This is to certify that the

thesis entitled

FINITE ELEMENT METHODS USED IN THE ANALYSIS OF RUNNING SHOE SOLES

presented by

Maureen Ann Clements

has been accepted towards fulfillment of the requirements for

M.S. degree in Mech. Eng.

R W Soutas - Little
Major professor

Date 10-29-85

O-7639

FINITE ELEMENT METHODS USED IN THE ANALYSIS OF RUNNING SHOE SOLES

By

Maureen Ann Clements

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1985

ABSTRACT

FINITE ELEMENT METHODS USED IN THE ANALYSIS OF RUNNING SHOE SOLES

By

Maureen Ann Clements

This thesis presents a design tool for the analysis of running shoe sole prototypes. A computer program was written to serve as a very specific preprocessor for ANSYS, a finite element program. The simplification of the finite element modeling process will be demonstrated. This preprocessor uses a general physical geometry to model a variety of shoe sole designs. Slightly different physical geometries can be artificially modeled by material property manipulation, as long as the basic configuration of the shoe is the same. Running shoe soles are made up of multiple layers, usually containing outsole, midsole, and wedge sections. The preprocessor allows for substitution of a layer while keeping the others constant. The material properties in any layer can be specified or changed easily leading to a versatile design tool. Displacement and stress plots show variations in material configurations and provide trends to guide design changes.

ACKNOWLEDGEMENTS

I would like to express my thanks and gratitude to my major professor Dr. Robert Soutas-Little for his acceptance, knowledge, and guidance.

I would like to thank; Wolverine World Wide Inc., for funding my research, Swanson Analysis Systems Inc., for the use of ANSYS, Chrysler Corporation for their patience and support, and the entire Case Center staff for their helpfulness.

I would also like to thank Drs. Eric Goodman, Larry Seegerlind, Clark Radcliffe, and M. V. Gandhi whose comments and advice on my research were greatly appreciated.

A special thanks to my family, especially my mom and dad, whose love and support always encouraged me to do my best.

And finally, I would like to thank all my friends, especially Brian Agar, Paul Zang, and Mary Ellen Zang whose love and moral support helped me through the rough spots.

TABLE OF CONTENTS

LIST OF FIGURES

Page

Chapter 1

INTRODUCTION

Running shoe design involves the study of the structure and dynamics of the human while running using the fields of biomechanics and exercise physiology. The goal of today's running shoe design is twofold; to reduce injury, and to improve performance by cutting down on muscular and cardio-vascular fatigue. An average runner takes approximately 500 steps per mile and eighty percent of the runners are regarded as heel strikers. They land on the outer part of their heel, roll to the midfoot, and push off with the ball of their foot (metatarsal) and toes. The most severe shock of the stride occurs when the foot first hits the ground. This vertical impact force constitutes approximately 90–95% of the total shock incurred by the runner and is in magnitude between two and three times the weight of the runner [1,3]. Controlling motion and shock during impact is necessary to reduce injury and provide comfort. The body has a natural motion during running which has been termed pronation (See Figure 1.1). As the weight shifts from heel to midfoot, the leg and hip flex to distribute the impact forces causing the foot to roll inward, or pronate. Ten degrees of pronation may be acceptable although it is felt that excessive pronation may lead to knee problems.

A more detailed look at the functional anatomy of the foot during strike is described subsequently. Most runners use the entire foot during a stride, starting with heel strike, then rolling onto the midfoot, and finally propelling off the forefoot. Before the heel strike, the foot supinates (or is in the state of inversion). That is, the foot is rotated at the ankle so that the inner edge of the foot is higher

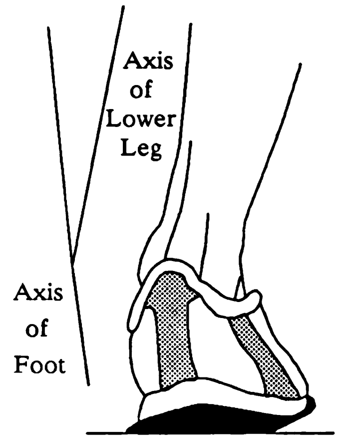


Figure 1.1 Pronation.

relative to the outer edge. At heel strike, the ankle, knee and hip all flex to cushion the shock. This flexing causes the foot to roll inward, or pronate. The foot pronates for 55–85% of the period of foot strike [1]. In the final phase of the step, the foot becomes more rigid to provide better lift. The hip and knee begin extending, the heel and midfoot leave the ground, and the toes propel the foot off the ground. The description of one step cycle explains the dynamics of the body to which the shoe is subjected.

The objective of footwear designers is to produce a shoe that is more responsive to the dynamics of the body. The goals of a running shoe design is to provide stability and cushioning. These goals, however, are often contradictory as soft shoe soles may provide good cushioning but bad stability and stiff shoe soles may provide good stability but poor cushioning. Soles absorbing too much shock return little energy to the body to enhance running. Shoe soles need the ability of the material to return or reflect impact energy, called resilience. This property helps the runner lift his foot at the end of a step. All of these factors are necessary considerations in the design of running shoe soles.

A running shoe consists of several components to accomodate cushioning and stability. The shoe consists of an "upper" of leather and synthetic fabric which constitutes the top and sides of the shoe. At the rear of the upper is the heel counter, a firm cup which cradles the back of the heel and centers the foot to keep it stable. The sole consists of a heel wedge of about 1/2" elevation to relieve strain on the Achilles tendon. Below the heel wedge is the midsole, a layer of material that provides both cushioning and stability. The bottom of the sole is an outsole consisting of a durable layer of rubber with treads or studs that provide traction and wear resistance.

The majority of the cushioning and stability is provided by the shoe sole consisting of the wedge, midsole and outsole. These layers are now described in detail.

The wedge and the midsole are usually made of EVA, a mixture of ethylene, vinyl, and acetate, each providing a particular function. Ethylene provides moldability, vinyl provides resilience and acetate provides structure and stiffness. The wedge design usually consists of a single material or possibly two materials. A common multiple material wedge consists of a heel section that contains two overlapping triangular sections. The Brooks Chariot utilizes this wedge design (See Figure 4.1).

The midsole is the most important part of the sole because it provides both cushioning and stability. There are a variety of designs used in the midsole. One example of a multidensity midsole is the Reebok Phase I Trainer which employs a soft EVA under the heel and ball of the foot for cushioning, firmer EVA at the inner or medial edge of the rear foot area to control pronation and hard EVA in the heel wedge [2]. Some midsole designs are even more complex such as Muzik

Challenger II which contains two polyurethane bags filled with mineral oil inserted into the sole under the ball and heel of the foot to emulate hydraulic shock absorbers and cushion the body by distributing and equalizing body weight [4].

The outsole is an important part of the shoe sole because it reduces wear, provides traction, and absorbs some impact. The outsole may be made of EVA, styrene butadiene rubber or more abrasion resistant rubber such as Vibram's Infinity compound or Goodyear's Indy 500 [2]. Recently shoe makers have started to manufacture outsoles with blown rubber of styrene butadiene or carborylated nitryl [2]. Blown rubber is lighter and provides better cushioning but is not as durable. Early outsole design consisted of rectangular studs similar to a waffle to absorb shock independently and to provide good traction. This design was modified because each stud acted as a miniature stilt and reduced stability. New waffle designs have lower stud profiles to retain shock absorbtion and traction but reduce instability. In some cases, center studs were eliminated creating a gap along the shoe's central axis. This gap was intended to make the foot fall toward the central axis in an effort to cut the tendency to fall off the axis and pronate.

As more information on how the body deals with shock and instability becomes available, the designer will be able to develop shoes that are more compatible with human motion.

Computers are being used in shoe design because of the increased competition between manufacturers, the improved quality due to computerized automation, and the increased sophistication of the design process. Puma and Adidas have designed shoes that couple with a personal computer to track speed, distance and compute calories burned [5]. Computers are also being used in the manufacturing process but only on a limited basis. Computerized cutters now can

do some of the grading and cutting of upper materials using laser beams or high velocity water. Moss Brown and Company are starting to use computerized stitchers but stitching complex curves over 3D surfaces is beyond current capabilities in manufacturing processes [2]. Computer aided design is also slowly appearing in the industry. Computers are used to simplify the design of lasts (the foot shape mold onto which the shoe is built) but the software has been difficult to obtain. Converse is currently developing this type of software independently [1]. Nike is using CAD/CAM in shoe design development. Fully detailed drawings are developed in 2D or 3D and these drawings can be scaled up or down automatically on the computer. The studs on the outsole are added interactively. This design information is then passed to the milling machines for cutting [6]. CAD can be taken one step further in that the performance of shoe soles can be modeled using finite element analysis. The shoe sole geometry, material properties, and load configurations are input into a finite element program and the performance is calculated. Different designs can be studied before actual prototypes are built.

This thesis discusses a design tool developed to aid in the finite element analysis of the running shoe sole and presents two specific applications. Chapter Two defines the finite element equations and the solution procedure employed by ANSYS, a finite element analysis code available from Swanson Analysis Systems Incorporated. Chapter Three describes a program written to speed the modeling procedure and to set up the ANSYS input file. Chapter Four discusses the modeling of two different shoe configurations; deflection and stress plots are presented. Chapter Five includes the conclusions and possible future work. This research was sponsored, in part, by a gift from the Brooks Shoe Company.

Chapter 2

FINITE ELEMENT METHODS

A purely analytical solution technique is not possible for a structure with an irregular shape or multiple material properties. With the development of high speed digital computers, complex structures could be analyzed using numerical techniques such as the finite element method. Finite element methods involve the discretization of a large continuous structure into a number of smaller elements. The equations for each element with its individual boundary and loading conditions can be assembled to describe the entire system. These equations were developed for this application using the finite element code ANSYS on a PRIME 750 linked to Tektronix graphics terminals. This chapter presents the equations used to study the running shoe sole and the solution techniques used by ANSYS.

A static analysis was used to solve for displacements, stresses, and forces in structures under the action of applied loads. The equilibrium equation for the static analysis can be written in the matrix form

$$[K] \{ u \} = \{ F \} \tag{2.1}$$

where $[K]$ = total stiffness matrix $\sum_{m=1}^{N} [ke]$

$\{ u \}$ = nodal displacement vector

N = number of elements

$[ke]$ = element stiffness matrix

The resulting set of simultaneous linear equations was solved by the wave front solution technique which will be described below.

The ANSYS program uses a wave front solution technique for the system of simultaneous linear equations obtained from the finite elements. The number of equations active at any point of the solution phase is the wave front size and the ordering of the elements is crucial for minimizing the wave front size. This minimization is important for reasons of efficiency because the computer time is proportional to the square of the mean wave front size. The wave front size is also dependent upon the sequence in which the elements are arranged. The node numbers of all the elements are scanned to determine which element is the last to use the node number. As the entire system of equations is assembled, the equations for a node occurring last are solved in terms of the remaining unknowns and eliminated from the matrix in core. The eliminated equations form the stiffness matrix. Equations which contain a new node are added to the assembled matrix. As nodes come and go during the solution process, the wave front size expands and contracts. The following describes the wave front solution procedure in equation form.

The active equations are:

$$\sum_{j=1}^{L} K_{kj} u_j = F_k \qquad (2.2)$$

where K_{kj} = stiffness term kj

u_j = nodal displacement j

F_k = nodal force k

k = row number

j = column number

L = number of equations

To eliminate an equation $i = k$, begin by normalizing

$$\sum_{j=1}^{L} K_{ij} / K_{ii}\, u_j = F_i / K_{ii} \tag{2.3}$$

This can be rewritten as

$$\sum_{j=1}^{L} K^*_{ij}\, u_j = F^*_i \tag{2.4}$$

where

$$K^*_{ij} = K_{ij} / K_{ii} \tag{2.5}$$

$$F^*_i = F_i / K_{ii} \tag{2.6}$$

This equation is written to a file for later back substitution. Note, K_{ii} is never zero if the model is properly developed. The remaining equations are modified as follows

$$K^*_{kj} = K_{kj} - K_{ki}\, K^*_{ij} \tag{2.7}$$

$$F^*_k = F_k - K_{ki}\, F^*_i \tag{2.8}$$

where $k \neq i$, so that

$$\sum_{j=1}^{L-1} K^*_{kj} \; u_j = F \tag{2.9}$$

where k varies from 1 to n – 1. Having eliminated row i, the other rows are eliminated by repeating the process. An example of this solution technique is demonstrated in Appendix A.

Chapter 3

PROGRAM DESCRIPTION

The purpose of this chapter is to describe in detail an interactive program developed as a design tool for the finite element analysis of running shoe soles. The program structure is discussed and the advantage of using this program as a specific preprocessor for ANSYS is delineated. The program was used to develop the models of two different shoe sole configurations and to study their characteristics.

A preprocessing program was written to overcome some of the labor intensive characteristics inherent in the use of the finite element analysis. The program uses a basic physical geometry and readily allows modeling of other geometries by manipulation of the material properties, thus eliminating the time consuming task of nodal coordinate definition. The program allows for the easy definition and change of material properties, thus reducing the labor intensive task of element property definition. The program discussed here interactively sets up an ANSYS command file ready for processing.

The basic model consists of six layers of elements. There are two layers of elements in the outsole, one layer of elements in the midsole, and three layers of elements in the wedge. There are six elements from medial to lateral, and 14 elements from posterior to anterior in each column of the bottom three layers. The wedge also has six elements per row but only extends eight elements per column (See Figure 3.1). Due to the size limitations inherent in the educational version of

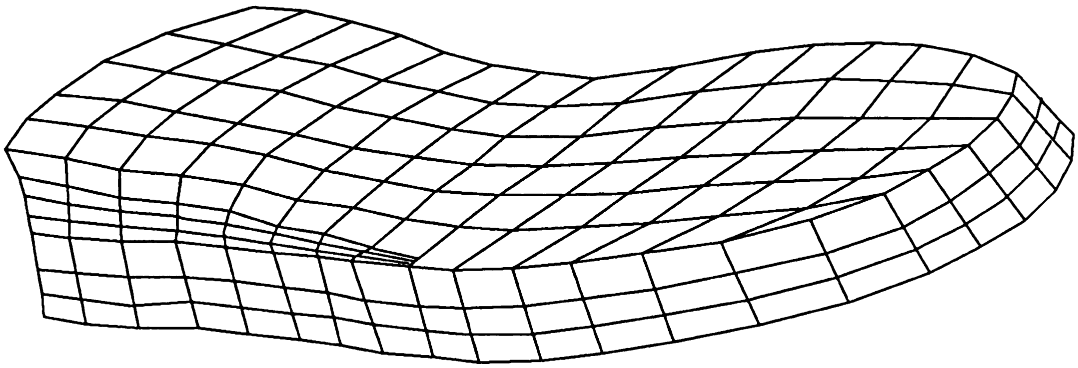


Figure 3.1 Element Model of a Shoe Sole.

ANSYS, the model is limited to six layers. The two layers of elements in the outsole section allow for carving out portions in this area which is common in outsole design. The three layers of elements in the wedge allow staircasing a change in materials. The coordinate geometry was measured from a size nine men's Brooks Chariot running shoe. The basic outside dimensions of the Brooks line of running shoes is very similar. This similarity allows modeling of a wide range of Brooks running shoes using the preprocessor.

The type of element used is a 3D isoparametric solid. The element is defined by eight nodes each having three degrees of freedom: translation in the x, y, and z directions.

The program sets up an ANSYS command file ready for analysis and is both prompt and menu driven. The program has four major sections; the initialization, the layer specifications, the element definition and the load specifications.

The first portion of the program consists of the initialization. The program will prompt for the name of the command file in which to store the ANSYS input file.

