1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

**STUDY OF TIRE-TERRAIN INTERACTION
USING THE FINITE ELEMENT METHOD**

by

Song-tao Chen B.Eng.

**A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering**

**Department of Mechanical
and Aeronautical Engineering
Carleton University
Ottawa, Ontario
Canada**

September 15, 1988

© copyright

1988, Song-tao Chen

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-46279-5

ACCEPTANCE SHEET

The undersigned recommend to the Faculty of Graduate Studies and Research the acceptance of the thesis: "Study of tire-terrain interaction using the finite element method" submitted by Song-tao Chen, in partial fulfillment of the requirements for the degree of Master of Engineering.

Thesis Supervisor

Chairman,

Department of Mechanical
and Aeronautical Engineering

ABSTRACT

In recent years, attempts have been made to apply the finite element method (FEM) to the analysis of tire-terrain interaction. However, almost all of the FEM developed to date appears to be suitable only for the study of terrain response to vehicular traffic. To develop a realistic method based on FEM for predicting tire performance, a new approach is being explored. In the new approach, a pneumatic tire is modelled as an elastic, hollow cylinder with basic dimensions corresponding to those of the tire. Its characteristics are analyzed using the FEM. In the analysis of tire-terrain interaction, the pressure-sinkage and shear stress-shear displacement relationships of the terrain, as measured by bevameter, are used. In addition, the response of the terrain to repetitive normal and shear loadings are taken into account. Using an incremental numerical technique and considering the conditions for the dynamic equilibrium of a moving tire, the deformed shape of the tire, the terrain sinkage, and the normal and shear stress distributions on the tire-terrain interface are determined. From the stress distributions on the interface at various slips, the tractive performance of a pneumatic tire, with a particular normal load and inflation pressure over specific terrain, can be more realistically predicted. The procedure is then extended to the prediction of the overall performance of an off-road wheeled vehicle.

ACKNOWLEDGEMENTS

The author wishes to extend his deepest appreciation to his advisory Professor J.Y.Wong for his invaluable guidance, encouragement and generous suggestions throughout the course of this investigation and in preparing the thesis.

The author also takes this opportunity to express his appreciation to his parents and family members for their encouragement throughout the period of his studies.

CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	x
LIST OF APPENDICES	xi
NOMENCLATURE	xii
 CHAPTER I	
INTRODUCTION	1
1.1 Review of the State of the Art	
in the Study of Tire-Terrain Interaction	1
1.2 Scope of the Present Work	17
CHAPTER II	
MODELLING TIRE CHARACTERISTICS	
USING THE FINITE ELEMENT METHOD	20
2.1 Description of the Tire Model	20
2.2 Deriving the Geometric Relationships and	
Elastic Properties of the Tire Model	24
2.2.1 Deriving the Geometric Relationships of	
the Tire Model on Undeformable Flat Surface	27
2.2.2 Deriving the Young's Modulus of	
the Tire Model	34

CHAPTER	III	ANALYSIS OF TIRE-TERRAIN INTERACTION	
		USING THE TIRE MODEL	40
	3.1	Terrain Model	40
	3.2	Tire Performance Prediction	44
	3.3	Front Tire Analysis	47
	3.3.1	Symmetric Loading Condition	48
	3.3.2	Asymmetric Loading Condition	49
	3.4	Rear Tire Analysis	55
	3.5	Vehicle Performance Prediction	57
CHAPTER	IV	APPLICATIONS OF THE TIRE MODEL TO	
		THE PREDICTION OF THE PERFORMANCE	
		OF THE FRONT AND REAR TIRES AND	
		THE OVERALL PERFORMANCE OF A WHEELED VEHICLE	62
	4.1	Introduction	62
	4.2	Front Tire Performance	64
	4.3	Rear Tire Performance	69
	4.4	Vehicle Performance	76
CHAPTER	V	CONCLUSIONS AND RECOMMENDATIONS	89
	5.1	Conclusions	89
	5.2	Recommendations	90
REFERENCES			91
APPENDICES			96

LIST OF FIGURES

Figure	Description	Page
1	Modes of operation for a tire according to Bekker's assumptions	3
2	Tire-terrain configuration according to Fujimoto's model	5
3	Tire-terrain system stiffness of Baladi's model	7
4	Tire-terrain configuration according to Baladi's model	8
5	Finite element mesh of Yong's model	11
6	Tire-terrain configuration according to Yong's model	12
7	Representation of the tire model	21
8	Three types of meshes of quadrilateral element	23
9	Finite element mesh of the tire model	25
10	Generalized Deflection Chart	26
11	Illustration of the incremental-iterative approach for deriving the relationship between contact length and deflection	29
12	Illustration of the calculated contact length at certain deflection	32
13	Comparison of the contact lengths predicted by FEM and the measured ones	33

