP(50),MAT(50),TP(50),NP,ZUPPER(51),D1111,D1122,
     &     D2222,D1212,D1112,D2212
      DIMENSION Q(50,6,6),QB(50,6,6),EC(6,6),T(6,6),
     + SP(6,6),QV(6),QVP(6),QQ(6),TT(3,3),QBR(6,6)
      COMMON/COMP/SA(6,6),QA(6,6)
C
C  LOOPING ON EACH PLY
C
      DO 50 L=1,NP
C
```

```
C  PLY PROPERTIES
C
        CALL MATS(Q,L)
C
        THP(L) = -THP(L)
C
C  DEFINING THE TRANSFORMATION MATRIX
C
        M=DCOS(THP(L))
        N=DSIN(THP(L))
C
        T(1,1)=M**4
        T(1,2)=2.*(M*N)**2
        T(1,3)=2.*M**3*N
        T(1,4)=N**4
        T(1,5)=2.*M*N**3
        T(1,6)=(M*N)**2
        T(2,1)=T(1,2)/2.
        T(2,2)=M**4+N**4
        T(2,3)=(T(1,5)-T(1,3))/2.
        T(2,4)=T(1,2)/2.
        T(2,5)=-T(2,3)
        T(2,6)=-T(2,1)
        T(3,1)=-T(1,3)
        T(3,2)=2.*T(2,5)
        T(3,3)=M**4-3.*(M*N)**2
        T(3,4)=T(1,5)
        T(3,5)=3.*(M*N)**2-N**4
        T(3,6)=T(2,5)
        T(4,1)=N**4
        T(4,2)=T(1,2)
        T(4,3)=-T(1,5)
        T(4,4)=M**4
        T(4,5)=-T(1,3)
        T(4,6)=(M*N)**2
        T(5,1)=-T(1,5)
        T(5,2)=-T(3,2)
        T(5,3)=T(3,5)
        T(5,4)=-T(4,5)
        T(5,5)=T(3,3)
        T(5,6)=-T(3,6)
        T(6,1)=4.*T(1,6)
        T(6,2)=-4.*T(1,2)
        T(6,3)=4.*T(2,3)
        T(6,4)=T(6,1)
        T(6,5)=-T(6,3)
        T(6,6)=(M**2-N**2)**2
C
C  STORING THE STIFFNESS MATRIX AS A VECTOR FOR IN-PLANE TRANSFORMS
C
        QQ(1)=Q(L,1,1)
        QQ(2)=Q(L,1,2)
        QQ(3)=Q(L,1,6)*2.
```

```
      QQ(4)=Q(L,2,2)
      QQ(5)=Q(L,2,6)*2.
      QQ(6)=Q(L,6,6)*4.
      DO 10 I=1,6
        QVP(I)=0.0
        DO 10 J=1,6
          QVP(I)=QVP(I)+T(I,J)*QQ(J)
 10   CONTINUE
      QB(L,1,1)=QVP(1)
      QB(L,1,2)=QVP(2)
      QB(L,1,6)=QVP(3)/2.
      QB(L,2,2)=QVP(4)
      QB(L,2,6)=QVP(5)/2.
      QB(L,6,6)=QVP(6)/4.
      QB(L,3,3)=Q(L,3,3)
C
C  DETERMINING THE OUT OF PLANE COUPLINGS
C
      T(1,1)=M**3
      T(1,2)=-M**2*N
      T(1,3)=M*N**2
      T(1,4)=-N**3
      T(1,5)=M**2*N
      T(1,6)=-M*N**2
      T(2,1)=-T(1,2)
      T(2,2)=T(1,1)
      T(2,3)=-T(1,4)
      T(2,4)=-T(1,6)
      T(2,5)=T(2,4)
      T(2,6)=T(2,1)
      T(3,1)=T(1,3)
      T(3,2)=-T(2,3)
      T(3,3)=T(1,1)
      T(3,4)=-T(1,5)
      T(3,5)=T(3,4)
      T(3,6)=-T(1,6)
      T(4,1)=N**3
      T(4,2)=T(1,3)
      T(4,3)=T(1,5)
      T(4,4)=T(1,1)
      T(4,5)=-T(2,5)
      T(4,6)=-T(2,6)
      T(5,1)=2.*T(1,2)
      T(5,2)=2.*T(4,2)
      T(5,3)=2.*T(4,3)
      T(5,4)=-T(5,2)
      T(5,5)=T(1,1)-M*N**2
      T(5,6)=N**3-M**2*N
      T(6,1)=-T(5,2)
      T(6,2)=-T(5,3)
      T(6,3)=T(5,2)
      T(6,4)=T(5,3)
      T(6,5)=-T(5,6)
```

```
      T(6,6)=T(5,5)
C
C  STORING THE STIFFNESS MATRIX AS A VECTOR FOR OUT-OF-PLANE TRANSFORMS
C
      QQ(1)=Q(L,1,4)
      QQ(2)=Q(L,1,5)
      QQ(3)=Q(L,2,4)
      QQ(4)=Q(L,2,5)
      QQ(5)=Q(L,4,6)*2.
      QQ(6)=Q(L,5,6)*2.
      DO 20 I=1,6
        QVP(I)=0.0
        DO 20 J=1,6
          QVP(I)=QVP(I)+T(I,J)*QQ(J)
 20   CONTINUE
      QB(L,1,4)=QVP(1)
      QB(L,1,5)=QVP(2)
      QB(L,2,4)=QVP(3)
      QB(L,2,5)=QVP(4)
      QB(L,4,6)=QVP(5)/2.
      QB(L,5,6)=QVP(6)/2.
C
C  CONTINUING WITH THE COUPLINGS
C
      T(1,1)=M**2
      T(1,2)=N**2
      T(1,3)=M*N
      T(2,1)=T(1,2)
      T(2,2)=T(1,1)
      T(2,3)=-T(1,3)
      T(3,1)=-2.*T(1,3)
      T(3,2)=-T(3,1)
      T(3,3)=M**2-N**2
C
C
C  STORING THE STIFFNESS MATRIX AS A VECTOR FOR OUT-OF-PLANE TRANSFORMS
C
      QQ(1)=Q(L,1,3)
      QQ(2)=Q(L,2,3)
      QQ(3)=Q(L,3,6)*2.
      DO 30 I=1,3
        QVP(I)=0.0
        DO 30 J=1,3
          QVP(I)=QVP(I)+T(I,J)*QQ(J)
 30   CONTINUE
      QB(L,1,3)=QVP(1)
      QB(L,2,3)=QVP(2)
      QB(L,3,6)=QVP(3)/2.
C
C  CONTINUING WITH THE COUPLINGS
C
      T(1,1)=M**2
      T(1,2)=-2.*M*N
```

```
      T(1,3)=N**2
      T(2,1)=T(1,2)/-2.
      T(2,2)=M**2-N**2
      T(2,3)=-T(2,1)
      T(3,1)=T(1,3)
      T(3,2)=-2.*T(2,3)
      T(3,3)=M**2
C
C  STORING THE STIFFNESS MATRIX AS A VECTOR FOR OUT-OF-PLANE TRANSFORMS
C
      QQ(1)=Q(L,4,4)
      QQ(2)=Q(L,4,5)
      QQ(3)=Q(L,5,5)
      DO 40 I=1,3
        QVP(I)=0.0
        DO 40 J=1,3
          QVP(I)=QVP(I)+T(I,J)*QQ(J)
 40   CONTINUE
      QB(L,4,4)=QVP(1)
      QB(L,4,5)=QVP(2)
      QB(L,5,5)=QVP(3)
C
 50   CONTINUE
C
C  DETERMINING THE AVERAGE QMATRIX & SMATRIX
C
      THK = 0.0
      DO 70 L=1,NP
         THK = THK+TP(L)
   70 CONTINUE
C
      CALL AVER(QB,TP,NP,THK)
C
      RETURN
      END
C
C ==================================================================
C
      SUBROUTINE MATS(Q,L)
C
C ==================================================================
C
C  THIS SUBROUTINE OBTAINS THE MATERIAL INFORMATION FROM THE FILE
C  'LIBRARY.BIN', CALCULATES THE D MATRIX, AND DOES SOME PROCESSING TO
C  THE DATA.
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/PLYDAT/THP(50),MAT(50),TP(50),NP,ZUPPER(51),D1111,D1122,
     &     D2222,D1212,D1112,D2212
      DIMENSION S(6,6),C(6,6),Q(50,6,6)
C
      OPEN(UNIT=7,IOSTAT=KOPEN7,ERR=5000,FILE='LIBRARY.BIN',
     &     STATUS='OLD',ACCESS='DIRECT',RECL=144)
```

```
C
      READ (7,REC=MAT(L)) E11,E22,E33,G23,G31,G12,ENU12,ENU13,ENU23
C
      CLOSE(UNIT=7)
C ------------------------------------------------------------------
C
C  CALCULATION OF THE D MATRIX
C
C ------------------------------------------------------------------
      ENU21 = ENU12*E22/E11
      XM = 1./(1. - ENU12*ENU21)
      E1111$ = XM*E11
      E2222$ = XM*E22
      E1122$ = XM*ENU21*E11
      E1212$ = G12
C
      XI1 = 0.25*(E1111$ + E2222$ + 2.*E1122$)
      XI2 = 0.125*(E1111$ + E2222$ - 2.*E1122$ + 4.*E1212$)
      R1 = 0.5*(E1111$ - E2222$)
      R2 = 0.125*(E1111$ + E2222$ - 2.*E1122$ - 4.*E1212$)
C
      E1111_ = XI1 + XI2 + R1*DCOS(2.*(-THP(L))) + R2*DCOS(4.*(-THP(L)))
      E2222_ = XI1 + XI2 - R1*DCOS(2.*(-THP(L))) + R2*DCOS(4.*(-THP(L)))
      E1122_ = XI1 - XI2 - R2*DCOS(4.*(-THP(L)))
      E1212_ = XI2 - R2*DCOS(4.*(-THP(L)))
      E1112_ = -0.5*R1*DSIN(2.*(-THP(L))) - R2*DSIN(4.*(-THP(L)))
      E2212_ = -0.5*R1*DSIN(2.*(-THP(L))) + R2*DSIN(4.*(-THP(L)))
C
      Z3Z33 = (ZUPPER(L)*ZUPPER(L)*ZUPPER(L)
     &         - ZUPPER(L+1)*ZUPPER(L+1)*ZUPPER(L+1))/3.
      D1111 = D1111 + E1111_*Z3Z33
      D2222 = D2222 + E2222_*Z3Z33
      D1122 = D1122 + E1122_*Z3Z33
      D1212 = D1212 + E1212_*Z3Z33
      D1112 = D1112 + E1112_*Z3Z33
      D2212 = D2212 + E2212_*Z3Z33
C ------------------------------------------------------------------
C ------------------------------------------------------------------
      DO 110 I=1,6
         DO 110 J=1,6
  110       S(I,J)=0.0
C
C DEFINING THE ENGINEERING COMPLIANCE MATRIX
C
      S(1,1)=1./E11
      S(1,2)=-ENU12/E11
      S(1,3)=-ENU13/E11
      S(2,1)=S(1,2)
      S(2,2)=1./E22
      S(2,3)=-ENU23/E22
      S(3,1)=S(1,3)
      S(3,2)=S(2,3)
      S(3,3)=1./E33
```

```
      S(4,4)=1./G23
      S(5,5)=1./G31
      S(6,6)=1./G12
C
C STORING OFF THE MATRIX FOR INVERSION TO OBTAIN THE E MATRIX
C
      DO 111 I=1,6
      DO 111 J=1,6
 111    C(I,J)=S(I,J)
C
C MATRIX INVERSION OF THE C MATRIX
C
      CALL INVSE (6,C)
C
      DO 112 I=1,6
         DO 112 J=1,6
  112       Q(L,I,J) = C(I,J)
C
      RETURN
 5000 WRITE(6,*)'Error in opening "LIBRARY.BIN."  Status = ',KOPEN7
      RETURN
      END
C
C ==================================================================
C
      SUBROUTINE AVER(QB,TP,NP,THK)
C
C ==================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION QB(50,6,1),C11(3,3),C12(3,3),C21(3,3),C22(3,3),CB21(3,3)
      DIMENSION DUM1(3,3),DUM2(3,3),CB11(3,3),CB12(3,3),CB22(3,3),TP(1)
      COMMON/COMP/SA(6,6),QA(6,6)
      DO 10 L=1,NP
C
        DO 15 I=2,6
        DO 15 J=1,I-1
        QB(L,I,J)=QB(L,J,I)
 15     CONTINUE
C
        C11(1,1)=QB(L,1,1)
        C11(1,2)=QB(L,1,2)
        C11(1,3)=QB(L,1,6)
        C11(2,2)=QB(L,2,2)
        C11(2,3)=QB(L,2,6)
        C11(3,3)=QB(L,6,6)
        C11(2,1)=C11(1,2)
        C11(3,1)=C11(1,3)
        C11(3,2)=C11(2,3)
C
        C12(1,1)=QB(L,1,3)
        C12(1,2)=QB(L,1,4)
        C12(1,3)=QB(L,1,5)
```

```
        C12(2,2)=QB(L,2,4)
        C12(2,3)=QB(L,2,5)
        C12(3,3)=QB(L,6,5)
        C12(2,1)=QB(L,2,3)
        C12(3,1)=QB(L,6,3)
        C12(3,2)=QB(L,6,4)
C
        C22(1,1)=QB(L,3,3)
        C22(1,2)=QB(L,3,4)
        C22(1,3)=QB(L,3,5)
        C22(2,2)=QB(L,4,4)
        C22(2,3)=QB(L,4,5)
        C22(3,3)=QB(L,5,5)
        C22(2,1)=C22(1,2)
        C22(3,1)=C22(1,3)
        C22(3,2)=C22(2,3)
C
        DO 20 I=1,3
        DO 20 J=1,3
        C21(I,J)=C12(J,I)
 20     CONTINUE
C
        CALL INVSE (3,C22)
C
        CALL MXMUL(3,C12,C22,DUM1)
        CALL MXMUL(3,DUM1,C21,DUM2)
C
        DO 50 I=1,3
        DO 50 J=1,3
          CB11(I,J)=CB11(I,J)+(C11(I,J)-DUM2(I,J))*TP(L)/THK
          CB12(I,J)=CB12(I,J)+DUM1(I,J)*TP(L)/THK
          CB22(I,J)=CB22(I,J)+C22(I,J)*TP(L)/THK
          CB21(I,J)=CB21(I,J)-DUM1(J,I)*TP(L)/THK
 50     CONTINUE
 10   CONTINUE
C
      CALL INVSE (3,CB22)
C
      CALL MXMUL(3,CB12,CB22,DUM1)
      CALL MXMUL(3,DUM1,CB21,DUM2)
C
      DO 60 I=1,3
      DO 60 J=1,3
        C11(I,J)=CB11(I,J)-DUM2(I,J)
        C12(I,J)=DUM1(I,J)
        C22(I,J)=CB22(I,J)
        C21(I,J)=DUM1(J,I)
 60   CONTINUE
C
      QA(1,1)=C11(1,1)
      QA(1,2)=C11(1,2)
      QA(1,6)=C11(1,3)
      QA(2,2)=C11(2,2)
```

```
      QA(2,6)=C11(2,3)
      QA(6,6)=C11(3,3)
      QA(2,1)=C11(2,1)
      QA(6,1)=C11(3,1)
      QA(6,2)=C11(3,2)
C
      QA(1,3)=C12(1,1)
      QA(1,4)=C12(1,2)
      QA(1,5)=C12(1,3)
      QA(2,4)=C12(2,2)
      QA(2,5)=C12(2,3)
      QA(6,5)=C12(3,3)
      QA(2,3)=C12(2,1)
      QA(6,3)=C12(3,1)
      QA(6,4)=C12(3,2)
C
      QA(3,3)=C22(1,1)
      QA(3,4)=C22(1,2)
      QA(3,5)=C22(1,3)
      QA(4,4)=C22(2,2)
      QA(4,5)=C22(2,3)
      QA(5,5)=C22(3,3)
      QA(4,3)=C22(2,1)
      QA(5,3)=C22(3,1)
      QA(5,4)=C22(3,2)
C
      QA(3,1)=C21(1,1)
      QA(3,2)=C21(1,2)
      QA(3,6)=C21(1,3)
      QA(4,2)=C21(2,2)
      QA(4,6)=C21(2,3)
      QA(5,6)=C21(3,3)
      QA(4,1)=C21(2,1)
      QA(5,1)=C21(3,1)
      QA(5,2)=C21(3,2)
C
      DO 70 I=1,6
      DO 70 J=1,6
      SA(I,J)=QA(I,J)
 70   CONTINUE
      CALL INVSE (6,SA)
      RETURN
      END
C
C ==================================================================
C
      SUBROUTINE MXMUL(N,PRE,POST,RES)
C
C ==================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PRE(N,1),POST(N,1),RES(N,1)
      DO 10 I=1,N
```

```
      DO 10 J=1,N
        RES(I,J)=0.
        DO 10 K=1,N
          RES(I,J)=RES(I,J)+PRE(I,K)*POST(K,J)
 10   CONTINUE
      RETURN
      END
C
C ==================================================================
C
      SUBROUTINE INVSE(N,A)
C
C==================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N,N)
      DO 10 I=1,N
        X=A(I,I)
        A(I,I)=1.
        DO 20 J=1,N
 20       A(I,J)=A(I,J)/X
        DO 10 K=1,N
          IF(K-I) 30,10,30
 30       X=A(K,I)
          A(K,I)=0.
          DO 40 J=1,N
 40         A(K,J)=A(K,J)-X*A(I,J)
 10   CONTINUE
      RETURN
      END
```

```
C
C ==================================================================
C
      SUBROUTINE OUTPUT
C
C ==================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      character*60  title1, title2, title3
      character*20  filnam
      character*6   jobnam
      character*5   ld$
      character*1   ans, outyp
C
      logical  local, genfil
C
      common /plydat/ THP(50),MAT(50),TP(50),NP,ZUPPER(51),D1111,D1122,
     &     D2222,D1212,D1112,D2212
      common /comp/ S(6,6),Q(6,6)
      common /inout/ THICK,NYN,KFR,TITLE1,TITLE2,TITLE3
      common /system/ jobnam,local,filnam(15),genfil(15)
C
      DIMENSION A(2,2),ALPHAA(10),TQA(10),EPT_MAX(3,3),EPC_MAX(3,3)
C
      if (.not.(local)) go to 1010
      WRITE(6,*) '  PRESS RETURN TO CONTINUE'
      READ(5,1) ANS
    1 FORMAT(A)
      WRITE(6,*)' '
      WRITE(6,*)'    THIS IS THE AVERAGE COMPLIANCE MATRIX  [INV(GPa)]'
      WRITE(6,*)' '
      DO 100 I=1,6
  100   WRITE (6,105) (S(I,J),J=1,6)
  105 FORMAT(1H ,6(F12.7,' '))
      WRITE(6,*)' '
      WRITE(6,*)' '
      WRITE(6,*)'    THIS IS THE AVERAGE STIFFNESS MATRIX  [GPa]'
      WRITE(6,*)' '
      DO 110 I=1,6
  110   WRITE (6,1000) (Q(I,J),J=1,6)
 1000 FORMAT(1H ,6(F9.4,' '))
      WRITE(6,*)' '
      WRITE(6,*)' '
      WRITE(6,*)'  PRESS RETURN TO CONTINUE'
      READ(5,1) ANS
      WRITE(6,*)' '
      WRITE(6,*)' '
      WRITE(6,*)'          THE AVERAGED CONSTITUTIVE PROPERTIES'
      WRITE(6,*)'**WARNING!  THESE RESULTS ARE ONLY A SIMPLE AVERAGE**'
      WRITE(6,*)' '
 1010 E11=1./S(1,1)
      ENU12=-S(1,2)*E11
      ENU13=-S(1,3)*E11
```

```
      E22=1./S(2,2)
      E33=1./S(3,3)
      G12=1./S(6,6)
      ENU23=-S(2,3)*E22
      G23=1./S(4,4)
      G13=1./S(5,5)
      ENU21=ENU12*E22/E11
      if (local) then
      WRITE(6,*)' '
 1050 FORMAT(A7,F8.3,A4)
 1060 FORMAT(A8,F9.5)
 1070 FORMAT(A7,F9.4,A4)
      WRITE(6,1050)'E11 =',E11,' GPa'
      WRITE(6,1050)'E22 =',E22,' GPa'
      WRITE(6,1050)'E33 =',E33,' GPa'
      WRITE(6,1060)'NU12 =',ENU12
      WRITE(6,1060)'NU21 =',ENU21
      WRITE(6,1060)'NU13 =',ENU13
      WRITE(6,1060)'NU23 =',ENU23
      WRITE(6,1070)'G12 =',G12,' GPa'
      WRITE(6,1070)'G13 =',G13,' GPa'
      WRITE(6,1070)'G23 =',G23,' GPa'
      WRITE(6,*)' '
      WRITE(6,*)' '
      end if
C
C ------------------------------------------------------------------
C
C  THIS SECTION CALCULATES ADDITIONAL VALUES FOR OUTPUT.
C
C ------------------------------------------------------------------
C
C
C  DEFINITION OF THE OUTPUT VARIABLES:
C
C  RECORD
C  NUMBER*   VARIABLE    DESCRIPTION                                UNITS
C
C     1       TITLE1     First title card (up to 60 characters)
C     2       TITLE2     Second title card (up to 60 characters)
C     3       TITLE3     Third title card (up to 60 characters)
C     4       NP         Number of plies in the laminate    MAXIMUM = 50
C                             (integer)
C     5       THICK      Thickness of the laminate                  [mm]
C     6       A(1,1)     A44                                        [MN/m]
C     6       A(1,2)     A45                                        [MN/m]
C     6       A(2,2)     A55                                        [MN/m]
C     7       E11        Etheta     - major axis                    [GPa]
C     7       E33        Ez         - major axis                    [GPa]
C     7       ENU12      nu r-theta - major axis
C     7       ENU13      nu r-z     - major axis
C     7       G13        G r-z      - major axis                    [GPa]
C     8       E22        Etheta     - minor axis                    [GPa]
```

```
C     8       E33        Ez         - minor axis                [GPa]
C     8       ENU21      nu r-theta - minor axis
C     8       ENU23      nu r-z     - minor axis
C     8       G23        G r-z      - minor axis                [GPa]
C     9       D1111      D1111                                  [Nm]
C     9       D1122      D1122                                  [Nm]
C     9       D2222      D2222                                  [Nm]
C     9       D1212      D1212                                  [Nm]
C     9       D1112      D1112                                  [Nm]
C     9       D2212      D2212                                  [Nm]
C
C     * RECORD NUMBER REFERS TO THE STORAGE LOCATION IN THE DIRECT
C       ACCESS FILE.  THIS INFORMATION IS USED IN ANOTHER SECTION OF
C       THIS SUBROUTINE.
C
C
C  CALCULATION OF THE SHEAR STIFFNESS CONSTANTS.
C
      A(1,1) = THICK * Q(4,4)
      A(1,2) = THICK * Q(4,5)
      A(2,2) = THICK * Q(5,5)
C
C  DEFINITION OF THE MATERIAL CONSTANTS USED IN THE MAJOR AXIS ANALYSIS:
C
C         Etheta     = E11
C         Ez         = E33
C         nu r-theta = NU12
C         nu r-z     = NU13
C         G r-z      = G13
C
C  DEFINITION OF THE MATERIAL CONSTANTS USED IN THE MINOR AXIS ANALYSIS:
C
C         Etheta     = E22
C         Ez         = E33
C         nu r-theta = NU21
C         nu r-z     = NU23
C         G r-z      = G23
C
C  THE D MATRIX IS CALCULATED IN SUBROUTINE MATS.
C
C ------------------------------------------------------------------
C
C  THIS SECTION PRINTS THE OUTPUT VARIABLES ON THE COMPUTER SCREEN.
C
C ------------------------------------------------------------------
C
      if (local) then
      WRITE(6,*)'  PRESS RETURN TO CONTINUE'
      READ(5,1) ANS
      WRITE(6,1100)
 1100 FORMAT(1H ,79(1H=)/)
      IF (KFR .EQ. 2) THEN
        WRITE(6,*)'  THE FOLLOWING OUTPUT DATA HAS BEEN SENT TO THE'
```

```
        WRITE(6,*)'  DIRECT ACCESS FILE NAMED ',filnam(2),'.'
        WRITE(6,*)'  THE CONTENTS OF THIS BINARY FILE ARE ALSO STORED'
        WRITE(6,*)'  IN "ROTATE.OUT" SO THAT YOU MAY OBTAIN A PRINTOUT'
        WRITE(6,*)'  OF YOUR DATA.'
        WRITE(6,*)' '
        WRITE(6,1100)
      END IF
      WRITE(6,1110)THICK
 1110 FORMAT(3X,'Thickness of the laminate =',F9.5,' mm'/)
      WRITE(6,1120)A(1,1),A(1,2),A(2,2)
 1120 FORMAT(3X,'A44 =',F8.4,' MN/m'/3X,'A45 =',F8.4,' MN/m'/
     *       3X,'A55 =',F8.4,' MN/m'/)
      WRITE(6,1130)E11,E33,ENU12,ENU13,G13
 1130 FORMAT(3X,'Etheta     - major axis =',F8.3,' GPa'/
     *       3X,'Ez         - major axis =',F8.3,' GPa'/
     *       3X,'nu r-theta - major axis =',F10.6/
     *       3X,'nu r-z     - major axis =',F10.6/
     *       3X,'G r-z      - major axis =',F8.4,' GPa'/)
      IF(NYN.EQ.1) THEN
        WRITE(6,*)'  PRESS RETURN TO CONTINUE'
        READ(5,1) ANS
      END IF
      WRITE(6,1140)E22,E33,ENU21,ENU23,G23
 1140 FORMAT(3X,'Etheta     - minor axis =',F8.3,' GPa'/
     *       3X,'Ez         - minor axis =',F8.3,' GPa'/
     *       3X,'nu r-theta - minor axis =',F10.6/
     *       3X,'nu r-z     - minor axis =',F10.6/
     *       3X,'G r-z      - minor axis =',F8.4,' GPa')
      WRITE(6,1141)D1111,D1122,D2222,D1212,D1112,D2212
 1141 FORMAT(//3X,'D1111 =',F10.5,' Nm'/3X,'D1122 =',F10.5,' Nm'/
     &         3X,'D2222 =',F10.5,' Nm'/3X,'D1212 =',F10.5,' Nm'/
     &         3X,'D1112 =',F10.5,' Nm'/3X,'D2212 =',F10.5,' Nm')
      end if
C
      IF (NYN.EQ.0 .OR. KFR.NE.2) RETURN
C
C ------------------------------------------------------------------
C
C  THE FOLLOWING CODE SENDS THE NECESSARY DATA TO THE DIRECT ACCESS
C  FILE DECLARED BY THE VARIABLE BFILE.
C
C ------------------------------------------------------------------
C
C  OPEN THE DIRECT ACCESS FILE
C
      OPEN(UNIT=12,IOSTAT=INPERR,ERR=2000,FILE=filnam(2),
     *     STATUS='NEW',ACCESS='DIRECT',RECL=110)
C
      WRITE(12,REC=1)TITLE1
      WRITE(12,REC=2)TITLE2
      WRITE(12,REC=3)TITLE3
      WRITE(12,REC=4)NP
      WRITE(12,REC=5)THICK
```

```
      WRITE(12,REC=6)A(1,1),A(1,2),A(2,2)
      WRITE(12,REC=7)E11,E33,ENU12,ENU13,G13
      WRITE(12,REC=8)E22,E33,ENU21,ENU23,G23
      WRITE(12,REC=9)D1111,D1122,D2222,D1212,D1112,D2212
C
C ------------------------------------------------------------
C
C  THIS SECTION TRANSFERS THE REMAINING DATA IN THE MASTER FILE TO THE
C  DIRECT ACCESS FILE.
C
C ------------------------------------------------------------
C
C  DEFINTION OF VARIABLES USED IN THIS SECTION:
C
C  RECORD
C  NUMBER    VARIABLE    DESCRIPTION                          UNITS
C
C     10       A$        Test specimen length in x-direction   [m]
C     10       B         Test specimen length in y-direction   [m]
C     11       RHO       Density of the test specimen          [kg/m↑3]
C     12       PNX       In-plane load in x-direction          [N/m]
C     12       PNY       In-plane load in y-direction          [N/m]
C     13       NX        B.C. number for x-direction (integer)
C     13       NY        B.C. number for y-direction (integer)
C     14       SHEAR     Shear correction factor (real)
C     15       BM        Impactor mass                         [kg]
C     15       VI        Impactor velocity                     [m/s]
C     16       RI        Impactor radius                       [m/s]
C     17       LDHZS     Input loading type HZS
C     18                 <not used>
C     19       R         Radius of the region to be analyzed   [mm]
C     20       NOHARK    Number of harmonics to be used for
C                           calculating the hertzian spring
C                           constant in program BESSEL.
C                           (integer)
C     20       NOHARS    Number of harmonics to be used for
C                           calculating the strain field in
C                           program BESSEL.  (integer)
C     21       NUMALP    Number of alphas to be entered     MAXIMUM = 10
C                            (integer)
C     21       ALPHAA    Array of alphas for K calculation     [mm]
C     22       TQA       Array of total load target values     [N]
C     23       LD$       Input loading type (PT, HZ, or HT)
C             (LOAD)     Variable read in from input file to obtain
C                        a value for LD$.
C                        LOAD --> LD$
C                        ----     ----
C                          1      HZ
C                          2      PT
C                          3      HT
C   For LD$ = HT, the following is also transfered to record 23:
C    (23)      XEP       Loading radius (for loading type HT)  [mm]
C     24       NRS       Number of radius divisions (integer)
```