ENTER THE NAME OF THE ANSYS COMMAND FILE TO STORE THIS INFORMATION. THE NAME SHOULD START WITH A "C_" TO INDICATE IT IS A COMMAND FILE.

The program then prompts for a title which is seen on all ANSYS outputs, such as plots or tables.

ENTER THE TITLE OF THIS RUN.

The program then writes some necessary ANSYS commands. Finally, the nodal coordinate information is written into the command file.

The next portion of the program consists of the prompts and menus for the specifications of each layer. The input prompts are virtually the same for each layer so only the specifications for the bottom of the outsole are shown as an example. Also, the menus for specifying the Poisson's ratio and elastic modulus are virtually the same, thus, only the menus for the elastic modulus are shown here. The program indicates to the user which layer is currently being specified. For example,

YOU ARE SPECIFYING THE PROPERTIES FOR THE BOTTOM OF THE OUTSOLE.

Then, the first menu appears.

DO YOU WANT THE ELASTIC MODULUS
1. **THE SAME ACROSS THE ENTIRE LAYER?**
2. **SPECIFIED AT EVERY ELEMENT?**
3. **VALUES READ FROM AN EXTERNAL FILE?**

If the user picks menu choice one, the program will prompt with

ENTER THE ELASTIC MODULUS

This elastic modulus will be written for every element in that layer. If the user enters menu choice two, the program will prompt with

ENTER THE ELASTIC MODULUS FOR ELEMENT 1

This process is repeated until the property is specified for the entire layer. The element specification begins at the lower left hand corner, proceeds to the right and then continues to the toe (See Figure 3.2). Menu choice three responds with a prompt to enter the name of the file in which the material properties are stored.

ENTER THE NAME OF THE FILE WITH THE ELASTIC MODULUS PROPERTIES

After the input method has been decided, the program inquires of the user whether the property information is to be permanently stored.

DO YOU WANT THIS ELASTIC MODULUS SPECIFICATION STORED FOR FUTURE USE?

If the user responds 'yes' the program will prompt for the name of the file.

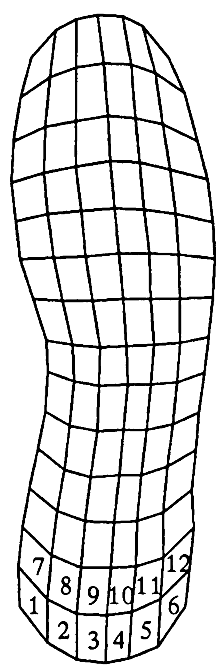


Figure 3.2 Element Ordering for Layer Specification.

ENTER THE NAME OF THE FILE TO STORE THIS LAYER'S ELASTIC MODULUS INFORMATION.

The material properties can be stored regardless of the input method. The program then writes the element definition and respective material property specifications into the command file. The elements are automatically reordered to reduce the wave front, however, the material specifications remain intact.

The program also allows for easy input of the load configuration. The load related menu is

DO YOU WANT TO
1. **DEFINE LOADS.**
2. **READ THE LOADS FROM AN EXTERNAL FILE.**
3. **EXIT.**

Menu choice one will respond with the question of whether the user wants the loads stored.

DO YOU WANT THIS LOAD DEFINITION STORED FOR FUTURE USE?

If the user responds 'yes' the program prompts for the name of the file.

ENTER THE NAME OF THE FILE TO STORE THIS LOAD INFORMATION

The program will then prompt for the total number of loads.

ENTER THE TOTAL NUMBER OF LOADS TO BE DEFINED

The next prompt is for the type of load, either deflection or force.

ENTER THE TYPE OF LOAD, D=DEFLECTION, F=FORCE.

Then, the label for the load is requested.

ENTER THE LABEL FOR THE LOAD. IF THE LOAD IS
- **A) DEFLECTION, THE OPTIONS ARE UX, UY, UZ, OR ALL.**
- **B) FORCE, THE OPTIONS ARE FX, FY, FZ, OR ALL.**

Once the type of load and its label are known, it is necessary to define which nodes these loads are applied to and the numeric value of the load.

ENTER THE STARTING NODE, ENDING NODE, NODE INCREMENT, AND THE VALUE OF THE LOAD.

This prompting sequence continues until an EXIT is entered. Menu choice two will ask for the file that holds the load configuration information.

ENTER THE NAME OF THE FILE WITH THE LOAD INFORMATION

The program then reads the information from the external file and writes it into the ANSYS input file. Menu choice three exits the user from the program completing the writing of the ANSYS input file.

This ANSYS preprocessor allows for a quick and easy initial study of model configurations. This analysis provides some quantitative information, but the main advantage of the analysis is the qualitative information it provides. It allows the designer to create a library of layer configurations to mix and match to create a wide range of shoe designs. The load definition is easy to define or change. Basically, this program gives the designer an analysis tool to lead to a more exact model definition for a complete quantitative analysis. The preprocessor program and its subroutines are listed in Appendix D.

Chapter 4

RESULTS

This chapter discusses two shoes that were modeled using the ANSYS preprocessor and solved using the ANSYS finite element program. The models were subjected to four different load conditions to simulate the cycle of a step. Stress and displacement plots are presented.

The first shoe modeled was the Brooks Chariot. The main area of design interest in the Chariot is the two material wedge shown below (See Figure 4.1).

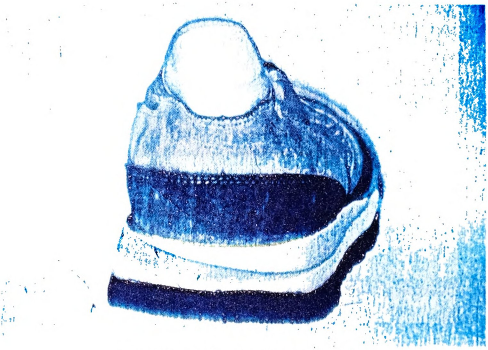


Figure 4.1 Brooks Chariot Wedge.

The wedge was modeled with a linear change in the elastic modulus in all the elements defining the wedge. The midsole was modeled with the elastic modulus the same across the entire layer of elements. Both layers in the outsole have the same elastic modulus. Although there is a tread pattern on the outsole, it is modeled as a solid because the majority of the outsole is in contact with the ground. The Poisson's ratio was assumed to be the same in all the materials in the shoe sole.

The second shoe modeled was the Brooks Trilogy. The wedge in the Trilogy is the same design used in the Chariot, but utilizes different materials. The model uses elastic modulus values obtained through testing rather than using a linear change in the property. These values were used in all three layers of elements representing the wedge. The midsole also has an area of design interest. The area under the metatarsal employs a material which is less stiff. The elements in this area have a different elastic modulus. The outsole has a number of areas of design interest. There are two wear plugs of a very stiff material. These plugs were modeled by defining a high elastic modulus value for the elements in that area. The Trilogy outsole also has a section that is carved out along the central axis of the shoe. This area was modeled by defining a very small elastic modulus in this area. The outsole also employs a number of different materials. Again, the Poisson's ratio was assumed to be the same for all materials. Figure 4.2 shows the actual outsole, and Figure 4.3 shows how this outsole was modeled. The material properties for each shoe are contained in Appendix B.

These two shoe models were subjected to four load configurations to simulate a cycle of one step; heel strike, full heel, full foot, and toe off. The target loading was 450 lbs. total force. Typical studies on shoes use a man with size nine feet weighing approximately 150 lbs. As stated previously, the impact force is about three times the weight of the runner resulting in a total force of 450 lbs. The loads were deflection loads applied to the top layer of nodes. The forces were then

Figure 4.2 Brooks Trilogy Outsole.

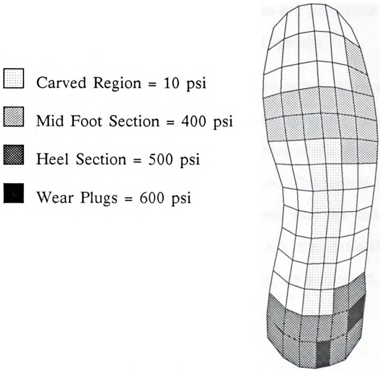


Figure 4.3 Brooks Trilogy Outsole Model.

summed to obtain the target force. The relationship between total deflection and total force is almost linear, so only one iteration was required to adjust the deflections so that a 450 lbs. total force loading was obtained. The nodes on the bottom were constrained in all directions to give realistic ground reactions.

The heel strike load is a linearly decreasing deflection loading from medial to lateral imposed on the top layer of nodes. The full heel load is a constant deflection load. The fore foot load employs a linearly decreasing deflection load from lateral to medial similar to the heel strike except imposed on different nodes. And finally, the toe off load is a linearly increasing deflection load from medial to lateral. Figure 4.4 shows the nodes which were subjected to the loading. The area lightly shaded represents the nodes on the top layer subjected to deflection. The area heavily shaded represents the nodes on the bottom layer which were constrained in all directions.

The results for the full heel load configuration are presented in this chapter. The results for the other loads are presented in Appendix C.

Stress plots are the first set of results to be presented. Figure 4.5 and Figure 4.6 show the stresses on the bottom of the shoe resulting from the full heel loading. As the figures show, the effect of the carved out region in the Trilogy outsole provides a more favorable stress distribution. The legend at the side of the plot indicates the maximum and minimum stresses and the increment between constant stress lines. The next set of results are the deflection plots. The deflections for each nodal row in the area of interest are presented. A nodal row is defined to be the nodal coordinate rows as the model goes from the heel to the toe. The resultant forces are depicted on the top layer of nodes. The effect of the carved out elements is apparent in both the deflection and resultant forces in nodal rows four and five (See Figures 4.7 through Figure 4.12).

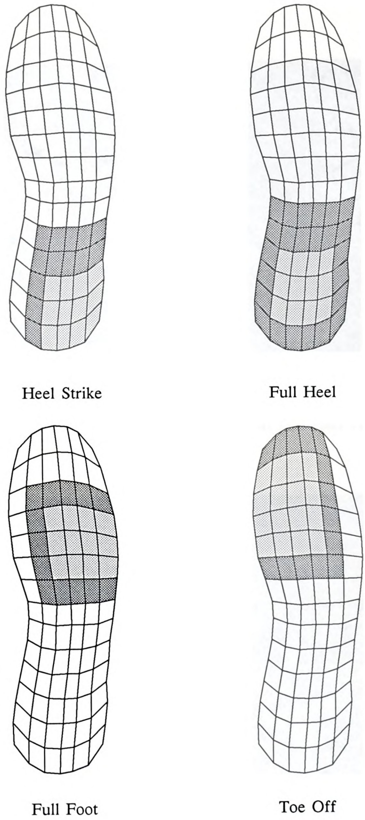


Figure 4.4 Load Configurations

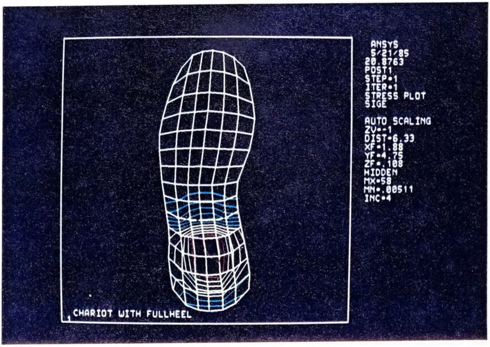


Figure 4.5 Chariot Stress Plot with Full Heel Load.

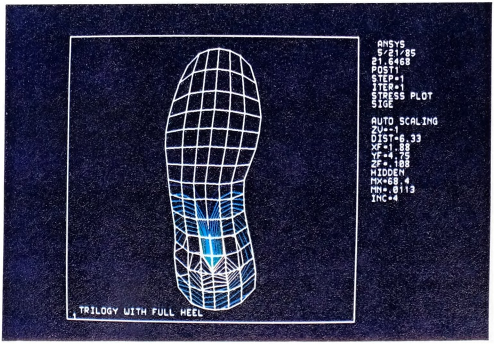


Figure 4.6 Trilogy Stress Plot with Full Heel Load.

Full Heel

CHARIOT

TRILOGY

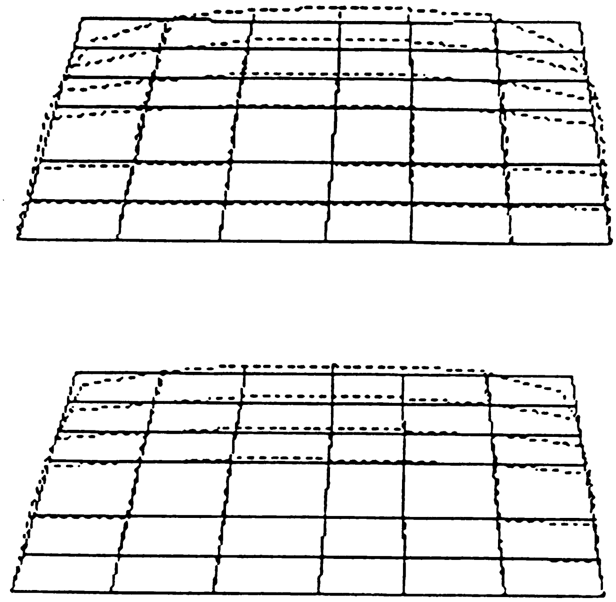


Figure 4.7 Nodal Row 1.

Full Heel

CHARIOT

TRILOGY

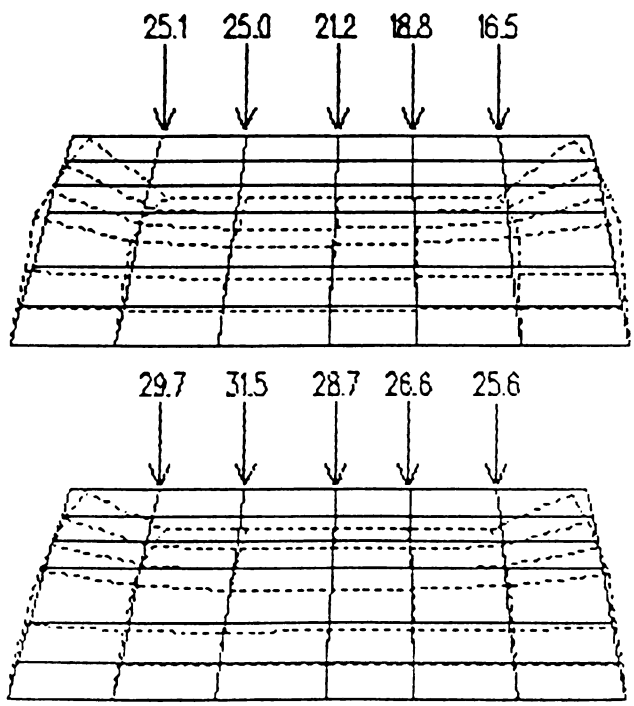


Figure 4.8 Nodal Row 2.