14	Comparison of contact widths predicted by FEM and the measured one	35
15	Young's modulus vs. normal load on the model	38
16	Representative pressure-sinkage relationship and response to repetitive loading	42
17	Illustration of the derivation of shear displacement	45
18	Vehicle free body diagram	60
19	Normal pressure and shear stress distributions of front tire at 1% slip	66
20	Normal pressure and shear stress distributions of front tire at 20% slip	67
21	Normal pressure and shear stress distributions of front tire at 40% slip	68
22	Illustration of front tire deformation, sinkage and elastic rebound of the terrain	70
23	Front tire performance at a particular tire inflation pressure, a specific slip and a given load	71
24	Normal pressure and shear stress distributions of rear tire at 7% slip	73
25	Normal pressure and shear stress distributions of rear tire at 20% slip	74
26	Normal pressure and shear stress distributions of rear tire at 60% slip	75

27	Illustration of rear tire deformation, sinkage and elastic rebound of the terrain	77
28	Rear tire performance at a particular tire inflation pressure, a specific slip and a given load	79
29	Deformations, sinkages and max. normal pressures and shear stresses of the front and rear tires of a vehicle at 10% slip	80
30	Deformations, sinkages and max. normal pressures and shear stresses of the front and rear tires of a vehicle at 40% slip	81
31	Comparison of vehicle performances by FEM and Wong's model	88

LIST OF TABLES

Table	Description	Page
I	Soft soil data	65
II	Front tire performance for normal load at 119.5 kN	72
III	Rear tire performance for normal load at 118.03 kN	78
IV	Vehicle performance at 10% slip	83
V	Vehicle performance at 20% slip	84
VI	Vehicle performance at 40% slip	85

LIST OF APPENDICES

Appendix	Description	Page
A	Outline of the formulation of two-dimensional finite element stress analysis using 8-node quadrilateral isoparametric elements	96
B	Constrained single-variable Golden section search method	105
C	Three-point Lagrangian interpolation	106
D	Computer codes	107

NOMENCLATURE

a	acceleration
b	the smaller dimension of a contact area
c	cohesion
DBP	Drawbar pull
E1	elastic rebound produced by front tire
E2	elastic rebound produced by rear tire
i	slip
j	shear displacement
K	shear deformation modulus
k_c	cohesive modulus of terrain deformation
k_f	frictional modulus of terrain deformation
n	exponent of terrain deformation
N1	normal load on front tire
N2	normal load on rear tire
P	normal pressure under a rectangular or circular plate
r	distance from tire centre to the point on the tire in tire-terrain contact region
R	radius of undeformed tire
R_m	motion resistance
R1	motion resistance of front tire
R2	motion resistance of rear tire
S	shear stress

S1	sinkage related to front tire
S2	sinkage related to rear tire
T1	thrust of front tire
T2	thrust of rear tire
V	translatory speed of tire centre
W	normal load, weight of a vehicle
Z	sinkage
α	angle between r and the tangent to the point on tire in tire-terrain contact region
θ	angular displacement
θ_0	front contact angle of a tire
θ_1	rear contact angle of a tire
ϕ	angle of internal shearing resistance
ω	angular speed

CHAPTER I INTRODUCTION

The prediction and evaluation of the tractive performance of off-road wheeled vehicle is one of the major topics in Terramechanics. To predict its performance, it is essential to have a basic understanding of the mechanics of tire-terrain interaction. A variety of methods of approach to the study of the mechanics of tires on unprepared terrain, ranging from entirely empirical to highly analytical, has been proposed. A brief review of these methods is presented below.

1.1 Review of the State of the Art

in the Study of Tire-Terrain Interaction

The tractive performance of a tire is usually defined by its thrust, motion resistance, sinkage, slip in relation to the applied normal load, torque and angular speed. Since the thrust, motion resistance and other parameters are determined by the normal and shear stress distributions along the tire-terrain interface, a central issue in the study of the mechanics of tire-terrain interaction is to establish the relationships between the stress distributions on the interface and tire parameters and terrain conditions (1).

Over the years, many methods have been proposed for determining the stress distributions on the tire-terrain interface. In the 1960's, Bekker (2) proposed a relatively simple method of evaluating the performance of a pneumatic tire on soil.

It was assumed that if the sum of the inflation pressure and the pressure produced by the carcass is greater than or equal to the maximum pressure exerted by the terrain at the lowest point of the tire circumference, the tire will remain undeformed like a rigid wheel. On the other hand, if the terrain is hard enough, the lower part of the tire will deform and the lower portion of the contact length will be equivalent to the chord of the circle representing the undeformed shape of the tire (see Fig.1). The normal pressure along the contact patch is calculated from Bekker's pressure-sinkage relationship. The pressure varies from zero at the surface to a constant value along the flat portion of the contact patch. The resistance due to soil compaction, the internal resistance due to the hysteresis of tire material, and sinkage are calculated from the normal stress distribution and equilibrium conditions. It should be mentioned, however, that the equations for evaluating the thrust and drawbar pull are not provided by this simple mathematical model developed by Bekker.