```
C                                   (For output of the strain field.)
C     25       OUTYP       Flag:  If OUTYP = Y, then the strain
C                                   field is calculated at each
C                                   theta division, each radius step,
C                                   and each ply.
C            (NOUTYP)      Variable read in to get a value for OUTYP.
C                          NOUTYP = 1 ——> OUTYP = Y
C                                 = 0 ——>       = N
C   For OUTYP = Y, the following is also transfered to record 25:
C    (25)      NTH         Theta divisions (integer)          MAXIMUM = 90
C    (25)      EPT_MAX     Matrix of allowable tensile strains
C                                   (tensor notation)
C    (25)      EPC_MAX     Matrix of allowable compressive strains
C                                   (tensor notation)
C     26       IX          Number of x-axis modes to be used in
C                                   program IMPACT.  (integer)
C     26       IY          Number of y-axis modes to be used in
C                                   program IMPACT.  (integer)
C     27       IFC         Flag:  If IFC = 1, then the rotary
C                                   inertia is condensed.  (integer)
C     27       MELIM       Flag:  If MELIM = 1, then the even modes are
C                          eliminated from the analysis in IMPACT
C     28       DELT        Time step length                      [micro_s]
C     28       NT          Number of time steps (integer)
C     29    (not used)     Will contain the final Hertzian Spring Constant
C                          after BESSEL has calculated it.  The variable
C                          name used by BESSEL is XKMEAN
C     30    (not used)     Will contain the peak force and corresponding
C                          acceleration computed by the global model.
C
      READ(11,*)A$,B
      WRITE(12,REC=10)A$,B
      READ(11,*)RHO
      WRITE(12,REC=11)RHO
      READ(11,*)PNX,PNY
      WRITE(12,REC=12)PNX,PNY
      READ(11,*)NX,NY
      WRITE(12,REC=13)NX,NY
      READ(11,*)SHEAR
      WRITE(12,REC=14)SHEAR
      READ(11,*)BM,VI
      WRITE(12,REC=15)BM,VI
      READ(11,*)RI
      WRITE(12,REC=16)RI
      READ(11,1148)LDHZS
 1145 FORMAT(A15)
 1146 FORMAT(A5)
 1147 FORMAT(A5)
 1148 FORMAT(A5)
      WRITE(12,REC=17)LDHZS
      read(12,rec=17)ldhzs
C      READ(11,1145)OFILE
C      WRITE(12,REC=18)OFILE
```

```
      READ(11,*)R
      WRITE(12,REC=19)R
      READ(11,*)NOHARK,NOHARS
      WRITE(12,REC=20)NOHARK,NOHARS
      READ(11,*)NUMALP
      READ(11,*)(ALPHAA(I),I=1,NUMALP)
      WRITE(12,REC=21)NUMALP,(ALPHAA(I),I=1,NUMALP)
      READ(11,*)(TQA(I),I=1,NUMALP)
      WRITE(12,REC=22)(TQA(I),I=1,NUMALP)
C     READ(11,1147)LD$  ------> CHANGED TO: READ(11,*)LOAD
      read(11,*)load
      if (load .eq. 1) then
        ld$='HZ'
      else if (load .eq. 2) then
        ld$='PT'
      else
        ld$='HT'
      end if
      IF (LD$ .EQ. 'HT') READ(11,*)XEP
      WRITE(12,REC=23)LD$
      IF (LD$ .EQ. 'HT') WRITE(12,REC=23)LD$,XEP
      READ(11,*)NRS
      WRITE(12,REC=24)NRS
C     READ(11,1146)OUTYP  ------> CHANGED TO: READ(11,*)NOUTYP
      read(11,*)noutyp
      outyp='N'
      if (noutyp .eq. 1) then
        outyp='Y'
      end if
      IF(OUTYP .EQ. 'Y') THEN
         READ(11,*)NTH
         READ(11,*)EPT_MAX(1,1),EPT_MAX(1,2),EPT_MAX(1,3)
         READ(11,*)EPT_MAX(2,2),EPT_MAX(2,3)
         READ(11,*)EPT_MAX(3,3)
         READ(11,*)EPC_MAX(1,1),EPC_MAX(1,2),EPC_MAX(1,3)
         READ(11,*)EPC_MAX(2,2),EPC_MAX(2,3)
         READ(11,*)EPC_MAX(3,3)
      END IF
      WRITE(12,REC=25)OUTYP
      read(12,rec=25)outyp
      IF(OUTYP .EQ. 'Y') WRITE(12,REC=25)OUTYP,NTH,
     &   EPT_MAX(1,1),EPT_MAX(1,2),EPT_MAX(1,3),
     &   EPT_MAX(2,2),EPT_MAX(2,3),EPT_MAX(3,3),
     &   EPC_MAX(1,1),EPC_MAX(1,2),EPC_MAX(1,3),
     &   EPC_MAX(2,2),EPC_MAX(2,3),EPC_MAX(3,3)
      READ(11,*)IX,IY
      WRITE(12,REC=26)IX,IY
      READ(11,*)IFC
      READ(11,*)MELIM
      WRITE(12,REC=27)IFC,MELIM
      READ(11,*)DELT,NT
      WRITE(12,REC=28)DELT,NT
      aa=0.
```

```
      write(12,rec=29) aa
      write(12,rec=30) aa,aa
C
C
C  CLOSE THE DIRECT ACCESS FILE AND THE INPUT FILE.
C
      CLOSE(UNIT=12)
      CLOSE(UNIT=11)
C
C ------------------------------------------------------------------
C
C  THIS SECTION PRINTS THE CONTENTS OF THE DIRECT ACCESS FILE IN THE
C  FILE 'ROTATE.OUT'.
C
C ------------------------------------------------------------------
C
C
C  OPEN THE FILE 'ROTATE.OUT'.
C
      OPEN(UNIT=30,IOSTAT=INPERR,ERR=2010,FILE='ROTATE.OUT',
     *     STATUS='NEW')
C
      WRITE(30,1151) filnam(2)
 1151 FORMAT(1H0///27X,'CONTENTS OF "ROTATE.OUT"'///8X,'This data is sto
     *red in binary form in the direct access file'/8X,a20//
     *)
      WRITE(30,1155)
 1155 FORMAT(8X,'RECORD'/8X,'NUMBER    CONTENTS'/8X,'------    --------'
     *//)
      WRITE(30,1160)TITLE1,TITLE2,TITLE3
 1160 FORMAT(8X,'1-3',5X,'Title Block:'/13X,A60/13X,A60/13X,A60//
     *       /13X,'Output from ROTATE'/13X,18(1H-)//)
      WRITE(30,1162)NP
 1162 FORMAT(10X,'4',5X,'Number of plies in the laminate =',I4/)
      WRITE(30,1165)THICK
 1165 FORMAT(10X,'5',5X,'Thickness of the laminate =',F9.5,' mm'/)
      WRITE(30,1170)A(1,1),A(1,2),A(2,2)
 1170 FORMAT(10X,'6',5X,'A44 =',F8.4,' MN/m'/10X,'6',5X,'A45 =',F8.4,
     *       ' MN/m'/10X,'6',5X,'A55 =',F8.4,' MN/m'/)
      WRITE(30,1175)E11,E33,ENU12,ENU13,G13
 1175 FORMAT(10X,'7',5X,'Etheta     - major axis =',F8.3,' GPa'/
     *       10X,'7',5X,'Ez         - major axis =',F8.3,' GPa'/
     *       10X,'7',5X,'nu r-theta - major axis =',F10.6/
     *       10X,'7',5X,'nu r-z     - major axis =',F10.6/
     *       10X,'7',5X,'G r-z      - major axis =',F8.4,' GPa'/)
      WRITE(30,1180)E22,E33,ENU21,ENU23,G23
 1180 FORMAT(10X,'8',5X,'Etheta     - minor axis =',F8.3,' GPa'/
     *       10X,'8',5X,'Ez         - minor axis =',F8.3,' GPa'/
     *       10X,'8',5X,'nu r-theta - minor axis =',F10.6/
     *       10X,'8',5X,'nu r-z     - minor axis =',F10.6/
     *       10X,'8',5X,'G r-z      - minor axis =',F8.4,' GPa'/)
      WRITE(30,1181)D1111,D1122,D2222,D1212,D1112,D2212
 1181 FORMAT(10X,'9',5X,'D1111 =',F10.5,' Nm'/
```

```
    *       10X,'9',5X,'D1122 =',F10.5,' Nm'/
    *       10X,'9',5X,'D2222 =',F10.5,' Nm'/
    *       10X,'9',5X,'D1212 =',F10.5,' Nm'/
    *       10X,'9',5X,'D1112 =',F10.5,' Nm'/
    *       10X,'9',5X,'D2212 =',F10.5,' Nm'//
    *       /13X,'Input from Master Input File'/13X,28(1H-)//)
     WRITE(30,1182)A$,B
1182 FORMAT(10X,'10',4X,'Test specimen length in x-direction =',F10.6,
    *  ' m'/10X,'10',4X,'Test specimen length in y-direction =',F10.6,
    *  ' m'/)
     WRITE(30,1184)RHO
1184 FORMAT(10X,'11',4X,'Density of the test specimen =',F8.1,
    &       ' kg/m↑3'/)
     WRITE(30,1186)PNX,PNY
1186 FORMAT(10X,'12',4X,'In-plane load in x-direction =',F8.4,' N/m'/
    *       10X,'12',4X,'In-plane load in y-direction =',F8.4,' N/m'/)
     WRITE(30,1188)NX,NY
1188 FORMAT(10X,'13',4X,'B.C. number for x-direction (integer) =',I3/
    &       10X,'13',4X,'B.C. number for y-direction (integer) =',I3/)
     WRITE(30,1189)SHEAR
1189 FORMAT(10X,'14',4X,'Shear correction factor (real) =',F9.5/)
     WRITE(30,1190)BM,VI
1190 FORMAT(10X,'15',4X,'Impactor mass =',F11.7,' kg'/
    *       10X,'15',4X,'Impactor velocity =',F8.3,' m/s'/)
     WRITE(30,1192)RI
1192 FORMAT(10X,'16',4X,'Impactor radius =',F8.3,' mm'/)
     WRITE(30,1193)LDHZS
1193 FORMAT(10X,'17',4X,'Input loading type HZS =',A5/)
     WRITE(30,1200)
1200 FORMAT(10X,'18',4X,'<unused record>')
     WRITE(30,1210)R
1210 FORMAT(10X,'19',4X,'Radius of the region to be analyzed =',F8.3,
    &       ' mm'/)
     WRITE(30,1220)NOHARK
1220 FORMAT(10X,'20',4X,'Number of harmonics to be used for'
    &  ' calculating the '/20X,'hertzian spring constant in program',
    &  ' BESSEL.'/20X,'(integer) =',I5)
     WRITE(30,1225)NOHARS
1225 FORMAT(10X,'20',4X,'Number of harmonics to be used for'
    &  ' calculating the '/20X,'strain field in program',
    &  ' BESSEL.  (integer) =',I5/)
     WRITE(30,1230)NUMALP
1230 FORMAT(10X,'21',4X,'Number of alphas to be entered (integer) =',
    &       I3)
     DO 1250 I=1,NUMALP
        WRITE(30,1240)I,ALPHAA(I)
1240    FORMAT(10X,'21',4X,'ALPHA(',I2,') =',F10.6,' mm')
1250 CONTINUE
     WRITE(30,*)' '
     DO 1270 I=1,NUMALP
        WRITE(30,1260)I,TQA(I)
1260    FORMAT(10X,'22',4X,'Total load target value (',I2,') =',
    &          F9.2,' N')
```

```
1270 CONTINUE
     WRITE(30,1280)LD$
1280 FORMAT(/10X,'23',4X,'Input loading type (PT, HZ, or HT) =  ',A2/)
     IF (LD$$ .EQ. 'HT') WRITE(30,1290)XEP
1290 FORMAT(10X,'23',4X,'Loading radius (for loading type HT) =',
    &       F8.3,' mm')
     WRITE(30,1300)NRS
1300 FORMAT(10X,'24',4X,'Number of radius divisions'/
    &       20X,'(For output of the strain field.)  (integer) =',I3/)
     WRITE(30,1310)OUTYP
1310 FORMAT(10X,'25',4X,'Flag = ',A1,'   If flag = Y, then the strain',
    &       ' field is'/27X,'calculated at each theta division, '
    &       /27X,'each radius step, and each ply.')
     IF(OUTYP .EQ. 'Y') THEN
        WRITE(30,1320)NTH
1320    FORMAT(10X,'25',4X,'Theta divisions (integer) =',I4)
        WRITE(30,1330)EPT_MAX(1,1),EPT_MAX(1,2),EPT_MAX(1,3),
    &                 EPT_MAX(2,2),EPT_MAX(2,3),EPT_MAX(3,3),
    &                 EPC_MAX(1,1),EPC_MAX(1,2),EPC_MAX(1,3),
    &                 EPC_MAX(2,2),EPC_MAX(2,3),EPC_MAX(3,3)
1330    FORMAT(10X,'25',4X,'Maximum allowable tensile strains ',
    &          '(tensor notation):'/
    &          20X,'Epsilon 11 tensile max =',F11.7/
    &          20X,'Epsilon 12 tensile max =',F11.7/
    &          20X,'Epsilon 13 tensile max =',F11.7/
    &          20X,'Epsilon 22 tensile max =',F11.7/
    &          20X,'Epsilon 23 tensile max =',F11.7/
    &          20X,'Epsilon 33 tensile max =',F11.7/
    &          10X,'25',4X,'Maximum allowable compressive strains ',
    &          '(tensor notation):'/
    &          20X,'Epsilon 11 compressive max =',F11.7/
    &          20X,'Epsilon 12 compressive max =',F11.7/
    &          20X,'Epsilon 13 compressive max =',F11.7/
    &          20X,'Epsilon 22 compressive max =',F11.7/
    &          20X,'Epsilon 23 compressive max =',F11.7/
    &          20X,'Epsilon 33 compressive max =',F11.7/)
     ENDIF
     WRITE(30,1340)IX,IY
1340 FORMAT(/10X,'26',4X,'Number of x-axis modes to be used in'/
    &       20X,'program IMPACT.  (integer) =',I4/
    &       10X,'26',4X,'Number of y-axis modes to be used in'/
    &       20X,'program IMPACT.  (integer) =',I4/)
     WRITE(30,1350)IFC,MELIM
1350 FORMAT(10X,'27',4X,'Flag = ',I1,'   If flag = 1, then the rotary'/
    &       20X,'inertia is condensed.'/
    &       10X,'27',4X,'Flag = ',I1,'   If flag = 1, then the even',
    &       ' modes are'/20X,'eliminated from the analysis in IMPACT.'
    &       /)
     WRITE(30,1360)DELT,NT
1360 FORMAT(10X,'28',4X,'Time step length =',F9.5,' micro_s'/
    &       10X,'28',4X,'Number of time steps (integer) =',I5/)
     WRITE(30,1370)
1370 FORMAT(10X,'29',4X,'Hertzian spring constant.  Added by BESSEL'/)
```

```
      WRITE(30,1380)
 1380 Format(10x,'30',4x,'Peak force computed by IMPACT.'/
     & 10x,'30',4x,'Acceleration at peak force computed by IMPACT.')
C
C  CLOSE THE FILE 'ROTATE.OUT'.
C
      CLOSE(UNIT=30)
C
      RETURN
 2000 WRITE(6,*)'ERROR IN OPENING ',filnam(2),'  STATUS = ',INPERR
      STOP
 2010 WRITE(6,*)'ERROR IN OPENING "ROTATE.OUT".  STATUS = ',INPERR
      STOP
      END
```

```
C=================================================================
C                 startup module
C
C  This module is used by ROTATE, BESSEL, and IMPACT to identify wether
C  or not the program is being executed as part of a complete analysis.
C  If so, variables are initialized according to the "system
C  configuration" and all user interaction is suppressed.
C
C  KEY VARIABLES:
C
C  LOCAL    (logical)  Flag:  If LOCAL = .TRUE. then the program is run
C                      interactively with the user.
C
C  FILNAM  (character) Array of filenames used by the program and by
C                      the complete analysis.
C  GENFIL   (logical)  Array of flags:  If GENFIL(N) = .TRUE. then the
C                      corresponding file contained in FILNAM is created
C
      subroutine startup

      character*20 filnam
      character*6 jobnam

      logical local, genfil, exst

      common /system/ jobnam,local,filnam(15),genfil(15)
C
C See if CONFIG.BIN exists.  If not, then the program is being run
C locally.
C
      inquire(file='config.bin', exist=exst)

C
C Read in CONFIGURATION parameters
C
      if (exst) then
        local = .false.
        open(unit=10, file='config.bin', status='old', access='direct',
     &     recl=30)
        read(10,rec=1) jobnam
        do i=1,15
                read(10,rec=i+1) filnam(i), genfil(i)
        end do
        close(unit=10)
      else
        local = .true.
   10   format(' ','Enter the desired jobname [max 6 characters]: ',$)
   20   format(a6)
   30   write(6,10)
        read(5,20) jobnam
        if (jobnam(1:1) .eq. ' ') then
          write(6,*)'*** Invalid jobname - please re-enter ***'
          go to 30
```

```
      end if
      n=1
40    if (jobnam(n:n) .ne. ' ' .and. n .lt. 7) then
        n=n+1
        go to 40
      end if
      n=n-1
      filnam(1)=jobnam(1:n)//'.dat'
      filnam(2)=jobnam(1:n)//'.bin'
      filnam(3)=jobnam(1:n)//'_angle.dat'
      filnam(4)=jobnam(1:n)//'_final.out'
      filnam(5)=jobnam(1:n)//'_hertz.out'
      filnam(6)=jobnam(1:n)//'_hpres.res'
      filnam(7)=jobnam(1:n)//'_hdisp.res'
      filnam(8)=jobnam(1:n)//'_amajr.res'
      filnam(9)=jobnam(1:n)//'_aminr.res'
      filnam(10)=jobnam(1:n)//'_dynam.out'
      filnam(11)=jobnam(1:n)//'_force.res'
      filnam(12)=jobnam(1:n)//'_displ.res'
      filnam(13)=jobnam(1:n)//'_accel.res'
      filnam(14)=jobnam(1:n)//'_dpres.res'
      filnam(15)=jobnam(1:n)//'_SS.bin'
      do i=1,15
        genfil(i)=.true.
      end do
    end if
    end
C-----&
```

```
      PROGRAM BESSEL
C
C  LOCAL1.FOR
C  CODE LAST EDITED:  7/10/89          VERSION 2  MOD 12
C  COMMENTS LAST EDITED:  9/5/89
C
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C ==================================================================
C  THIS PROGRAM IS FOR CHECKING THE PRESSURE DISTRIBUTION BEING MODELED
C  AS A TRUNCATED SERIES OF BESSEL FUNCTIONS.
C  MOD 7:   THE SECOND LOOP ON THE INPUT LOAD HAS BEEN ELIMINATED.
C           THE OPTIONS OF PLOTTING THE PRESSURE VS. RADIAL POSITION
C           AND LOAD VS. APPROACH FOR SEVERAL # OF HARMONICS HAS BEEN
C           ADDED TO THE KEYBOARD INPUT OPTIONS.
C  MOD 8:   THE TOTAL FORCE FROM THE FOURIER-BESSEL SERIES MODEL OF THE
C           PRESSURE DISTRIBUTION IS NOW CALCULATED.
C  MOD 9:   ADDED THE OPTION OF INPUTING AN ESTIMATED LOAD FOR KEYBOARD
C           INPUT.  IT USED TO ASSUME 1.0 N AUTOMATICALLY.
C  MOD 10:  CHANGED THE INPUTING OF DATA INTO THE *_AMAJR.RES AND
C           *_AMINR.RES FILES SO THAT THEY MAY BE USED WITH PROGRAM
C           KCON.  THIS MAKES IT EASIER TO CHECK THE CONVERGENCE OF K.
C  MOD 11:  DELETED OUTPUT OF IMPACT RADIUS.
C  MOD 12:  WRITE RECOVED FORCE TO DATA FILE.
C ==================================================================
C
C  THIS IS THE SECOND PROGRAM OF ACPI (ANALYSIS OF COMPOSITE PLATE
C  IMPACT.)
C
C ==================================================================
C
C              THIS PROGRAM CALCULATES THE STRESSES AND
C         STRAINS IN A RADIALLY ORTHOTROPIC PLATE DUE TO A
C                           CONTACT LOAD
C
C ==================================================================
C    THIS IS THE SECOND GENERATION PROGRAM FOR THE LOCAL MODEL OF THE
C    IMPACT ANALYSIS PACKAGE ORIGINALLY WRITTEN BY D.S. CAIRNS.  THIS
C    REVISION WAS WRITTEN BY M.J. KRAFT AND MANY CHANGES HAVE BEEN
```

```
C    INCORPORATED.  ADDITION REVISIONS HAVE BEEN MADE BY K.F. RYAN.
C
C        ++++      DO NOT USE THIS PROGRAM BLINDLY, IT WILL       ++++
C        ++++ GIVE YOU STUPID ANSWERS FOR STUPID INPUT PARAMETERS ++++
C ==================================================================
C ==================================================================
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (N1=60,N2=300,N3=40,N4=100)
C
C  PARAMETERS FOR ARRAYS
C
C      N1 = MAXIMUM NUMBER OF LAMINATE PLIES
C      N2 = MAXIMUM NUMBER OF HARMONICS
C      N3 = MAXIMUM NUMBER OF RADIUS DIVISIONS
C      N4 = MAXIMUM NUMBER OF POINTS THAT CAN BE USED TO PLOT PRESSURE
C           VS. RADAIL POSITON
C
      REAL*8
     & EPL(3,3),EP(3,3),THET(0:N2),XT(3,3),TT(3,3),
     & ETT(0:N3,0:N1),ERR(0:N3,0:N1),EZZ(0:N3,0:N1),
     & EPT_MAX(3,3), EPC_MAX(3,3),
     & GA(0:N3,0:N1),STT(0:N3,0:N1),SZZ(0:N3,0:N1),
     & SRR(0:N3,0:N1),SRZ(0:N3,0:N1),SIP(0:N1),A(4,4),B(4),
     & NU1, NU2, NU12, NU22, LDD, K, J0, J1, JO, JP, JD, JT
C
      real output(8)
      CHARACTER*60 TITLE1,TITLE2,TITLE3
      CHARACTER*20 IFILE, filnam
      character*6  jobnam
      character*5 TLD$
      CHARACTER*3 TEST(3,3),LD$
      CHARACTER*1 OUTYP,FLAG1,FLAG3,FLAG4
      CHARACTER*80 XFER
C
      LOGICAL local, genfil
C
      INTEGER V
      EXTERNAL FUNC1,FUNC2
C
      DIMENSION ALPHAA(10),TQA(10),XK(10,2),XKAVG(2),TLOAD(10,2),
     &   PRESS(N4)
C
      common /system/ jobnam, local, filnam(15), genfil(15)
C
C  THE FOLLOWING FILES ARE OPENED IN THIS PROGRAM:
C
C  UNIT    FILE           COMMENTS
C  ----    ----           --------
C
C   12     BINARY INPUT   THE DIRECT ACCESS FILE THAT CONTAINS THE
C                             INPUT DATA FOR THIS PROGRAM AND THE
C                             REMAINING PROGRAMS IN ACPI.
```

```
C   13     '_ANGLE.DAT'    CONTAINS THE ANGLE (IN DEGREES) OF EACH PLY.
C   14     '_HERTZ.OUT'    Text output for Hertzian calculation.
C   14     '_FINAL.OUT'    Text output for stress-strain calculations.
C   16     '_HPRES.RES'    Data file for plotting the pressure at R = 0
C                              VS. # of harmonics in the Hertzian spring
C                              constant calculation.
C   16     '_DPRES.RES'    Data file for plotting the pressure at R = 0
C                              VS. # of harmonics in the stress-strain
C                              calculations.
C   17     '_HDISP.RES'    Data file for plotting the approach alpha VS.
C                              # of harmonics.
C                              (VALID FOR LOADING OPTION HZS ONLY.)
C   27... '_AMAJR.RES'      DATA FILE FOR PLOTTING THE TOTAL LOAD VS. THE
C                              APPROACH FOR THE MAJOR AXIS.
C   37... '_AMINR.RES'      DATA FILE FOR PLOTTING THE TOTAL LOAD VS. THE
C                              APPROACH FOR THE MINOR AXIS.
C   25       (SCRATCH)      SCRATCH FILE USED TO HELP FORMAT FINAL OUTPUT
C                              MAJOR AXIS OUTPUT DATA
C   26       (SCRATCH)      SCRATCH FILE USED TO HELP FORMAT FINAL OUTPUT
C                              MINOR AXIS OUTPUT DATA
C
C ==================================================================
C
C  NOMENCLATURE     (equations referenced are in D. S. Cairns' thesis.)
C
C  A          matrix in Ax=B to find Am, Bm, Cm, and Dm
C  AA         a in Eq.(3.2.8)
C  AAA        Lower limit of integration for subroutine QROMB
C  ACC        acceleration at the center of the laminate
C  ALPHA      input indentation
C  ALPHAA     array of input indentations
C  ALPHAI     geometric indentation
C  AM         coefficients of the stress function
C  A1         a11 in Eq.(3.2.4)
C  A2         a12 in Eq.(3.2.4)
C  A3         a13 in Eq.(3.2.4)
C  A4         a33 in Eq.(3.2.4)
C  A5         a44 in Eq.(3.2.4)
C  B          vector in Ax=B to find Am, Bm, Cm, and Dm
C  BB         b in Eq.(3.2.8)
C  BBB        Upper limit of integration for subroutine QROMB
C  BETAM      coefficients in the Fourier-Bessel expansion of the
C             contact pressure  Ref: Eq.(3.2.16)
C  BGM        parameter in Simpson's integration routine
C  BM         coefficients of the stress function
C  CC         c in Eq.(3.2.8)
C  CM         coefficients of the stress function
C  CS1Z       = COSH(S1*WM*Z)
C  CS2Z       = COSH(S2*WM*Z)
C  DAM$       ='Y' if damage prediction is required
C             ='N' if damage prediction in not required
C  DD         d in Eq.(3.2.8)
C  DELALP     delta alpha
```

```
C  DISP       relative displacement between the top and bottom
C             surface at the center of the laminate
C  DISPBOT    displacement of the bottom surface at the center of the
C             laminate
C  DISPTOP    displacement of the top surface at the center of the
C             laminate
C  DM         coefficients of the stress function
C  EP         strains relative to ply axes
C  EPC_MAX    ply strengths in compression
C  EPL        strains (axisymmetric) relative to laminate axes
C  EPT_MAX    ply strengths in tension
C  ERR        epsilon rr
C  ETT        epsilon theta-theta
C  EXST       flag:  if exst = .true. then 'config.bin' exists
C  EZZ        epsilon zz
C  FACT1      coefficient of Eq. (3.2.16b)
C  FACT2      coefficient of Eq. (3.2.30b)
C  FB         total force from the Fourier-Bessel series model of the
C             pressure distribution
C  FD         second derivative of f(z)
C  FILNAM     array of filenames used by the program and by
C             the complete analysis
C  FLAG1      ='Y' when an input file is being used
C             ='N' when data is inputed via the keyboard
C  FORCE      recovered max. contact pressure [p(r=0) in Eq. 3.2.16]
C  FP         first derivative of f(z)
C  FT         third derivative of f(z)
C  FUNC1      integrand of Eq. (3.2.16b) with PZ factored out
C  FUNC2      integrand of Eq. (3.2.30b)
C  FZ         f(z) in the stress fuction
C  G          transverse shear modulus of the laminate  G r-z  [Pa]
C  GA         shear strain
C  GAM        inertia loading at the center of the laminate
C  GAMMAM     coefficients in the Fourier-Bessel expansion of the
C             inertia loading (Eq. 3.2.30)
C  GENFIL     array of flags:  If GENFIL(N) = .TRUE. then the
C             corresponding file contained in FILNAM is created
C  G2         minor axis G
C  HZ         stress/strain and damage prediction option
C  HZS        static indentation option
C  I          counter
C  IFILE
C  JD         second derivative of the Bessel function at a given r
C  JO         Bessel function of order zero at a given r
C  JOBNAM     jobname used for the analysis
C  JP         first derivative of the Bessel function at a given r
C  JT         third derivative of the Bessel function at a given r
C  J0         Bessel function of order zero
C  J1         Bessel function of order one
C  K
C  KFLAG2     If KFLAG2 = 1, then the Hertzian spring constant is
C             being calculated.
C             If KFLAG2 = 2, then the stresses and strains are
```

```
C              being calculated.
C  KFLAG4      If KFLAG4 = 1, then data is created for plotting the
C              pressure vs. radial position.
C  KFLAG3      If KFLAG3 = -1, then only data for the major axis is
C              generated.
C  KOUNT1      Do loop counter - major/minor axis
C  KOUNT2      Do loop counter - various alphas
C  KWFLAG      flag used in formating data output
C  L           counter
C  LDD
C  LD$         Input loading type:
C                 Stress/strain options:
C                     'PT'  Point load
C                     'HZ'  Hertzian distribution
C                     'HT'  Hat load
C                 Hertzian spring constant option:
C                     'HZS'  Hertzian, force finding option
C  LOCAL       Flag:  If LOCAL = .TRUE. then the program is run
C              interactively with the user.
C  M           # of modes used
C  MJMN        MJMN = KFLAG3 + 2
C  N           counter
C  NRA         counter for radius divisions
C  NRS         # of radius divisions
C  NTH         # of angle divisions
C  NUMALP      number of alphas used to find the Hertzian spring constant
C  NU1         in-plane Poisson's ratio  nu r-theta
C  NU12        minor axis NU1
C  NU2         throuth-the-thickness Poissin's ratio  nu r-z
C  NU22        minor axis NU2
C  NZE         counter for the thickness divisions
C  NZT         # of thickness divisions (= 2 x # of plies + 1)
C  N1          max. # of plies
C  N2          max. # of modes (harmonics)
C  N3          max. # of radius divisions
C  OUTPUT
C  OUTYP       Flag:  if OUTYP = 'Y', then the strain is calculated at
C              each theta division, each raduis step, and each ply.
C  PI          3.14159265359
C  PZ          max. contact pressure (Ref: p0 in Fig. 3.2)  [Pa]
C  P1          function for contact pressure
C  P2          function for inertia loading
C  R           radius of the local region (Ref: Rd)  [m]
C  RD          radial coordinate (r)  [m]
C  RHO         density of the laminate
C  RI          radius of the impactor (Ref: Ri)
C  RINT        radius position (dummy variable)
C  SIP
C  SRR         sigma rr
C  SRZ         sigma rz
C  SS*         integral of the function FUNC* from AAA to BBB
C  SSRR        contribution to sigma rr of each mode
C  SSRZ        contribution to sigma rz of each mode
```