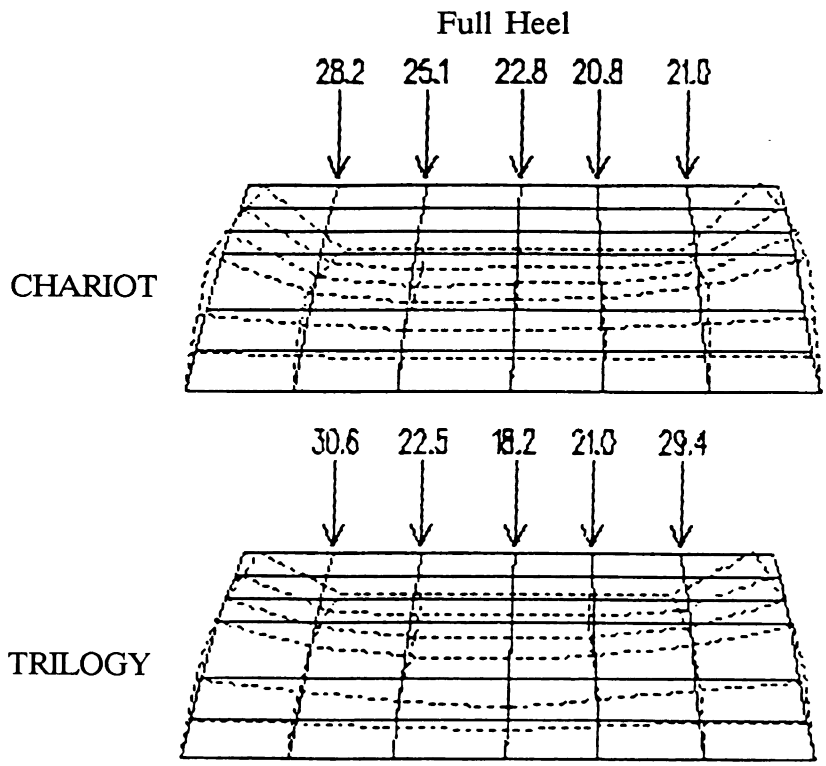


Figure 4.9 Nodal Row 3.

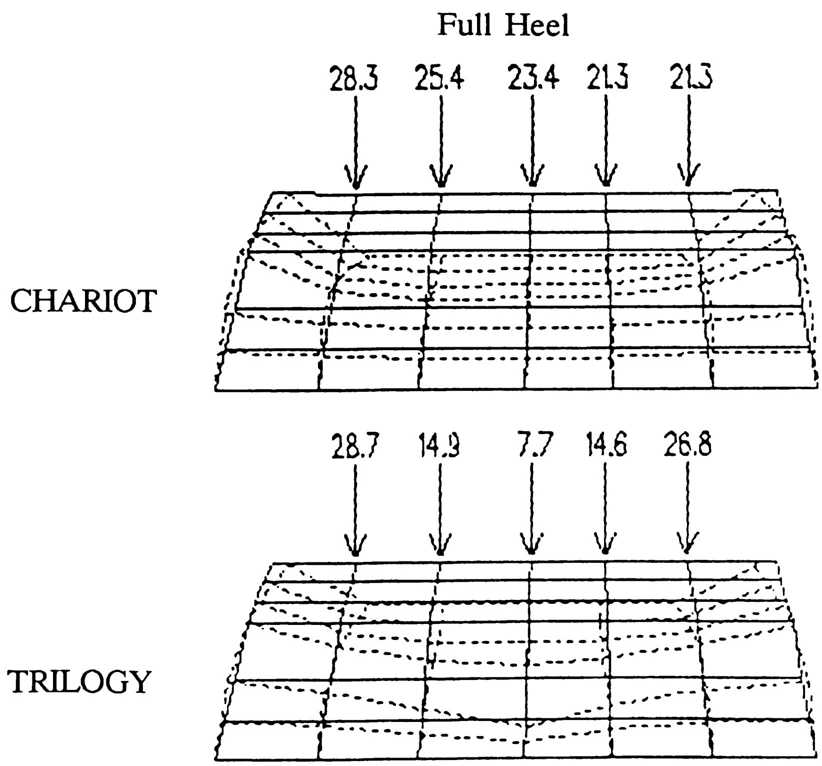


Figure 4.10 Nodal Row 4.

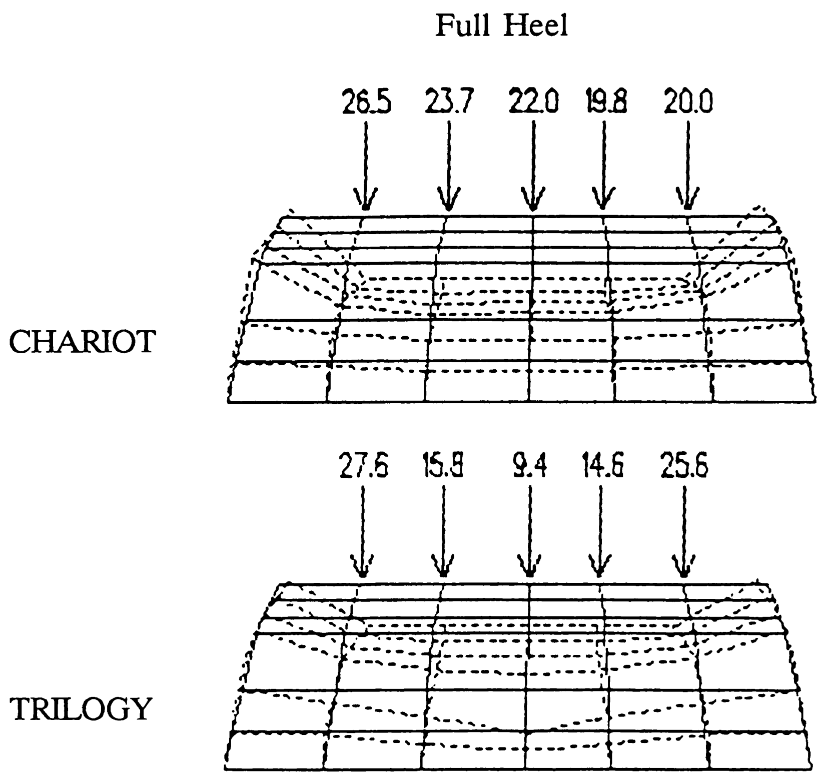


Figure 4.11 Nodal Row 5.

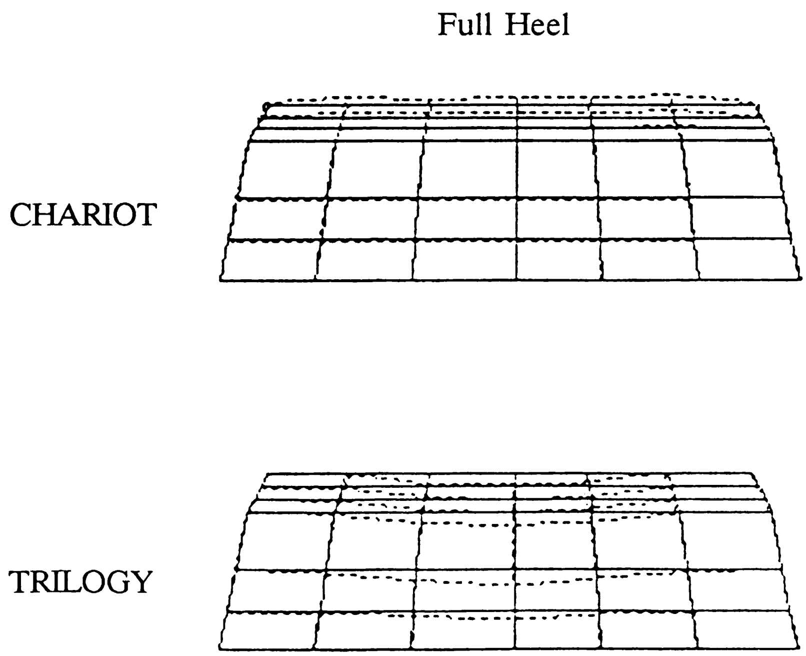


Figure 4.12 Nodal Row 6.

Chapter 5

CONCLUSIONS

This thesis has shown the advantages of using this preprocessor as a tool for the finite element analysis of running shoe soles. The program uses a basic physical geometry and readily allows modeling of other geometries by manipulation of the material properties. The program is very flexible and easily accepts changes in loads or material properties. A library of layer designs can be created to mix and match to produce a number of design prototypes. Although some quantitative information is obtained through this analysis procedure, the main benefit is the qualitative information it can provide. Design trends can be identified by creating and analyzing a number of models with slight changes in material properties, configurations, or layer definition. Once the design trends have been identified, a precise model of the shoe can be created manually to obtain quantitative information.

Future work could include using this same approach with the industrial version of ANSYS which permits a greater number of element layers to be defined, allowing for the development of more precise models. Also work could be done to make the program more user friendly. These improvements could include improved error checking, "oops" features (to allow the designer to back up and redefine a material property), or graphical displays. Graphics could be implemented to indicate to the designer which element is under present consideration.

There are a number of areas not addressed by this thesis. The program does not account for any nonlinearity in the material properties. Also, the effect of the interaction between the shoe upper and shoe sole has not been examined.

This thesis has presented a tool to aid the designer in the creation of a shoe sole to reduce injury and improve performance. The preprocessor minimizes time constraints placed on the designer allowing him to be more creative in the development of shoe sole prototypes

REFERENCES

REFERENCES

1. Baer, Tony. "Designing for the Long Run." *Mechanical Engineering"* September 1984, pp. 66–75.
2. Tucker, Jonathan B. "Running Shoe Technology Picks Up the Pace." *High Technology*, March 1985, pp. 28–34.
3. Cavanagh, Peter C. "The Shoe-Ground Interface in Running." *American Academy of Orthopeadic Surgeons Symposium on the Foot and Leg in Running Sports*. The C. V. Mosby Company, 1982, pp. 30–44.
4. "Athletic, Leisure Shoes Aid Fancy Footwork." *Design News* April 4, 1984, pp. 42–43.
5. Ward, Sam. "New Computerized Shoe Gives You a Run for Your Money." *USA Today*, July 8, 1985, p. 8.
6. "CAD/CAM Helps Nike Put Best Foot Forward", *IEEE Computer Graphics and Applications* February 1985, pp. 76–77.
7. Kohnke, Peter C. "ANSYS Engineering Analysis System Theoretical Manual", Swanson Analysis Systems Incorporated, 1983.
8. Hull, Andrew J. "Modelling Cylindrical Discontinuities in Running Shoe Soles Using a Condensed Finite Element Formulation." M. S. Thesis, Michigan State University, 1985.

APPENDICES

APPENDIX A

WAVE FRONT SOLUTION EXAMPLE

ANSYS uses a wave front solution procedure. This appendix shows the solution of a three by three system of equations utilizing this solution technique.

The equations are

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix} \tag{A.1}$$

The first equation is normalized so that (A.2)

$$u_1 + (k_{12} / k_{11})\, u_2 + (k_{13} / k_{11})\, u_3 = f_1 / k_{11} \tag{A.3}$$

The normalized components are then

$$k^*_{12} = k_{12} / k_{11} \tag{A.4}$$

$$k^*_{13} = k_{13} / k_{11} \tag{A.5}$$

$$f^* = f_1 / k_{11} \tag{A.6}$$

The second iteration produces the following components

$$k^*_{22} = [k_{22} - k_{21}(k_{12} / k_{11})] \tag{A.7}$$

$$k^*_{23} = [k_{23} - k_{21}(k_{13} / k_{11})] \tag{A.8}$$

$$k^*_{32} = [k_{32} - k_{31}(k_{12} / k_{11})] \qquad \text{(A.9}$$

$$k^*_{33} = [k_{33} - k_{31}(k_{13} / k_{11})] \qquad \text{(A.10}$$

$$f^*_2 = [f_2 - k_{21}(f_1 / k_{11})] \qquad \text{(A.11}$$

$$f^*_3 = [f_3 - k_{31}(f_1 / k_{11})] \qquad \text{(A.12}$$

which are assembled as

$$\begin{bmatrix} k^*_{22} & k^*_{23} \\ k^*_{32} & k^*_{33} \end{bmatrix} \begin{Bmatrix} u_2 \\ u_3 \end{Bmatrix} = \begin{Bmatrix} f^*_2 \\ f^*_3 \end{Bmatrix} \qquad \text{(A.13}$$

The first equation becomes

$$u_2 + \Big[[k_{23} - k_{21}(k_{13} / k_{11})] \;/\; [k_{22} - k_{21}(k_{12} / k_{11})]\Big]\, u_3$$

$$= [f_2 - k_{21}(f_1 / k_{11})] \;/\; [k_{22} - k_{21}(k_{12} / k_{11})] \qquad \text{(A.14}$$

while the second equation becomes

$$k^{**}_{33}\, u_3 = f^{**}_3 \qquad \text{(A.15}$$

u_3 is solved for as

$$u_3 = f^{**}_3 / k^{**}_3 \qquad \text{(A.16}$$

where

$$k^{**}_3 = \Big[[k_{22} - k_{21}(k_{12} / k_{11})]\; [k_{33} - k_{31}(k_{13} / k_{11})]\Big]$$
$$- \Big[[k_{32} - k_{31}(k_{12} / k_{11})]\; [k_{23} - k_{21}(k_{13} / k_{11})]\Big] \qquad \text{(A.17}$$

$$f^{**}_3 = \Big[[k_{22} - k_{21}(k_{12} / k_{11})]\; [f_3 - k_{13}(f_1 / k_{11})]\Big]$$
$$- \Big[[k_{32} - k_{31}(k_{12} / k_{11})]\; [f_2 - k_{21}(f_1 / k_{11})]\Big] \qquad \text{(A.18}$$

Solving this system of equations using Cramer's Rule would yield the same result. The advantage of using a wave front solution technique is the optimization of computer time during the equation solution phase [8].

APPENDIX B

MATERIAL PROPERTIES FOR THE CHARIOT AND TRILOGY MODELS

This appendix contains the elastic modulus values in psi specified for each element of the Chariot model and the Trilogy model. Poisson's ratio was 0.25 for all of the elements in both models. Note, the element numbers have been reordered to reduce the size of the wave front. The elements start at the lower left corner, proceed vertically, to the right, and finally toward the toe. The element properties as specified remains intact, just the element numbers are changed.

ELASTIC MODULUS VALUES

ELEMENT	CHARIOT	TRILOGY
1.	260.0	500.0
2.	260.0	300.0
3.	130.0	350.0
4.	260.0	450.0
5.	260.0	450.0
6.	260.0	450.0
7.	260.0	500.0
8.	260.0	300.0
9.	130.0	350.0
10.	234.0	450.0
11.	234.0	450.0
12.	234.0	450.0
13.	260.0	500.0
14.	260.0	300.0
15.	130.0	350.0
16.	208.0	500.0
17.	208.0	500.0
18.	208.0	500.0
19.	260.0	600.0
20.	260.0	600.0
21.	130.0	350.0
22.	182.0	500.0
23.	182.0	500.0
24.	182.0	500.0
25.	260.0	500.0
26.	260.0	300.0
27.	130.0	350.0
28.	156.0	400.0
29.	156.0	400.0
30.	156.0	400.0
31.	260.0	500.0
32.	260.0	300.0
33.	130.0	350.0
34.	130.0	400.0
35.	130.0	400.0
36.	130.0	400.0
37.	260.0	500.0
38.	260.0	300.0
39.	130.0	350.0

ELEMENT	CHARIOT	TRILOGY
40.	260.0	450.0
41.	260.0	450.0
42.	260.0	450.0
43.	260.0	500.0
44.	260.0	300.0
45.	130.0	350.0
46.	234.0	450.0
47.	234.0	450.0
48.	234.0	450.0
49.	260.0	500.0
50.	260.0	300.0
51.	130.0	350.0
52.	208.0	500.0
53.	208.0	500.0
54.	208.0	500.0
55.	260.0	500.0
56.	260.0	300.0
57.	130.0	350.0
58.	182.0	500.0
59.	182.0	500.0
60.	182.0	500.0
61.	260.0	500.0
62.	260.0	300.0
63.	130.0	350.0
64.	156.0	400.0
65.	156.0	400.0
66.	156.0	400.0
67.	260.0	600.0
68.	260.0	600.0
69.	130.0	350.0
70.	130.0	400.0
71.	130.0	400.0
72.	130.0	400.0
73.	260.0	300.0
74.	260.0	300.0
75.	130.0	350.0
76.	260.0	450.0
77.	260.0	450.0
78.	260.0	450.0
79.	260.0	300.0

ELEMENT	CHARIOT	TRILOGY
80.	260.0	300.0
81.	130.0	350.0
82.	234.0	450.0
83.	234.0	450.0
84.	234.0	450.0
85.	260.0	10.0
86.	260.0	10.0
87.	130.0	350.0
88.	208.0	500.0
89.	208.0	500.0
90.	208.0	500.0
91.	260.0	10.0
92.	260.0	10.0
93.	130.0	350.0
94.	182.0	500.0
95.	182.0	500.0
96.	182.0	500.0
97.	260.0	500.0
98.	260.0	300.0
99.	130.0	350.0
100.	156.0	400.0
101.	156.0	400.0
102.	156.0	400.0
103.	260.0	500.0
104.	260.0	300.0
105.	130.0	350.0
106.	130.0	400.0
107.	130.0	400.0
108.	130.0	400.0
109.	260.0	300.0
110.	260.0	300.0
111.	130.0	350.0
112.	260.0	450.0
113.	260.0	450.0
114.	260.0	450.0
115.	260.0	300.0
116.	260.0	300.0
117.	130.0	350.0
118.	234.0	450.0
119.	234.0	450.0

ELEMENT	CHARIOT	TRILOGY
120.	234.0	450.0
121.	260.0	10.0
122.	260.0	10.0
123.	130.0	350.0
124.	208.0	500.0
125.	208.0	500.0
126.	208.0	500.0
127.	260.0	10.0
128.	260.0	10.0
129.	130.0	350.0
130.	182.0	500.0
131.	182.0	500.0
132.	182.0	500.0
133.	260.0	300.0
134.	260.0	300.0
135.	130.0	350.0
136.	156.0	400.0
137.	156.0	400.0
138.	156.0	400.0
139.	260.0	300.0
140.	260.0	300.0
141.	130.0	350.0
142.	130.0	400.0
143.	130.0	400.0
144.	130.0	400.0
145.	260.0	300.0
146.	260.0	300.0
147.	130.0	350.0
148.	260.0	450.0
149.	260.0	450.0
150.	260.0	450.0
151.	260.0	300.0
152.	260.0	300.0
153.	130.0	350.0
154.	234.0	450.0
155.	234.0	450.0
156.	234.0	450.0
157.	260.0	10.0
158.	260.0	10.0
159.	130.0	350.0

ELEMENT	CHARIOT	TRILOGY
160.	208.0	500.0
161.	208.0	500.0
162.	208.0	500.0
163.	260.0	10.0
164.	260.0	10.0
165.	130.0	350.0
166.	182.0	500.0
167.	182.0	500.0
168.	182.0	500.0
169.	260.0	300.0
170.	260.0	300.0
171.	130.0	350.0
172.	156.0	400.0
173.	156.0	400.0
174.	156.0	400.0
175.	260.0	300.0
176.	260.0	300.0
177.	130.0	350.0
178.	130.0	400.0
179.	130.0	400.0
180.	130.0	400.0
181.	260.0	300.0
182.	260.0	300.0
183.	130.0	350.0
184.	260.0	400.0
185.	260.0	400.0
186.	260.0	400.0
187.	260.0	300.0
188.	260.0	300.0
189.	130.0	350.0
190.	234.0	400.0
191.	234.0	400.0
192.	234.0	400.0
193.	260.0	10.0
194.	260.0	10.0
195.	130.0	350.0
196.	208.0	400.0
197.	208.0	400.0
198.	208.0	400.0
199.	260.0	10.0

ELEMENT	CHARIOT	TRILOGY
200.	260.0	10.0
201.	130.0	350.0
202.	182.0	400.0
203.	182.0	400.0
204.	182.0	400.0
205.	260.0	300.0
206.	260.0	300.0
207.	130.0	350.0
208.	156.0	350.0
209.	156.0	350.0
210.	156.0	350.0
211.	260.0	300.0
212.	260.0	300.0
213.	130.0	350.0
214.	130.0	350.0
215.	130.0	350.0
216.	130.0	350.0
217.	260.0	300.0
218.	260.0	300.0
219.	130.0	350.0
220.	260.0	400.0
221.	260.0	400.0
222.	260.0	400.0
223.	260.0	300.0
224.	260.0	300.0
225.	130.0	350.0
226.	234.0	400.0
227.	234.0	400.0
228.	234.0	400.0
229.	260.0	10.0
230.	260.0	10.0
231.	130.0	350.0
232.	208.0	400.0
233.	208.0	400.0
234.	208.0	400.0
235.	260.0	10.0
236.	260.0	10.0
237.	130.0	350.0
238.	182.0	400.0
239.	182.0	400.0

ELEMENT	CHARIOT	TRILOGY
240.	182.0	400.0
241.	260.0	300.0
242.	260.0	300.0
243.	130.0	350.0
244.	156.0	350.0
245.	156.0	350.0
246.	156.0	350.0
247.	260.0	300.0
248.	260.0	300.0
249.	130.0	350.0
250.	130.0	350.0
251.	130.0	350.0
252.	130.0	350.0
253.	260.0	300.0
254.	260.0	300.0
255.	130.0	350.0
256.	260.0	400.0
257.	260.0	400.0
258.	260.0	400.0
259.	260.0	300.0
260.	260.0	300.0
261.	130.0	350.0
262.	234.0	400.0
263.	234.0	400.0
264.	234.0	400.0
265.	260.0	10.0
266.	260.0	10.0
267.	130.0	350.0
268.	208.0	400.0
269.	208.0	400.0
270.	208.0	400.0
271.	260.0	10.0
272.	260.0	10.0
273.	130.0	350.0
274.	182.0	400.0
275.	182.0	400.0
276.	182.0	400.0
277.	260.0	300.0
278.	260.0	300.0
279.	130.0	350.0

ELEMENT	CHARIOT	TRILOGY
280.	156.0	350.0
281.	156.0	350.0
282.	156.0	350.0
283.	260.0	300.0
284.	260.0	300.0
285.	130.0	350.0
286.	130.0	350.0
287.	130.0	350.0
288.	130.0	350.0
289.	260.0	300.0
290.	260.0	300.0
291.	130.0	350.0
292.	260.0	300.0
293.	260.0	300.0
294.	130.0	350.0
295.	260.0	10.0
296.	260.0	10.0
297.	130.0	350.0
298.	260.0	10.0
299.	260.0	10.0
300.	130.0	350.0
301.	260.0	400.0
302.	260.0	400.0
303.	130.0	350.0
304.	260.0	400.0
305.	260.0	400.0
306.	130.0	350.0
307.	260.0	400.0
308.	260.0	400.0
309.	130.0	300.0
310.	260.0	400.0
311.	260.0	400.0
312.	130.0	300.0
313.	260.0	400.0
314.	260.0	400.0
315.	130.0	300.0
316.	260.0	400.0
317.	260.0	400.0
318.	130.0	300.0
319.	260.0	400.0

ELEMENT	CHARIOT	TRILOGY
320.	260.0	400.0
321.	130.0	300.0
322.	260.0	400.0
323.	260.0	400.0
324.	130.0	300.0
325.	260.0	400.0
326.	260.0	400.0
327.	130.0	250.0
328.	260.0	400.0
329.	260.0	400.0
330.	130.0	250.0
331.	260.0	400.0
332.	260.0	400.0
333.	130.0	250.0
334.	260.0	400.0
335.	260.0	400.0
336.	130.0	250.0
337.	260.0	400.0
338.	260.0	400.0
339.	130.0	250.0
340.	260.0	400.0
341.	260.0	400.0
342.	130.0	250.0
343.	260.0	300.0
344.	260.0	300.0
345.	130.0	200.0
346.	260.0	300.0
347.	260.0	300.0
348.	130.0	200.0
349.	260.0	300.0
350.	260.0	300.0
351.	130.0	200.0
352.	260.0	300.0
353.	260.0	300.0
354.	130.0	200.0
355.	260.0	300.0
356.	260.0	300.0
357.	130.0	200.0
358.	260.0	400.0
359.	260.0	400.0

ELEMENT	CHARIOT	TRILOGY
360.	130.0	200.0
361.	260.0	300.0
362.	260.0	300.0
363.	130.0	150.0
364.	260.0	300.0
365.	260.0	300.0
366.	130.0	150.0
367.	260.0	300.0
368.	260.0	300.0
369.	130.0	150.0
370.	260.0	300.0
371.	260.0	300.0
372.	130.0	150.0
373.	260.0	300.0
374.	260.0	300.0
375.	130.0	150.0
376.	260.0	400.0
377.	260.0	400.0
378.	130.0	150.0
379.	260.0	300.0
380.	260.0	300.0
381.	130.0	150.0
382.	260.0	300.0
383.	260.0	300.0
384.	130.0	150.0
385.	260.0	300.0
386.	260.0	300.0
387.	130.0	150.0
388.	260.0	300.0
389.	260.0	300.0
390.	130.0	150.0
391.	260.0	300.0
392.	260.0	300.0
393.	130.0	150.0
394.	260.0	300.0
395.	260.0	300.0
396.	130.0	150.0

APPENDIX C

STRESS AND DISPLACEMENT PLOTS

This appendix contains the stress and displacement plots for three load configurations; heel strike, full foot, and toe off. The stress plots shown are the stresses occurring on the bottom of the shoe sole. The effect of the carved out elements in the outsole of the Trilogy is apparent in all of the Trilogy stress plots. This effect is also apparent in the nodal deflection plots and resulting forces. The legend at the side of the stress plots indicates the maximum and minimum stresses and the increment between constant stress lines.

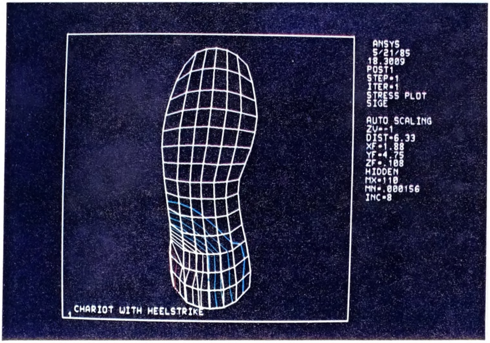


Figure C.1 Chariot Stress Plot with Heel Strike Load.

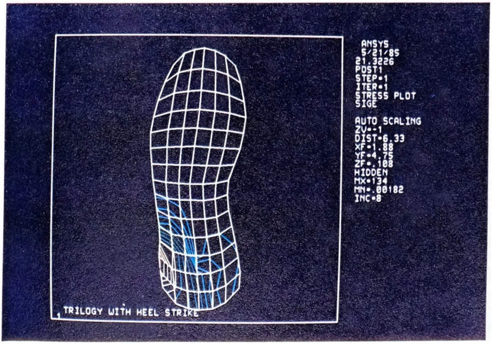


Figure C.2 Trilogy Stress Plot with Heel Strike Load.

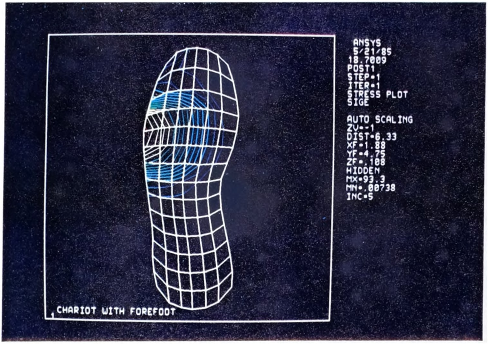


Figure C.3 Chariot Stress Plot with Full Foot Load.

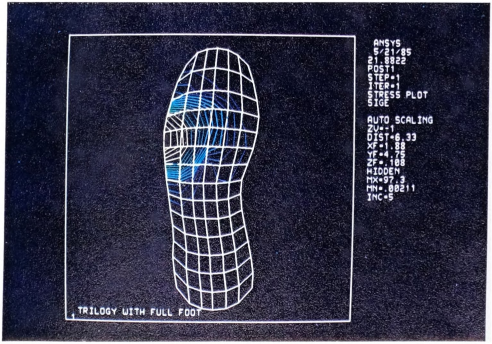


Figure C.4 Trilogy Stress Plot with Full Foot Load.

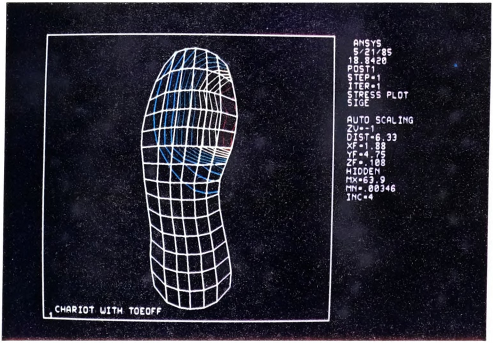


Figure C.5 Chariot Stress Plot with Toe Off Load.

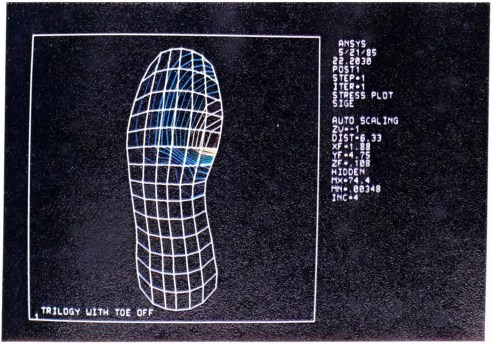


Figure C.6 Trilogy Stress Plot with Toe Off Load.

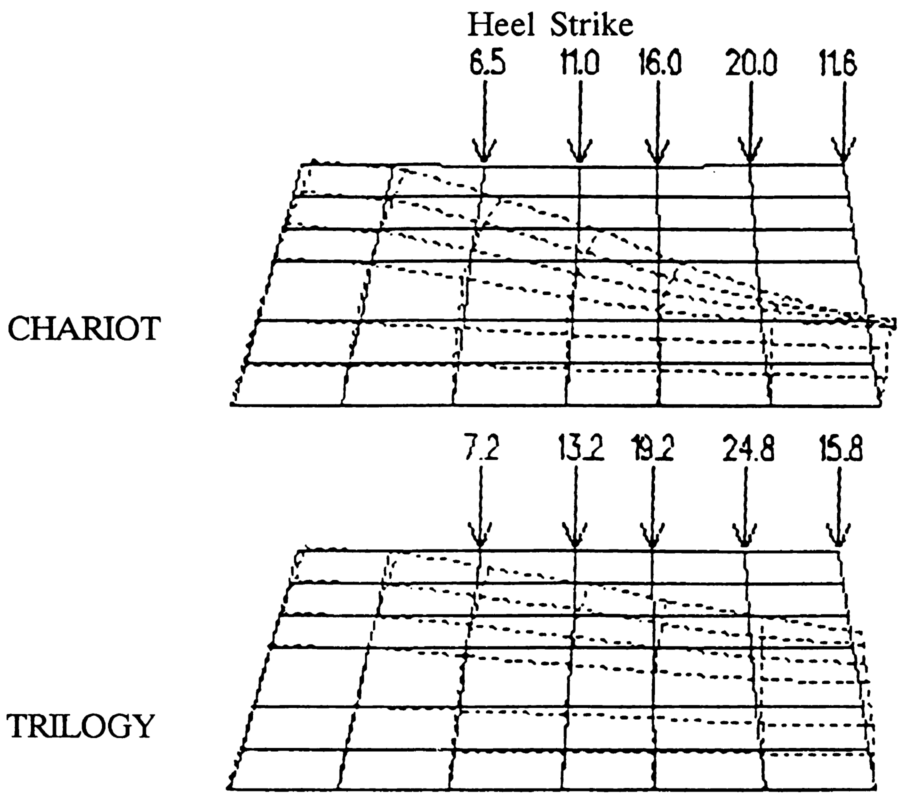


Figure C.7 Nodal Row 1.