```
C  SSTT       contribution to sigma theta-theta of each mode
C  SSZZ       contribution to sigma zz of each mode
C  SS1Z       = SINH(S1*WM*Z)
C  SS2Z       = SINH(S2*WM*Z)
C  STT        sigma theta-theta
C  SZZ        sigma zz
C  S1         s1 in Eq.(3.2.12)
C  S2         s2 in Eq.(3.2.12)
C  T          thickness of the laminate  [m]
C  TEST       marker for damage prediction
C  THET       ply angles
C  THP        angle between ply angle and angle division
C  THT        angle division
C  TITLE1     First title card (up to 60 characters)
C  TITLE2     Second title card (up to 60 characters)
C  TITLE3     Third title card (up to 60 characters)
C  TLD$
C  TLOAD
C  TQ         total load applied (Ref: P)  [N]
C  TQA        array of input trial loads  [N]
C  TT
C  V          counter for mode #
C  WM         omega-m in Eq.(3.2.14)
C  XEP        radius of contact  (Ref: Rc in Fig. 3.3)  [m]
C  XET        in-plane stiffness  E in-plane  [Pa]
C  XET2       minor axis XET
C  XEZ        through-the-thickness stiffness  Ez  [Pa]
C  XEZ2       minor axis XEZ
C  XFER
C  XK         Hertzian spring constant [N/m↑1.5]
C  XKAVG      average Hertzian spring constant for major axis [N/m↑1.5]
C             average Hertzian spring constant for minor axis [N/m↑1.5]
C  XKMEAN     average of the major axis and minor axis
C             Hertzian spring constants [N/m↑1.5]
C  XT
C  Z          through-the-thickness coordinate with z=0 at the midplane
C             of the laminate
C  ZEROJ0     roots of the Bessel function of order zero
C
      PI = 3.1415926535897932D0
      OPEN(UNIT=1,FILE='RFORCE.RES',STATUS='NEW')
C
C PROGRAM INTRODUCTION
C
      call startup
      if (local) then
        WRITE(6,10)
   10   FORMAT(1H1,79(1H=)//5X,'THIS IS THE SECOND',
     &   ' PROGRAM OF ACPI (ANALYSIS OF',
     &   ' COMPOSITE PLATE IMPACT).'//79(1H=)//5X,
     &   'PROGRAM BESSEL'//5X,'THIS PROGRAM',
     &   ' COMPUTES THE HERTZIAN SPRING CONSTANT',
     &   ' PLUS THE STRESSES AND'/5X,'STRAINS',
```

```
     &   ' IN A RADIALLY ORTHOROPIC PLATE DUE',
     &   ' TO A CONTACT LOAD.'//79(1H=))
        WRITE(6,20)
   20   FORMAT(1H0,'Is this program using a binary input',
     &' file? (Y/N):',$)
        READ(5,30)FLAG1
   30   FORMAT(A)
      end if
C
C  FLAG1 = 'Y':  AN INPUT FILE IS BEING USED.
C  FLAG1 = 'N':  DATA IS INPUTED VIA THE KEYBOARD.
C
C ==================================================================
C
C  THIS IS THE INPUT SECTION USED WHEN AN INPUT FILE IS BEING USED.
C
C ==================================================================
C
      if (.not.(local)) flag1 = 'Y'
      IF (FLAG1 .EQ. 'Y' .or. FLAG1 .EQ. 'y') THEN
         if (local) then
               WRITE(6,40)
   40          FORMAT(//' IF YOU ARE CALCULATING THE',
     &           ' HERTZIAN SPRING CONSTANT, ENTER',
     &           ' "1".'//' IF YOU ARE CALCULATING THE',
     &           ' STRESSES AND STRAINS, ENTER "2".'/)
               READ(5,*)KFLAG2
         end if
C
C  KFLAG2 = 1:  CALCULATING THE HERTZIAN SPRING CONSTANT
C  KFLAG2 = 2:  CALCULATING THE STRESSES AND STRAINS
C

   45    OPEN(UNIT=12,IOSTAT=KOPN16,ERR=60,FILE=filnam(2),
     &      STATUS='OLD',ACCESS='DIRECT',RECL=110)
         read(12,rec=4) np
         if (.not. (local)) then
               read(12,rec=30) aa, bb
               kflag2=1
               if (aa .ne. 0. .and. bb .ne. 0.) kflag2=2
         end if
C
         READ(12,REC=1)TITLE1
         READ(12,REC=2)TITLE2
         READ(12,REC=3)TITLE3
         READ(12,REC=5)T
C  CONVERT FROM mm TO m.
         T = T*1.D-3
         READ(12,REC=19)R
C  CONVERT FROM mm TO m.
         R = R*1.D-3
         READ(12,REC=7)XET,XEZ,NU1,NU2,G
C  CONVERT FROM GPa TO Pa.
```

```
      XET = XET*1.D9
      XEZ = XEZ*1.D9
      G = G*1.D9
C ------------------------------------------------------------------
C  THIS SECTION READS THE ADDITIONAL INPUT NEEDED TO CALCULATE THE
C  HERTZIAN SPRING CONSTANT.
C ------------------------------------------------------------------
      IF (KFLAG2 .EQ. 1) THEN
         READ(12,REC=17)TLD$
             LD$=TLD$(2:4)
         READ(12,REC=20)M
         NUMHAR = 4
         DELHAR = M/NUMHAR
         READ(12,REC=21)NUMALP,(ALPHAA(I),I=1,NUMALP)
         DO 50 I=1,NUMALP
C  CONVERT FROM mm TO m.
50          ALPHAA(I) = ALPHAA(I)*1.D-3
         READ(12,REC=16)RI
C  CONVERT FROM mm TO m.
         RI = RI*1.D-3
         READ(12,REC=22)(TQA(I),I=1,NUMALP)
      ENDIF
C ------------------------------------------------------------------
C  THIS SECTION READS THE ADDITIONAL INPUT NEEDED TO CALCULATE THE
C  STRESSES AND STRAINS.
C ------------------------------------------------------------------
      IF (KFLAG2 .EQ. 2) THEN
         READ(12,REC=23)LD$
         read(12,rec=20) M, M
C ------------------------------------------------------------------
C  READ INPUT FOR LOAD TYPE PT.
C ------------------------------------------------------------------
         IF (LD$ .EQ. 'PT') THEN
            READ(12,REC=30)LDD,ACC
            READ(12,REC=11)RHO
         ENDIF
C ------------------------------------------------------------------
C  READ INPUT FOR LOAD TYPE HZ.
C ------------------------------------------------------------------
         if (ld$ .eq. 'HZ') then
             read(12,rec=29) H
             read(12,rec=16) RI
C Convert from mm to m
             RI = RI * 1.D-3
             read(12,rec=11) RHO
             read(12,rec=30) LDD, ACC
         end if
C------------------------------------------------------------------
C  READ INPUT FOR LOAD TYPE HT.
C------------------------------------------------------------------
         if (ld$ .eq. 'HT') then
             read(12,rec=23) LD$,XEP
         end if
```

```
C
C-----------------------------------------------------------------
C  READ INPUT COMMON FOR LOAD TYPES PT, HZ, AND HT
C-----------------------------------------------------------------
            read(12,rec=24) NRS
            read(12,rec=4) NZT
            read(12,rec=25) OUTYP
            if (OUTYP .eq. 'Y') then
                  open(unit=13, file=filnam(3), status='old',
     &            access='sequential')
                  do i=1,NZT
                        read(13,*) THET(i)
                  end do
                  close(unit=13)
                  read(12,rec=25) OUTYP, NTH,
     &                  EPT_MAX(1,1), EPT_MAX(1,2), EPT_MAX(1,3),
     &                  EPT_MAX(2,2), EPT_MAX(2,3), EPT_MAX(3,3),
     &                  EPC_MAX(1,1), EPC_MAX(1,2), EPC_MAX(1,3),
     &                  EPC_MAX(2,2), EPC_MAX(2,3), EPC_MAX(3,3)
            end if
         ENDIF
C
         GOTO 90
   60    if (local) then
               write(6,65)
   65          format('Enter input filename: ',$)
               read(5,66) filnam(2)
   66          format(a20)
               go to 45
         else
            STOP
         end if
      END IF
C
C ==================================================================
C
C  THIS INPUT SECTION IS USED WHEN THE DATA IS INPUTED VIA THE KEYBOARD.
C
C ==================================================================
C
      IF (LOCAL) THEN
         WRITE(6,40)
         READ(5,*)KFLAG2
      END IF
C
C  KFLAG2 = 1:  CALCULATING THE HERTZIAN SPRING CONSTANT
C  KFLAG2 = 2:  CALCULATING THE STRESSES AND STRAINS
C
      WRITE(6,67)
   67 FORMAT('$Do you want to see how well the pressure ',
     &      'distribution is being modeled?(Y/N): ')
      READ(5,30)FLAG4
      IF (FLAG4 .EQ. 'Y' .or. FLAG4 .EQ. 'y') THEN
```

```
         KFLAG4 = 1
         OPEN(UNIT=2,FILE='PRESS.RES',STATUS='NEW')
         OPEN(UNIT=3,FILE='PR.RES',STATUS='NEW')
         WRITE(6,*)'Please enter the # of points to calculate ',
     &      'the pressure at (Max = 100):'
         READ(5,*)KIO
      ELSE
         KFLAG4 = 0
      ENDIF
      IF (KFLAG2 .EQ. 1) THEN
         WRITE(6,2112)
 2112    FORMAT(' Number of convergance curves to use in the plot'/
     &      '$   of Load vs. Approach  (Max = 10): ')
         READ(5,*)NUMHAR
         WRITE(6,2114)
 2114    FORMAT(' Increment in total number of harmonics'/
     &      '$   between convergance curves (integer): ')
         READ(5,*)DELHAR
         M = DELHAR*NUMHAR
      ENDIF
      WRITE(6,68)
   68 FORMAT(/,' Please enter three title lines.',
     &       /,' (Maximum number of characters in each line = 60.)',/)
      WRITE(6,2120)1
 2120 FORMAT(' ',I1,':"',60X,'"')
      WRITE(6,2130)1
 2130 FORMAT('+',I1,':"',$)
      READ(5,2200)TITLE1
      WRITE(6,2120)2
      WRITE(6,2130)2
      READ(5,2200)TITLE2
      WRITE(6,2120)3
      WRITE(6,2130)3
      READ(5,2200)TITLE3
 2200 FORMAT(A60)
      WRITE(6,*)' '
      WRITE(6,*)' '
      WRITE(6,*)'Please enter the following parameters ',
     &          'with appropriate units.'
      WRITE(6,71)
   71 FORMAT(//,'$Plate thickness [mm]:  ')
      READ(5,*)T
C CONVERT FROM mm TO m.
      T = T*1.D-3
      WRITE(6,72)
   72 FORMAT('$Radius of the region to be analyzed [mm]:  ')
      READ(5,*)R
C CONVERT FROM mm TO m.
      R = R*1.D-3
      WRITE(6,*)'MAJOR AXIS:'
      WRITE(6,73)
   73 FORMAT('$     E in-plane [GPa]:  ')
      READ(5,*)XET
```

```
      WRITE(6,74)
   74 FORMAT('$     Ez [GPa]:  ')
      READ(5,*)XEZ
      WRITE(6,75)
   75 FORMAT('$     nu r-theta:  ')
      READ(5,*)NU1
      WRITE(6,76)
   76 FORMAT('$     nu r-z:  ')
      READ(5,*)NU2
      WRITE(6,77)
   77 FORMAT('$     G r-z [GPa]:  ')
      READ(5,*)G
C CONVERT FROM GPa TO Pa.
      XET = XET*1.D9
      XEZ = XEZ*1.D9
      G = G*1.D9
      WRITE(6,1460)
 1460 FORMAT(/,'$Do you wish to calculate data for the',
     &' minor axis? (Y/N):')
      READ(5,30)FLAG3
      IF (FLAG3 .EQ. 'Y' .or. FLAG3 .EQ. 'y') THEN
         KFLAG3 = 0
         WRITE(6,*)'MINOR AXIS:'
         WRITE(6,1473)
 1473    FORMAT('$     E in-plane [GPa]:  ')
         READ(5,*)XET2
         WRITE(6,1474)
 1474    FORMAT('$     Ez [GPa]:  ')
         READ(5,*)XEZ2
         WRITE(6,1475)
 1475    FORMAT('$     nu r-theta:  ')
         READ(5,*)NU12
         WRITE(6,1476)
 1476    FORMAT('$     nu r-z:  ')
         READ(5,*)NU22
         WRITE(6,1477)
 1477    FORMAT('$     G r-z [GPa]:  ')
         READ(5,*)G2
C CONVERT FROM GPa TO Pa.
         XET2 = XET2*1.D9
         XEZ2 = XEZ2*1.D9
         G2 = G2*1.D9
      ELSE
         KFLAG3 = -1
      ENDIF
C ------------------------------------------------------------------
C  THIS SECTION READS THE ADDITIONAL INPUT NEEDED TO CALCULATE THE
C  HERTZIAN SPRING CONSTANT.
C ------------------------------------------------------------------
      IF(KFLAG2 .EQ. 1) THEN
         LD$ = 'HZS'
         WRITE(6,2079)
 2079    FORMAT('$Number of alphas to use in finding the Hertzian ',
```

```
     &      'spring constant (Max = 10):  ')
        READ(5,*)NUMALP
        WRITE(6,2080)
 2080   FORMAT('$Increment in alpha [mm]  (suggest 0.025 mm):  ')
        READ(5,*)DELALP
        ALPHAA(1) = DELALP*1.D-3
        DO 2081 L=2,NUMALP
           ALPHAA(L) = ALPHAA(L-1) + DELALP*1.D-3
 2081   CONTINUE
        WRITE(6,2082)
 2082   FORMAT('$Estimate the required load [N]  (suggest 1.0 N):  ')
        READ(5,*)TQA(1)
        DO 79 I=1,NUMALP
   79      TQA(I) = TQA(1)
        WRITE(6,81)
   81   FORMAT('$Impactor radius [mm]:  ')
        READ(5,*)RI
C CONVERT FROM mm TO m.
        RI = RI*1.D-3
      ENDIF
C ------------------------------------------------------------------
C  THIS SECTION READS THE ADDITIONAL INPUT NEEDED TO CALCULATE THE
C  STRESSES AND STRAINS.
C ------------------------------------------------------------------
      IF(KFLAG2 .EQ. 2) THEN
        WRITE(6,82)
   82   FORMAT(//)
        WRITE(6,*)'THE AVAILABLE LOADING TYPES FOR THIS ANALYSIS:'
        WRITE(6,*)' '
        WRITE(6,*)'   PT  -  POINT LOAD'
        WRITE(6,*)'   HZ  -  HERTZIAN DISTRIBUTION'
        WRITE(6,*)'   HT  -  HAT LOAD'
        WRITE(6,*)' '
        WRITE(6,*)'PLEASE INPUT THE LOADING TYPE PT, HZ, OR HT'
        READ(5,70)LD$
   70   FORMAT(A)
C ------------------------------------------------------------------
C  READ INPUT FOR LOAD TYPE PT.
C ------------------------------------------------------------------
        IF (LD$.EQ.'PT') THEN
           WRITE(6,*)'-----------------------------------------------'
           WRITE(6,*)'                LOAD TYPE PT OPTION'
           WRITE(6,*)'-----------------------------------------------'
           WRITE(6,*)' '
           WRITE(6,*)'Please enter the following parameters ',
     &                'with appropriate units.'
           WRITE(6,83)
   83      FORMAT('$Peak force computed by IMPACT [N]:  ')
           READ(5,*)LDD
           WRITE(6,84)
   84      FORMAT('$Acceleration at peak force computed by IMPACT',
     &             ' [m/s*2]:  ')
           READ(5,*)ACC
```

```
            WRITE(6,85)
   85       FORMAT('$Density of the test specimen [kg/m↑3]:  ')
            READ(5,*)RHO
         ENDIF
C ------------------------------------------------------------------
C  READ INPUT FOR LOAD TYPE HZ.
C ------------------------------------------------------------------
         IF(LD$.EQ.'HZ') THEN
            WRITE(6,*)'------------------------------------------------'
            WRITE(6,*)'                LOAD TYPE HZ OPTION'
            WRITE(6,*)'------------------------------------------------'
            WRITE(6,*)' '
            WRITE(6,*)'Please enter the following parameters ',
     &                'with appropriate units.'
            WRITE(6,*)' '
            WRITE(6,*)' '
            WRITE(6,86)
   86       FORMAT('$Hertzian spring constant [N/m↑1.5]:  ')
            READ(5,*)H
            WRITE(6,81)
            READ(5,*)RI
C CONVERT FROM mm TO m.
            RI = RI*1.D-3
            WRITE(6,85)
            READ(5,*)RHO
            WRITE(6,83)
            READ(5,*)LDD
            WRITE(6,84)
            READ(5,*)ACC
         ENDIF
C------------------------------------------------------------------
C  READ INPUT FOR LOAD TYPE HT.
C------------------------------------------------------------------
         IF (LD$.EQ.'HT') THEN
            WRITE(6,*)'------------------------------------------------'
            WRITE(6,*)'                LOAD TYPE HT OPTION'
            WRITE(6,*)'------------------------------------------------'
            WRITE(6,*)' '
            WRITE(6,*)' '
            WRITE(6,87)
   87       FORMAT('$Please enter the loading radius [mm]:  ')
            READ(5,*)XEP
C CONVERT FROM mm TO m.
            XEP = XEP*1.D-3
         ENDIF
C------------------------------------------------------------------
C  READ INPUT COMMON FOR LOAD TYPES PT, HZ, AND HT
C------------------------------------------------------------------
         WRITE(6,*)'Please enter the following parameters:'
         WRITE(6,*)' '
         WRITE(6,*)' '
         WRITE(6,88)
   88    FORMAT(' Number of radius divisions',/,
```

```
     &          '$(For output of the strain field.)  [integer]:  ')
        READ(5,*)NRS
        WRITE(6,89)
   89   FORMAT('$Number jof plies in the laminate:  ')
        READ(5,*)NZT
        WRITE(6,91)
   91   FORMAT(' Do you want the strain field calculated at various ',
     &          /,'$theta divisions?  [Y/N]:  ')
        READ(5,*)OUTYP
        IF(OUTYP .EQ. 'Y') THEN
           DO I=1,NZT
              WRITE(6,92) I
   92         FORMAT('$Angle of ply ',I2,' [deg]:  ')
              READ(5,*)THET(I)
           END DO
           WRITE(6,93)
   93      FORMAT(' Enter the number of theta divisions',/,
     &            '$(the first quadrant is divided by this number):  ')
           READ (5,*)NTH
C
C  INPUT THE STRAIN FAILURE CRITERIA
C
           WRITE(6,*)' '
           WRITE(6,*)'PLY TENSILE STRAIN TO FAILURE CRITERIA [STRAIN]',
     &        ' [TENSOR NOTATION]'
           WRITE(6,*)' '
           WRITE(6,*)'EPT_MAX(1,1), EPT_MAX(1,2), EPT_MAX(1,3)'
           READ(5,*)  EPT_MAX(1,1), EPT_MAX(1,2), EPT_MAX(1,3)
C
           WRITE(6,*)'EPT_MAX(2,2), EPT_MAX(2,3)'
           READ(5,*)  EPT_MAX(2,2), EPT_MAX(2,3)
C
           WRITE(6,*)'EPT_MAX(3,3)'
           READ(5,*)  EPT_MAX(3,3)
           WRITE(6,*)' '
C
           WRITE(6,*)'PLY COMPRESSIVE STRAIN TO FAILURE CRITERIA ',
     &                '[STRAIN] [TENSOR NOTATION]'
           WRITE(6,*)' '
           WRITE(6,*)'EPC_MAX(1,1), EPC_MAX(1,2), EPC_MAX(1,3)'
           READ(5,*)  EPC_MAX(1,1), EPC_MAX(1,2), EPC_MAX(1,3)
C
           WRITE(6,*)'EPC_MAX(2,2), EPC_MAX(2,3)'
           READ(5,*)  EPC_MAX(2,2), EPC_MAX(2,3)
C
           WRITE(6,*)'EPC_MAX(3,3)'
           READ(5,*)  EPC_MAX(3,3)
C
        ENDIF
     ENDIF
C
C ==================================================================
C  END OF THE INPUT SECTION
```

```
C ==================================================================
C
C
C ==================================================================
C
C  INITILIZE SOME VALUES
C
C ==================================================================
C
   90 IF(LD$ .EQ. 'HZS') THEN

         GAM = 0.0D0
         ACC = 0.0D0
         OUTYP = 'N'
         NRS = 0
         NZT = 2
         XKAVG(1) = 0.0D0
         XKAVG(2) = 0.0D0
      else if (LD$ .eq. 'PT') then
                GAM = -RHO*ACC
                QIMP = LDD
      else if (LD$ .eq. 'HZ') then
                TQ = LDD
                ALPHA = (TQ / H)**(2.0D0/3.0D0)
                GAM = -RHO*ACC
                XEP = sqrt(RI**2.D0-(RI-ALPHA)**2.D0)
                PZ = 3.0D0*TQ/(2.0D0*PI*XEP**2.D0)
      ENDIF
      if (LD$ .ne. 'HZS' .and. OUTYP .eq. 'Y') then
        do ii=1,3
                do jj=1,ii
                        EPT_MAX(ii,jj) = EPT_MAX(jj,ii)
                        EPC_MAX(ii,jj) = EPC_MAX(jj,ii)
                end do
        end do
      end if
C
      KOUNT1 = 0
C
C  KWFLAG IS AN OUTPUT FLAG FOR FORMATTING THE FINAL OUTPUT IF
C  WE CALCULATE THE HERTZIAN SPRING CONSTANTS.  THE OUTPUT FORMAT IS:
C  PAGE 1: TITLE INFORMATION, PLATE THICKNESS, RADIUS OF ANALYZED
C          REGION, NUMBER OF HARMONICS, IMPACTOR RADIUS, IMPACT RADIUS,
C          AND THE HERTZIAN SPRING CONSTANTS FOR EACH ALPHA WITH THE
C          CORRESPONDING LOAD FOR EACH MAJOR AND MINOR AXIS CALCULATIONS
C
C  PAGE 2: TITLE INFORMATION; MAJOR AXIS INFORMATION (Etheta, Ez,
C          nu r-theta, nu r-z, and G r-z); AND APPROACH, PRESSURE,
C          AND LOAD INFORMATION FOR EACH ALPHA
C
C  PAGE 3: SAME AS PAGE 2 EXCEPT INFORMATION IS FOR THE MINOR AXIS
C
C  KWFLAG = 1 INDICATES THAT THE FIRST PAGE HAS BEEN SENT TO A FILE
```

```
C
      KWFLAG = 0

      if (kflag2 .eq. 1) then
        if (genfil(5)) open(unit=14, file=filnam(5), status='new')
        if (genfil(6)) THEN
           open(unit=16,file=filnam(6),status='new')
           WRITE(16,*) 0.D0,0.D0
        ENDIF
      else
        open(unit=14, file=filnam(4), status='new')
        if (genfil(14)) THEN
           open(unit=16, file=filnam(14), status='new')
           WRITE(16,*) 0.D0,0.D0
        ENDIF
      end if
      if (ld$ .ne. 'HZS') go to 100
      if (kflag2 .eq. 1) then
        if (genfil(7)) THEN
           open(unit=17, file=filnam(7), status='new')
           WRITE(17,*) 0.D0,0.D0
        ENDIF
        open(unit=25, status='scratch')
        open(unit=26, status='scratch')
      end if
         IF(GENFIL(8)) THEN
            DO 95 I=1,NUMHAR
               J=26+I
               OPEN(UNIT=J,FILE=FILNAM(8),STATUS='NEW')
               WRITE(J,*) 0.0, 0.0
               IF(KFLAG3 .EQ. 0) THEN
                  IF(GENFIL(9)) THEN
                     L=36+I
                     OPEN(UNIT=L,FILE=FILNAM(9),STATUS='NEW')
                     WRITE(L,*) 0.0, 0.0
                  ENDIF
               ENDIF
   95       CONTINUE
         ENDIF
C
C  START OF DO LOOPS FOR HERTZIAN SPRING CONSTANT CALCULATION
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C  FIRST LOOP FOR CALCULATING THE HERTZIAN SPRING CONSTANT.
C
      MJMN = KFLAG3 + 2
      DO 447 KOUNT1=1,MJMN
C
C  WHEN KOUNT1 = 1, THE LOOP IS MADE FOR THE MAJOR AXIS.
C  WHEN KOUNT1 = 2, THE LOOP IS MADE FOR THE MINOR AXIS.
C
         IF(KOUNT1 .EQ. 2) THEN
```

```
            IF(FLAG1 .EQ. 'Y' .OR. FLAG1 .EQ. 'y') THEN
               READ(12,REC=8)XET,XEZ,NU1,NU2,G
C  CONVERT FROM GPa TO Pa
               XET = XET*1.D9
               XEZ = XEZ*1.D9
               G = G*1.D9
            ELSE
               XET = XET2
               XEZ = XEZ2
               NU1 = NU12
               NU2 = NU22
               G = G2
            ENDIF
         ENDIF
C
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C  SECOND LOOP FOR CALCULATING THE HERTZIAN SPRING CONSTANT.
C  LOOP ON EACH INPUTED APPROACH ALPHA.
C
         DO 445 KOUNT2=1,NUMALP

               ALPHA = ALPHAA(KOUNT2)
               TQ = TQA(KOUNT2)
C
C ==================================================================
C
C  CORE OF THE THREE DO LOOPS
C
C ==================================================================
C
  100          IF((LD$.EQ.'HZ') .OR. (LD$.EQ.'HZS')) THEN
C
C  COMPUTE FOOTPRINT SIZE FROM GEOMETRIC RELATIONSHIP - REF:  Eqs.
C  (3.2.22) and (3.2.24)
C
                  XEP = SQRT(RI**2.D0-(RI-ALPHA)**2.D0)
                  PZ = 3.D0*TQ/(2.D0*PI*XEP**2.D0)
               ENDIF
C  Eq. (3.2.4)
               A1 = 1.D0/XET
               A2 = -NU1/XET
               A3 = -NU2/XET
               A4 = 1.D0/XEZ
               A5 = 1.D0/G
C
C DETERMINING THE COEFFICIENTS - REF: Eqs.(3.2.8)
C
               AA = A3*(A1-A2)/(A1*A4-A3**2.D0)
               BB = (A3*(A3+A5)-A2*A4)/(A1*A4-A3**2.D0)
               CC = (A3*(A1-A2)+A1*A5)/(A1*A4-A3**2.D0)
               DD = (A1**2.D0-A2**2.D0)/(A1*A4-A3**2.D0)
C
```

```
C DETERMINING THE EIGENVALUES
C
              S1 = SQRT((AA+CC+SQRT((AA+CC)**2.D0-4.D0*DD))/(2.D0*DD))
              S2 = SQRT((AA+CC-SQRT((AA+CC)**2.D0-4.D0*DD))/(2.D0*DD))
C
C INITIALIZE DISPLACEMENT AND SUM MODES
C
              DISP = 0.0D0
              FORCE = 0.0D0
              FB = 0.0D0
              DO 99 KIE=1,KI0
                 PRESS(KIE) = 0.0D0
   99         CONTINUE
C
      if (local) then
              WRITE(6,*)' '
              WRITE(6,*)'     MODE       APPROACH ALPHA'
              WRITE(6,*)' '
      end if
      IF (KFLAG2 .EQ. 1) KV=1
C
              DO 190 V = 1,M
                 WM = ZEROJ0(V)/R
C
C DETERMINING THE NON-HOMOGENEOUS PORTION
C
                 IF (LD$.EQ.'PT') THEN
                    BETAM = LDD / (PI * J1(ZEROJ0(V))**2 * R**2)
                    FORCE = BETAM + FORCE
                 ENDIF
C
                 IF (LD$.EQ.'PA') THEN
                    U = WM*XEP
                    C0 = -2./(R**2*ZEROJ1(V)**2)*(1./WM**2*U*J0(X)-
     &                 1./(U**2*WM**2)*(U*(U**2-4)*J1(X)+2.*U**2*J0(X)))
                 ENDIF
C
                 IF (LD$.EQ.'HT') THEN
                    C0 = -2.*J1(X)*10.*X/(XEP**2*R**2*PI*ZEROJ1(V)**2
     &                 *WM**5*XEP**2*PI)
                 ENDIF
C
C   NUMERICAL INTEGRATION FOR A HERTZIAN LOADING DISTRIBUTION
C      TO DETERMINE COEFFICIENT BETAm -  Eq. (3.2.16b)
C
                 XLO = 0.0D0
                 IF ((LD$.EQ.'HZ') .OR. (LD$.EQ.'HZS')) THEN
                    AAA = 0.D0
                    BBB = XEP
                    CALL QROMB(FUNC1,AAA,BBB,SS1,XEP,WM)
                    FACT1 = 2.D0 / (J1(ZEROJ0(V))*R)**2.D0
                    BETAM = FACT1 * PZ * SS1
C      COMPUTE RECOVERED PRESSURE AT CENTER OF PLATE (r=0)
```