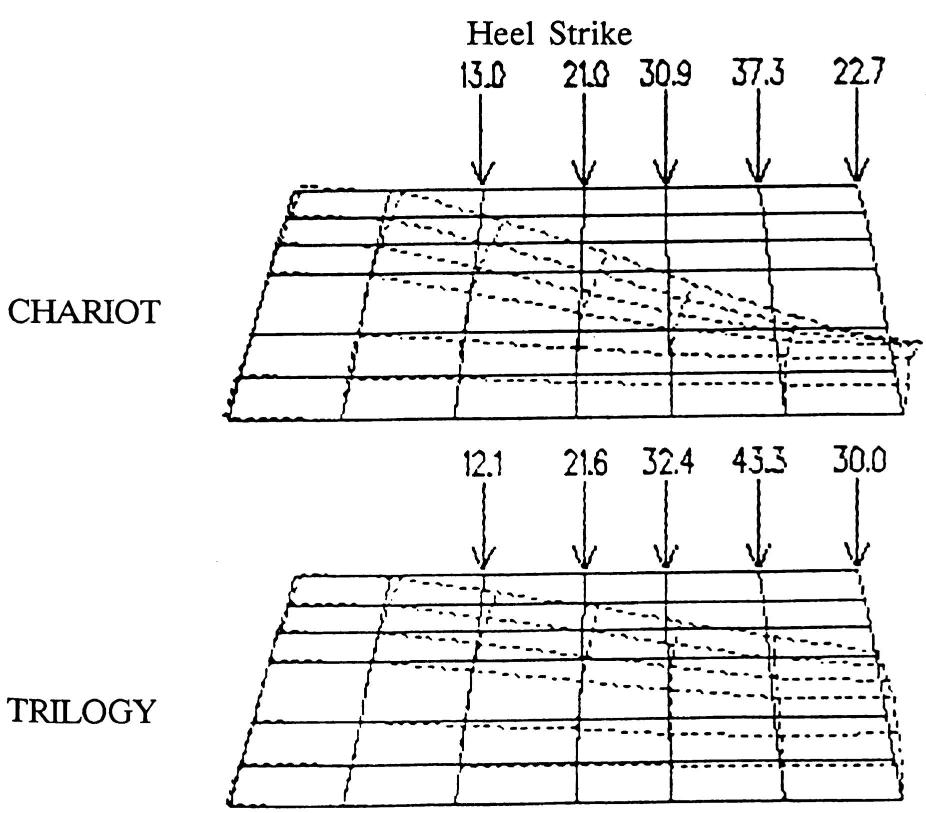
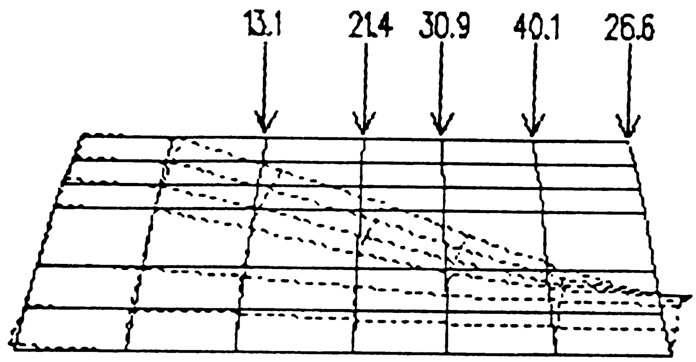


Figure C.8 Nodal Row 2.

Heel Strike

CHARIOT



TRILOGY

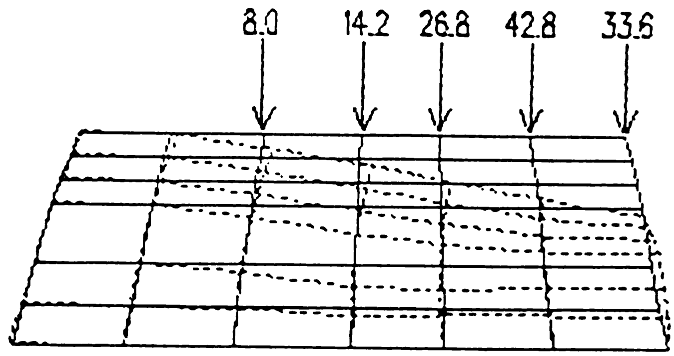
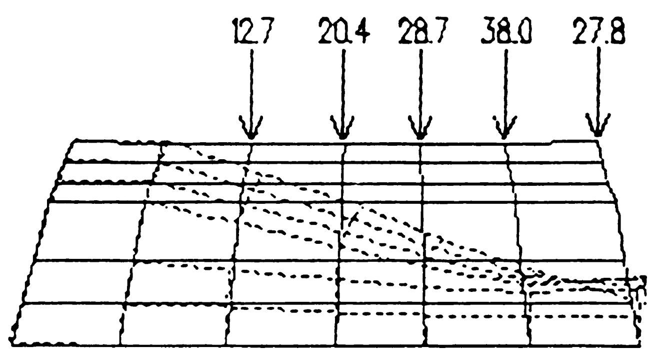


Figure C.9 Nodal Row 3.

Heel Strike

CHARIOT



TRILOGY

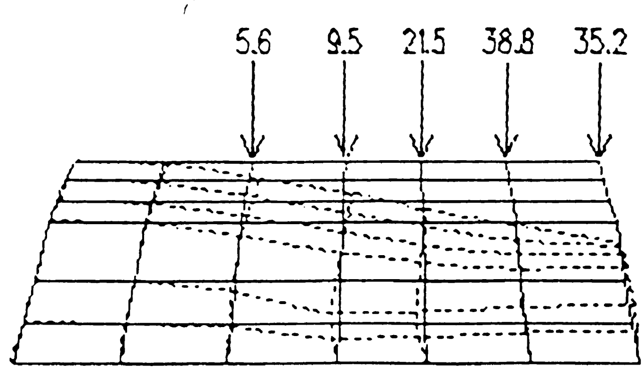
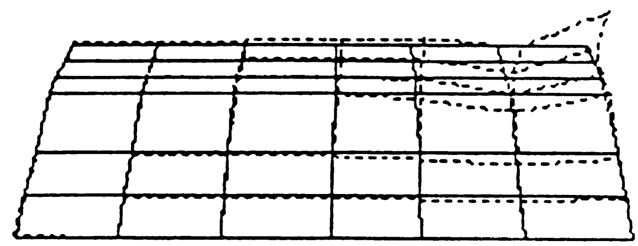


Figure C.10 Nodal Row 4.

Heel Strike

CHARIOT



TRILOGY

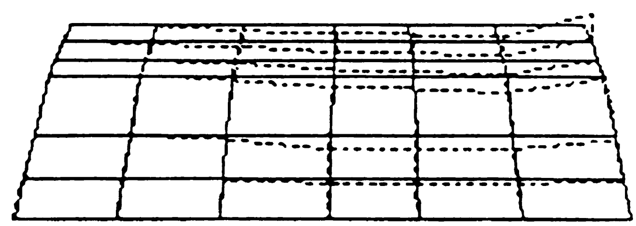
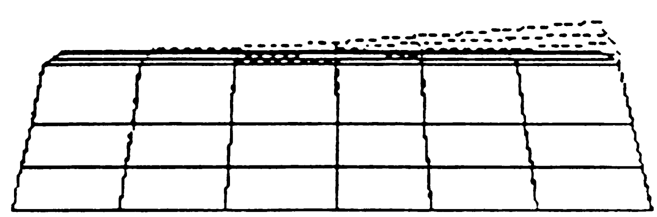


Figure C.11 Nodal Row 5.

Full Foot

CHARIOT



TRILOGY

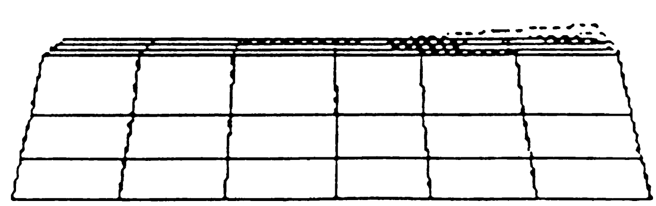


Figure C.12 Nodal Row 8.

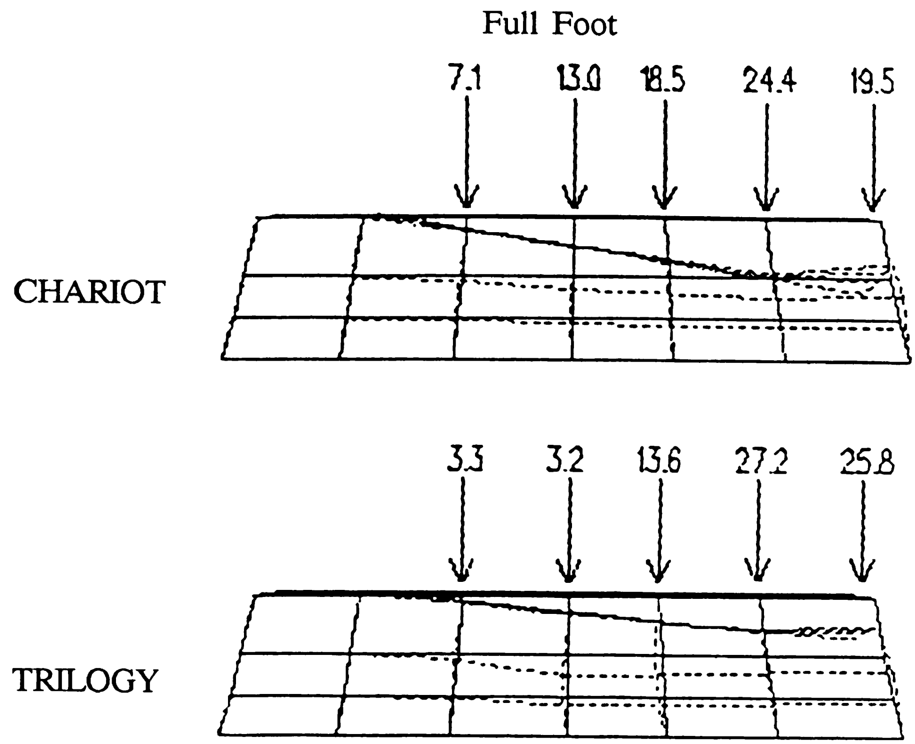


Figure C.13 Nodal Row 9.

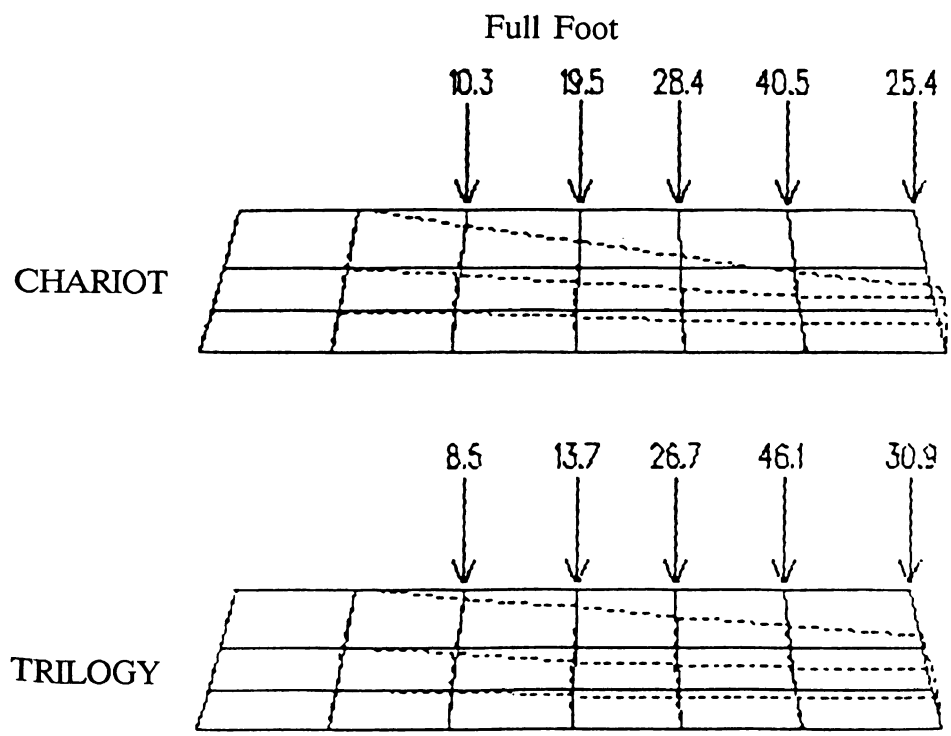


Figure C.14 Nodal Row 10.

Full Foot

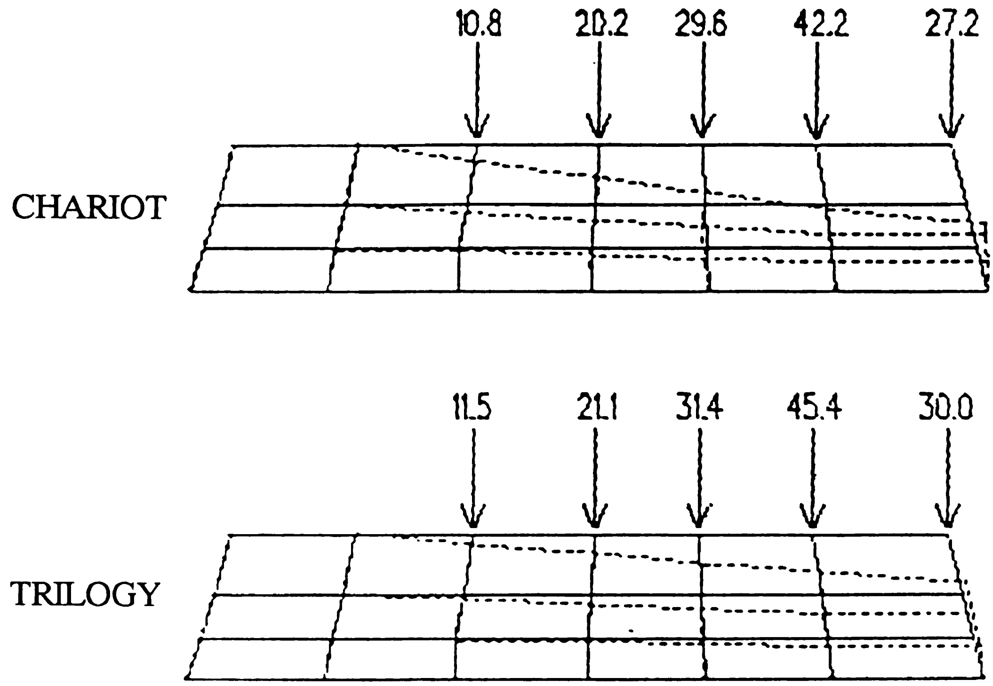


Figure C.15 Nodal Row 11.