```
C       REF:  EQ. (3.2.16a) - NOTE THAT J0(0)=1
                      FORCE = FORCE + BETAM
C       COMPUTE RECOVERED FORCE (FB).
                      FB = FB + 2.D0*PI*R*BETAM*J1(WM*R)/WM
                      WRITE(1,*) V,FB
C     COMPUTE DATA TO PLOT THE PRESSURE VS. RADIAL POSITION (IF DESIRED).
                      IF ((KFLAG4 .EQ. 1) .AND. ((KOUNT1 .EQ. 1 .AND.
     & KOUNT2 .EQ. INT(NUMALP/2)+1) .or. (kflag2 .eq. 2))) THEN
                         DO 159 KIE=1,KIO
                            YR = (REAL(KIE)/REAL(KIO))**3.D0*R
                            PRESS(KIE) = PRESS(KIE) + BETAM*J0(WM*YR)
  159                    CONTINUE
                       ENDIF
                    ENDIF
C
C          NUMERICAL INTEGRATION FOR INERTIAL LOADING BODY FORCE
C          DISTRIBUTION TO DETERMINE COEFFICIENT GAMMAm - Eq. (3.2.30b)
                    IF(LD$ .EQ. 'HZS') THEN
                       GAMMAM = 0.0D0
                    ELSE
                       AAA = 0.D0
                       BBB = XEP
                       CALL QROMB(FUNC2,AAA,BBB,SS2,XEP,WM)
                       FACT2 = -2.D0*GAM / (J1(ZEROJ0(V))*R)**2.D0
                       GAMMAM = FACT2 * SS2
                    ENDIF
C
C SOLVE FOR MODAL AMPLITUDES AM, BM, CM, AND DM FOR FM(Z) SOLUTION
C
C     SET UP MATRIX FOR Ax = B SOLUTION FOR MODAL AMPLITUDES
C
                    A(1,1) = (-CC*S1 + DD*S1**3.D0) * COSH(-WM*S1*T/2.D0)
                    A(1,2) = (-CC*S1 + DD*S1**3.D0) * SINH(-WM*S1*T/2.D0)
                    A(1,3) = (-CC*S2 + DD*S2**3.D0) * COSH(-WM*S2*T/2.D0)
                    A(1,4) = (-CC*S2 + DD*S2**3.D0) * SINH(-WM*S2*T/2.D0)
C
                    A(2,1) =  A(1,1)
                    A(2,2) = -A(1,2)
                    A(2,3) =  A(1,3)
                    A(2,4) = -A(1,4)
C
                    A(3,1) = ( 1.D0 - AA*S1**2.D0) * SINH( WM*S1*T/2.D0)
                    A(3,2) = ( 1.D0 - AA*S1**2.D0) * COSH( WM*S1*T/2.D0)
                    A(3,3) = ( 1.D0 - AA*S2**2.D0) * SINH( WM*S2*T/2.D0)
                    A(3,4) = ( 1.D0 - AA*S2**2.D0) * COSH( WM*S2*T/2.D0)
C
                    A(4,1) = -A(3,1)
                    A(4,2) =  A(3,2)
                    A(4,3) = -A(3,3)
                    A(4,4) =  A(3,4)
C
                    B(1) = ( -BETAM + GAMMAM*T/2.D0 ) / WM**3.D0
                    B(2) =  -GAMMAM*T/2.D0 / WM**3.D0
```

```
                B(3) = 0.0D0
                B(4) = 0.0D0
C
C  CALL GAUSS-JORDON ELIMINATION SUBROUTINE TO FIND AMPLITUDES
C
                CALL GAUSSJ(A,4,4,B,1,1)
C
                AM = B(1)
                BM = B(2)
                CM = B(3)
                DM = B(4)
C
                DO 170 NRA = 0, NRS
                   DO 160 NZE = 1, NZT
C
C     CALCULATE FUNCTION VALUES AT PLY MIDPLANE, UNLESS RUNNING
C  ANALYSIS FOR FORCE GIVEN ALPHA, WHERE FUNCTIONS ARE CALCULATED
C                     ONLY AT +H/2 AND -H/2
C
                        IF (LD$ .NE. 'HZS') THEN
                           Z = -T/2.D0+T/(2.D0*NZT)+T*(NZE-1)/NZT
                        ELSE
                           Z = -T/2.D0 + (NZE-1)*T
                        ENDIF
C
C CALCULATE THE VALUES OF F, F', F'', F'''
C
                        SS1Z = SINH(S1*WM*Z)
                        CS1Z = COSH(S1*WM*Z)
                        SS2Z = SINH(S2*WM*Z)
                        CS2Z = COSH(S2*WM*Z)

                        FZ = AM*SS1Z + BM*CS1Z + CM*SS2Z + DM*CS2Z
C
                        FP = WM*(S1*(AM*CS1Z + BM*SS1Z) +
     &                           S2*(CM*CS2Z + DM*SS2Z))
C
                        FD = WM**2.D0*(S1**2.D0*(AM*SS1Z + BM*CS1Z) +
     &                                 S2**2.D0*(CM*SS2Z + DM*CS2Z))
C
                        FT = WM**3.D0*(S1**3.D0*(AM*CS1Z + BM*SS1Z) +
     &                                 S2**3.D0*(CM*CS2Z + DM*SS2Z))
C
C  DEFINE BESSEL FUNCTION AND DERIVATIVES JP, JD, JT
C
                        IF (NRA .NE. 0) THEN
                           RD = (R / NRS) * NRA
C
                           JO = J0(WM*RD)
                           JP = -WM * J1(WM*RD)
                           JD = WM**2.D0 * (-JO + J1(WM*RD)/(WM*RD))
                           JT = WM**3.D0 * ( JO/(WM*RD) +
     &                          (1.D0 - 2.D0/(WM*RD)**2.D0)*J1(WM*RD))
```

```
                        ELSE
C
C  LIMIT VALUES OF JO, JP, JD, AND JT AT RD=0
C
                           JO =  1.D0
                           JP =  0.D0
                           JD = -0.5D0 * WM**2.D0
                           JT =  0.D0
                        ENDIF
C
C-----------------------------------------------------------------
C   RECOVER DISPLACEMENTS AND ALPHA FOR NONLINEAR CONTACT PROBLEM
C-----------------------------------------------------------------
C
C   COMPUTE DISPLACEMENTS AT THE ZERO RADIUS POSITION ONLY
C
                        IF (NRA .EQ. 0) THEN
C
                           IF ((NZE .EQ. 1) .OR. (NZE .EQ. NZT)) THEN
                              disptop = 0.
                              dispbot = 0.
                              IF  (NZE .EQ. 1)  THEN
                                 DISPTOP = -2.D0*A3*((BB+1.D0)*JD*FZ +
     &                              AA*FD) + A4*(2.D0*CC*JD*FZ + DD*FD)
                              ENDIF
                              IF (NZE .EQ. NZT) THEN
                                 DISPBOT = -2.D0*A3*((BB+1.D0)*JD*FZ +
     &                              AA*FD) + A4*(2.D0*CC*JD*FZ + DD*FD)
                              ENDIF
                              DISP = DISP + (DISPTOP-DISPBOT)
C
                           ENDIF
                        ENDIF
C
                        IF (LD$ .EQ. 'HZS') GOTO 160
C-----------------------------------------------------------------
C  COMPUTE THE STRESSES IN THE PLATE AXES
C-----------------------------------------------------------------
C
                        IF (NRA .EQ. 0) THEN
C
C  LIMIT VALUES OF STRESSES AT RD=0
C
                           STT(NRA,NZE) = STT(NRA,NZE) +
     &                        (-( BB*JD*FP ) - ( JD*FP ) -
     &                        ( AA*JO*FT ))
C
                           SZZ(NRA,NZE) = SZZ(NRA,NZE) +
     &                        ( CC*JD*FP ) + (CC*JD*FP) +
     &                        ( DD*JO*FT )
C
                           SRR(NRA,NZE) = SRR(NRA,NZE) +
     &                        (-( JD*FP ) - ( BB*JD*FP ) -
```

```
     &                        ( AA*JO*FT ))
C
                           SRZ(NRA,NZE) = SRZ(NRA,NZE) +
     &                        FZ*JT + 0.5*JT*FZ +
     &                        JT*FZ + AA*JP*FD
                        ELSE
C
                           STT(NRA,NZE) = STT(NRA,NZE) - (
     &                        ( BB*JD*FP ) + ( 1./RD*JP*FP ) +
     &                        ( AA*JO*FT ) )
C
                           SZZ(NRA,NZE) = SZZ(NRA,NZE) + (
     &                        ( CC*JD*FP ) + ( CC/RD*JP*FP ) +
     &                        ( DD*JO*FT ) )
C
                           SRR(NRA,NZE) = SRR(NRA,NZE) - (
     &                        (    JD*FP ) + ( BB/RD*JP*FP ) +
     &                        ( AA*JO*FT ) )
C
                           SRZ(NRA,NZE) = SRZ(NRA,NZE) + (
     &                        (    JT*FZ ) - ( 1./RD**2*JP*FZ ) +
     &                        ( 1.0/RD*JD*FZ ) + ( AA*JP*FD ) )
C
                        ENDIF
C
  160                 CONTINUE
  170               CONTINUE
C
C  WRITE DISPLACEMENT AND FORCE TO OUTPUT SOURCES FOR CONVERGENCE CHECK
C  ONLY WRITE TO FILE IF WE ARE OPERATING ON THE MIDDLE ALPHA.
C  WE WANT TO AVOID EXCESS FILE GENERATION.
C
                  if (local) WRITE(6,180) V,DISP
  180             FORMAT('+',5X,I4,8X,E12.5,' m')
C IF:  MIDDLE ALPHA BEING CALCULATED FOR THE MAJOR AXIS
C THEN: SEND DATA TO FILE
C
                  IF ((KOUNT1.EQ.1 .AND. KOUNT2.EQ.INT(NUMALP/2)+1)
     &               .or. (kflag2 .eq. 2))THEN
                     if (kflag2 .eq. 1 .and. (genfil(6))) then
                        WRITE(16,*) V,FORCE
                     else if (kflag2 .eq. 2 .and. (genfil(14))) then
                        write(16,*) V,FORCE
                     end if
                     IF (LD$ .EQ. 'HZS' .and. (genfil(7)))
     &                  WRITE(17,*) V,DISP
                  ENDIF
                  IF(KOUNT1.EQ.1 .AND. V.EQ.INT(KV*DELHAR)) THEN
                     IF(GENFIL(8)) THEN
                        LMJR=26+KV
                        WRITE(LMJR,*)ALPHA,TQ*(ALPHA/DISP)
                        KV=KV+1
                     ENDIF
```

```
                 ENDIF
                 IF(KOUNT1.EQ.2 .AND. V.EQ.INT(KV*DELHAR)) THEN
                    IF(GENFIL(9)) THEN
                       LMNR=36+KV
                       WRITE(LMNR,*)ALPHA,TQ*(ALPHA/DISP)
                       KV=KV+1
                    ENDIF
                 ENDIF
  190         CONTINUE
C
C CALCULATE THE GEOMETRIC INDENTATION
C
                 ALPHAI = RI-SQRT(RI**2.D0-XEP**2.D0)
C
                 IF ((KFLAG4 .EQ. 1) .AND. ((KOUNT1 .EQ. 1 .AND.
     & KOUNT2 .EQ. INT(NUMALP/2)+1) .or. (kflag2 .eq. 2))) THEN
                    WRITE(2,*)0,FORCE
                    DO 195 KIE=1,KIO
                       YR = (REAL(KIE)/REAL(KIO))**3.D0*R
                       WRITE(2,*)YR,PRESS(KIE)
                       IF(((YR/XEP)**2.D0) .GT. 1.D0) THEN
                          PR = 0.D0
                       ELSE
                          PR = PZ*SQRT(1.D0-(YR/XEP)**2.D0)
                       ENDIF
                       WRITE(3,*)YR,PR
  195               CONTINUE
                 ENDIF
C
C-----------------------------------------------------------------
C  WRITING INPUT INFORMATION TO THE OUTPUT FILE
C-----------------------------------------------------------------
C
        if (kflag2 .eq. 1 .and. (.not.(genfil(5)))) go to 420
                IF (KWFLAG .EQ. 0) THEN
                 WRITE(14,200)
  200            FORMAT(1H0//23X,'FOURIER - BESSEL IMPACT ANALYSIS'///)
                 WRITE(14,210)TITLE1,TITLE2,TITLE3
  210            FORMAT(8X,A60/8X,A60/8X,A60//)
                 WRITE(14,220)LD$
  220            FORMAT(8X,'Analysis for loading type ',A3/)
                 WRITE(14,230)'Plate thickness =',T*1.E3,' mm'
  230            FORMAT(8X,A31,F9.5,A3)
                 WRITE(14,240)'Radius of the region analyzed =',
     &            R*1.E3,' mm'
  240            FORMAT(8X,A31,F8.3,A3)
                 WRITE(14,250)'Number of harmonics =',M
                 IF(LD$ .EQ. 'HZS') then
                  WRITE(14,280)RI*1.E3
  250             FORMAT(8X,A31,I4/)
                  WRITE(14,253)'MAJOR AXIS'
  253             FORMAT(' ',30X,A10/)
                  WRITE(14,255)'   Alpha   ','    Load    ',
```

```
     &          '        K        '
255            FORMAT(' ',15X,A11,3X,A12,3X,A17)
               WRITE(14,255)' ————————','————————————',
     &          '————————————————'
               KWFLAG=1
              end if
             END IF
             if (LD$ .eq. 'HZS') then
              IF(KOUNT1 .EQ. 1 .AND. KMAJ .EQ. 0) THEN
               WRITE(25,200)
               WRITE(25,210)TITLE1,TITLE2,TITLE3
               WRITE(25,260)'MAJOR AXIS:'
               WRITE(25,270)XET*1.E-9,XEZ*1.E-9,NU1,NU2,
     &          G*1.E-9
               KMAJ=1
              END IF
              IF(KOUNT1 .EQ. 2 .AND. KMIN .EQ. 0) THEN
               WRITE(26,200)
               WRITE(26,210)TITLE1,TITLE2,TITLE3
               WRITE(26,260)'MINOR AXIS:'
               WRITE(26,270)XET*1.E-9,XEZ*1.E-9,NU1,NU2,
     &          G*1.E-9
               KMIN=1
              END IF
             else
              if (ld$ .eq. 'HZ') then
               write(14,285)alpha*1000.0
              end if
             end if
260          FORMAT(27X,A10)
270          FORMAT(27X,'Etheta     =',F8.3,' GPa'/
     &               27X,'Ez         =',F8.3,' GPa'/
     &               27X,'nu r-theta =',F10.6/
     &               27X,'nu r-z     =',F10.6/
     &               27X,'G r-z      =',F8.4,' GPa')
280          FORMAT(22X,'Impactor radius =',F8.3,' mm'//)
285          format(27x,'alpha      =',f8.4,' mm'/)
             IF(LD$ .EQ. 'PT') WRITE(14,290)LDD,RHO,ACC
290          FORMAT(27X,'Total load =',E12.5,' N'/
     &               30X,'Density =',E12.5,' kg/m↑3'/
     &               18X,'Center acceleration =',E12.5,' m/s↑2'//)
             IF(LD$ .EQ. 'HZ') WRITE(14,300)RHO,RI*1.E3,XEP*1.E3,
     &        TQ,ACC,FB,PZ*1.E-6,FORCE*1.E-6
300          FORMAT(11X,'Density of test specimen   = ',F8.1,
     &        ' kg/m↑3'/
     &        11X,'Impactor radius              = ',F8.3,' mm'/
     &        11X,'Impact radius                = ',F8.3,' mm'/
     &        11X,'Maximum impact load          = ',E12.5,' N'/
     &        11X,'Corresponding acceleration   = ',E12.5,' m/s↑2'/
     &        11X,'Recovered load               = ',E12.5,' N'/
     &        11x,'Input pressure (at r=0)      = ',f8.2,' MPa'/
     &        11x,'Recovered pressure (at r=0) = ',f8.2,' MPa'//)
C
```

```
C If calculating the Hertzian spring constant then
C Write ALPHA information to:  KOUNT1 = 1        MAJOR AXIS scratch file
C                                     = 2        MINOR AXIS scratch file
C
                 If (ld$ .eq. 'HZS') then
                  IF (KOUNT1 .EQ.1) THEN
                   if (kount2 .eq. 5 .or. kount2 .eq. 9) then
                    write(25,*)char(12)
                    write(25,905)title1,title2,title3,'MAJOR AXIS'
                   end if
                   WRITE(25,*)' '
                   WRITE(25,305)ALPHA*1.E3
  305              FORMAT(' ','ALPHA = ',F5.4,' mm')
                   WRITE(25,310)ALPHAI*1.E3,DISP*1.E3,PZ*1.E-6,
     &              FORCE*1.E-6,TQ,FB,TQ*(ALPHAI/DISP)
                  ELSE
                   if (kount2 .eq. 5 .or. kount2 .eq. 9) then
                    write(26,*)char(12)
                    write(26,905)title1,title2,title3,'MINOR AXIS'
                   end if
                   WRITE(26,*)' '
                   WRITE(26,305)ALPHA*1.E3
                   WRITE(26,310)ALPHAI*1.E3,DISP*1.E3,PZ*1.E-6,
     &              FORCE*1.E-6,TQ,FB,TQ*(ALPHAI/DISP)
                  END IF
                  IF (KOUNT1 .EQ. 1) then
                   WRITE(6,309)'MAJOR AXIS'
                  ELSE
                   WRITE(6,309)'MINOR AXIS'
                  END IF
  309             format(/27x,a10/)
                  WRITE(6,310)ALPHAI*1.E3,DISP*1.E3,PZ*1.E-6,
     &              FORCE*1.E-6,TQ,FB,TQ*(ALPHAI/DISP)
  310             FORMAT(/
     &              9X,'Geometric approach (input)     =',F16.13,' mm'/
     &              9X,'Displacement approach (output) =',F16.13,' mm'/
     &              9X,'Input pressure (at r=0)        =',F16.10,' MPa'/
     &              9X,'Recovered pressure (at r=0)    =',F16.10,' MPa'/
     &              9X,'Input load                     =',F16.9,' N'/
     &              9X,'Recovered load                 =',F16.9,' N'/
     &              9X,'Scaled load for given alpha    =',F16.9,' N')
                 ENDIF
C
C ------------------------------------------------------------------
C
C  END OF CORE SECTION
C
  420           IF(LD$ .NE. 'HZS') GOTO 449
C
C ------------------------------------------------------------------
            TLOAD(KOUNT2,KOUNT1) = TQ*(ALPHAI/DISP)
C  CALCULATE THE HERTZIAN SPRING CONSTANT
            XK(KOUNT2,KOUNT1) = TLOAD(KOUNT2,KOUNT1)/((ALPHA)**1.5)
```

```
        if (local) then
            WRITE(6,*)' '
            WRITE(6,442)KOUNT2,XK(KOUNT2,KOUNT1)*1.E-9
  442       FORMAT(10X,'K(',I2,') =',F8.4,' GN/m↑1.5')
        end if
            XKAVG(KOUNT1) = XKAVG(KOUNT1) + XK(KOUNT2,KOUNT1)
  445    CONTINUE
C ------------------------------------------------------------------
C ------------------------------------------------------------------
         XKAVG(KOUNT1) = XKAVG(KOUNT1)/REAL(NUMALP)
        if (local) then
         WRITE(6,*)' '
         WRITE(6,*)'KAVG = ',XKAVG(KOUNT1)*1.E-9,' GN/m↑1.5'
        end if
  447 CONTINUE
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C ------------------------------------------------------------------
C
C  CALCULATE THE AVERAGE OF THE MAJOR AND MINOR AXIS K's.
C
      IF(MJMN .EQ. 2) THEN
         XKMEAN = (XKAVG(1) + XKAVG(2))/2.
      ELSE
         XKMEAN = XKAVG(1)
      ENDIF
      if (local) then
        WRITE(6,*)' '
        WRITE(6,*)'AVERAGE K FOR THE MAJOR AND MINOR AXIS = ',XKMEAN*
     &1.E-9,'GN/m↑1.5'
      end if
C
C  WRITE ALPHA,LOAD, AND K TO OUTPUT FILE
C
      if (genfil(5)) then
      DO KKOUNT=1,NUMALP
        WRITE(14,448)ALPHAA(KKOUNT)*1.E3,TLOAD(KKOUNT,1),
     & XK(KKOUNT,1)*1.E-9
      END DO
  950 FORMAT(' ',/22X,'AVERAGE K = ',F8.4,' GN/M↑1.5'//)
      WRITE(14,950)XKAVG(1)*1.E-9
      WRITE(14,253)'MINOR AXIS'
      WRITE(14,255)'   Alpha    ','    Load     ','        K         '
      WRITE(14,255)' ----------','------------','----------------'
      DO KKOUNT=1,NUMALP
        WRITE(14,448)ALPHAA(KKOUNT)*1.E3,TLOAD(KKOUNT,2),
     & XK(KKOUNT,2)*1.E-9
      END DO
      WRITE(14,950)XKAVG(2)*1.E-9
  448 FORMAT(' ',15X,f8.4,' mm',3X,f10.2,' N',3X,f10.4,' GN/m↑1.5')

C  write the average alpha to the file
```

```
      write(14,903)xkmean*1.E-9
  903 format(/
     & 14X'Average K for major and minor axis = ',f8.4,' GN/mt1.5')
  904 format(///14x,'jobname_HDISP.RES contains results for the ',
     &          'major axis'/
     &          14x,'axis with Alpha = ',f8.4,' mm')
      write(14,904)alphaa(int(numalp/2)+1)*1.E3
C=================================================================

C
C  REWIND SCRATCH FILES AND MOVE THEIR DATA INTO THE OUTPUT FILE
C
      WRITE(14,*)CHAR(12)
      REWIND(UNIT=25)
      REWIND(UNIT=26)
      READ(25,910)XFER
      IF (MJMN .EQ. 2) READ(26,910)XFER
      NBLOCK=0
  900 READ(UNIT=25,FMT=910,END=920)XFER
  905 format(//23x,'FOURIER - BESSEL IMPACT ANALYSIS'///
     & 8x,a60/8x,a60/8x,a60//
     & 27x,a10///)
      WRITE(14,915)XFER
      GO TO 900
  910 FORMAT(A80)
  915 format(a)
  920 WRITE(14,*)CHAR(12)
      IF (MJMN .EQ. 2) THEN
  925    READ(UNIT=26,FMT=910,END=930)XFER
         WRITE(14,*)XFER
         GO TO 925
      ENDIF
  930 CLOSE (UNIT=25)
      CLOSE (UNIT=26)
      CLOSE (UNIT=14)
      end if
C SEND THE FINAL HERTZIAN SPRING CONSTANT (XKMEAN) TO THE BINARY FILE
C IN RECORD 29.
      IF(FLAG1 .EQ. 'Y' .OR. FLAG1 .EQ. 'y') THEN
         WRITE(12,REC=29)XKMEAN
         CLOSE(UNIT=12)
      END IF
      if (kflag2 .eq. 1 .and. (genfil(6))) CLOSE(UNIT=16)
      if (kflag2 .eq. 2 .and. (genfil(14))) close(unit=16)
      if (kflag2 .eq. 1 .and. (genfil(7))) CLOSE(UNIT=17)
C =================================================================
C STOP PROGRAM IF IT WAS A STATIC ANALYSIS
C =================================================================
C
  449 IF (LD$ .EQ. 'HZS') STOP
C
      WRITE(14,450)
  450 FORMAT(1X,'RAD. ',1X,'PLY',5X,'SIGMA r',7X,'SIGMA theta',6X
```

```
     & 'SIGMA z',7X,'TAU r-z')
C
      DO 500 I=0,NRS
         DO 400 J = 1,NZT
            WRITE(14,550)I,J,SRR(I,J),STT(I,J),SZZ(I,J),SRZ(I,J)
  550       FORMAT(1X,I2,4X,I2,5X,E10.4,5X,E10.4,5X,E10.4,5X,E10.4)
  400    CONTINUE
  500 CONTINUE
C
C CALCULATING THE STRAINS FROM THE STRESSES
C
      WRITE(14,*)' '
      WRITE(14,*)' '
      WRITE(14,455)
  455 FORMAT(1X,'RAD',1X,'PLY',2X,'EPS. r',5X,'EPS. theta',
     & 4X,'EPS. z',5X,'GAMMA r-z')
C
      open(unit=35,file=filnam(15),status='new',access='direct',
     &     recl=40)
      nrec=1
      write(35,rec=nrec) np,nrs
      DO 700 I=0,NRS
         DO 600 J=1,NZT
            nrec=nrec+1
            ERR(I,J) = A1*SRR(I,J)+A2*STT(I,J)+A3*SZZ(I,J)
            ETT(I,J) = A2*SRR(I,J)+A1*STT(I,J)+A3*SZZ(I,J)
            EZZ(I,J) = A3*(SRR(I,J)+STT(I,J))+A4*SZZ(I,J)
            GA(I,J) = A5*SRZ(I,J)
            output(1)=srr(i,j)
            output(2)=stt(i,j)
            output(3)=szz(i,j)
            output(4)=srz(i,j)
            output(5)=err(i,j)
            output(6)=ett(i,j)
            output(7)=ezz(i,j)
            output(8)=ga(i,j)
            write(35,rec=nrec) i,j,(output(ii),ii=1,8)
            WRITE(14,555)I,J,ERR(I,J),ETT(I,J),EZZ(I,J),GA(I,J)
  555       FORMAT(1X,I2,2X,I2,2X,E10.4,2X,E10.4,2X,E10.4,2X,E10.4)
  600    CONTINUE
  700 CONTINUE
C
      close(unit=35)
      WRITE(14,*)' '
      WRITE(14,*)' '
C
      IF (OUTYP.EQ.'N') GOTO 1100
      DO 830 I=0,NTH
        DO 820 J=0,NRS
          DO 810 K=1,NZT
             EPL(1,1) = ERR(J,K)
             EPL(2,2) = ETT(J,K)
             EPL(3,3) = EZZ(J,K)
```

```
            EPL(1,3) = GA(J,K)*0.5
            EPL(3,1) = EPL(1,3)
            THT = 90./FLOAT(NTH) * FLOAT(I)
            THP = (THT - THET(K))
            THP = THP * PI/180.
            XT(1,1) = COS(THP)
            TT(1,1) = XT(1,1)
            XT(2,2) = XT(1,1)
            TT(2,2) = XT(2,2)
            XT(1,2) = -SIN(THP)
            TT(2,1) = XT(1,2)
            XT(2,1) = +XT(1,2)
            TT(1,2) = XT(2,1)
            XT(3,3) = 1.0
            XT(1,3) = 0.0
            XT(2,3) = 0.0
            TT(1,3) = 0.0
            TT(2,3) = 0.0
C
C  TENSOR TRANSFORMATIONS
C
            DO 875 L=1,3
              DO 865 M=1,3
                EP(L,M)=(XT(1,L)*XT(1,M)*EPL(1,1)+
     &           XT(1,L)*XT(2,M)*EPL(1,2)+XT(1,L)*
     &           XT(3,M)*EPL(1,3))+(XT(2,L)*XT(1,M)*
     &           EPL(2,1)+XT(2,L)*XT(2,M)*EPL(2,2)+
     &           XT(3,L)*XT(3,M)*EPL(2,3))+(XT(3,L)*
     &           XT(1,M)*EPL(3,1)+XT(3,L)*XT(2,M)*
     &           EPL(3,2)+XT(3,L)*XT(3,M)*EPL(3,3))
  865         CONTINUE
  875       CONTINUE
C
            RTHP = 90./FLOAT(NTH)*FLOAT(I)
C
            WRITE(14,*)'THETA              =',RTHP
            WRITE(14,*)'RADIUS STEP        =',J
            WRITE(14,*)'PLY                =',K
            WRITE(14,*)
            WRITE(14,*)'EPSILON(I,J) IN PRINCIPAL AXES'
C
C  TEST FOR STRAINS GREATER THAN MAXIMUM VALUES AND MARK WITH ****
C
            DO II= 1, 3
               DO JJ= 1, 3
                  IF ((EP(II,JJ) .LT. EPT_MAX(II,JJ)) .AND.
     &                (EP(II,JJ) .GT. EPC_MAX(II,JJ))) THEN
                     TEST(II,JJ)='   '
                  ELSE
                     TEST(II,JJ)='***'
                  ENDIF
               ENDDO
            ENDDO
```

```
C
            DO II= 1, 3
               WRITE(14,805) EP(II,1),TEST(II,1),
     &                       EP(II,2),TEST(II,2),
     &                       EP(II,3),TEST(II,3)
            ENDDO
  805          FORMAT(1X,E12.5,A,1X,E12.5,A,1X,E12.5,A)
C
            WRITE(14,*)' '
  810     CONTINUE
  820   CONTINUE
  830 CONTINUE
 1100 STOP
      END
```