Full Foot

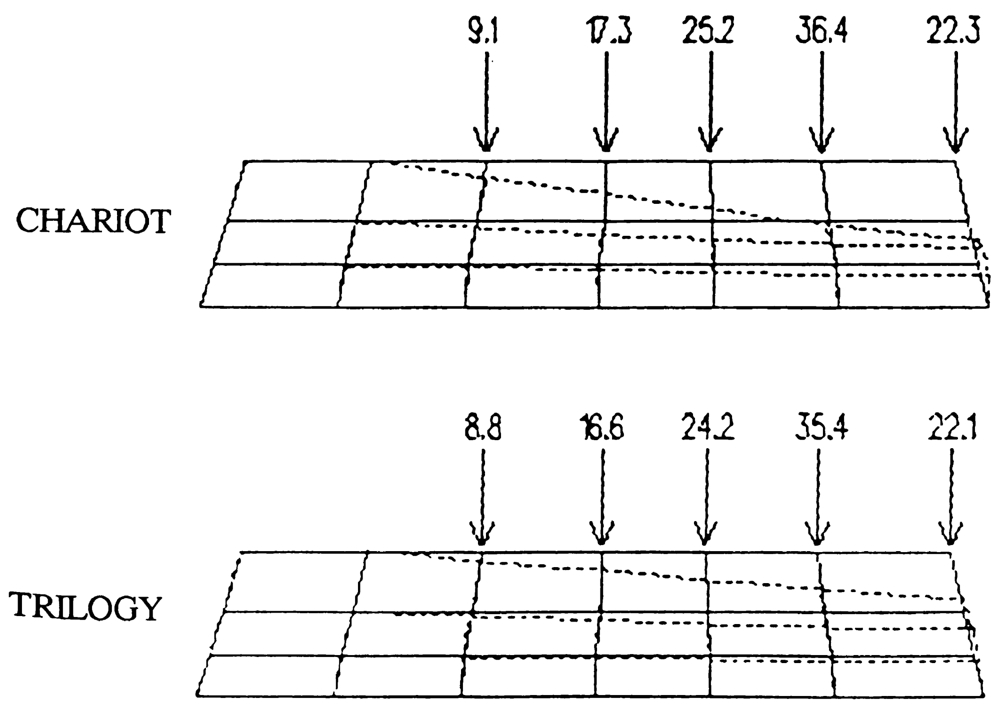


Figure C.16 Nodal Row 12.

Full Foot

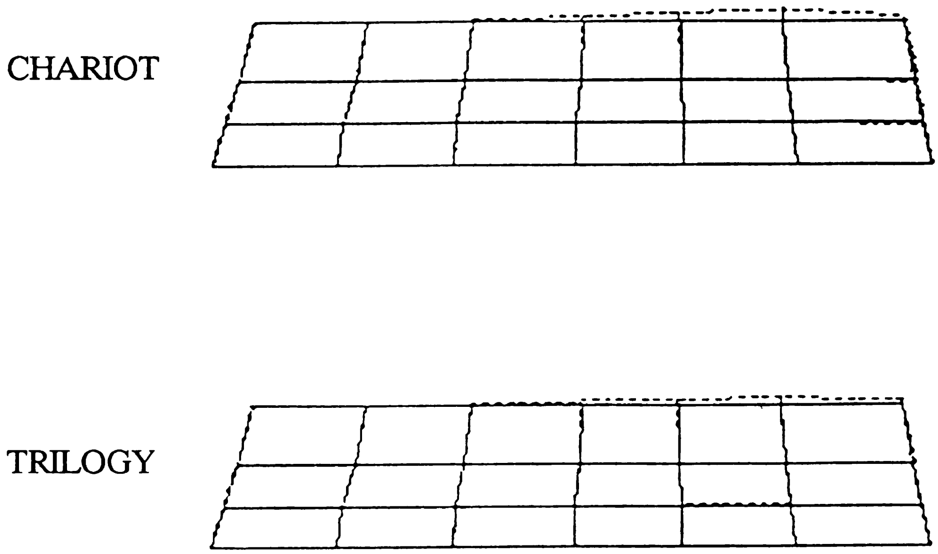


Figure C.17 Nodal Row 13.

Toe Off

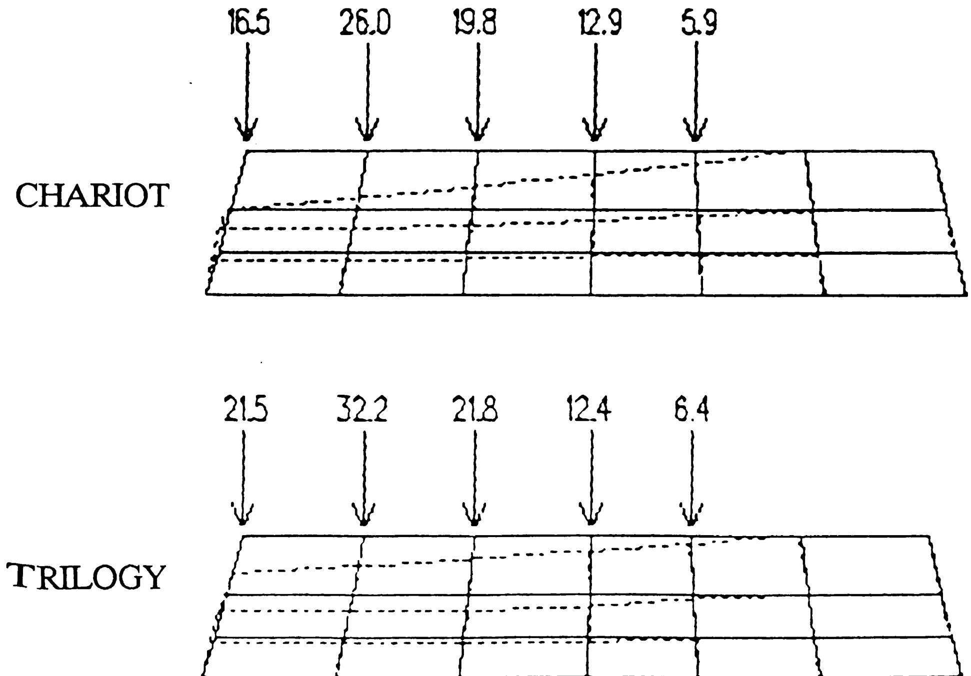


Figure C.18 Nodal Row 10.

Toe Off

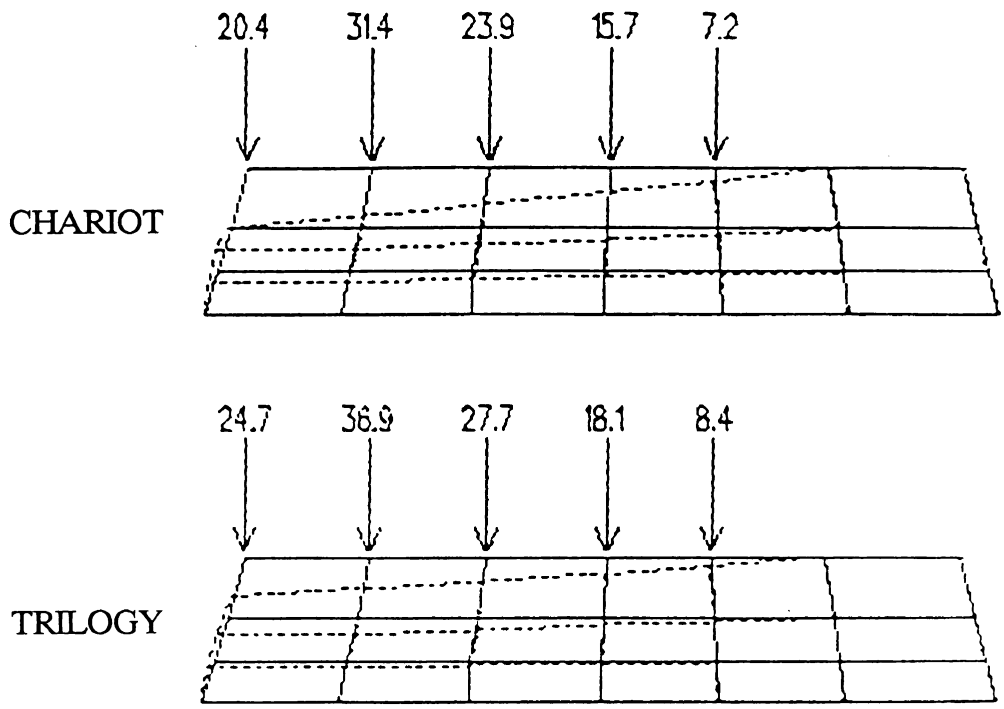


Figure C.19 Nodal Row 11.

Toe Off

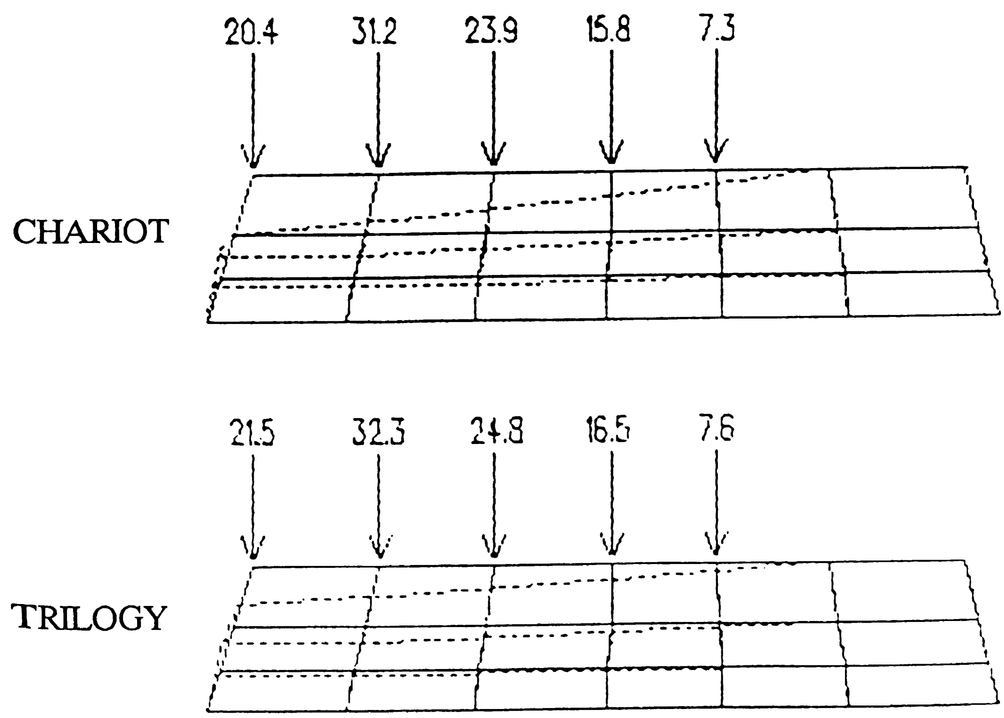


Figure C.20 Nodal Row 12.

Toe Off

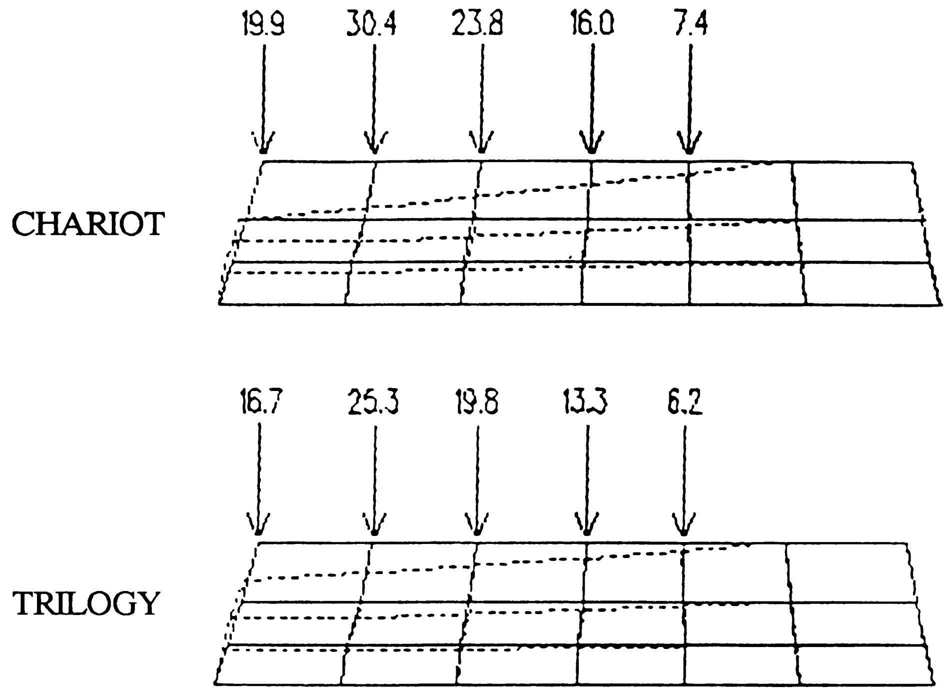


Figure C.21 Nodal Row 13.

Toe Off

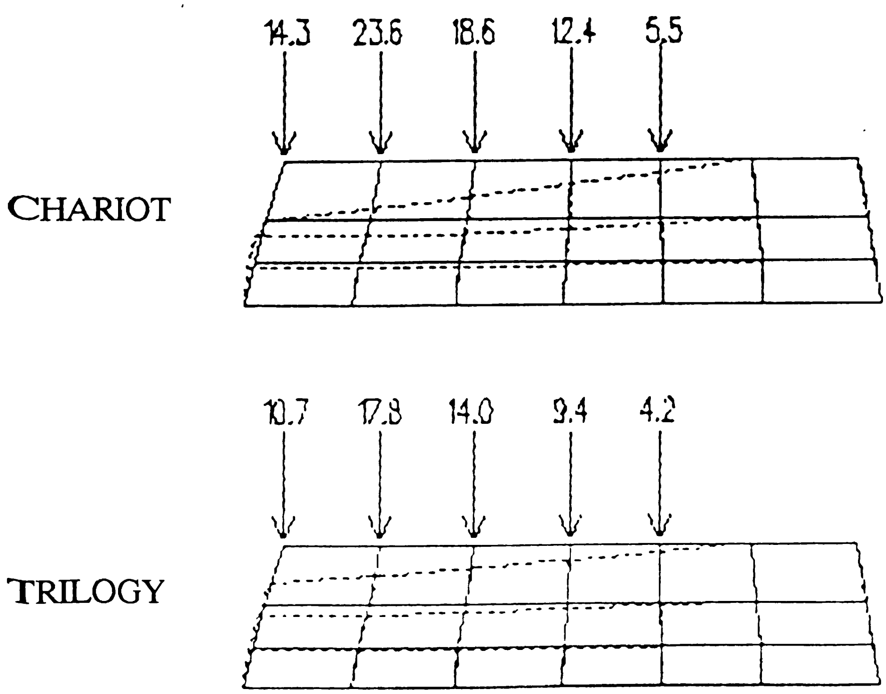
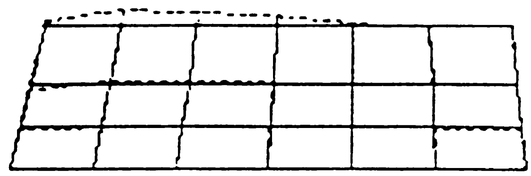


Figure C.22 Nodal Row 14.

Toe Off

CHARIOT



TRILOGY

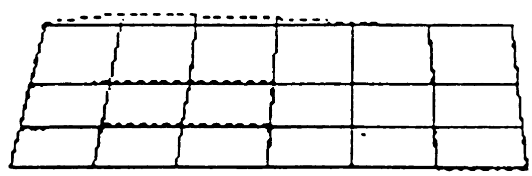


Figure C.23 Nodal Row 15.