```
C   LOCAL2.FOR                                                    4/17/89
C*****************************************************************
C
C       FUNCTION J0:  USED FOR DEFINING BESSEL FUNCTIONS
C                 TAKEN FROM 'NUMERICAL RECIPES'
C
C*****************************************************************
C
      REAL*8 FUNCTION J0(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DATA P1,P2,P3,P4,P5 /
     &     1.D0,    -.1098628627D-2,  .2734510407D-4, -.2073370639D-5,
     &      .2093887211D-6 /,
     &     Q1,Q2,Q3,Q4,Q5 /
     &     -.1562499995D-1,  .1430488765D-3, -.6911147651D-5,
     &      .7621095161D-6, -.934945152D-7 /
      DATA R1,R2,R3,R4,R5,R6 /
     &     57568490574.D0, -13362590354.D0,  651619640.7D0,
     &     -11214424.18D0,   77392.33017D0,  -184.9052456D0 /,
     &     S1,S2,S3,S4,S5,S6 /
     &     57568490411.D0,  1029532985.D0,  9494680.718D0,
     &      59272.64853D0,  267.8532712D0,  1.D0 /
C
      IF (ABS(X) .LT. 8.) THEN
         Y = X**2
            J0 = (R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6)))))
     &           /(S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6)))))
      ELSE
         AX = ABS(X)
         Z = 8./AX
         Y = Z**2
         XX = AX-0.785398164
            J0 = SQRT(0.636619772/AX)*
     &           (COS(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y*P5))))-
     &          Z*SIN(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5)))))
         ENDIF
         RETURN
         END
C
C*****************************************************************
C
C       FUNCTION J1:  USED FOR DEFINING BESSEL FUNCTIONS
C                 TAKEN FROM 'NUMERICAL RECIPES', pp. 173-4.
C
C*****************************************************************
C
      REAL*8 FUNCTION J1(X)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DATA R1,R2,R3,R4,R5,R6 /
     &     72362614232.D0, -7895059235.D0, 242396853.1D0,
     &     -2972611.439D0,  15704.48260D0, -30.16036606D0 /,
```

```
     &     S1,S2,S3,S4,S5,S6 /
     &     144725228442.D0, 2300535178.D0, 18583304.74D0,
     &       99447.43394D0, 376.9991397D0, 1.D0 /
      DATA P1,P2,P3,P4,P5 /
     &               1.D0,  .183105D-2,  -.3516396496D-4,
     &     .2457520174D-5,  -.240337019D-6 /,
     &     Q1,Q2,Q3,Q4,Q5 /
     &     .04687499995D0,  -.2002690873D-3, .8449199096D-5,
     &      -.88228987D-6,   .105787412D-6 /
C
      IF (ABS(X) .LT. 8.) THEN
         Y = X**2
            J1 = X*(R1+Y*(R2+Y*(R3+Y*(R4+Y*(R5+Y*R6)))))
     &            /(S1+Y*(S2+Y*(S3+Y*(S4+Y*(S5+Y*S6)))))
      ELSE
         AX = ABS(X)
         Z = 8./AX
         Y = Z**2
         XX = AX-2.356194491
            J1 = SQRT(.636619772/AX)*
     &            (COS(XX)*(P1+Y*(P2+Y*(P3+Y*(P4+Y*P5))))-
     &           Z*SIN(XX)*(Q1+Y*(Q2+Y*(Q3+Y*(Q4+Y*Q5)))))
         ENDIF
         RETURN
         END
C
C*************************************************************
C
C  FUNCTION TO DEFINE ITH ZERO OF J0
C
C*************************************************************
C
      REAL*8 FUNCTION ZEROJ0(I)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 XM(20)
      DATA (XM(I),I=1,20) /
     &    2.4048255577D0,  5.520078110 3D0,
     &    8.6537279129D0, 11.7915344391D0,
     &   14.9309177086D0, 18.0710639679D0,
     &   21.2116366299D0, 24.3524715308D0,
     &   27.4934791320D0, 30.6346064684D0,
     &   33.7758202136D0, 36.9170983537D0,
     &   40.0584257646D0, 43.1997917132D0,
     &   46.3411883717D0, 49.4826098974D0,
     &   52.6240518411D0, 55.7655107550D0,
     &   58.9069839261D0, 62.0484691902D0 /
C
      PI = 3.1415926535897932D0
C
C  ZEROS FOR J0 BESSEL FUNCTION
C
      IF (I .LT. 21) THEN
        ZEROJ0 = XM(I)
```

```
C
      ELSE
        S = FLOAT(I)
        A = 4.*S-1.
        AX = A*PI
C
        ZEROJ0 = 0.25*AX*
     &           ( 1.D0+ 2.D0/AX**2 - 62.D0/(3.D0*AX**4) +
     &           15116.D0/(15.*AX**6) - 12554474.D0/(105.D0*AX**8) +
     &           8368654292.D0/(315.D0*AX**10) )
C
      ENDIF
      RETURN
      END
C
C*************************************************************
C
C  FUNCTION TO DEFINE ITH ZERO OF J1
C
C*************************************************************
C
      REAL*8 FUNCTION ZEROJ1(I)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      PI = 3.14159265359D0
C
      S = FLOAT(I)
      B = 4.*S+1
      BX = B*PI
C
      ZEROJ1 = 0.25*BX*
     &         ( 1.D0 - 6.D0/BX**2 + 6.D0/BX**4 -
     &         4716.D0/(5.D0*BX**6) + 3902418.D0/(35.D0*BX**8) -
     &         895167324.D0/(35.D0*BX**10) )
C
      RETURN
      END
C=============================================================
      REAL*8 FUNCTION FUNC1(RINT,XEP,WM)
C        Integration of eq. (3.2.16b) with PZ factored out.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 J0
      FUNC1=SQRT(1.D0-(RINT/XEP)**2.D0)*RINT*J0(WM*RINT)
      RETURN
      END
C=============================================================
      REAL*8 FUNCTION FUNC2(RINT,XEP,WM)
c        Integration of eq. (3.2.20).
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 J0
      FUNC2=RINT*J0(WM*RINT)
      RETURN
      END
```

```
C=================================================================
      SUBROUTINE QROMB(FUNC,A,B,SS,XEP,WM)
C        Taken from Numerical Recipes, pp. 114-5.
C        Returns as SS the integral of the function FUNC from A to B.
C        Integration is performed by Romberg's method of order 2K,
C        where, e.g., K=2 is Simpson's rule.
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (EPS=1.D-8, JMAX=20, JMAXP=JMAX+1, K=5, KM=K-1)
C        Here EPS is the fractional accuracy desired, as determined by
C        the extrapolation error estimate; JMAX limits the total number
C        steps; K is the number of points used in the extraploation.
      DIMENSION S(JMAXP),H(JMAXP)
C        These store the successive trapezoidal approximations and their
C        relative step-sizes.
      EXTERNAL FUNC1,FUNC2
      H(1)=1.D0
      DO 100 J=1,JMAX
         CALL TRAPZD(FUNC,A,B,S(J),J,XEP,WM)
         IF (J.GE.K) THEN
            CALL POLINT(H(J-KM),S(J-KM),K,0.D0,SS,DSS)
            IF (ABS(DSS).LT.EPS*ABS(SS)) RETURN
         ENDIF
         S(J+1)=S(J)
         H(J+1)=0.25D0*H(J)
C           This is a key step:  The factor is 0.25 even though the
C           step-size is decreased by only 0.5.  This makes the
C           extrapolation a polynomial in h↑2 as allowed by equation
C           (4.2.1), not just a polynomial in h.
  100 CONTINUE
      PAUSE 'Too many steps.'
      RETURN
      END
C=================================================================
      SUBROUTINE TRAPZD(FUNC,A,B,S,N,XEP,WM)
C        Taken from Numerical Recipes, p. 111.
C        This routine computes the N'th stage of refinement of an
C        extended trapezoidal rule.  FUNC is input as the name of the
C        function to be integrated between linits A and B, also input.
C        When called with N=1, the routine returns as S the crudest
C        estimate of [INTEGRAL from a to b of f(x)dx].  Subsequent calls
C        with N=2,3,... (in that sequential order) will improve the
C        accuracy S by adding 2↑(N-2) additional interior points.  S
C        should not be modified between sequential calls.
      IMPLICIT REAL*8 (A-H,O-Z)
      IF (N.EQ.1) THEN
         S=0.5D0*(B-A)*(FUNC(A,XEP,WM)+FUNC(B,XEP,WM))
         IT=1
C           IT is the number of points to be added on the next call.
      ELSE
         TNM=IT
         DEL=(B-A)/TNM
C           This is the spacing of the points to be added.
         X=A+0.5D0*DEL
```

```
         SUM=0.D0
         DO 100 J=1,IT
            SUM=SUM+FUNC(X,XEP,WM)
            X=X+DEL
  100    CONTINUE
         S=0.5D0*(S+(B-A)*SUM/TNM)
C           This replaces S by its refined value.
         IT=2*IT
      ENDIF
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE POLINT(XA,YA,N,X,Y,DY)
C        Taken from Numerical Recipes, p. 82.
C        Given arrays XA and YA, each of length N, and given a value X,
C        this routine returns a value Y, and an error estimate DY.  If
C        P(x) is the polynomial of degree N-1 such that P(XAi)=YAi,
C        i=1,...,N, then the returned value Y=P(X).
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (NMAX=10)
C        Change NMAX as desired to be the largest anticipated value of
C        N.
      DIMENSION XA(N),YA(N),C(NMAX),D(NMAX)
      NS=1
      DIF=ABS(X-XA(1))
      DO 100 I=1,N
C        Here we find the index NS of the closest table entry.
         DIFT=ABS(X-XA(I))
         IF (DIFT.LT.DIF) THEN
            NS=I
            DIF=DIFT
         ENDIF
         C(I)=YA(I)
C           Initialize the tableau of C's and D's.
         D(I)=YA(I)
  100 CONTINUE
      Y=YA(NS)
C        This is the intial approximation to Y.
      NS=NS-1
      DO 300 M=1,N-1
C        For each column of the tableau, we loop over the current C's
C        and D's and update them.
         DO 200 I=1,N-M
            HO=XA(I)-X
            HP=XA(I+M)-X
            W=C(I+1)-D(I)
            DEN=HO-HP
            IF(DEN.EQ.0.D0)PAUSE
C              This error can occur only if two input XA's are (to
C              within roundoff) identical.
            DEN=W/DEN
            D(I)=HP*DEN
C              Here the C's and D's are updated.
```

```
            C(I)=HO*DEN
  200     CONTINUE
          IF (2*NS.LT.N-M)THEN
C            After each cloumn in the tableau is completed, we decide
C            which correction, C or D, we want to add to our accumulating
C            value of Y, i.e. which path to take through the tableau -
C            forking up or down.  We do this in such a way as to take the
C            most "straight line" route through the tableau to its apex,
C            updating NS accordingly to keep track of where we are.  This
C            route keeps the partial approximations centered (insofar as
C            possible) on the target X.  The last DY added is thus the
C            error indication.
             DY=C(NS+1)
          ELSE
             DY=D(NS)
             NS=NS-1
          ENDIF
          Y=Y+DY
  300 CONTINUE
      RETURN
      END
C
      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
C*************************************************************
C
C  THIS IS A GENERAL ROUTINE FOR GAUSS-JORDAN ELIMINATION WITH
C     WITH FULL PIVOTING - TAKEN FROM 'NUMERICAL RECIPES'
C
C*************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      PARAMETER (NMAX = 50)
      DIMENSION A(NP,NP), B(NP,MP), IPIV(NMAX),
     &          INDXR(NMAX), INDXC(NMAX)
C
      DO 11 J=1, N
         IPIV(J) = 0.0
 11   CONTINUE
C
      DO 22 I=1, N
         BIG = 0.
         DO 13 J=1, N
            IF (IPIV(J) .NE. 1) THEN
               DO 12 K=1, N
                  IF (IPIV(K) .EQ. 0) THEN
                      IF (ABS(A(J,K)) .GE. BIG) THEN
                         BIG = ABS(A(J,K))
                         IROW = J
                         ICOL = K
                      ENDIF
                  ELSE IF (IPIV(K) .GT. 1) THEN
                      PAUSE 'SINGULAR MATRIX'
                  ENDIF
```

```
 12            CONTINUE
            ENDIF
 13      CONTINUE
C
         IPIV(ICOL) = IPIV(ICOL)+1
C
         IF (IROW .NE. ICOL) THEN
            DO 14 L=1, N
               DUM = A(IROW,L)
               A(IROW,L) = A(ICOL,L)
               A(ICOL,L) = DUM
 14         CONTINUE
C
            DO 15 L=1,M
               DUM = B(IROW,L)
               B(IROW,L) = B(ICOL,L)
               B(ICOL,L) = DUM
 15         CONTINUE
         ENDIF
         INDXR(I) = IROW
         INDXC(I) = ICOL
         IF (A(ICOL,ICOL) .EQ. 0.) PAUSE 'SINGULAR MATRIX'
         PIVINV = 1./A(ICOL,ICOL)
         A(ICOL,ICOL) = 1.
         DO 16 L=1, N
            A(ICOL,L) = A(ICOL,L)*PIVINV
 16      CONTINUE
         DO 17 L=1, M
            B(ICOL,L) = B(ICOL,L)*PIVINV
 17      CONTINUE
         DO 21 LL=1, N
            IF (LL .NE. ICOL) THEN
               DUM = A(LL,ICOL)
               A(LL,ICOL) = 0.
               DO 18 L=1, N
                  A(LL,L) = A(LL,L) - A(ICOL,L)*DUM
 18            CONTINUE
               DO 19 L=1, M
                  B(LL,L) = B(LL,L) - B(ICOL,L)*DUM
 19            CONTINUE
            ENDIF
 21      CONTINUE
 22   CONTINUE
C
      DO 24 L=N,1,-1
         IF (INDXR(L) .NE. INDXC(L)) THEN
            DO 23 K=1,N
               DUM = A(K,INDXR(L))
               A(K,INDXR(L)) = A(K,INDXC(L))
               A(K,INDXC(L)) = DUM
 23         CONTINUE
         ENDIF
 24   CONTINUE
```

```
RETURN
END
```

```
C   LOCAL3.FOR                                                    4/17/89
C=====================================================================
C                startup module
C
C  This module is used by ROTATE, BESSEL, and IMPACT to identify whether
C  or not the program is being executed as part of a complete analysis.
C  If so, variables are initialized according to the "system
C  configuration" and all user interaction is suppressed.
C
C  KEY VARIABLES:
C
C  LOCAL    (logical)  Flag:  If LOCAL = .TRUE. then the program is run
C                      interactively with the user.
C
C  FILNAM  (character) Array of filenames used by the program and by
C                      the complete analysis.
C  GENFIL   (logical)  Array of flags:  If GENFIl(N) = .TRUE. then the
C                      corresponding file contained in FILNAM is created
C
      subroutine startup

      character*20 filnam
      character*6  jobnam

      logical local, genfil, exst

      common /system/ jobnam,local,filnam(15),genfil(15)
C
C See if CONFIG.BIN exists.  If not, then the program is being run
C locally.
C
      inquire(file='config.bin', exist=exst)

C
C Read in CONFIGURATION parameters
C
      if (exst) then
        local = .false.
        open(unit=10, file='config.bin', status='old', access='direct',
     &     recl=30)
        read(10,rec=1) jobnam
        do i=1,15
                read(10,rec=i+1) filnam(i), genfil(i)
        end do
        close(unit=10)
      else
        local = .true.
   10   format(' ','Enter the desired jobname [max 6 characters]: ',$)
   20   format(a6)
   30   write(6,10)
        read(5,20) jobnam
        if (jobnam(1:1) .eq. ' ') then
          write(6,*)'*** Invalid jobname - please re-enter ***'
```

```
        go to 30
      end if
      n=1
40    if (jobnam(n:n) .ne. ' ' .and. n .lt. 7) then
        n=n+1
        go to 40
      end if
      n=n-1
      filnam(1)=jobnam(1:n)//'.dat'
      filnam(2)=jobnam(1:n)//'.bin'
      filnam(3)=jobnam(1:n)//'_angle.dat'
      filnam(4)=jobnam(1:n)//'_damge.out'
      filnam(5)=jobnam(1:n)//'_hertz.out'
      filnam(6)=jobnam(1:n)//'_hpres.res'
      filnam(7)=jobnam(1:n)//'_hdisp.res'
      filnam(8)=jobnam(1:n)//'_amajr.res'
      filnam(9)=jobnam(1:n)//'_aminr.res'
      filnam(10)=jobnam(1:n)//'_dynam.out'
      filnam(11)=jobnam(1:n)//'_force.res'
      filnam(12)=jobnam(1:n)//'_displ.res'
      filnam(13)=jobnam(1:n)//'_accel.res'
      filnam(14)=jobnam(1:n)//'_dpres.res'
      filnam(15)=jobnam(1:n)//'_ss.bin'
      do i=1,15
        genfil(i)=.true.
      end do
    end if
    end
C-----&
```

```
      PROGRAM KNPUT
C   Puts a value for K and n in the Hertzian contact law
C   into the master binary file.
C   7/4/89
C Program written by Kiernan F. Ryan 1989.
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C23456789112345678921234567893123456789412345678951234567896123456789712
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*15 FILNAM
      WRITE(6,*)'     Enter the binary file name (EX: CHEN.BIN):'
      READ(5,10)FILNAM
   10 FORMAT(A)
      WRITE(6,*)'     Enter the value of K [GN/m+n]:'
      READ(5,*)HK
      WRITE(6,*)'     Enter the value of the exponent n (F=Ka+n):'
      READ(5,*)XN
      OPEN(UNIT=12,FILE=FILNAM,STATUS='OLD',ACCESS='DIRECT',RECL=110)
      WRITE(12,REC=29)HK*1.E9,XN
      CLOSE(UNIT=12)
      STOP
      END
```

```
      PROGRAM IMPACT
C   PFG1.FOR                                                    4/11/89
C
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C*********************************************************************
C   THIS PROGRAM IS THE MAIN CONTROLLING PROGRAM FOR THE PLATE IMPACT
C          ANALYSIS OF THE TIME HISTORY OF AN IMPACT EVENT.
C
C   INFORMATION ON THE FILES AND SUBROUTINES USED BY THIS PACKAGE MAY
C                 BE FOUND IN THE FILE 'IMPACT.DOC'
C
C   ORIGINAL PROGRAM WRITTEN BY:  WILSON TSANG  19 APRIL 1988
C              MODIFICATIONS BY:  MICHAEL KRAFT
C              MODIFICATIONS BY:  KIERNAN RYAN     APRIL 1989
C
C*********************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 MASS
C
C    INCLUDE COMMON PARAMETER VALUES AND COMMON BLOCKS
C
        INCLUDE 'IMPACT.INC'
C
        DIMENSION XBETA(M1),XB(M1),YBETA(M2),YB(M2),
     &            PSI(3,3,M1,M1),PHI(3,3,M2,M2),WX(M1),WY(M2),
     &            MASS(M3),STIFF(M4),W(M3),AA1(M3),BB1(M3),CC1(M3),
     &            COSI(M3),SINI(M3),ACBS(M3),WCOS(M3),WSIN(M3),
     &            OMEGASQ(M3),AA2(M3),BB2(M3),CC2(M3),DD2(M3)
      EQUIVALENCE (MASS(1),AA1(1)),(STIFF(1),BB1(1)),(W(1),CC1(1)),
     &            (STIFF(M3+1),COSI(1)),(STIFF(2*M3+1),SINI(1)),
     &            (STIFF(3*M3+1),ACBS(1)),(STIFF(4*M3+1),WCOS(1)),
     &            (STIFF(5*M3+1),WSIN(1)),(STIFF(6*M3+1),OMEGASQ(1)),
     &            (STIFF(7*M3+1),AA2(1)),(STIFF(8*M3+1),BB2(1)),
     &            (STIFF(9*M3+1),CC2(1)),(STIFF(10*M3+1),DD2(1))
C
        COMMON /INVSE/ WORK(M3),KP(M3),DET(2),INERT(3)
C
C    initialize file configuration
```

```
C
      call startup
C
C    OPEN ALL FILES USED FOR THIS ANALYSIS
C
C       UNIT        FILE            TYPE
C       21          FORCE.RES       OUTPUT
C       22          DISP.RES        OUTPUT
C       23          ACCEL.RES       OUTPUT
C
     if (genfil(11)) OPEN(UNIT=21,FILE=filnam(11),STATUS='NEW')
     if (genfil(12)) OPEN(UNIT=22,FILE=filnam(12),STATUS='NEW')
     if (genfil(13)) OPEN(UNIT=23,FILE=filnam(13),STATUS='NEW')
C
      CALL INPUT(NX,NY,IX,IY,IFC,IEEMFA,H,RHO,
     &           A,B,D11,D12,D16,D22,D26,D66,
     &           A44,A45,A55,S,HK,BM,DELT,NT,VI,PNX,PNY,XN)
C
      CALL BCONS(XBETA,XTHETA,XA,XB,IX,NX,IEEMFA)
C
      CALL BCONS(YBETA,YTHETA,YA,YB,IY,NY,IEEMFA)
C
      IF (NX.EQ.4) THEN
         CALL INTGL1(XBETA,XTHETA,XA,XB,IX,PSI,WX,IEEMFA)
      ELSE
         CALL INTGL(XBETA,XTHETA,XA,XB,IX,PSI,WX)
      ENDIF
C
      IF (NY.EQ.4) THEN
         CALL INTGL1(YBETA,YTHETA,YA,YB,IY,PHI,WY,IEEMFA)
      ELSE
         CALL INTGL(YBETA,YTHETA,YA,YB,IY,PHI,WY)
      ENDIF
C
      CALL MATRIX(MASS,STIFF,W,IX,IY,PSI,PHI,WX,WY,
     &            H,RHO,A,B,D11,D12,D16,D22,D26,D66,
     &            A44,A45,A55,S,PNX,PNY)
      N=3*IX*IY
C
      IF(NX.EQ.4.OR.NY.EQ.4) CALL ARRAN(STIFF,MASS,W,N,NLIST)
C
      IF (IFC.EQ.1) CALL CONDENSE(MASS,STIFF,W,N,NLIST)
C
      CALL MARCH(MASS,STIFF,W,N,HK,BM,DELT,NT,VI,AA1,BB1,CC1,COSI,
     &  SINI,ACBS,WCOS,WSIN,OMEGASQ,AA2,BB2,CC2,DD2,XN)
C
      STOP
      END
C
C-----------------------------------------------------------------
```

```
      SUBROUTINE INPUT(NX, NY, IX, IY, IFC, IEEMFA, H, RHO, A, B,
     &                 D11, D12, D16, D22, D26, D66,
     &                 A44, A45, A55, S, HK, BM, DELT,
     &                 NT, VI, PNX, PNY, XN)
C*****************************************************************************
C            THIS SUBROUTINE READS IN THE REQUIRED DATA FOR THE
C             PLATE IMPACT ANALYSIS PROGRAM FROM THE DATA FILE
C                NAMED 'IMPACT.DAT' (sequential file)
C                   OR 'DIRACC.BIN' (binary file)
C*****************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*80 TITLE1,TITLE2,TITLE3
      CHARACTER*20 BINFIL
      logical exst
C
C  INCLUDE ALL COMMON PARAMETERS AND COMMON BLOCKS
C
      INCLUDE 'IMPACT.INC'
      COMMON /BINARY/nfile,binfil
C
C   If NFILE = 1, then the input file is DIRACC.BIN
C
C*****************************************************************************
C    THE REQUIRED INPUT AND A DESCRIPTION OF THE UNITS REQUIRED:
C
C    DESCRIPTION            VARIABLE                UNITS
C
C    title #1               TITLE1                max 80 char
C    title #2               TITLE2
C    title #3               TITLE3
C
C    x-direction b.c.         NX                    integer
C    y-direction b.c.         NY                    integer
C
C    * where the index numbers for the boundary conditions are:
C          1 FOR SS-SS
C          2 FOR CL-FR
C          3 FOR CL-CL
C          4 FOR FR-FR
C          5 FOR SS-CL
C          6 FOR SS-FR
C
C    number of x-axis modes    IX                   integer
C    number of y-axis modes    IY                   integer
C    * note that IX = IY for this version
C
C    condense rotary inertia   IFC                  integer
C    * 1 = yes, 0 = no
C
C    eliminate even modes      IEEMFA               integer
C    from analysis
C    * 1 = yes, 0 = no
C
```

```
C     plate thickness           H                    meters
C     plate density             RHO                  kg/m↑3
C
C     x-dimension               A                    meters
C     y-dimension               B                    meters
C
C     D11                       D11                  N*m
C     D12                       D12
C     D16                       D16
C     D22                       D22
C     D26                       D26
C     D66                       D66
C
C     A44                       A44                  N/m
C     A45                       A45
C     A55                       A55
C
C     shear correction factor   S                    real
C
C     hertzian spring constant  HK                   N/m↑1.5
C     exponent in Hertz law     XN
C     impactor mass             BM                   kg
C     time step length          DELT                 seconds
C     number of time steps      NT                   integer
C     impactor velocity         VI                   m/s
C
C     x-direction load          PNX                  N/m
C     y-direction load          PNY                  N/m
C
C     binary filename (input)   FILNAM(2)
C*****************************************************************************
C

        if (local) then
        write(6,*)'Input file: [0 = IMPACT.DAT; 1 = binary file]:'
        read(5,*) nfile
        else
                nfile=1
        end if

        if (nfile .eq. 0) then
      OPEN(UNIT=40,FILE='IMPACT.DAT',STATUS='OLD')
        READ(40,100) TITLE1
        READ(40,100) TITLE2
        READ(40,100) TITLE3
 100    FORMAT(A)
C
        READ(40,*) NX
        READ(40,*) NY
        READ(40,*) IX
        READ(40,*) IY
        READ(40,*) IFC
        READ(40,*) IEEMFA
```

```
C
        READ(40,*) H, RHO, A, B
        READ(40,*) D11, D12, D16, D22, D26, D66
        READ(40,*) A44, A45, A55, S
        READ(40,*) HK, XN, BM, DELT, NT, VI
C
        READ(40,*) PNX, PNY
C
C Read data from FILNAM(2)
C
        else
        if (local) then
  102           inquire(file=filnam(2), exist=exst)
                if (.not.(exst)) then
                        write(6,104)
  104                   format(/' ','Enter the complete name ',
     &                          'of the binary file: ',$)
                        read(5,105) filnam(2)
                        go to 102
                end if
        end if
 105    format(a)
                open(unit=12,file=filnam(2),status='old',
     &               access='direct',recl=110)
                read(12,rec=1) TITLE1
                read(12,rec=2) TITLE2
                read(12,rec=3) TITLE3
C
                read(12,rec=13) NX, NY
                read(12,rec=26) IX, IY
                read(12,rec=27) IFC, IEEMFA
C
                read(12,rec=5) H
C CONVERT TO METERS
                H = H*1.E-3
                read(12,rec=11) RHO
                read(12,rec=10) A, B
                read(12,rec=9) D11, D12, D22, D66, D16, D26
                read(12,rec=6) A44, A45, A55
C CONVERT TO FROM MN/m TO N/m
                A44 = A44*1.E6
                A45 = A45*1.E6
                A55 = A55*1.E6
                read(12,rec=14) S
                read(12,rec=29) HK, XN
                read(12,rec=15) BM, VI
                read(12,rec=28) DELT, NT
C CONVERT FROM micro_s TO s
                DELT = DELT*1.E-6
C
                read(12,rec=12) PNX, PNY
        end if
      if (genfil(10)) then
```

```
      WRITE(6,110) TITLE1
      WRITE(6,110) TITLE2
      WRITE(6,110) TITLE3
 110  FORMAT(1X,A)
C
      WRITE(6,*)' '
      WRITE(6,*)'NUMBER OF X-DIRECTION MODES =',IX
      WRITE(6,*)'NUMBER OF Y-DIRECTION MODES =',IY
      WRITE(6,*)' '
      WRITE(6,*)'             TIME INCREMENT =',DELT
      WRITE(6,*)'       NUMBER OF TIME STEPS =',NT
      WRITE(6,*)' '
     end if
C
      CLOSE(UNIT=12)


      RETURN
      END
C
C-----------------------------------------------------------------------
```

```
      SUBROUTINE BCONS(BETA,THETA,A,B,I,N,IEEMFA)
C*****************************************************************************
C
C
C
C
C*****************************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C    INCLUDE ALL COMMON PARAMETERS AND COMMON BLOCKS
C
      INCLUDE 'IMPACT.INC'
C
      DIMENSION BETA(1), B(1)
C
      if (genfil(10)) WRITE (6,*) 'SETTING UP BOUNDARY CONDITIONS'
C
      IF (N.EQ.1) THEN
         DO 10 J=1,I
            IF(IEEMFA.EQ.1) THEN
               M=2*J-1
            ELSE
               M=J
            ENDIF
           BETA(J)=M*3.141592654
           B(J)=0.
 10      CONTINUE
         THETA=0.
         A=0.
         RETURN
      ENDIF
C
      IF (N.EQ.2) THEN
         DO 20 J=1,I
            IF(IEEMFA.EQ.1) THEN
               M=2*J-1
            ELSE
               M=J
            ENDIF
            BETA(J)=(M-.5)*3.141592654
            B(J)=2*MOD(M,2)-1
 20      CONTINUE
         THETA=-.785398163
         A=1.
         RETURN
      ENDIF
C
      IF (N.EQ.3) THEN
         DO 30 J=1,I
            IF(IEEMFA.EQ.1) THEN
               M=2*J-1
            ELSE
```

```
            M=J
         ENDIF
         BETA(J)=(M+.5)*3.141592654
         B(J)=2*MOD(M,2)-1
 30      CONTINUE
         THETA=-.785398163
         A=1.
         RETURN
      ENDIF
C
      IF (N.EQ.4) THEN
         IF(IEEMFA.EQ.1) THEN
            BETA(1)=0.
            BETA(2)=1.5*3.14159265359
            B(1)=0.
            B(2)=1.
         ELSE
            BETA(1)=0.
            BETA(2)=0.
            B(1)=0.
            B(2)=0.
         ENDIF
         DO 40 J=3,I
            IF(IEEMFA.EQ.1) THEN
               M=2*J-1
            ELSE
               M=J
            ENDIF
            BETA(J)=(M-2.+.5)*3.141592654
            B(J)=2*MOD(M-2,2)-1
 40      CONTINUE
         THETA=2.35619449
         A=1.
         RETURN
      ENDIF
C
      IF (N.EQ.5) THEN
         DO 50 J=1,I
            IF(IEEMFA.EQ.1) THEN
               M=2*J-1
            ELSE
               M=J
            ENDIF
            BETA(J)=(M+.25)*3.141592654
            B(J)=2*MOD(M,2)-1
 50      CONTINUE
         THETA=0.
         A=0.
         RETURN
      ENDIF
C
      IF (N.EQ.6) THEN
         DO 60 J=1,I
```

```
            IF(IEEMFA.EQ.1) THEN
               M=2*J-1
            ELSE
               M=J
            ENDIF
            BETA(J)=(M+.25)*3.141592654
            B(J)=2*MOD(M+1,2)-1
 60       CONTINUE
          THETA=0.
          A=0.
          RETURN
       ENDIF
C
       END
C
C
C
C-----------------------------------------------------------------------
C
C
C
      SUBROUTINE INTGL(BETA,THETA,A,B,II,F,W)
C*****************************************************************************
C
C
C
C*****************************************************************************
       IMPLICIT REAL*8 (A-H,O-Z)
       INTEGER P,Q
C
C   INCLUDE COMMON PARAMETER VALUES AND COMMON BLOCKS
C
       INCLUDE 'IMPACT.INC'
C
       DIMENSION BETA(1),B(1),F(3,3,M1,1),W(1)
C
      if (genfil(10))
     & WRITE (6,*) 'CALCULATING INTEGRALS OF BEAM FUNCTIONS'
C
       PI=3.141592654
       DO 10 I=1,3
          DO 10 J=1,3
             DO 10 M=1,II
                DO 10 N=1,II
C
                   IF (I.GT.J) THEN
                      F(I,J,M,N)=F(J,I,N,M)
                      GOTO 10
                   ENDIF
C
                   P=I-1
                   Q=J-1
                   D1=BETA(M)-BETA(N)
```

```
                    D2=BETA(M)+BETA(N)
                    D3=BETA(M)*BETA(M)+BETA(N)*BETA(N)
                    D4=2*MOD(P+1,2)-1
                    D5=2*MOD(Q+1,2)-1
                    D6=THETA+P*PI/2.
                    D7=THETA+Q*PI/2.
C
                    IF (M.EQ.N) THEN
                       D8=COS(D6-D7)
                    ELSE
                       D8=(SIN(D1+D6-D7)-SIN(D6-D7))/D1
                    ENDIF
C
                       D9=(SIN(D6+D7)-SIN(D2+D6+D7))/D2
                       D10=D4*A*(BETA(N)*COS(D7)+BETA(M)*SIN(D7))
                       D11=B(M)*(BETA(M)*SIN(BETA(N)+D7)-
     &                           BETA(N)*COS(BETA(N)+D7))
                       D12=D5*A*(BETA(M)*COS(D6)+BETA(N)*SIN(D6))
                       D13=B(N)*(BETA(N)*SIN(BETA(M)+D6)-
     &                           BETA(M)*COS(BETA(M)+D6))
                       D14=(D4*D5*A*A+B(M)*B(N))/D2
C
                       F(I,J,M,N)=(BETA(M)**P)*(BETA(N)**Q)*
     &                            (D8+D9+D14+1.414213562*
     &                            (D10+D11+D12+D13)/D3)
 10     CONTINUE
C
        X=.5
        DO 20 I=1,II
           D15=BETA(I)*X+THETA
           D16=EXP(-BETA(I)*X)
           D17=EXP(-BETA(I)*(1.-X))
           W(I)=1.414213562*SIN(D15)+A*D16+B(I)*D17
 20     CONTINUE
        RETURN
        END
C
C
C
C-----------------------------------------------------------------------
C
C
C
      SUBROUTINE INTGL1(BETA,THETA,A,B,II,F,W,IEEMFA)
C***********************************************************************
C
C
C
C
C***********************************************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        INTEGER P,Q
C
```

```
C   INCLUDE COMMON PARAMETER VALUES AND COMMON BLOCKS
C
        INCLUDE 'IMPACT.INC'
C
        DIMENSION BETA(1),B(1),F(3,3,M1,1),W(1)
C
        if (genfil(10))
     &  WRITE (6,*) 'CALCULATING INTEGRALS OF BEAM FUNCTIONS'
C
        PI=3.141592654
        R2=1.414213562
        R3=1.732050808
C
        IF(IEEMFA.EQ.1) THEN
          K=2
        ELSE
          K=3
        ENDIF
C
        DO 10 I=1,3
           DO 10 J=1,3
              DO 20 M=1,K-1
                 DO 20 N=1,II
                    F(I,J,M,N)=0.
 20           CONTINUE
              DO 30 M=1,II
                 DO 30 N=1,K-1
                    F(I,J,M,N)=0.
 30           CONTINUE
              DO 10 M=K,II
                 DO 10 N=K,II
C
                    IF (I.GT.J) THEN
                       F(I,J,M,N)=F(J,I,N,M)
                       GOTO 10
                    ENDIF
C
                    P=I-1
                    Q=J-1
                    D1=BETA(M)-BETA(N)
                    D2=BETA(M)+BETA(N)
                    D3=BETA(M)*BETA(M)+BETA(N)*BETA(N)
                    D4=2*MOD(P+1,2)-1
                    D5=2*MOD(Q+1,2)-1
                    D6=THETA+P*PI/2.
                    D7=THETA+Q*PI/2.
C
                    IF (M.EQ.N) THEN
                       D8=COS(D6-D7)
                    ELSE
                       D8=(SIN(D1+D6-D7)-SIN(D6-D7))/D1
                    ENDIF
C
```

```
                    D9=(SIN(D6+D7)-SIN(D2+D6+D7))/D2
                    D10=D4*A*(BETA(N)*COS(D7)+BETA(M)*SIN(D7))
                    D11=B(M)*(BETA(M)*SIN(BETA(N)+D7)-
     &                        BETA(N)*COS(BETA(N)+D7))
                    D12=D5*A*(BETA(M)*COS(D6)+BETA(N)*SIN(D6))
                    D13=B(N)*(BETA(N)*SIN(BETA(M)+D6)-
     &                        BETA(M)*COS(BETA(M)+D6))
                    D14=(D4*D5*A*A+B(M)*B(N))/D2
                    F(I,J,M,N)=(BETA(M)**P)*(BETA(N)**Q)*
     &                          (D8+D9+D14+R2*(D10+D11+D12+D13)/D3)
C
 10     CONTINUE
C
        DO 40 I=K,II
           D1=SIN(BETA(I)+THETA)+SIN(THETA)
           D2=SIN(BETA(I)+THETA)-SIN(THETA)
           D3=COS(BETA(I)+THETA)-COS(THETA)
           F(1,2,1,I)=R2*D2+B(I)-A
           F(2,1,I,1)=F(1,2,1,I)
           F(1,2,2,I)= R3*(-R2*D1-B(I)-A+2./BETA(I)*(-R2*D3+A+B(I)))
           F(2,1,I,2)=F(1,2,2,I)
           F(2,1,2,I)=2.*R3/BETA(I)*(R2*D3-A-B(I))
           F(1,2,I,2)=F(2,1,2,I)
           F(1,3,1,I)=BETA(I)*(R2*D3+A+B(I))
           F(3,1,I,1)=F(1,3,1,I)
           F(1,3,2,I)=R3*F(1,3,1,I)-2.*R3*(BETA(I)*(R2*COS(BETA(I)+
     &     THETA)+B(I))-R2*D2-B(I)+A)
           F(3,1,I,2)=F(1,3,2,I)
           F(2,3,2,I)=-2.*R3*BETA(I)*(R2*D3+A+B(I))
           F(3,2,I,2)=F(2,3,2,I)
           F(2,2,2,I)=2.*R3*(-R2*D2-B(I)+A)
           F(2,2,I,2)=F(2,2,2,I)
C
 40     CONTINUE
C
        IF(IEEMFA.EQ.1) THEN
          F(1,1,1,1)=1.
        ELSE
          F(1,1,1,1)=1.
          F(1,1,2,2)=1.
          F(1,2,1,2)=-2.*R3
          F(2,1,2,1)=-2.*R3
          F(2,2,2,2)=12.
        ENDIF
C
        X=0.5
C
C  ASSIGN RIGID BODY MODE AMPLITUDES AT CENTER OF PLATE
C
        IF(IEEMFA.EQ.1) THEN
          W(1)=1.
        ELSE
          W(1)=1.
```

```
          W(2)=0.
        ENDIF
        DO 50 I=K,II
           D15=BETA(I)*X+THETA
           D16=EXP(-BETA(I)*X)
           D17=EXP(-BETA(I)*(1.-X))
           W(I)=1.414213562*SIN(D15)+A*D16+B(I)*D17
 50     CONTINUE
        RETURN
        END
C
C-----------------------------------------------------------------------
```

```
      SUBROUTINE MATRIX(MASS, STIFF, W, IX, IY, PSI, PHI, WX, WY,
     &                  H, RHO, A, B, D11, D12, D16, D22, D26, D66,
     &                  A44, A45, A55, S, PNX, PNY)
C
C*****************************************************************************
C
C
C
C
C*****************************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MASS

C
C   INCLUDE COMMON PARAMETER VALUES AND COMMON BLOCKS
C
      INCLUDE 'IMPACT.INC'
C
      DIMENSION PHI(3,3,M2,1), PSI(3,3,M1,1),
     &          MASS(1),STIFF(1),WX(1),WY(1),W(1)
C
      if (genfil(10)) WRITE (6,*) 'SETTING UP STIFFNESS MATRIX'
      II=IX*IY
      DO 10 I=1,II
         DO 10 J=1,II
            M=(I-1)/IY+1
            N=I-IY*(M-1)
            K=(J-1)/IY+1
            L=J-IY*(K-1)
C
C  CALCULATE COMPONENTS OF STIFFNESS MATRIX AND STORE IMMEDIATELY
C                          IN VECTOR FORM
C
C       KAC(I,J)
C
      STIFF((J+2*II)*(J+2*II-1)/2+I) =
     &          S*S*(A55 * B * PSI(2,2,M,K) * PHI(1,1,N,L) +
     &               A45 * A * PSI(2,1,M,K) * PHI(1,2,N,L))
C
C       KAB(I,J)
C
      STIFF((J+II)*(J+II-1)/2+I) =
     &          D12 * PSI(3,1,M,K) * PHI(1,3,N,L) +
     &          D16 * B/A * PSI(3,2,M,K) * PHI(1,2,N,L) +
     &          D26 * A/B * PSI(2,1,M,K) * PHI(2,3,N,L) +
     &          D66 * PSI(2,2,M,K) * PHI(2,2,N,L) +
     &       S*S*A45 * A*B * PSI(1,1,M,K) * PHI(2,2,N,L)
C
C       KBC(I,J)
C
      STIFF((J+2*II)*(J+2*II-1)/2+I+II) =
     &       S*S*(A45 * B * PSI(1,2,M,K) * PHI(2,1,N,L) +
     &            A44 * A * PSI(1,1,M,K)*PHI(2,2,N,L))
```

```
C
      IF (I.GT.J) GOTO 10
C
C     KAA(I,J)
C
      STIFF(J*(J-1)/2+I) =
     &            D11 * B/A * PSI(3,3,M,K) * PHI(1,1,N,L) +
     &            D16 * PSI(3,2,M,K) * PHI(1,2,N,L) +
     &            D16 * PSI(2,3,M,K) * PHI(2,1,N,L) +
     &            D66 * A/B * PSI(2,2,M,K) * PHI(2,2,N,L) +
     &        S*S*A55 * B*A * PSI(2,2,M,K) * PHI(1,1,N,L)
C
C     KBB(I,J)
C
      STIFF((J+II)*(J+II-1)/2+I+II) =
     &            D22 * A/B * PSI(1,1,M,K) * PHI(3,3,N,L) +
     &            D26 * PSI(1,2,M,K) * PHI(3,2,N,L) +
     &            D26 * PSI(2,1,M,K) * PHI(2,3,N,L) +
     &            D66 / A*B * PSI(2,2,M,K) * PHI(2,2,N,L) +
     &        S*S*A44 * A*B * PSI(1,1,M,K) * PHI(2,2,N,L)
C
C     KCC(I,J)
C
      STIFF((J+2*II)*(J+2*II-1)/2+I+2*II) =
     &        S*S*(A55 * B/A * PSI(2,2,M,K) * PHI(1,1,N,L) +
     &             A44 * A/B * PSI(1,1,M,K) * PHI(2,2,N,L) +
     &             A45 * PSI(2,1,M,K) * PHI(1,2,N,L) +
     &             A45 * PSI(1,2,M,K) * PHI(2,1,N,L)) +
     &             PNX * B/A * PSI(2,2,M,K) * PHI(1,1,N,L) +
     &             PNY * A/B * PSI(1,1,M,K) * PHI(2,2,N,L)
C
 10   CONTINUE
C
      if (genfil(10)) WRITE(6,*) 'SETTING UP MASS MATRIX'
      DO 30 I=1,II
         M=(I-1)/IY+1
         N=I-IY*(M-1)
C
C        MA(I)
C
         MASS(I) = RHO * H*H*H/12. * B*A * PSI(2,2,M,M)
C
C        MB(I)
C
         MASS(II+I) = RHO * H*H*H/12. * A*B * PHI(2,2,N,N)
C
C        MC(I)
C
         MASS(2*II+I) = RHO * H * A*B
C
 30   CONTINUE
C
      if (genfil(10)) WRITE(6,*) 'WORKING OUT FORCE VECTOR'
```

```
      DO 40 I=1,II
         M=(I-1)/IY+1
         N=I-IY*(M-1)
         W(I)=0.
         W(I+II)=0.
         W(I+2*II)=WX(M)*WY(N)
 40   CONTINUE
C
      RETURN
      END
C
C-----------------------------------------------------------------------
```

```
      SUBROUTINE ARRAN(STIFF,MASS,W,N,NLIST)
C*****************************************************************************
C
C
C
C*****************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MASS
C
C   INCLUDE COMMON PARAMETER VALUES AND COMMON BLOCKS
C
      INCLUDE 'IMPACT.INC'
C
      DIMENSION STIFF(1), MASS(1), W(1), ILIST(M6)
C
      DO 5 I=1,N
         IF(MASS(I).EQ.0.) THEN
            NLIST=NLIST+1
            ILIST(NLIST)=I
         ENDIF
 5    CONTINUE
      if (genfil(10)) WRITE(6,*) 'NLIST=',NLIST
      DO 7 I=1,NLIST
         if (genfil(10)) WRITE(6,*) 'ILIST(',I,')=',ILIST(I)
 7    CONTINUE
C
      if (genfil(10)) WRITE(6,*) 'REARRANGING MASS'
      IND1=1
      I=1
      IDUM=ILIST(IND1)
 10   CONTINUE
      IF(I.EQ.IDUM) THEN
         IND1=IND1+1
         DO 20 J=I,N+1-IND1
            MASS(J)=MASS(J+1)
 20      CONTINUE
         IDUM = ILIST(IND1)-IND1+1
         I=I-1
      ENDIF
C
      IF(IND1.EQ.NLIST+1) THEN
         GOTO 15
      ENDIF
C
      I=I+1
      GOTO 10
C
 15   if (genfil(10)) WRITE(6,*) 'REARRANGING STIFF'
      IND1=1
      IND2=1
      IND3=1
      IND4=1
      DO 40 I=1,N
```

```
            IF (I.EQ.ILIST(IND3)) THEN
               IND3=IND3+1
               IND2=IND2+I
               GOTO 40
            ENDIF
            DO 50 J=1,I
               IF(J.EQ.ILIST(IND4)) THEN
                  IND2=IND2+1
                  IND4=IND4+1
                  GOTO 50
               ENDIF
               STIFF(IND1)=STIFF(IND2)
               IND1=IND1+1
               IND2=IND2+1
 50         CONTINUE
            IND4 = 1
 40      CONTINUE
C
         if (genfil(10)) WRITE(6,*) 'REARRANGING FORCE VECTOR'
         IND1=1
         I=1
         IDUM=ILIST(IND1)
 60      CONTINUE
         IF(I.EQ.IDUM) THEN
            IND1=IND1+1
            DO 120 J=I,N+1-IND1
               W(J)=W(J+1)
 120        CONTINUE
            IDUM = ILIST(IND1)-IND1+1
            I=I-1
         ENDIF
C
         IF(IND1.EQ.NLIST+1) THEN
           GOTO 25
         ENDIF
C
         I=I+1
         GOTO 60
C
 25      N=N-NLIST
         RETURN
         END
C
C-----------------------------------------------------------------------
```

```
      SUBROUTINE CONDENSE(MASS,STIFF,W,N,NLIST)
C*****************************************************************************
C
C
C
C*****************************************************************************
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 MASS
C
C    INCLUDE COMMON PARAMETER VALUES AND COMMON BLOCKS
C
      INCLUDE 'IMPACT.INC'
C
      DIMENSION MASS(1),STIFF(1),W(1),WX(1),WY(1)
      DIMENSION H(M5)
C
      COMMON /INVSE/ WORK(M3),KP(M3),DET(2),INERT(3)
C
      N2=(N+NLIST)/3
      N1=N-N2
C
      if (genfil(10)) WRITE(6,*) 'INVERTING K2'
      CALL DSPFA(STIFF,N1,KP,INFO)
      CALL DSPDI(STIFF,N1,KP,DET,INERT,WORK,001)
C
      if (genfil(10)) then
      WRITE(6,*)' '
      WRITE(6,*) 'K1T*K2-1*K1'
      WRITE(6,*)' '
      end if
      DO 10 J=1,N2
         if (local) WRITE(6,90) J
         DO 20 K=1,N1
            WORK(K)=0.
            DO 20 L=1,N1
            IF(K.LE.L) WORK(K)=WORK(K)+STIFF((L-1)*L/2+K)
     &         *STIFF((J+N1)*(J+N1-1)/2+L)
            IF(K.GT.L) WORK(K)=WORK(K)+STIFF((K-1)*K/2+L)
     &         *STIFF((J+N1)*(J+N1-1)/2+L)
 20      CONTINUE
         DO 30 I=J,N2
            DUM=0.
            DO 40 K=1,N1
               DUM=DUM+STIFF((I+N1)*(I+N1-1)/2+K)*WORK(K)
 40         CONTINUE
            H((I-1)*I/2+J)=DUM
 30      CONTINUE
 10   CONTINUE
      if (.not.(local)) write(6,91) j
C
      if (genfil(10)) WRITE(6,*)
      if (genfil(10)) WRITE(6,*) 'CONDENSING STIFF'
      DO 50 I=1,N2
```

```
         DO 50 J=I,N2
         STIFF((J-1)*J/2+I) =
     &          STIFF((J+N1-1)*(J+N1)/2+I+N1)-H((J-1)*J/2+I)
 50      CONTINUE
C
         if (genfil(10)) WRITE(6,*) 'CONDENSING MASS & W'
         DO 60 I=1,N2
         MASS(I)=MASS(I+N1)
         W(I)=W(I+N1)
 60      CONTINUE
         N=N2
 90      FORMAT('+',I4)
 91      format(' ',i4)
         RETURN
         END
C
C-----------------------------------------------------------------------
```

```
      SUBROUTINE MARCH(MASS,STIFF,W,N,HK,BM,DELT,NT,VI,A1,B1,C1,COSI,
     &   SINI,ACBS,WCOS,WSIN,OMEGASQ,A2,B2,C2,D2,XN)
C   Pierre's fast time marching routine
C   PFG7.FOR                                                    9/5/89
C=================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 MASS
      CHARACTER*20 BINFIL
C
C   Include common parameter values and common blocks.
C
      INCLUDE 'IMPACT.INC'
      COMMON /BINARY/NFILE,BINFIL
C
C   Variable Definitions:
C
C   R        Contact force (N)
C   WC       Displacement of plate center (m)
C   WD       Velocity of plate center (m/s)
C   WDD      Acceleration of plate center (m/s↑2)
C   U        Displacement of the impactor (m)
C   UD       Velocity of the impactor (m/s)
C
C   Note that the positive z-axis points downward, in the direction of
C   positive impactor velocity.
C
C   If NFILE=1, then a binary input/output file is being used.
C
C=================================================================
      DIMENSION MASS(1),STIFF(1),W(1),OMEGA(M3),EPHI(M3,M3),FV1(M3),
     &   CW(M3),A1(1),B1(1),C1(1),D1(M3),COSI(1),SINI(1),ACBS(1),
     &   WCOS(1),WSIN(1),OMEGASQ(1),A2(1),B2(1),C2(1),D2(1)
      EQUIVALENCE (FV1(1),D1(1))
      NDELW = 10
C
C   If the rotary inertia is condensed, N = IX * IY
C   If not, N = 3 * IX * IY
C
      II = N*(N+1)/2
C
C   Obtain [M]↑-1/2 and store in MASS.  Hence, I no longer have [M].
C
      IF(GENFIL(10)) WRITE(6,*)' Solving for [M]↑-1/2'
      DO 100 I=1,N
         MASS(I) = 1./MASS(I)**0.5
  100 CONTINUE
C
C   Calculate [K] = [M]↑-1/2 * [Kcc↑*] * [M]↑-1/2 and store in STIFF.
C   Hence, I no longer have [Kcc↑*].
C
      IF(GENFIL(10)) WRITE(6,*)
     &   ' Calculating [K] = [M]↑-1/2 * [Kcc↑*] * [M]↑-1/2'
      DO 200 I=1,N
```

```
         DO 200 J=I,N
            LL = J*(J-1)/2 + I
            STIFF(LL) = MASS(I) * MASS(J) * STIFF(LL)
  200 CONTINUE
C
C   Obtain the eigenvalues (OMEGA) and eigenvectors (EPHI) of STIFF.
C
      IF(GENFIL(10)) WRITE(6,*)' Obtaining the eignvalues (OMEGA) and ei
     &genvectors (EPHI) of [K] (STIFF).'
      CALL RSP(M3,N,II,STIFF,OMEGA,1,EPHI,FV1,CW,IERR)
      DO 300 I=1,N
C   OMEGA returned from RSP is really OMEGA squared.
         OMEGASQ(I) = OMEGA(I)
         OMEGA(I) = SQRT(OMEGA(I))
  300 CONTINUE
C
C   Calculate CWi = {PHI}↑(i)↑T * [M]↑-1/2 * {W}
C
      IF(GENFIL(10)) WRITE(6,*)' Obtaining CWi = {PHI}↑(I)↑T * [M]↑-1/2
     &* {W}.'
      DO 400 I=1,N
         FV1(I) = MASS(I) * W(I)
  400 CONTINUE
      DO 500 I=1,N
         CW(I) = 0.
         DO 500 J=1,N
            CW(I) = CW(I) + EPHI(J,I)*FV1(J)
  500 CONTINUE
C
C   Obtain the initial values.
C
      IF(GENFIL(10)) WRITE(6,*)' Obtaining the initial values.'
      U1 = EPSILON*DELT*VI
      UD1 = VI
      WC = 0.D0
      WD = 0.D0
      WDD = 0.D0
      R = HK*(U1)**XN
      RD = XN*HK*(U1)**(XN-1.)*VI
      DO 600 I=1,N
         C1(I) = R*CW(I)/OMEGA(I)**2
         D1(I) = C1(I)/R*RD
         A1(I) = -1.*C1(I)
         B1(I) = -1.*D1(I)/OMEGA(I)
         SINI(I) = OMEGA(I)*DELT
         COSI(I) = COS(SINI(I))
         SINI(I) = SIN(SINI(I))
         WCOS(I) = OMEGA(I)*COSI(I)
         WSIN(I) = OMEGA(I)*SINI(I)
  600 CONTINUE
C
C   Set the initial values of the maximum force and corresponding
C   acceleration equal to zero.
```

```
C
      FORCEMAX = 0.D0
      ACCELMAX = 0.D0
C
C   Set the time counter equal to zero.
C
      J = 0
      IF(GENFIL(11)) WRITE(21,*) 0.,0.
      IF(GENFIL(12)) WRITE(22,*) 0.,0.
      IF(GENFIL(13)) WRITE(23,*) 0.,0.
C
C   Time marching
C
      IF(GENFIL(10)) WRITE(6,*)' '
      IF(GENFIL(10)) WRITE(6,*)' Time marching'
      IF(GENFIL(10)) WRITE(6,*)' '
  700 IF(NT .GT. 1) THEN
         IF(J .GE. NT) GOTO 1300
      ELSE IF (NT .EQ. 0) THEN
         IF(UD .LT. 0.) GOTO 1300
      ELSE IF (NT .EQ. 1) THEN
         IF((WC.LE.0.) .AND. J.GT.5) GOTO 1300
      END IF
      J = J + 1
      IF(LOCAL) WRITE(6,800)J
  800 FORMAT('+',I6)
C
C   Calculation of:
C      The displacement of the impator (U)
C      The velocity of the impactor (UD)
C      The displacement of the plate's center (WC)
C      The velocity of the plate's center (WD)
C      The acceleration of the plate's center (WDD)
C      The contact force (R)
C   Note that a 1 after a variable is for time step J and a 2 after a
C   variable is for time step J+1.
C
      U1WC = U1 - WC
      IF(U1WC .LT. 0.) U1WC = 0.D0
      U2 = -1.*R*DELT*DELT/2./BM - XN*HK/6./BM*(U1WC)**(XN-1.)*(UD1-WD)*
     &     DELT**3 + UD1*DELT + U1
      UD2 = -1.*R*DELT/BM -
     &      XN*HK/2./BM*(U1WC)**(XN-1.)*(UD1-WD)*DELT*DELT + UD1
      WC = 0.D0
      WD = 0.D0
      WDD = 0.D0
      DO 900 I=1,N
         ACBS(I) = A1(I)*COSI(I) + B1(I)*SINI(I)
         WC = WC + (ACBS(I) + C1(I) + D1(I)*DELT) * CW(I)
         WD = WD + (B1(I)*WCOS(I) - A1(I)*WSIN(I) + D1(I)) * CW(I)
         WDD = WDD - OMEGASQ(I)*ACBS(I)*CW(I)
  900 CONTINUE
      U2WC = U2 - WC
```

```
      IF(U2WC .LT. 0.) U2WC = 0.D0
      R = HK*(U2WC)**XN
      DO 1000 I=1,N
         C2(I) = R*CW(I)/OMEGASQ(I)
         D2(I) = XN*HK*(U2WC)**(XN-1.)*(UD2-WD)*CW(I)/OMEGASQ(I)
         A2(I) = A1(I)*COSI(I) + B1(I)*SINI(I) + C1(I) + D1(I)*DELT -
     &           C2(I)
         B2(I) = B1(I)*COSI(I) - A1(I)*SINI(I) + (D1(I)-D2(I))/OMEGA(I)
         A1(I) = A2(I)
         B1(I) = B2(I)
         C1(I) = C2(I)
         D1(I) = D2(I)
 1000 CONTINUE
      U1 = U2
      UD1 = UD2
      IF(MOD(J,NDELW).EQ. 0) THEN
         IF(GENFIL(11)) WRITE(21,*) DELT*J,R
         IF(GENFIL(12)) WRITE(22,*) DELT*J,WC
         IF(GENFIL(13)) WRITE(23,*) DELT*J,WDD
      END IF
C
C   Sample for maximum force.
C
      IF(R .GT. FORCEMAX) THEN
         FORCEMAX = R
         ACCELMAX = WDD
         INDEX = J
      END IF
      GOTO 700
C
C   End of time marching
C
 1300 CONTINUE
      IF(.NOT. (LOCAL)) WRITE(6,1400) J
 1400 FORMAT(' ',I6)
      IF(GENFIL(10)) WRITE(6,1500) FORCEMAX,ACCELMAX,
     &   INDEX*DELT*10**6
 1500 FORMAT(/' Maximum force',14X,'= ',F13.5,'    N'/' Corresponding ac
     &celeration = ',F14.3,'   m/s+2'/' Occurring at time',10X,'= ',F11.
     &3,'      micro_s')
C
C   Write FORCEMAX and ACCELMAX to the binary file if NFILE = 1.
C
      IF(NFILE .EQ. 1) THEN
         OPEN(UNIT=12,FILE=FILNAM(2),STATUS='OLD',ACCESS='DIRECT',
     &      RECL=110)
         WRITE(12,REC=30) FORCEMAX, ACCELMAX
         CLOSE(UNIT=12)
      END IF
C=================================================================
      RETURN
      END
```

```
C=================================================================
C                 startup module
C
C  This module is used by ROTATE, BESSEL, and IMPACT to identify wether
C  or not the program is being executed as part of a complete analysis.
C  If so, variables are initialized according to the "system
C  configuration" and all user interaction is suppressed.
C
C  KEY VARIABLES:
C
C  LOCAL    (logical)  Flag:  If LOCAL = .TRUE. then the program is run
C                      interactively with the user.
C
C  FILNAM  (character) Array of filenames used by the program and by
C                      the complete analysis.
C  GENFIL   (logical)  Array of flags:  If GENFIl(N) = .TRUE. then the
C                      corresponding file contained in FILNAM is created
C
      subroutine startup

      character*20 filnam
      character*10 jobnam

      logical local, genfil, exst

      common /system/ jobnam,local,filnam(15),genfil(15)
C
C See if CONFIG.BIN exists.  If not, then the program is being run
C locally.
C
      inquire(file='config.bin', exist=exst)