APPENDIX D

ANSYS PREPROCESSOR ROUTINES FOR SHOE SOLE MODELING

This appendix contains the program and its subroutines developed as a preprocessor for ANSYS. These routines were written on a Prime 750 computer and compiled under Fortran77. Also included is the command file used to compile and load the program and its associated subroutines.

```
      PROGRAM DRIVER
C
C THIS PROGRAM IS THE DRIVER FOR THE CREATION OF AN ANSYS INPUT
C FILE OF A RUNNING SHOE SOLE MODEL.
C
      COMMON/ LOG / ASKEVERYPR, ASKEVERYEM, GETPRL, GETEML, PUTPRL, PUTE
     +ML
      COMMON/ PREMSPEC / ELASMOD(400), POISRATIO(400)
C
      LOGICAL ASKEVERYPR, ASKEVERYEM, GETPRL, GETEML, PUTPRL, PUTPRL
C
C SET UP THE INITIALIZATION FOR THE ANSYS COMMAND FILE, THIS INCLUDES
C WRITING THE NODAL COORDINATE INFORMATION.
C
      CALL INIT
C
C DEFINE THE MATERIAL SPECIFICATIONS FOR ALL LAYERS IN THE OUTSOLE
C
      CALL BOTOSOLE
      CALL TOPOSOLE
      CALL MIDSOLE
      CALL BOTWEDGE
      CALL MIDWEDGE
      CALL TOPWEDGE
C
C REORDER THE MATERIAL PROPERTY INFORMATION FOR EACH ELEMENT IN ORDER
C TO REDUCE THE WAVE FRONT.  WRITE THE ELEMENT ORDER INTO THE ANSYS FILE.
C
      CALL WRITELEM
C
C DEFINE THE LOAD SPECIFICATIONS
C
      CALL LOADS
C
C WRITE THE FINAL INFORMATION NEEDED TO COMMENCE THE SOLVING PROCESS
C
      CALL SOLVER
      CLOSE(6)
      END
```

```
      SUBROUTINE INIT
C
C  THIS SUBROUTINE WRITES THE INITIAL INFORMATION NEEDED FOR THE ANSYS
C  COMMAND FILE. THIS INFORMATION INCLUDES, A NUMBER OF PREP7 COMMANDS,
C  THE NAME OF THE COMMAND FILE, THE TITLE OF THE RUN, AND THE NODAL
C  COORDINATES.
C
      CHARACTER*32 CFILENAME, TITLE
C
C  PROMPT THE COMMAND FILE NAME
C
      PRINT*,'    ENTER THE NAME OF THE ANSYS COMMAND FILE TO STORE THIS
     +MODEL INFORMATION.'
      PRINT*,'    THE NAME SHOULD START WITH A "C_" TO INDICATE IT IS A
     +COMMAND FILE.'
      READ(1,'(A)') CFILENAME
      OPEN(6,FILE=CFILENAME)
C
C  WRITE THE INITIAL SECTION OF THE COMMAND FILE
C
      WRITE(6,1000)
1000  FORMAT('ANSYS')
      WRITE(6,2000)
2000  FORMAT('/INTER,NO')
      WRITE(6,2500)
2500  FORMAT('/PREP7')
C
C  PROMPT FOR THE TITLE, WRITE IT TO THE FILE, AND CONTINUE WITH PREP7
C  COMMANDS
C
      PRINT*,' ENTER THE TITLE OF THIS RUN'
      READ(1,'(A)') TITLE
      WRITE(6,3000) TITLE
3000  FORMAT('/TITLE, ',A32)
      WRITE(6,4000)
4000  FORMAT('ET,1,45')
C
C  WRITE THE NODAL COORDINATE INFORMATION
C
      OPEN(10,FILE='XYZNODES')
      DO 10 I=1,609
      READ(10,*) X,Y,Z
      WRITE(6,5000) I,X,Y,Z
5000  FORMAT('N,',I3,',',F10.5,',',F10.5,',',F10.5)
10    CONTINUE
      CLOSE(10)
      RETURN
      END


      SUBROUTINE CLSALL
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO CLOSE ALL UNITS THAT WERE
C  OPENED FOR THE LAYER SPECIFICATIONS BEFORE THEY ARE USED AGAIN
C
      CLOSE(7)
      CLOSE(8)
      CLOSE(11)
      CLOSE(12)
      RETURN
      END


      SUBROUTINE ASK
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE THE METHOD OF
C  INPUT FOR THE MATERIAL PROPERTY SPECIFICATIONS AND TO DETERMINE
C  IF THESE PROPERTIES NEED TO BE STORED FOR FUTURE USE.
C
      CHARACTER*1 ANS
C
      COMMON/ LOG / ASKEVERYPR, ASKEVERYEM, GETPRL, GETEML, PUTPRL, PUTE
     +ML
```

```
C
      LOGICAL ASKEVERYPR,ASKEVERYEM,GETPRL,GETEML,PUTPRL,PUTEML
C
      ASKEVERYPR = .FALSE.
      ASKEVERYEM = .FALSE.
C
C  PROMPT FOR THE INPUT METHOD FOR THE POISSON'S RATIO INFORMATION
C
1     PRINT*,' DO YOU WANT THE POISSON'S RATIO'
      PRINT*,'       1. THE SAME ACROSS THE ENTIRE LAYER?'
      PRINT*,'       2. SPECIFIED AT EVERY ELEMENT?'
      PRINT*,'       3. VALUES READ FROM AN EXTERNAL FILE?'
      READ(1,*) IANSWER
C
C  CHECK FOR VALID INPUT
C
      IF(IANSWER.NE.1 .AND. IANSWER.NE.2 .AND. IANSWER.NE.3) THEN
      PRINT*,' ANSWER OUT OF RANGE. '
      GO TO 1
      ENDIF
C
C  SET THE APPROPRIATE LOGICAL VARIABLES
C
      IF(IANSWER.EQ.2) ASKEVERYPR = .TRUE.
      IF(IANSWER.EQ.3) THEN
      CALL GETPR
      GETPRL = .TRUE.
      ENDIF
C
C  ASK IF THE POISSON'S RATIO VALUES SHOULD BE STORED FOR FUTURE USE
C
3     PRINT*,' DO YOU WANT THIS POISON'S RATIO SPECIFICATION STORED FOR
     + FUTURE USE?'
      READ(1,'(A)') ANS
C
C  CHECK FOR VALID INPUT
C
      IF(ANS.NE.'Y' .AND. ANS.NE.'N') THEN
      PRINT*,' ENTER EITHER Y OR N. '
      GO TO 3
      ENDIF
C
C  SET THE APPROPRIATE LOGICAL VARIABLES
C
      IF(ANS.EQ.'Y') THEN
      PUTPRL = .TRUE.
      CALL PUTPR
      ENDIF
C
C  PROMPT FOR THE INPUT METHOD FOR THE ELASTIC MODULAS INFORMATION
C
2     PRINT*,' DO YOU WANT THE ELASTIC MODULAS'
      PRINT*,'       1. THE SAME ACROSS THE ENTIRE LAYER?'
      PRINT*,'       2. SPECIFIED AT EVERY ELEMENT?'
      PRINT*,'       3. VALUES READ FROM AN EXTERNAL FILE?'
      READ(1,*) IANSWER
C
C  CHECK FOR VALID INPUT
C
      IF(IANSWER.NE.1 .AND. IANSWER.NE.2 .AND. IANSWER.NE.3) THEN
      PRINT*,' ANSWER OUT OF RANGE. '
      GO TO 2
      ENDIF
C
C  SET THE APPROPRIATE LOGICAL VARIABLES
C
      IF(IANSWER.EQ.2) ASKEVERYEM = .TRUE.
      IF(IANSWER.EQ.3) THEN
      CALL GETEM
      GETEML = .TRUE.
      ENDIF
C
C  ASK IF THE ELASTIC MODULAS VALUES SHOULD BE STORED FOR FUTURE USE
```

```
C
4     PRINT*,' DO YOU WANT THIS ELASTIC MODULAS SPECIFICATION STORED FOR
     + FUTURE USE?'
      READ(1,'(A)') ANS
C
C  CHECK FOR VALID INPUT
C
      IF(ANS.NE.'Y' .AND. ANS.NE.'N') THEN
      PRINT*,' ENTER EITHER Y OR N. '
      GO TO 4
      ENDIF
C
C  SET THE APPROPRIATE LOGICAL VARIABLES
C
      IF(ANS.EQ.'Y') THEN
      PUTEML = .TRUE.
      CALL PUTEM
      ENDIF
      RETURN
      END


      SUBROUTINE MATSPECS(ISTART,IEND,INC)
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO PROMPT THE USER TO INPUT
C  THE MATERIAL PROPERTIES BASED ON THE METHOD OF INPUT CHOSEN AND
C  STORED IN LOGICAL VARIABLES
C
      COMMON/ PREMSPEC / ELASMOD(400), POISRATIO(400)
      COMMON/ LOG / ASKEVERYPR, ASKEVERYEM, GETPRL, GETEML, PUTPRL, PUTE
     +ML
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
C
      LOGICAL FIRST,ASKEVERYPR,ASKEVERYEM,GETPRL,GETEML,PUTPRL,PUTEML
C
C  THE METHOD OF INPUT IS TO SPECIFY A MATERIAL PROPERTY WHICH IS
C  THE SAME ACROSS THE ENTIRE LAYER.  PROMPT FOR THAT VALUE.
C
      IF((.NOT.ASKEVERYPR).AND.(.NOT.GETPRL).AND.(FIRST)) THEN
      PRINT*,' ENTER POISSON'S RATIO '
      READ(1,*) PRALL
      ENDIF
C
      IF((.NOT.ASKEVERYEM).AND.(.NOT.GETEML).AND.(FIRST)) THEN
      PRINT*,' ENTER THE ELASTIC MODULAS '
      READ(1,*) EMALL
      ENDIF
C
C  SPECIFYING THE POSSON'S RATIO FOR THE LAYER
C
      DO 10 I=ISTART,IEND,INC
      ICNTER = ICNTER + 1
C
C  PROMPT FOR A VALUE AT EVERY ELEMENT
C
      IF(ASKEVERYPR) THEN
      WRITE(1,1000) ICNTER
1000  FORMAT(' ENTER POISSON'S RATIO FOR ELEMENT ',I3)
      READ(1,*) POISRATIO(I)
C
C  READ THE VALUE FROM AN EXTERNAL FILE
C
      ELSEIF(GETPRL) THEN
      READ(7,*) POISRATIO(I)
C
C  THE VALUE IS THE SAME ACROSS THE LAYER, WRITE THAT VALUE INTO THE
C  ARRAY
C
      ELSE
      POISRATIO(I) = PRALL
      ENDIF
C
C  IF THE VALUES ARE TO BE SAVED, WRITE TO AN EXTERNAL FILE
C
```

```
      IF(PUTPRL) WRITE(11,*) POISRATIO(I)
C
C  SPECIFYING THE POSSON'S RATIO FOR THE LAYER
C
C
C  PROMPT FOR A VALUE AT EVERY ELEMENT
C
      IF(ASKEVERYEM) THEN
      WRITE(1,2000) ICNTER
2000  FORMAT(' ENTER ELASTIC MODULAS FOR ELEMENT ',I3)
      READ(1,*) ELASMOD(I)
C
C  READ THE VALUE FROM AN EXTERNAL FILE
C
      ELSEIF(GETEML) THEN
      READ(8,*) ELASMOD(I)
      ELSE
C
C  THE VALUE IS THE SAME ACROSS THE LAYER. WRITE THAT VALUE INTO THE
C  ARRAY
C
      ELASMOD(I) = EMALL
      ENDIF
C
C  IF THE VALUES ARE TO BE SAVED, WRITE TO AN EXTERNAL FILE
C
      IF(PUTEML) WRITE(12,*) ELASMOD(I)
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PUTPR
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO PROMPT FOR THE FILE NAME IN
C  WHICH THE POISON'S RATIO PROPERTIES ARE TO BE WRITTEN AND OPEN
C  THAT FILE ON UNIT 11
C
      CHARACTER*32 PRPUTNAME
C
      PRINT*,' ENTER THE NAME OF THE FILE TO STORE THIS LAYER'S POISON'S
     +RATIO INFORMATION.'
      READ(1,'(A)') PRPUTNAME
      OPEN(11,FILE=PRPUTNAME)
      RETURN
      END

      SUBROUTINE GETPR
C
C  THE PUPOSE OF THIS SUBROUTINE IS TO PROMPT FOR THE FILE WITH
C  THE POISON'S RATIO PROPERTIES AND TO OPEN THAT FILE ON UNIT 7
C
      CHARACTER*32 PRFILE
C
      PRINT*,' ENTER THE NAME OF THE FILE WITH THE POISSON'S RATIO PROPE
     +RTIES.'
      READ(1,'(A)') PRFILE
      OPEN(7,FILE=PRFILE)
      RETURN
      END

      SUBROUTINE PUTEM
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO PROMPT FOR THE FILE NAME IN
C  WHICH THE ELASTIC MODULAS PROPERTIES ARE TO BE WRITTEN AND OPEN
C  THAT FILE ON UNIT 12
C
      CHARACTER*32 EMPUTNAME
C
      PRINT*,' ENTER THE NAME OF THE FILE TO STORE THIS LAYER'S ELASTIC
     +MODULAS INFORMATION.'
      READ(1,'(A)') EMPUTNAME
      OPEN(12,FILE=EMPUTNAME)
      RETURN
      END

      SUBROUTINE GETEM
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO PROMPT FOR THE FILE WITH
C  THE ELASTIC MODULAS PROPERTIES AND TO OPEN THAT FILE ON UNIT 8
C
      CHARACTER*32 EMFILE
C
      PRINT*,' ENTER THE NAME OF THE FILE WITH THE ELASTIC MODULAS PROPE
     +RTIES.'
      READ(1,'(A)') EMFILE
      OPEN(8,FILE=EMFILE)
      RETURN
      END
```

```
      SUBROUTINE BOTOSOLE
C
C  THIS SUBROUTINE PROMPT FOR THE MATERIAL PROPERTIES OF THE BOTTOM OF
C  THE OUTSOLE AND STORES THIS INFORMATION UNTIL THE ELEMENTS ARE WRITTEN
C  INTO THE COMMAND FILE.
C
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
      LOGICAL FIRST
C
C  INDICATE TO THE USER WHICH LAYER HE IS DEFINING
C
      PRINT*,' YOU ARE SPECIFYING PROPERTIES FOR THE BOTTOM OF THE OUTSO
     +LE'
C
C  FIND OUT HOW THE MATERIAL PROPERTIES WILL BE SPECIFIED
C
      ICNTER = 0
      CALL ASK
C
C  DEFINE THE MATERIAL PROPERTIES
C  THIS SUBROUTINE CALL OCCURS TWICE BECAUSE OF THE WAY THE ELEMENTS
C  ARE ORDERED.
C
      ISTART = 1
      IEND = 283
      INC = 6
      FIRST = .TRUE.
      CALL MATSPECS(ISTART,IEND,INC)
      ISTART = 289
      IEND = 394
      INC = 3
      FIRST = .FALSE.
      CALL MATSPECS(ISTART,IEND,INC)
      CALL CLSALL
      RETURN
      END

      SUBROUTINE TOPOSOLE