C
C Read in CONFIGURATION parameters
C
      if (exst) then
        local = .false.
        open(unit=10, file='config.bin', status='old', access='direct',
     &     recl=30)
        read(10,rec=1) jobnam
        do i=1,15
                read(10,rec=i+1) filnam(i), genfil(i)
        end do
        close(unit=10)
      else
        local = .true.
   10   format(' ','Enter the desired jobname [max 10 characters]: ',$)
   20   format(a10)
   30   write(6,10)
        read(5,20) jobnam
        if (jobnam(1:1) .eq. ' ') then
          write(6,*)'*** Invalid jobname - please re-enter ***'
          go to 30
```

```
      end if
      n=1
40    if (jobnam(n:n) .ne. ' ' .and. n .lt. 11) then
        n=n+1
        go to 40
      end if
      n=n-1
      filnam(1)=jobnam(1:n)//'.dat'
      filnam(2)=jobnam(1:n)//'.bin'
      filnam(3)=jobnam(1:n)//'_angle.dat'
      filnam(4)=jobnam(1:n)//'_final.out'
      filnam(5)=jobnam(1:n)//'_hertz.out'
      filnam(6)=jobnam(1:n)//'_hpres.res'
      filnam(7)=jobnam(1:n)//'_hdisp.res'
      filnam(8)=jobnam(1:n)//'_amajr.res'
      filnam(9)=jobnam(1:n)//'_aminr.res'
      filnam(10)=jobnam(1:n)//'_dynam.out'
      filnam(11)=jobnam(1:n)//'_force.res'
      filnam(12)=jobnam(1:n)//'_displ.res'
      filnam(13)=jobnam(1:n)//'_accel.res'
      do i=1,15
        genfil(i)=.true.
      end do
    end if
    end
C-----&
```

```
C*********************************************************************
C                                                                 4/6/89
C  THIS IS THE INCLUDE FILE FOR THE PROGRAM IMPACT.FOR
C
C*********************************************************************
C
C   THE PARAMETERS USED IN DIMENSIONING THE ARRAYS ARE:
C
C       M1 = MAXIMUM NUMBER OF MODES IN X-DIRECTION
C       M2 = MAXIMUM NUMBER OF MODES IN Y-DIRECTION
C       M3 = (M1 * M2 * 3)
C       M4 = (M3 * (M3 + 1)) / 2
C       M5 = (M1 * M2)*(M1 * M2 +1) / 2
C       M6 = (M1 * M2)
C
C-------------------------------------------------------------------
C       17 MODES(X) BY 17 MODES(Y):
C
        character*20 filnam
        character*10 jobnam
        logical local,genfil

        common /system/ jobnam,local,filnam(15),genfil(15)

        PARAMETER(M1 = 17, M2 = 17,
     &            M3 = (M1 * M2 * 3),
     &            M4 = (M3 * (M3 + 1)) / 2. ,
     &            M5 = (M1 * M2)*(M1 * M2 + 1.) / 2. ,
     &            M6 = (M1 * M2),
     &       EPSILON = 0.01 )
C*********************************************************************
```

```
      PROGRAM LIBRARY
C
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C Program written by Kiernan F. Ryan and Chris Park, 1988.
C ==================================================================
C ==================================================================
C
C This program allows you to do the following:
C
C    1)  Add a new material to the library of material ply properties.
C    2)  Delete a material from the library of ply properties.
C    3)  Edit a material in the library of ply properties.
C    4)  Display the materials currently in the library.
C
C ** NOTE **
C
C To allow for easy expansion, the file record length has been set to
C 144 characters to allow for the addition of 3 REAL*8 variables.  So,
C all programs accessing the file LIBRARY.BIN should specify RECL=144.
C
C ** RECORD FORMAT **
C
C The material information is stored in the following order:
C
C   E11,E22,E33,G23,G31,G12,ENU12,ENU13,ENU23,NAME,THICK,RHO,???,???
C
C Where ??? denotes unspecified variables.
C
C ==================================================================
C ==================================================================
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER NAME*40, CREATE*1, a$*40
      integer current
      logical exst,mflag,mkflag,opned

      common /block1/e11,e22,e33,g23,g31,g12,enu12,enu13,enu23,
     &               name,thick,rho
      common /block2/last,nlast,current
```

```
      common /block3/mkflag
C
C SEARCH FOR THE LAST RECORD.  THE PROGRAM INPUT SEARCHES THE FILE
C LIBRARY.BIN FOR A NEGATIVE THICKNESS TO INDICATE THE END OF THE FILE
C SINCE THE FILE CONTAINS BLANK RECORDS BETWEEN ONES WITH MATERIAL
C INFORMATION.
C
        DATA E11,E22,E33,G23,G31,G12,ENU12,ENU13,ENU23,NAME,THICK,
     &  rho /0.,0.,0.,0.,0.,0.,0.,0.,0.,'LAST RECORD',-1.,0./

        INQUIRE(FILE='LIBRARY.BIN',EXIST=EXST)
        IF (EXST) THEN
                THICK=0.
                LAST=0
                OPEN(UNIT=7,FILE='LIBRARY.BIN',STATUS='OLD',
     &                  ACCESS='DIRECT',RECL=144)
                DO WHILE (THICK .GT. 0. .OR. THICK .EQ. 0.)
                        LAST=LAST+1
                        READ(7,REC=LAST)E11,E22,E33,G23,G31,G12,
     &                          ENU12,ENU13,ENU23,NAME,THICK
                END DO
                CLOSE(7,STATUS='KEEP')
        ELSE
    4           WRITE(6,*)'File LIBRARY.BIN does not exist in this ',
     &          'directory.'
                READ(5,5) CREATE
    5           FORMAT(A1)
    6           FORMAT(' ','Create it [y/n]: ',$)
                IF (CREATE .EQ. 'N' .OR. CREATE .EQ. 'n') THEN
                        STOP
                ELSE IF (CREATE .EQ. 'Y' .OR. CREATE .EQ. 'y') THEN
                        OPEN(UNIT=7,FILE='LIBRARY.BIN',STATUS='NEW',
     &                  ACCESS='DIRECT',RECL=144)
                        WRITE(7,REC=1)E11,E22,E33,G23,G31,G12,ENU12,
     &                          ENU13,ENU23,NAME,THICK,E11,E11,E11
                        LAST=1
                        CLOSE(7,STATUS='KEEP')
                ELSE
                        GO TO 4
                END IF
        END IF


C
C Program introduction:
C
      open(unit=7, file='library.bin', status='old', access='direct',
     & recl=144)
    9 WRITE(6,10)
   10 FORMAT(1H1,79(1H=)//5X,'PROGRAM LIBRARY'///5X,
     & 'This program allows you to do the following:'//
     & 8X,'1)  Add a new material to the library of material ply ',
     & 'properties.'/
```

```
     &  8X,'2)  Delete a material from the library of ply properties.'/
     &  8X,'3)  Edit a material in the library of ply properties.'/
     &  8X,'4)  Display the materials currently in the library.'/
     &  8X,'5)  Exit.'/1H0,79(1H=)///////)
   19 WRITE(6,20)
   20 FORMAT(5X,'Enter your option (1,2,3,4 or 5):  ',$)
      READ(5,*) KOPTN
      write(6,*)' '
      write(6,*)' '
      write(6,*)' '
      IF (KOPTN .LT. 1 .OR. KOPTN .GT. 5) THEN
        GO TO 19
      END IF
      GO TO (29,150,200,550,21)KOPTN
   21 close(7,status='keep')
      stop
C
C ------------------------------------------------------------------
C
C Option 1:  Adding an new material to the library of material ply
C            properties.
C
C ------------------------------------------------------------------
C
   29   WRITE(6,30)
   30   FORMAT(1H1,79(1H=)///3X,'Option 1:  Adding a new material to ',
     &    'the library of material ply properties.'/1H0,79(1H=)//)
C
C This section obtains the input data.
C
   34   FORMAT(' ','"',40X,'"')
        WRITE(6,35)
   35   FORMAT(5X,'Enter the name of the material (maximum of 40 ',
     &    'characters):  '/)
        WRITE(6,34)
        WRITE(6,36)
   36   FORMAT('+','"',$)
        READ(5,37) NAME
   37   FORMAT(A40)
        WRITE(6,38)
   38   FORMAT(/5X,'Enter the material number to be used for this ',
     &    'material:  '$)
        READ(5,*) MATNUM
        WRITE(6,39)
   39   FORMAT(5x,'Enter the material thickness [mm]: ',$)
        READ(5,*) THICK
   45   format(/5x,'Enter the material density [kg/mt3]: ',$)
        write(6,45)
        read(5,*) RHO
        WRITE(6,40)
   40   FORMAT(/5X,'Enter E11 [GPa]:  '$)
        READ(5,*) E11
        WRITE(6,50)
```

```
   50   FORMAT(5X,'Enter E22 [GPa]:  '$)
        READ(5,*) E22
        WRITE(6,60)
   60   FORMAT(5X,'Enter E33 [GPa]:  '$)
        READ(5,*) E33
        WRITE(6,70)
   70   FORMAT(5X,'Enter G23 [GPa]:  '$)
        READ(5,*) G23
        WRITE(6,80)
   80   FORMAT(5X,'Enter G31 [GPa]:  '$)
        READ(5,*) G31
        WRITE(6,90)
   90   FORMAT(5X,'Enter G12 [GPa]:  '$)
        READ(5,*) G12
        WRITE(6,100)
  100   FORMAT(5X,'Enter NU12:  '$)
        READ(5,*) ENU12
        WRITE(6,110)
  110   FORMAT(5X,'Enter NU13:  '$)
        READ(5,*) ENU13
        WRITE(6,120)
  120   FORMAT(5X,'Enter NU23:  '$)
        READ(5,*) ENU23
C
C This section stores the inputed information in the direct access file
C "LIBRARY.BIN."
C
      mkflag=.false.
      current=matnum
      call save(matnum)
      if (mkflag) then
        call fmark
      end if
C
C
C
C ==================================================================
      go to 9

C==================================================================
C      OPTION 2: DELETE A MATERIAL FROM THE FILE
C
C      MDEL      MATERIAL NUMBER TO DELETE (-1 EXITS, 0 SHOWS MATERIALS)
C
C      LIST      DISPLAYS MATERIAL NAMES IN THE LIBRARY
C
C      SEE(N)    DISPLAYS RECORD N FROM MATERIAL LIST
C
C      BLANK(N)  DELETES RECORD N FROM MATERIAL LIST AND UPDATES LAST
C==================================================================

  150 WRITE(6,160)
  160 FORMAT(1h1,79(1h=),//' ',10x,'Option 2: Delete Material'//
```

```
     & 1h0,79(1h=)//////////////)
  165 write(6,170)
  170 format(' ','Material # to be deleted [0=list ,-1=exit]: ',$)
      read(5,*)mdel
      if (mdel .eq. 0) then
        call list
      else if (mdel .eq. -1) then
        go to 9
      else if (mdel .gt. 0 .and. mdel .lt. last) then
        call get(mdel)
        write(6,240)'Delete Material #',mdel,name,thick,rho,e11,g23,
     &   e22,g31,e33,g12,enu12,enu13,enu23
        write(6,180)
  180   format(' ',//,' Delete this material [y/n]: ',$)
        read(5,5)create
        if (create .eq. 'Y' .or. create .eq. 'y') then
                mkflag=.false.
                current=mdel
                call blank(mdel)
                if (mkflag) then
                        call fmark
                end if
                go to 9
        else
                go to 150
        end if
      else
        write(6,*)'*** INVALID CHOICE ***'
        go to 165
      end if
      go to 150

C=================================================================
C     OPTION 3: EDIT MATERIAL IN THE LIBRARY
C
C     MATN        Material number to edit (0=list, -1=exit)
C
C     NOPT        Edit option number (0=exit)
C
C     GET(N)      Gets material properties in record N
C
C     SAVE(N)     Saves material properties in record N
C=================================================================
  200 write(6,210)
  210 format(1h1,79(1h=)//15x,'Option 3: Edit Material Properties'/1h0,
     & 79(1h=)//////////////)
  215 write(6,220)
  220 format(' ','Enter material # to edit [0=see list, -1=exit]: ',$)
      read(5,*) matn
      if (matn .eq. 0) then
        call list
      else if (matn .eq. -1) then
        go to 9
```

```
      else if (matn .gt. 0 .and. matn .lt. last) then
        go to 230
      else
        write(6,*)'*** INVALID NUMBER ***'
        go to 215
      end if
      go to 200
 230  call get(matn)
      mflag=.false.
 235  write(6,240)'Edit Material #',matn,name,thick,rho,e11,g23,e22,
     & g31,e33,g12,enu12,enu13,enu23
 240  format (1h1,//' ',15x,a17,i2,//
     &'  1. Name:',a40,/
     &'  2. Thickness: ',f8.4,' [mm]',t35,'3. Density: ',f8.2,
     &' [kg/mt3]'//
     &'  4. E11: ',f12.6,' [GPa]',5x,'7. G23: ',f12.6,' [GPa]'/
     &'  5. E22: ',f12.6,' [GPa]',5x,'8. G31: ',f12.6,' [Gpa]'/
     &'  6. E33: ',f12.6,' [Gpa]',5x,'9. G12: ',f12.6,' [GPa]'//
     &' 10. NU12: ',f8.4,/
     &' 11. NU13: ',f8.4,/
     &' 12. NU23: ',f8.4,///////)
 250 write(6,260)
 260 format(' ','Enter item # to change [0=exit]: ',$)
      read(5,*)nopt
      if (nopt .gt. 12 .or. nopt .lt. 0) then
        write(6,*)'*** INVALID OPTION # ***'
        go to 250
      else if (nopt .eq. 0) then
        if (mflag) then
 270            write(6,280)
 280            format(' ','Save material changes [y/n]: ',$)
                read(5,5)create
                if (create .eq. 'Y' .or. create .eq. 'y') then
                        mkflag=.false.
                        current=matn
                        call save(matn)
                        if (mkflag) then
                                call fmark
                        end if
                else if (create .eq. 'N' .or. create .eq. 'n') then
                        go to 200
                else
                        go to 270
                end if
        else
                go to 200
        end if
      else
C=================================================================
C This section inputs value changes for the selected material
C=================================================================
        mflag=.true.
        go to(300,330,345,350,370,390,410,430,450,470,490,510)nopt
```

```
      end if
      go to 200
C=================== Change Material Name ==================

  300 write(6,*)'Name [40 characters max]'
  310 format(' ','"',40x,'"')
      write(6,310)
      write(6,320)
  320 format('+','"',$)
      read(5,325)name
  325 format(a40)
      go to 235

C=================== Change Thickness ==================

  330 write(6,340)
  340 format(' ','Thickness [mm]: ',$)
      read(5,*)thick
      if (thick .lt. 0) then
        write(6,*)'*** INVALID THICKNESS ***'
        go to 330
      else
        go to 235
      end if
C=================== Change Density ==================
  345 write(6,347)
  347 format(' ','Density [kg/m↑3]: ',$)
      read(5,*) rho
      if (rho .gt. 0) then
        go to 235
      else
        write(6,*)'*** INVALID DENSITY ***'
        go to 345
      end if

C=================== Change E11 ==================

  350 write(6,360)
  360 format(' ','E11 [Gpa]: ',$)
      read(5,*)E11
      go to 235

C=================== Change E22 ==================

  370 write(6,380)
  380 format(' ','E22 [GPa]: ',$)
      read(5,*)E22
      go to 235

C=================== Change E33 ==================

  390 write(6,400)
  400 format(' ','E33 [GPa]: ',$)
```

```
      read(5,*)E33
      go to 235

C=================== Change G23 ===================

  410 write(6,420)
  420 format(' ','G23 [GPa]: ',$)
      read(5,*)G23
      go to 235

C=================== Change G31 ===================

  430 write(6,440)
  440 format(' ','G31 [GPa]: ',$)
      read(5,*)G31
      go to 235

C=================== Change G12 ===================

  450 write(6,460)
  460 format(' ','G12 [GPa]: ',$)
      read(5,*)G12
      go to 235

C=================== Change NU12 ===================

  470 write(6,480)
  480 format(' ','NU12: ',$)
      read(5,*)ENU12
      go to 235

C=================== Change NU13 ===================

  490 write(6,500)
  500 format(' ','NU13: ',$)
      read(5,*)ENU13
      go to 235

C=================== Change NU23 ===================

  510 write(6,520)
  520 format(' ','NU23: ',$)
      read(5,*)ENU23
      go to 235

C============== END OF OPTION #3 ==============================

C=============================================================
C                OPTION #4: VIEW MATERIAL PROPERTIES
C=============================================================
  550 call list
      go to 9
      end
```

```
C=================================================================
C                          SUBROUTINE FMARK
C
C This section updates the end of the material data in the file
C LIBRARY.BIN by setting THICK=-1 in the record after the last material.
C This marker is looked for by parts of this program and by part of
C the program INPUT.
C=================================================================
      subroutine fmark

      implicit real*8 (a-h,o-z)
      character a$*40
      integer current

      common /block2/last,nlast,current

  700 a=0.
      a$=' '
      t=0.
      if (last .ne. current) then
        write(7,rec=last)a,a,a,a,a,a,a,a,a,a$,a,a,a
      end if
      if (nlast .eq. 0) then
        do while (.not. (t .gt. 0. .or. last .eq. 1))
                last = last-1
                read(7,rec=last)a,a,a,a,a,a,a,a,a,a$,t
        end do
        if (t .gt. 0.) then
                last=last+1
C       else
C               last = last   (implied, but not needed in code)
        end if
      else
        last=nlast
      end if
      a=0.
      a$=' '
      t=-1.
      write(7,rec=last)a,a,a,a,a,a,a,a,a,a$,t,a,a,a
      end
C=================================================================
C                          SUBROUTINE GET(N)
C
C  This routine gets the material information for material N out of
C  LIBRARY.BIN and stores it into the common variables:
C       E11, E22, E33, G23, G31, G12, ENU12, ENU13, ENU23, NAME,
C       THICK, ???, ???, ???
C          (??? denotes unassigned and NOT retrieved variables)
C  Make sure that any routine which utilizes GET(N) designates the
C  above variables as COMMON.
C=================================================================
      subroutine get(n)
```

```
      implicit real*8 (a-h,o-z)
      character name*40

      common /block1/e11,e22,e33,g23,g31,g12,enu12,enu13,enu23,
     &               name,thick,rho

      read(7,rec=n) e11,e22,e33,g23,g31,g12,enu12,enu13,enu23,
     *              name,thick,rho
      end

C=================================================================
C                        SUBROUTINE SAVE(N)
C
C This routine is the same as GET(N) except that the COMMON variables
C are stored in the file LIBRARY.BIN.
C=================================================================
      subroutine save(n)

      implicit real*8 (a-h,o-z)
      logical mkflag
      character name*40,query

      common /block1/e11,e22,e33,g23,g31,g12,enu12,enu13,enu23,
     &               name,thick,rho
      common /block2/last,nlast
      common /block3/mkflag
      write(7,rec=n)e11,e22,e33,g23,g31,g12,enu12,enu13,enu23,
     &              name,thick,rho
      if (n .eq. last .or. n .gt. last) then
        nlast=n+1
        mkflag=.true.
      end if
      end
C=================================================================
C                        SUBROUTINE BLANK(N)
C
C This routine DELETES a material from LIBRARY.BIN by writing a blank
C record to the file.  It also updates the variable LAST which marks
C the end of the file.
C=================================================================
      subroutine blank(n)

      logical mkflag
      common /block2/last,nlast
      common /block3/mkflag

      write(7,rec=n),0.,0.,0.,0.,0.,0.,0.,0.,0.,' ',0.,0.,0.,0.
      if (n .eq. last-1) then
        nlast=0
        mkflag=.true.
      end if
      end
```

```
C=================================================================
C                        SUBROUTINE LIST
C
C  This routine displays the material properties stored in the file
C  LIBRARY.BIN
C=================================================================

      subroutine list

      implicit real*8 (a-h,o-z)
      character name*40,tname*40,query*1

      common /block2/last,nlast

      n=1

  730 write(6,740)
      ncycle=1
      nstart=n
  740 format(' '////' ',12x,'Material Library: Material Names'//
     &' #   Name'/
     &' --  ----------------------------------------')
      do while(ncycle .lt. 16)
        if (n .lt. last) then
                read(7,rec=n)e1,e2,e3,e4,e5,e6,en12,en13,
     &                       en23,name,thick
        else
                do i=ncycle,15
                        write(6,*)' '
                end do
                ncycle=16
                go to 755
        end if
        if (thick .gt. 0.) then
                write(6,750)n,name
                ncycle=ncycle+1
        end if
        n=n+1
  750   format(' ',i2,3x,a40,g12.6)
  755 end do
  760 if (last .gt. n) then
        write(6,770)
  770   format(' [Enter -1=next page, 0=done, #=see properties]: ',$)
      else
        write(6,780)
  780   format(' [Enter 0=done, #=see material properties]: ',$)
      end if
      read(5,*)nop
      write(6,*)' '
      write(6,*)' '
      write(6,*)' '
      if (nop .eq. 0) then
        return
```

```
      else if (nop .eq. -1 .and. n .lt. last) then
        go to 730
      else if (nop .gt. 0 .and. nop .lt. last) then
        n=nstart
        read(7,rec=nop)e1,e2,e3,e4,e5,e6,en12,en13,en23,
     &                 name,thick,rho
        if (thick .gt. 0.) then
                write(6,790)'Material #',nop,name,thick,rho,e1,
     &          e4,e2,e5,e3,e6,en12,en13,en23
        else
                name='<undefined material>'
                write(6,790)'Material #',nop,name,0.,0.,
     &          0.,0.,0.,0.,0.,0.,0.,0.
        end if
  790 format(1h1,//,' ',15x,a17,i2,//
     &'     Name: ',a40,/
     &'     Thickness: ',g12.6,' [mm]',t35,'   Density: ',f8.2,
     &' [kg/m+3]'//
     &'     E11: ',g12.6,' [GPa]',5x,'   G23: ',g12.6,' [GPa]'/
     &'     E22: ',g12.6,' [GPa]',5x,'   G31: ',g12.6,' [GPa]'/
     &'     E33: ',g12.6,' [GPa]',5x,'   G12: ',g12.6,' [GPa]'//
     &'     NU12: ',g12.6,/
     &'     NU13: ',g12.6,/
     &'     NU23: ',g12.6,///////)
        write(6,800)
  800   format('0','[Hit RETURN when done]: ',$)
        read(5,810)query
  810   format(a1)
        go to 730
      else
        write(6,*)'*** INVALID NUMEBER ***'
        do n=1,1000
        end do
        n=nstart
        go to 730
      end if
      end
C
```

```
      program view
C
C Copyright c1989 Massachusetts Institute of Technology
C
C Permission to use, copy and modify this software and its documentation
C for internal purposes only and without fee is hereby granted provided
C that the above copyright notice and this permission appear on all
C copies of the code and supporting documentation.  For any other use of
C this software, in original or modified form, including but not limited
C to, adaptation as the basis of a commerical software or hardware
C product, or distribution in whole or in part, specific prior
C permission and/or the appropriate license must be obtained from MIT.
C This software is provided "as is" without any warranties whatsoever,
C either express or implied, including but not limited to the implied
C warranties of merchantability and fitness for a particular purpose.
C This software is a research program, and MIT does not represent that
C it is free of errors or bugs or suitable for any particular task.
C
C Program written by Chris Park 1988, modifications by Kiernan F. Ryan.
C  This program displays the contents of the binary file
C  used by the programs BESSEL and GLOBAL.  Limited editing
C  to the binary file can be accomplished using the program
C  INPUT.

      implicit real*8 (a-h,o-z)
      real*8 idd
      character*60 title1,title2,title3
      character filnam*20, status*23, bc*35, ofile*20,
     & idhzs*5, id$*2, outyp, q,atemp*3

      dimension a(3,3),alphaa(10),tqa(10),ept_max(3,3),
     & epc_max(3,3), bc(6), que(0:1)

      logical exst

C
C input the filename and check to see that it exists in the
C current directory
C
      write(6,10)
   10 format(////' ',t20,'Binary File Viewer'///
     &' ','Filename: ',$)
      read(5,20)filnam
   20 format(a)
      inquire(file=filnam,exist=exst)
      if (.not.(exst)) then
      write(6,30)
   30 format(//' ','That file does not exist in the current directory'/
     &' ','Aborting Program')
      stop
      end if
C
C read in data from the binary file
```

```
C
C REC #   VARIABLE   UNITS   COMMENTS
C ------  --------   -----   --------
C   1     TITLE1            First line of title block
C   2     TITLE2            Second line of title block
C   3     TITLE3            Third line of title block
C   4     NP                Number of plies  [integer]
C   5     THICK      mm     Thickness of laminant
C   6     A(1,1)     MN/m   A44
C   6     A(1,2)     MN/m   A45
C   6     A(2,2)     MN/m   A55
C   7     E11        GPa    Etheta        - major axis
C   7     E1Z        GPa    Ez            - major axis
C   7     ENU12             nu r-theta    - major axis
C   7     ENU13             nu r-z        - major axis
C   7     G13        GPa    G r-z         - major axis
C   8     E22        GPa    Etheta        - minor axis
C   8     E2Z        GPa    Ez            - minor axis
C   8     ENU21             nu r-theta    - minor axis
C   8     ENU23             nu r-z        - minor axis
C   8     G23        GPa    G r-z         - minor axis
C   9     D1111      Nm     D1111
C   9     D1111      Nm     D1111
C   9     D1122      Nm     D1122
C   9     D2222      Nm     D2222
C   9     D1212      Nm     D1212
C   9     D1112      Nm     D1112
C   9     D2212      Nm     D2212
C  10     A$         m      Specimen length (X-direction)
C  10     B          m      Specimen length (Y-direction)
C  11     RHO        kg/m↑3 Specimen density
C  12     PNX        N/m    In-plane load (X-direction)
C  12     PNY        N/m    In-plane load (Y-direction)
C  13     NX                B.C. number (X-direction)  [integer]
C  13     NY                B.C. number (Y-direction)  [integer]
C  14     SHEAR             Shear correction factor
C  15     BM         kg     Impactor mass
C  15     VI         m/s    Impactor velocity
C  16     RI         m      Impactor radius
C  17     LDHZS             Input loading type HZS
C  18     CM         kg     Concentrated mass at center of plate
C  19     R          mm     Radius of the region to be analyzed
C  20     NOHARK            Number of harmonics to be used for
C                                 calculating the hertzian spring
C                                 constant in BESSEL  [integer]
C  20     NOHARS            Number of harmonics to be used for
C                                 calculating the strain field in
C                                 program BESSEL  [integer]
C  21     NUMALP            Number of alphas to be entered.  MAX = 10
C                                 [integer]
C  21     ALPHAA(N)  mm     Array of alphas for K calculation
C  22     TQA(N)     N      Array of total load target values
C  23     LD$               Input loading type (PT, HZ, or HT)
```

```
C (23)     XEP        mm    Loading radius  (For LD$ = HT)
C  24      NRS              Number of radius divisions for the output
C                                 of the strain field  [integer]
C  25      OUTYP            Flag.  If OUTYP = Y, then the strain field
C                                 is calculated at each theta division,
C                                 radius step, and ply.
C (25)     NTH              Theta divisions  (For OUTYP = Y)  [integer]
C (25)     EPT_MAX          Matrix of allowable tensor strains; tensor
C                                 notation  (For OUTYP = Y)
C                                 [order: (1,1), (1,2), (1,3), (2,2),
C                                         (2,3),(3,3)]
C (25)     EPC_MAX          Matrix of allowable tensor strains; tensor
C                                 notation  (For OUTYP = Y)
C                                 [order: (1,1), (1,2), (1,3), (2,2),
C                                         (2,3), (3,3)]
C  26      IX               Number of X-axis modes to be used in program
C                                 IMPACT  [integer]
C  26      IY               Number of Y-axis modes to be used in program
C                                 IMPACT  [integer]
C  27      IFC              Flag.  If IFC = 1, then the rotary inertia
C                                 is condensed  [integer]
C  27      MELIM            Flag.  If MELIM = 1, then the even modes are
C                                 eliminated from the analysis in IMPACT
C                                 [integer]
C  28      DELT  micro_s    Time step length
C  28      NT               Number of time steps  [integer]
C  29      HK    N/m↑1.5    Hertzian spring constant
C  30      LDD      N       Peak force calculated in IMPACT
C  30      ACC    m/s↑2     Acceleration corresponding to the peak force
C
C  OTHER VARIABLES:
C
C  FILNAM        Name of binary file to view
C  STATUS        Contains the next step of analysis for which the file
C                contains sufficient data. [BESSEL (Hertzian), Global,
C                or BESSEL (Stress-Strain)]
C  BC(N)         Array containing B.C. descriptions
C
      write(6,5) filnam
    5 format(//t20,'Reading data from ',a20//)
      open(unit=12, file=filnam, status='old',access='direct',
     &     recl=110)
      read(12,rec=1) title1
      read(12,rec=2) title2
      read(12,rec=3) title3
      read(12,rec=4) np
      read(12,rec=5) thick
      read(12,rec=6) a(1,1), a(1,2), a(2,2)
      read(12,rec=7) e11, e1z, enu12, enu13, g13
      read(12,rec=8) e22, e2z, enu21, enu23, g23
      read(12,rec=9) d1111,d1122,d2222,d1212,d1112,d2212
      read(12,rec=10) a$, b
      read(12,rec=11) rho
```

```
      read(12,rec=12) pnx, pny
      read(12,rec=13) nx, ny
      read(12,rec=14) shear
      read(12,rec=15) bm, vi
      read(12,rec=16) ri
      read(12,rec=17) idhzs
      read(12,rec=18) CM
      read(12,rec=19) r
      read(12,rec=20) nohark, nohars
      read(12,rec=21) numaip,(alphaa(i), i=1,numaip)
      read(12,rec=22) (tqa(i), i=1,numaip)
      read(12,rec=23) atemp
      if (atemp(1:1) .eq. ' ') then
        id$=atemp(2:3)
      else
        id$=atemp(1:2)
      end if
      if (id$ .eq. 'HT') then
        read(12,rec=23) id$,xep
      end if
      read(12,rec=24) nrs
      read(12,rec=25) atemp
      if (atemp(1:1) .eq. ' ') then
         outyp=atemp(2:2)
      else
        outyp=atemp(1:1)
      end if
      if (outyp .eq. 'Y') then
        read(12,rec=25) outyp,nth,((ept_max(i,j),j=i,3),i=1,3),
     &          ((epc_max(i,j),j=i,3), i=1,3)
      end if
      read(12,rec=26) ix, iy
      read(12,rec=27) ifc, melim
      read(12,rec=28) delt, nt
      read(12, err=50, rec=29) hk
      read(12, err=60, rec=30) idd, acc

C
C assign a STATUS value
C
      STATUS='BESSEL (Stress-Strain)'
      go to 70
   50 STATUS='BESSEL (Hertzian)'
      go to 70
   60 STATUS='IMPACT'
   70 close(unit=12)
C
C initialize B.C. array
C
      bc(1)='Simply Supported - Simply Supported'
      bc(2)='Clamped - Free'
      bc(3)='Clamped - Clamped'
      bc(4)='Free - Free'
```

```
      bc(5)='Simply Supported - Clamped'
      bc(6)='Simply Supported - Free'
C
C Initialize QUE array
C
      que(0)='No'
      que(1)='Yes'
C
C write the file contents to the screen in pages
C
C write page 1
C
   80 write(6,160) filnam, status
      write(6,90) title1,title2,title3,np,thick,
     &a(1,1), a(1,2), a(2,2)
   90 format(' ','Title Block:'//
     &' ','     (1) ',a60/
     &' ','     (2) ',a60/
     &' ','     (3) ',a60///
     &' ','(4) # plies: ',i2,t30,'(5) Laminant thickness: ',f8.4,' mm'/
     &/' (6): A44: ',f8.4,' MN/m',4x,'A45: ',f8.4,' MN/m',4x,'A55: ',
     &f8.4,' MN/m'//////)
      write(6,150)
  100 format(a1)
      read(5,100) q
      if (q .eq. 'Q' .or. q .eq. 'q') go to 1000
C
C write page 2
C
      write(6,160) filnam,status
      write(6,110) e11,e22,e1z,e2z,enu12,enu21,
     &enu13,enu23,g13,g23,d1111,d1212,d1122,d1112,d2222,d2212
  110 format(' ','(7) Major Axis:',t30,'(8) Minor Axis:'/
     &' ','    ----------',t30,'    ----------'/
     &' ','    E11:   ',f8.4,' GPa',t35,'E22:   ',f8.4,' GPa'/
     &' ','    E33:   ',f8.4,' GPa',t35,'E33:   ',f8.4,' GPa'/
     &' ','    ENU12: ',f8.4,'    ',t35,'ENU21: ',f8.4/
     &' ','    ENU13: ',f8.4,'    ',t35,'ENU23: ',f8.4/
     &' ','    G13:   ',f8.4,' GPa',t35,'G23:   ',f8.4,' GPa'//
     &' ','(9) D - Matrix:'//
     &' ','    D1111: ',f8.4,' Nm',t35,'D1212: ',f8.4,' Nm'/
     &' ','    D1122: ',f8.4,' Nm',t35,'D1112: ',f8.4,' Nm'/
     &' ','    D2222: ',f8.4,' Nm',t35,'D2212: ',f8.4,' Nm'//)
      write(6,150)
      read(5,100) q
      if (q .eq. 'Q' .or. q .eq. 'q') go to 1000
C
C write page 3
C
      write (6,160) filnam, status
      write (6,120) a$, b, rho, pnx, pny, nx, bc(nx),
     & ny, bc(ny), shear, bm, vi, ri
  120 format(' ','(10) Specimen length:  X-direction: ',f8.4,
```

```
     &' m',t50,'Y-direction: ',f8.4,' m'/
     &' ','(11) Specimen density: ',f8.2,' kg/m↑3'//
     &' ','(12) In-plane load: X-direction: ',f8.4,' N/m',t50,
     &  'Y-direction: ',f8.4,' N/m'//
     &' ','(13) Boundary conditions:'/
     &' ',t10,'X-direction: [',i1,'] ',a35/
     &' ',t10,'Y-direction: [',i1,'] ',a35//
     &' ','(14) Shear correction factor: ',f8.4//
     &' ','Impactor:'/
     &' (15) mass: ',f8.4,' kg',3x,'(15) velocity: ',f8.4,' m/s',3x,
     &'(16) radius: ',f8.4,' mm'//)
      write(6,150)
      read(5,100) q
      if (q .eq. 'Q' .or. q .eq. 'q') go to 1000
C
C write page 4
C
      write(6,160) filnam, status
      write(6,130) idhzs, CM, r, nohark, nohars,
     & numaip
  130 format(' ','(17) Input loading type: ',a5/
     &' ','(18) Concentrated mass: ',F8.6,' kg'/
     &' ','(19) Radius of analyzed region: ',f8.4,' mm'/
     &' ','(20) # harmonics:   Hertzian: ',i3,t45,'Strain field: ',
     & i3//
     &' ','(21) # alphas: ',i2/
     &' ','       (21) alpha [mm]',t30,'(22) target load [N]'/
     &' ','       ---------------',t30,'--------------------')
      do i=1,10
        if (i .gt. numaip) then
                write(6,*)
        else
                write(6,140) alphaa(i),tqa(i)
        end if
      end do
  140 format(' ',10x,f8.4,t35,f8.4)
      write(6,150)
  150 format(' ',t10,'[Enter "Q" to quit or hit RETURN',
     &' for next page]',$)
      read(5,100) q
      if (q .eq. 'Q' .or. q .eq. 'q') goto 1000
  160 format(///25x,'Binary File Viewer',//
     &' ','File: ',a20/
     &' ','Next phase: ',a23/)
C
C write page 5
C
      write(6,160) filnam, status
      write(6,170) id$
  170 format(' ','(23) Input loading type: ',a2,$)
      if (id$ .eq. 'HT') write(6,180) xep
  180 format('+','         loading radius: ',f8.4,' mm')
      write(6,190) nrs
```

```
  190 format(' ','(24) # Radius divisions: ',i3/)
      if (outyp .eq. 'N') then
        write(6,*)'(25) Calculate strain field at ',
     &          'each theta division, ply, and step: No'
      else if(outyp .eq. 'Y') then
        write(6,195) nth, (EPT_MAX(1,j),j=1,3),(EPC_MAX(1,j),j=1,3),
     &  EPT_MAX(1,2),EPT_MAX(2,2),EPT_MAX(2,3),EPC_MAX(1,2),
     &  EPC_MAX(2,2),EPC_MAX(2,3),EPT_MAX(1,3),EPT_MAX(2,3),
     &  EPT_MAX(3,3),EPC_MAX(1,3),EPC_MAX(2,3),EPC_MAX(3,3)
  195   format(' ','(25) Calculate strain field at each theta ',
     &  'division, ply, and step: Yes'/
     &  ' ','# theta divisions: ',i2/
     &  'Max tensile strains (tensor):',t35,'Max compressive strains',
     &  ' (tensor):'//' ',f8.4,t12,f8.4,t24,f8.4,t40,f8.4,t52,f8.4,
     &  t64,f8.4/
     &  f8.4,t12,f8.4,t24,f8.4,t40,f8.4,t52,f8.4,t64,f8.4/
     &  f8.4,t12,f8.4,t24,f8.4,t40,f8.4,t52,f8.4,t64,f8.4//)
      end if
      write(6,200) ix,iy,ifc,que(ifc),melim,que(melim),delt,nt,
     & hk*1.E-9,idd,acc
  200 format(/' ','(26) IMPACT modes:     X-axis: ',i3,t45,'Y-axis: ',
     & i3//' ','(27) Condense rotary inertia:       [',i1,'] ',a3/
     &' ','(27) Eliminate even modes in IMPACT: [',i1,'] ',a3//
     &' ','(28) Time step length: ',f8.4,' micro_s',t45,
     &'# time steps: ',i4//
     &' ','(29) Hertzian spring constant: ',f8.4,' GN/m+1.5'//
     &' ','(30) Peak force from IMPACT: ',f16.4,' N'/
     &' ','(30) Acceleration: ',f16.4,' m/s+2'/)
      write(6,150)
      read(5,100) q
      if (q .eq. 'Q' .or. q .eq. 'q') go to 1000
      go to 80
C
C ------------------------------------------------------------------
C
C  THIS SECTION PRINTS THE CONTENTS OF THE DIRECT ACCESS FILE IN THE
C  FILE 'VIEW.OUT'.
C
C ------------------------------------------------------------------
C
C
C  OPEN THE FILE 'VIEW.OUT'.
C
 1000 OPEN(UNIT=11,IOSTAT=INPERR,ERR=2010,FILE='VIEW.OUT',
     *     STATUS='NEW')
C
      WRITE(11,1151) filnam
 1151 FORMAT(1H0///27X,'CONTENTS OF "VIEW.OUT"'///8X,'This data is sto
     *red in binary form in the direct access file'/8X,'"'a'".'//
     *)
      WRITE(11,1155)
 1155 FORMAT(8X,'RECORD'/8X,'NUMBER    CONTENTS'/8X,'------    --------'
     *//)
```

```
      WRITE(11,1160)TITLE1,TITLE2,TITLE3
 1160 FORMAT(8X,'1-3',5X,'Title Block:'/13X,A60/13X,A60/13X,A60//
     *       /13X,'Output from ROTATE'/13X,18(1H-)//)
      WRITE(11,1162)NP
 1162 FORMAT(10X,'4',5X,'Number of plies in the laminate =',I4/)
      WRITE(11,1165)THICK
 1165 FORMAT(10X,'5',5X,'Thickness of the laminate =',F9.5,' mm'/)
      WRITE(11,1170)A(1,1),A(1,2),A(2,2)
 1170 FORMAT(10X,'6',5X,'A44 =',F8.4,' MN/m'/10X,'6',5X,'A45 =',F8.4,
     *       ' MN/m'/10X,'6',5X,'A55 =',F8.4,' MN/m'/)
      WRITE(11,1175)E11,E1Z,ENU12,ENU13,G13
 1175 FORMAT(10X,'7',5X,'Etheta     - major axis =',F8.3,' GPa'/
     *       10X,'7',5X,'Ez         - major axis =',F8.3,' GPa'/
     *       10X,'7',5X,'nu r-theta - major axis =',F10.6/
     *       10X,'7',5X,'nu r-z     - major axis =',F10.6/
     *       10X,'7',5X,'G r-z      - major axis =',F8.4,' GPa'/)
      WRITE(11,1180)E22,E2Z,ENU21,ENU23,G23
 1180 FORMAT(10X,'8',5X,'Etheta     - minor axis =',F8.3,' GPa'/
     *       10X,'8',5X,'Ez         - minor axis =',F8.3,' GPa'/
     *       10X,'8',5X,'nu r-theta - minor axis =',F10.6/
     *       10X,'8',5X,'nu r-z     - minor axis =',F10.6/
     *       10X,'8',5X,'G r-z      - minor axis =',F8.4,' GPa'/)
      WRITE(11,1181)D1111,D1122,D2222,D1212,D1112,D2212
 1181 FORMAT(10X,'9',5X,'D1111 =',F10.5,' Nm'/
     *       10X,'9',5X,'D1122 =',F10.5,' Nm'/
     *       10X,'9',5X,'D2222 =',F10.5,' Nm'/
     *       10X,'9',5X,'D1212 =',F10.5,' Nm'/
     *       10X,'9',5X,'D1112 =',F10.5,' Nm'/
     *       10X,'9',5X,'D2212 =',F10.5,' Nm'//
     *       /13X,'Input from Master Input File'/13X,28(1H-)//)
      WRITE(11,1182)A$,B
 1182 FORMAT(10X,'10',4X,'Test specimen length in x-direction =',F10.6,
     *  ' m'/10X,'10',4X,'Test specimen length in y-direction =',F10.6,
     *  ' m'/)
      WRITE(11,1184)RHO
 1184 FORMAT(10X,'11',4X,'Density of test the specimen =',F8.1,
     &       ' kg/m+3'/)
      WRITE(11,1186)PNX,PNY
 1186 FORMAT(10X,'12',4X,'In-plane load in x-direction =',F8.4,' N/m'/
     *       10X,'12',4X,'In-plane load in y-direction =',F8.4,' N/m'/)
      WRITE(11,1188)NX,NY
 1188 FORMAT(10X,'13',4X,'B.C. number for x-direction (integer) =',I3/
     &       10X,'13',4X,'B.C. number for y-direction (integer) =',I3/)
      WRITE(11,1189)SHEAR
 1189 FORMAT(10X,'14',4X,'Shear correction factor (real) =',F9.5/)
      WRITE(11,1190)BM,VI
 1190 FORMAT(10X,'15',4X,'Impactor mass =',F11.7,' kg'/
     *       10X,'15',4X,'Impactor velocity =',F8.3,' m/s'/)
      WRITE(11,1192)RI
 1192 FORMAT(10X,'16',4X,'Impactor radius =',F8.3,' mm'/)
      WRITE(11,1193)LDHZS
 1193 FORMAT(10X,'17',4X,'Input loading type HZS =',A5/)
      WRITE(11,1200)CM
```

```
1200 FORMAT(10X,'18',4X,'Concentrated mass at center of plate = ',
    &       F8.6,' kg'/)
     WRITE(11,1210)R
1210 FORMAT(10X,'19',4X,'Radius of the region to be analyzed =',F8.3,
    &       ' mm'/)
     WRITE(11,1220)NOHARK
1220 FORMAT(10X,'20',4X,'Number of harmonics to be used for'
    &  ' calculating the '/20X,'hertzian spring constant in program',
    &  ' BESSEL.'/20X,'(integer) =',I5)
     WRITE(11,1225)NOHARS
1225 FORMAT(10X,'20',4X,'Number of harmonics to be used for'
    &  ' calculating the '/20X,'strain field in program',
    &  ' BESSEL.  (integer) =',I5/)
     WRITE(11,1230)NUMALP
1230 FORMAT(10X,'21',4X,'Number of alphas to be entered (integer) =',
    &       I3)
     DO 1250 I=1,NUMALP
        WRITE(11,1240)I,ALPHAA(I)
1240    FORMAT(10X,'21',4X,'ALPHA(',I2,') =',F10.6,' mm')
1250 CONTINUE
     WRITE(11,*)' '
     DO 1270 I=1,NUMALP
        WRITE(11,1260)I,TQA(I)
1260    FORMAT(10X,'22',4X,'Total load target value (',I2,') =',
    &          F9.2,' N')
1270 CONTINUE
     WRITE(11,1280)LD$
1280 FORMAT(/10X,'23',4X,'Input loading type (PT, HZ, or HT) =  ',A2/)
     IF (LD$$ .EQ. 'HT') WRITE(11,1290)XEP
1290 FORMAT(10X,'23',4X,'Loading radius (for loading type HT) =',
    &       F8.3,' mm')
     WRITE(11,1300)NRS
1300 FORMAT(10X,'24',4X,'Number of radius divisions'/
    &       20X,'(For output of the strain field.)  (integer) =',I3/)
     WRITE(11,1310)OUTYP
1310 FORMAT(10X,'25',4X,'Flag = ',A1,'   If flag = Y, then the strain',
    &       ' field is'/27X,'calculated at each theta division, '
    &       /27X,'each radius step, and each ply.')
     IF(OUTYP .EQ. 'Y') THEN
        WRITE(11,1320)NTH
1320    FORMAT(10X,'25',4X,'Theta divisions (integer) =',I4)
        WRITE(11,1330)EPT_MAX(1,1),EPT_MAX(1,2),EPT_MAX(1,3),
    &                 EPT_MAX(2,2),EPT_MAX(2,3),EPT_MAX(3,3),
    &                 EPC_MAX(1,1),EPC_MAX(1,2),EPC_MAX(1,3),
    &                 EPC_MAX(2,2),EPC_MAX(2,3),EPC_MAX(3,3)
1330    FORMAT(10X,'25',4X,'Maximum allowable tensile strains ',
    &          '(tensor notation):'/
    &          20X,'Epsilon 11 tensile max =',F11.7/
    &          20X,'Epsilon 12 tensile max =',F11.7/
    &          20X,'Epsilon 13 tensile max =',F11.7/
    &          20X,'Epsilon 22 tensile max =',F11.7/
    &          20X,'Epsilon 23 tensile max =',F11.7/
    &          20X,'Epsilon 33 tensile max =',F11.7/
```

```
     &          10X,'25',4X,'Maximum allowable compressive strains ',
     &          '(tensor notation):'/
     &          20X,'Epsilon 11 compressive max =',F11.7/
     &          20X,'Epsilon 12 compressive max =',F11.7/
     &          20X,'Epsilon 13 compressive max =',F11.7/
     &          20X,'Epsilon 22 compressive max =',F11.7/
     &          20X,'Epsilon 23 compressive max =',F11.7/
     &          20X,'Epsilon 33 compressive max =',F11.7/)
      ENDIF
      WRITE(11,1340)IX,IY
 1340 FORMAT(/10X,'26',4X,'Number of x-axis modes to be used in'/
     &       20X,'program IMPACT.  (integer) =',I4/
     &       10X,'26',4X,'Number of y-axis modes to be used in'/
     &       20X,'program IMPACT.  (integer) =',I4/)
      WRITE(11,1350)IFC,MELIM
 1350 FORMAT(10X,'27',4X,'Flag = ',I1,'   If flag = 1, then the rotary'/
     &       20X,'inertia is condensed.'/
     &       10X,'27',4X,'Flag = ',i1,'   If flag = 1, then the even',
     &       ' modes are'/20X,'eliminated from the analysis in IMPACT.'
     &       /)
      WRITE(11,1360)DELT,NT
 1360 FORMAT(10X,'28',4X,'Time step length =',F9.5,' micro_s'/
     &       10X,'28',4X,'Number of time steps (integer) =',I5/)
      if (status .ne. 'BESSEL (Hertzian)') then
      write(11,1365)
 1365 format(//13x,'Output from BESSEL'/
     &         13x,'------------------'//)
      WRITE(11,1370) HK*1.E-9
      end if
 1370 FORMAT(10X,'29',4X,'Hertzian spring constant = ',f8.4,
     &       ' GN/m**1.5'/)
      if (status .eq. 'BESSEL (Stress-Strain)')then
      write(11,1375)
      WRITE(11,1380) LDD, ACC
 1375 format(//13x,'Output from IMPACT'/
     &         13x,'------------------'//)
      end if
 1380 Format(10x,'30',4x,'Peak force computed by IMPACT = ',f16.4,
     &' N'/10x,'30',4x,'Acceleration at peak force = ',f16.4,
     &' m/s**2')
C
C  CLOSE THE FILE 'VIEW.OUT'.
C
      CLOSE(UNIT=11)
C
      write(6,*)
      write(6,*) 'Binary file information sent to VIEW.OUT.'
      stop
 2000 WRITE(6,*)'ERROR IN OPENING "DIRACC.BIN".  STATUS = ',INPERR
      STOP
 2010 WRITE(6,*)'ERROR IN OPENING "ROTATE.OUT".  STATUS = ',INPERR
      STOP
      END
```

```
$ FILE = F$SEARCH ("ANALYSIS.COM")          ! See if ANALYSIS.COM exists.
$ IF FILE .NES. "" THEN SET PROTECT=(S=RWED,O=RWED) ANALYSIS.COM;*
$ IF FILE .NES. "" THEN DEL ANALYSIS.COM;*  ! If so, delete all versions.
$ FILE = F$SEARCH ("CONFIG.BIN")            ! Same with CONFIG.BIN.
$ IF FILE .NES. "" THEN SET PROTECT=(S=RWED,O=RWED) CONFIG.BIN;*
$ IF FILE .NES. "" THEN DEL CONFIG.BIN;*
$ DEFINE/USER_MODE SYS$INPUT SYS$COMMAND:   ! Set input as keyboard.
$ RUN INPUT                                 ! Run the controlling program.
$ FILE = F$SEARCH ("CONFIG.BIN")            ! See if CONFIG.BIN exists.
$ IF FILE .EQS. "" THEN EXIT                ! If not, don't run analysis.
$ ! PROTECT THE ANALYSIS FILES FROM ACCIDENTAL DELETION DURING RUN
$ SET PROTECT = (S=RWE,O=RWE,G=RE,W=RE) ANALYSIS.COM
$ SET PROTECT = (S=RWE,O=RWE,G=RE,W=RE) CONFIG.BIN
$ SUBMIT/QUE=GUS_BATCH ANALYSIS             ! Otherwise, run the analysis.
```

APPENDIX B

Operation of ACPI (Analysis of Composite Plate Impact)

This appendix is provided for those who wish to run the computer codes listed in Appendix A. A flow chart illustrating the order in which the codes are run, and the data files generated, is shown in Figure B.1. The system may look complex, but its operation is very easy.

Program INPUT is the program which asks for the inputs needed for all of the programs in ACPI. It creates a sequential file called '*.DAT', where the asterisk represents a job name given by the user. The file LIBRARY.BIN is needed to run this program. This binary file contains material properties of several material systems. Additional material systems may be added to LIBRARY.BIN by running program LIBRARY.

Program ROTATE reads the sequential file '*.DAT' and calculates the average material properties and the D matrix for the laminate. It then places all of the information into a binary file called '*.BIN'. The output file 'ROTATE.OUT' is created. This file lists all of the users inputs and the output data from program ROTATE so that the user may have a hard copy of his or her work. The file '*.ANGLE.DAT' is also created by program ROTATE. This file contains the ply angle information and is only needed for the stress/strain calculation part of the local analysis.

Program BESSEL contains the local model analysis. The file '*_HERTZ.OUT' is the output file. It lists the loads for the inputted indentations. Detailed information on the modeled pressure distribution is also sum-

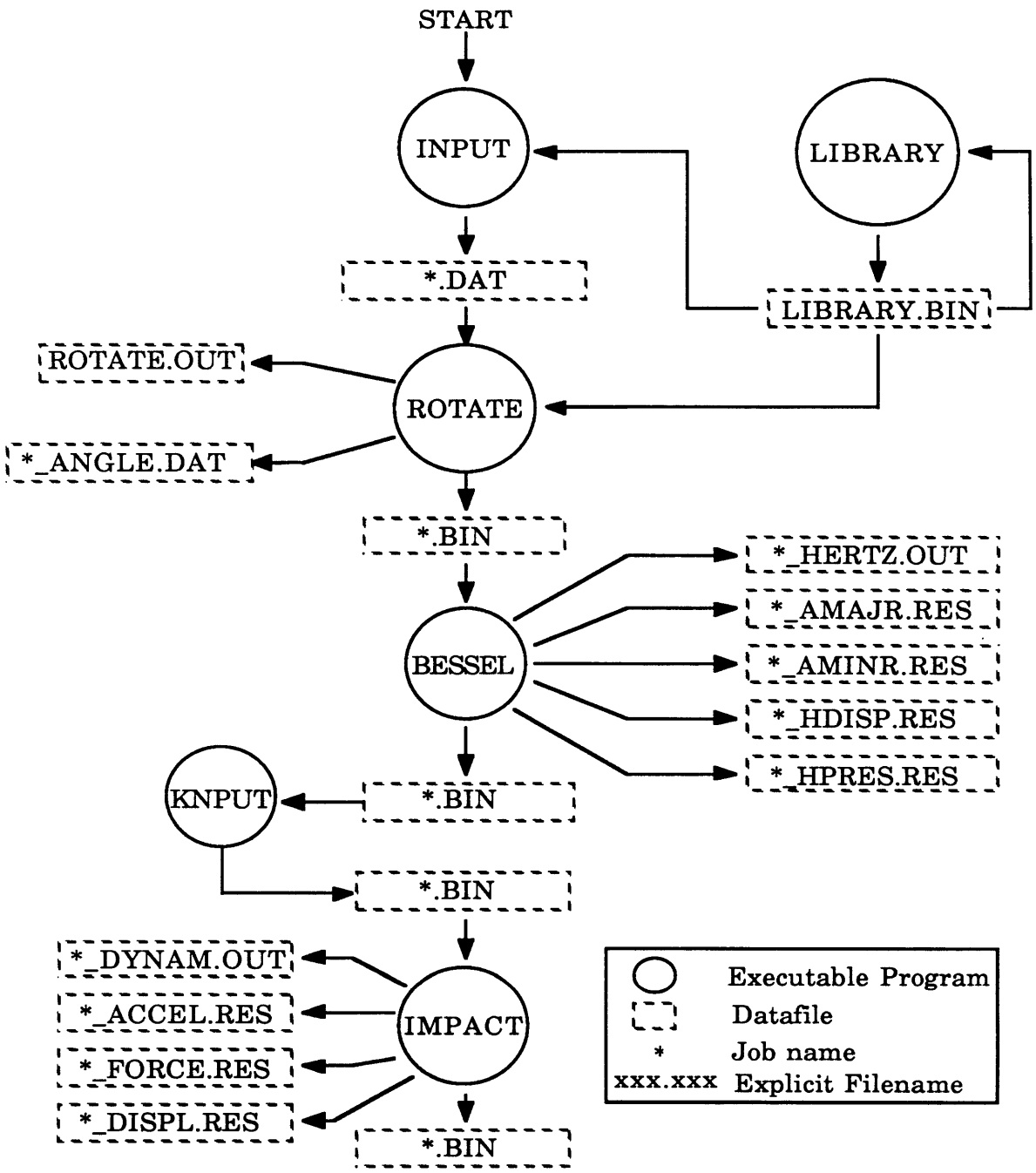


FIGURE B.1 Impact analysis flow chart.

marized to allow the user to make a quick check on the convergence of the solution. The files '*_AMAJR.RES' and '*_AMINR.RES' contain the indentation and load information for the major axis analysis and the minor axis analysis, respectively. The files '*_HDISP.RES' and '*_HPRES.RES' may be created to check the convergence of the solution. The file '*_HDISP.RES' contains the information needed to plot the recovered indentation (indentation calculated as the top surface displacement minus the bottom surface displacement) versus the number of modes. This number should converge to a final value. The file '*_HPRES.RES' contains the information needed to plot the pressure at the center of the plate against the number of modes used in the analysis. This plot should also converge to a final value. If these files are insufficient to confirm convergence, then the program may be run interactively. More options are available in the interactive mode since it was being used as a research tool.

Program KNPUT is a simple program that is run interactively. It places the values of K and n in Eq. (2.1) into the binary file '*_BIN'.

Program IMPACT contains the global analysis. The file '*_DYNAM.OUT' is the output file listing the maximum force, the corresponding acceleration, and the time at which they occur. Additional information is provided which indicates the progress of the program. The files '*_ACCEL.RES', '*_FORCE.RES', and '*_DISPL.RES' contain the information needed to plot the acceleration, force, and displacement histories, respectively. These files can become quite large, so program IMPACT is programmed to dump information at every tenth time step rather than every time step. This can be changed by changing the value of NDELW at the beginning of the subroutine MARCH. The peak force and corresponding acceleration information is placed in '*.BIN' for future use.

Program VIEW (which is not shown in the flow chart) allows the user to view the contents of '*.BIN'. It is run interactively and is useful if one wishes to check on the contents of a binary file. Note that the binary file may not be looked at without this program since the data is stored in binary form. The information is printed to the file 'VIEW.OUT' so that the user may have a hard copy.

To run the programs, the FORTRAN source codes must be compiled into executable programs. The file ACPI.COM is the file which controls the execution of the programs. It needs to be in the same directory as the executable programs. This controlling file is activated by typing @ACPI.

The first program run in ACPI is program INPUT. This program has a main menu from which the user has several options. The first option allows the user to define the parameters needed to run all of the analyses. The computer asks for the input parameters in a user-friendly manner. Default values may be chosen for some of the questions. Other questions, which deal with how the analysis is run, suggest appropriate inputs. Once all of the questions are answered, the information is stored in the data file '*.DAT'. Any incorrect inputs may be corrected with an edit mode option. This option is also useful if one wishes to run parametric studies. Rather than having to answer all of the questions again, the edit option may be used to change one or more variables.

Once the data file is created, the user may choose the starting point and the exit point of the analysis. If it is the first time that the analysis is being run, then program ROTATE must be run first. If the analysis has already been run, then the user may start with a program other than ROTATE. For example, if one runs the analysis through program IMPACT and then decides to change the time step, he of she does not have to run programs ROTATE and BESSEL a second time, but may start the analysis with program IMPACT. In

addition to choosing the starting and exit points, the user may also choose which output files to create. This option is intended to allow the user to suppress the generation of unnecessary files, and is very useful if storage space is limited.

Ideally, one could start the analysis with the material properties and end with the dynamic analysis. However, recent changes in the global analysis has not permitted time to fully adjust the rest of the system. What must therefore be done is to run the analysis through program BESSEL to obtain the contact relation. Once the exponent n and stiffness K in Eq. (2.1) are calculated, the program KNPUT must be run. This program places the values of n and K into the binary file. This program must also be run after making changes with the edit option of ACPI. Once the values of n and K are in place, the user may return to the ACPI system and run the dynamic analysis.

A file called CONFIG.BIN is created when the ACPI system is run. This file is used as a flag by the individual programs to let them know that they are running as part of a system. The entire analysis is automatically submitted as a batch job. Once the analysis is complete, ACPI deletes CONFIG.BIN. If the programs are run separately, one should be sure that this file does not exist in the directory.

All of the programs may be run separately. This is done by typing RUN, a space, the program name, and hitting the return key. All programs, except program IMPACT, are run interactively with keyboard input from the user. Program IMPACT requires a sequential file with all of the inputs, when it is not run as a part of ACPI. The details on the inputs for this file, which must be called IMPACT.DAT, are listed in the comment section of subroutine INPUT of program IMPACT.