C
C  THIS SUBROUTINE PROMPT FOR THE MATERIAL PROPERTIES OF THE TOP OF
C  THE OUTSOLE AND STORES THIS INFORMATION UNTIL THE ELEMENTS ARE WRITTEN
C  INTO THE COMMAND FILE.
C
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
      LOGICAL FIRST
C
C  INDICATE TO THE USER WHICH LAYER HE IS DEFINING
C
      ICNTER = 0
      PRINT*,' YOU ARE SPECIFYING PROPERTIES FOR THE TOP OF THE OUTSOLE.
     +'
C
C  FIND OUT HOW THE MATERIAL PROPERTIES WILL BE SPECIFIED
C
      CALL ASK
C
C  DEFINE THE MATERIAL PROPERTIES
C  THIS SUBROUTINE CALL OCCURS TWICE BECAUSE OF THE WAY THE ELEMENTS
C  ARE ORDERED.
C
      ISTART = 2
      IEND = 284
      INC = 6
      FIRST = .TRUE.
      CALL MATSPECS(ISTART,IEND,INC)
      ISTART = 290
      IEND = 395
      INC = 3
      FIRST = .FALSE.
      CALL MATSPECS(ISTART,IEND,INC)
      CALL CLSALL
      RETURN
      END
```

```
      SUBROUTINE MIDSOLE
C
C  THIS SUBROUTINE PROMPT FOR THE MATERIAL PROPERTIES OF THE MIDSOLE
C  AND STORES THIS INFORMATION UNTIL THE ELEMENTS ARE WRITTEN INTO
C  THE COMMAND FILE.
C
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
      LOGICAL FIRST
C
C  INDICATE TO THE USER WHICH LAYER HE IS DEFINING
C
      ICNTER = 0
      PRINT*,' YOU ARE SPECIFYING PROPERTIES FOR THE MIDSOLE.'
C
C  FIND OUT HOW THE MATERIAL PROPERTIES WILL BE SPECIFIED
C
      CALL ASK
C
C  DEFINE THE MATERIAL PROPERTIES
C  THIS SUBROUTINE CALL OCCURS TWICE BECAUSE OF THE WAY THE ELEMENTS
C  ARE ORDERED.
C
      ISTART = 3
      IEND = 285
      INC = 6
      FIRST = .TRUE.
      CALL MATSPECS(ISTART,IEND,INC)
      ISTART = 291
      IEND = 396
      INC = 3
      FIRST = .FALSE.
      CALL MATSPECS(ISTART,IEND,INC)
      CALL CLSALL
      RETURN
      END

      SUBROUTINE BOTWEDGE
C
C  THIS SUBROUTINE PROMPT FOR THE MATERIAL PROPERTIES OF THE BOTTOM OF
C  THE WEDGE AND STORES THIS INFORMATION UNTIL THE ELEMENTS ARE WRITTEN
C  INTO THE COMMAND FILE.
C
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
      LOGICAL FIRST
C
C  INDICATE TO THE USER WHICH LAYER HE IS DEFINING
C
      ICNTER = 0
      PRINT*,' YOU ARE SPECIFYING PROPERTIES FOR THE BOTTOM OF THE WEDGE
     +.'
C
C  FIND OUT HOW THE MATERIAL PROPERTIES WILL BE SPECIFIED
C
      CALL ASK
C
C  DEFINE THE MATERIAL PROPERTIES
C
      ISTART = 4
      IEND = 286
      INC = 6
      FIRST = .TRUE.
      CALL MATSPECS(ISTART,IEND,INC)
      FIRST = .FALSE.
      CALL CLSALL
      RETURN
      END
```

```
      SUBROUTINE MIDWEDGE
C
C  THIS SUBROUTINE PROMPT FOR THE MATERIAL PROPERTIES OF THE MIDDLE OF
C  THE WEDGE AND STORES THIS INFORMATION UNTIL THE ELEMENTS ARE WRITTEN
C  INTO THE COMMAND FILE.
C
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
      LOGICAL FIRST
C
C  INDICATE TO THE USER WHICH LAYER HE IS DEFINING
C
      ICNTER = 0
      PRINT*,' YOU ARE SPECIFYING PROPERTIES FOR THE MIDDLE OF THE WEDGE
     +.'
C
C  FIND OUT HOW THE MATERIAL PROPERTIES WILL BE SPECIFIED
C
      CALL ASK
C
C  DEFINE THE MATERIAL PROPERTIES
C
      ISTART = 5
      IEND = 287
      INC = 6
      FIRST = .TRUE.
      CALL MATSPECS(ISTART,IEND,INC)
      FIRST = .FALSE.
      CALL CLSALL
      RETURN
      END

      SUBROUTINE TOPWEDGE
C
C  THIS SUBROUTINE PROMPT FOR THE MATERIAL PROPERTIES OF THE TOP OF
C  THE WEDGE AND STORES THIS INFORMATION UNTIL THE ELEMENTS ARE WRITTEN
C  INTO THE COMMAND FILE.
C
      COMMON / STUFF / FIRST,PRALL,EMALL,ICNTER
      LOGICAL FIRST
C
C  INDICATE TO THE USER WHICH LAYER HE IS DEFINING
C
      ICNTER = 0
      PRINT*,' YOU ARE SPECIFYING PROPERTIES FOR THE TOP OF THE WEDGE.'
C
C  FIND OUT HOW THE MATERIAL PROPERTIES WILL BE SPECIFIED
C
      CALL ASK
C
C  DEFINE THE MATERIAL PROPERTIES
C
      ISTART = 6
      IEND = 288
      INC = 6
      FIRST = .TRUE.
      CALL MATSPECS(ISTART,IEND,INC)
      FIRST = .FALSE.
      CALL CLSALL
      RETURN
      END
```

```
      SUBROUTINE WRITELEM
C
C   THE PURPOSE OF THIS SUBROUTINE IS TO WRITE THE PROPER MATERIAL
C   PROPERTIES FOR EACH ELEMENT, AND THEN DEFINE THAT ELEMENT
C
      COMMON/ PREMSPEC / ELASMOD(400), POISRATIO(400)
C
      INTEGER ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT
C
C   OPEN THE FILE WITH THE ELEMENT DEFINITION
C
      OPEN(9,FILE='ELEMENTS')
C
C   WRITE THE MATERIAL PROPERTIES FOR EACH ELEMENT
C
      DO 10 I=1,396
      WRITE(6,500) I
500   FORMAT('MAT,',I3)
      WRITE(6,1000) I,POISRATIO(I)
1000  FORMAT('NUXY,',I3,',',F10.4)
      WRITE(6,2000) I,ELASMOD(I)
2000  FORMAT('EX,',I3,',',F10.4)
C
C   WRITE THE ELEMENT DEFINITION
C
      READ(9,*) ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT
      WRITE(6,3000) ONE,TWO,THREE,FOUR,FIVE,SIX,SEVEN,EIGHT
3000  FORMAT('E,',I4,',',I4,',',I4,',',I4,',',I4,',',I4,',',I4,',',I4)
10    CONTINUE
      CLOSE(9)
      RETURN
      END


      SUBROUTINE LOADS
C
C   THE PURPOSE OF THIS SUBROUTINE IS TO DEFINE THE LOADS TO APPLY
C   TO THE MODEL
C
      CHARACTER*1 LOADTYPE
      CHARACTER*1 ANS
      CHARACTER*4 LABEL
      CHARACTER*32 LOADFILE
      CHARACTER*32 STRLOAD
C
      LOGICAL WRIT
C
      WRIT = .FALSE.
      ICNT = 0
200   CONTINUE
      ICNT = ICNT + 1
C
C   PROMPT FOR THE METHOD OF INPUT FOR THE LOAD DEFINTION
C
1     PRINT*,' DO YOU WANT TO '
      PRINT*,'     1. DEFINE LOADS.'
      PRINT*,'     2. READ THE LOADS FROM AN EXTERNAL FILE.'
      PRINT*,'     3. EXIT.'
      READ(1,*) IANS
C
C   CHECK FOR VALID INPUT
C
      IF(IANS.NE.1 .AND. IANS.NE.2 .AND. IANS.NE.3) THEN
      PRINT*,' ANSWER OUT OF RANGE. '
      GO TO 1
      ENDIF
C
C   THE FIRST TIME THROUGH, ASK IF THE LOAD DEFINITION NEEDS TO BE
C   STORED FOR FUTURE USE
C
      IF((IANS.EQ.1).AND.(ICNT.EQ.1)) THEN
3     PRINT*,' DO YOU WANT THIS LOAD DEFINITION STORED FOR FUTURE USE?'
      READ(1,'(A)') ANS
```

```
C
C   CHECK FOR VALID INPUT
C
        IF(ANS.NE.'Y' .AND. ANS.NE.'N') THEN
        PRINT*,' ENTER EITHER Y OR N. '
        GO TO 3
        ENDIF
C
C   SET APPROPRIATE LOGICAL VARIABLES
C
        IF(ANS.EQ.'Y') THEN
        WRIT = .TRUE.
C
C   PROMPT FOR THE NAME OF THE FILE TO STORE THIS INFORMATION
C   AND OPEN UNIT 14
C
        PRINT*,' ENTER THE NAME OF THE FILE TO STORE THIS LOAD INFORMATION
      +.'
        READ(1,'(A)') STRLOAD
        OPEN(14,FILE=STRLOAD)
C
C   PROMPT FOR THE NUMBER OF LOADS TO BE DEFINED AND WRITE INTO THE FILE
C
        PRINT*,' ENTER THE TOTAL NUMBER OF LOADS TO BE DEFINED.'
        READ(1,*) NUMLOADS
        WRITE(14,*) NUMLOADS
        ENDIF
        ENDIF
C
C   THE LOAD IS TO BE USER DEFINED. PROMPT FOR THE LOAD DEFINITION
C
        IF(IANS.EQ.1) THEN
        PRINT*,' ENTER THE TYPE OF LOAD. D = DEFLECTION, F = FORCE.'
        READ(1,'(A)') LOADTYPE
4       PRINT*,' ENTER THE LABEL FOR THE LOAD. IF THE LOAD IS'
        PRINT*,'     A) DEFLECTION  THE OPTIONS ARE UX, UY, UZ OR ALL'
        PRINT*,'     B) FORCE.  THE OPTIONS ARE FX, FY, FZ, OR ALL'
        READ(1,'(A)') LABEL
C
C   CHECK FOR VALID INPUT
C
        IF(LOADTYPE.EQ.'D' .AND. LABEL.NE.'UX' .AND. LABEL.NE.'UY' .AND. L
      +ABEL.NE.'UZ' .AND. LABEL.NE.'ALL') THEN
        PRINT*,' INCORRECT LABEL. '
        GO TO 4
        ENDIF
        IF(LOADTYPE.EQ.'F' .AND. LABEL.NE.'FX' .AND. LABEL.NE.'FY' .AND. L
      +ABEL.NE.'FZ' .AND. LABEL.NE.'ALL') THEN
        PRINT*,' INCORRECT LABEL. '
        GO TO 4
        ENDIF
C
C   PROMPT FOR WHICH NODES THIS LOAD IS TO BE APPLIED
C   AND WRITE THIS INTO THE COMMAND FILE
C
        PRINT*,' ENTER THE STARTING NODE, ENDING NODE, NODE INCREMENT, AND
      +THE VALUE OF THE LOAD.'
        READ(1,*) ISTART, IEND, INC, VALUE
        WRITE(6,1000) LOADTYPE,ISTART,LABEL,VALUE,IEND,INC
1000    FORMAT(A1,',',I3,',',A4,',',F10.4,',,',I3,',',I3)
C
C   WRITE THE LOAD INFORMATION TO A FILE IF REQUESTED
C
        IF(WRIT) WRITE(14,1000) LOADTYPE,ISTART,LABEL,VALUE, IEND,INC
C
C   THE LOADS ARE TO BE READ FROM AN EXTERNAL FILE. PROMPT FOR THE FILE
C   WHICH CONTAINS THE VALUES AND WRITE IT TO THE COMMAND FILE
C
        ELSEIF(IANS.EQ.2) THEN
        PRINT*,' ENTER THE NAME OF THE FILE WITH THE LOAD INFORMATION.'
        READ(1,'(A)') LOADFILE
        OPEN(13,FILE=LOADFILE)
        READ(13,*) NUMLOADS
```

```
      DO 10 I=1,NUMLOADS
      READ(13,2000) LOADTYPE,ISTART,LABEL,VALUE,IEND,INC
2000  FORMAT(A1,1X,I3,1X,A4,1X,F10.4,2X,I3,1X,I3)
      WRITE(6,1000) LOADTYPE,ISTART,LABEL,VALUE,IEND,INC
      IF(WRIT) WRITE(14,1000) LOADTYPE,ISTART,LABEL,VALUE, IEND,INC
10    CONTINUE
      CLOSE(13)
      GO TO 100
      ELSE
      GO TO 100
      ENDIF
      GO TO 200
100   CONTINUE
      CLOSE(14)
      RETURN
      END

      SUBROUTINE SOLVER
C
C  THE PURPOSE OF THIS SUBROUTINE IS TO WRITE THE ANSYS
C  INFORMATION THAT IS NEEDED FOR SOLUTION TO THE COMMAND FILE
C
      WRITE(6,500)
500   FORMAT('KRF,1')
      WRITE(6,1000)
1000  FORMAT('ITER,1,1,1')
      WRITE(6,2000)
2000  FORMAT('AFWRIT')
      WRITE(6,3000)
3000  FORMAT('FINISH')
      WRITE(6,4000)
4000  FORMAT('/EXEC')
      WRITE(6,5000)
5000  FORMAT('/INPUT,27')
      WRITE(6,6000)
6000  FORMAT('FINISH')
      RETURN
      END
```

```
FTN77 DRIVER -DEBUG -FULLCHECK
FTN77 INIT -DEBUG -FULLCHECK
FTN77 CLSALL -DEBUG -FULLCHECK
FTN77 ASK -DEBUG -FULLCHECK
FTN77 MATSPECS -DEBUG -FULLCHECK
FTN77 PUTPR -DEBUG -FULLCHECK
FTN77 PUTEM -DEBUG -FULLCHECK
FTN77 GETPR -DEBUG -FULLCHECK
FTN77 GETEM -DEBUG -FULLCHECK
FTN77 BOTOSOLE -DEBUG -FULLCHECK
FTN77 TOPOSOLE -DEBUG -FULLCHECK
FTN77 MIDSOLE -DEBUG -FULLCHECK
FTN77 TOPWEDGE -DEBUG -FULLCHECK
FTN77 MIDWEDGE -DEBUG -FULLCHECK
FTN77 BOTWEDGE -DEBUG -FULLCHECK
FTN77 WRITELEM -DEBUG -FULLCHECK
FTN77 LOADS -DEBUG -FULLCHECK
FTN77 SOLVER -DEBUG -FULLCHECK
SEG
LOAD #DRIVER
LO B_DRIVER
LO B_INIT
LO B_CLSALL
LO B_ASK
LO B_MATSPECS
LO B_PUTPR
LO B_PUTEM
LO B_GETPR
LO B_GETEM
LO B_BOTOSOLE
LO B_TOPOSOLE
LO B_MIDSOLE
LO B_TOPWEDGE
LO B_MIDWEDGE
LO B_BOTWEDGE
LO B_WRITELEM
LO B_LOADS
LO B_SOLVER
LI F77LIB
LI
MA 3
Q
DELETE B_DRIVER
DELETE B_INIT
DELETE B_CLSALL
DELETE B_ASK
DELETE B_MATSPECS
DELETE B_PUTPR
DELETE B_PUTEM
DELETE B_GETPR
DELETE B_GETEM
DELETE B_BOTOSOLE
DELETE B_TOPOSOLE
DELETE B_MIDSOLE
DELETE B_TOPWEDGE
DELETE B_MIDWEDGE
DELETE B_BOTWEDGE
DELETE B_WRITELEM
DELETE B_LOADS
DELETE B_SOLVER
CO -TTY
```