It should be pointed out that while the assumptions regarding tire deformation described above may be reasonable under certain conditions, they certainly can not be regarded as universally true.

Following Bekker's model, in 1978 and 1986 Wong (3) developed a comprehensive model for wheeled vehicle performance. It can be used to predict both the normal and shear stress distributions under a tire. A new criterion for evaluating the operating mode of a tire, whether behaving like a rigid or

Figure 1 Modes of operation for a tire according to Bekker's assumption

elastic wheel, has been introduced, and effects of lugs on the tractive performance has been taken into consideration.

In 1977, Fujimoto (4) proposed a tire performance model for operating over cohesive soil. A uniform normal pressure distribution along the contact patch was assumed. The shear stress was expressed as an exponential function of the shear strain of the soil. The contact geometry was considered as part of a rigid wheel with a radius greater than that of the undeformed tire (see Fig.2). The front and rear contact angles were used in the analysis. The front contact angle was determined from the equilibrium of the tire, while the rear contact angle was expressed as a function of the ratio of tire stiffness and soil strength. The equivalent radius was calculated from the equilibrium equations. Making use of the equations for rigid wheel performance, the drawbar pull, motion resistance, torque and sinkage of the tire can be predicted.

This model appears to be an improvement over the Bekker model in that the shear stress on the tire-terrain interface has been taken into account. The torque, thrust and drawbar pull can then be determined from shear stress distribution. However the prediction procedure is more complicated than that of the Bekker model. Furthermore, the assumption that the normal pressure is uniformly distributed over the contact patch is not realistic in many cases.

In 1984, Baladi and Rohani (5) developed a model, in which, the tire-terrain interaction was treated as two springs in

Figure 2 Tire-terrain configuration according to Fujimoto's model

series, one representing the tire and the other representing the terrain, as shown in Figs.3 and 4. The stiffness of the tire spring was derived from the characteristics of the tire on hard ground. The contact length was assumed as the chord of a circle representing the tire circumference, and the contact width was assumed to remain constant. The rigidity of the terrain, on the other hand, was obtained by assuming that at certain sinkage the normal stress exerted by terrain is equal to the radial stress inside an expanding cavity. The radial stress is expressed analytically in terms of the shear strength and the volume change characteristics of soil. From these, the normal pressure on the interface was expressed as a function of contact angle and terrain properties. The shear stress was then derived in terms of normal pressure, slip, and the shear modulus, cohesion and angle of internal friction of the terrain. An assumption similar to Fujimoto's (4) about the contact geometry was made. The performance is determined by integrating the normal and shear stresses over the contact area.

Baladi and Rohani showed that there was a good agreement between the tire performance predicted using their model and the laboratory test data. It should be mentioned that according to this model the maximum normal pressure always occurs at mid-point of the contact patch. This does not always agree with the existing experimental data, which indicate that the location of the maximum normal pressure varies with the slip of the tire.

The Finite Element Method (FEM) was first introduced into

a. SPRING CONSTANT FOR FLEXIBLE TIRE (k_t)

b. SPRING CONSTANT FOR SOIL (k_s)

c. SPRING CONSTANT FOR SOIL-TIRE SYSTEM $\frac{k_s k_t}{k_s + k_t}$

Figure 3 Tire-terrain system stiffness of Baladi's model

the study of tire-terrain interaction by J.V.Perumpral et al (6) in 1971 in an attempt to predict the stress distributions and the soil deformation within the soil mass under a tractor tire.

For a stationary tire the elliptical contact area was approximated by a circular area and the load was applied normal to the circular area uniformly. The non-linear behaviour of the soil was considered, and the non-linear equations were solved using an iterative method. The elastic constants were different for different elements and were updated in each iteration according to the stress-strain relationship from experimental data. The difference between the new and previous elastic constants served as a criterion for convergence. The Poisson's ratio was assumed to be constant throughout the analysis and a value of 0.4 was chosen for convenience. This means that Young's modulus is the only variable. The results were compared with those from elastic theory and showed good agreement.

For a rolling tire, it was considered to be a plane strain problem. The tire was assumed as a rigid wheel to simplify the analysis. The assumptions regarding the elastic properties were the same as those for a stationary tire. Sinkage, radial and tangential stresses were specified as input for the finite element analysis. The outputs were the strain, stress and displacement fields of the soil mass under the tire with a given load.

This kind of model requires the geometry of contact patch and the stress distributions on the tire-soil interface to be

specified. It can only be used to analyze the strain, stress and displacement fields within the soil mass. It can not be used to predict the stress distributions on the interface, the shape of the interface and consequently the performance of the tire.

In 1977 and 1980, Yong et al (7) developed a more sophisticated finite element model (see Fig.5 and 6). They assumed the tire was a linear elastic cylindrical body (plane strain case). The terrain was treated as another cylindrical body with an infinite radius and as a piecewise linear elastic material. Only the terrain was discretized by finite elements. The contact length and deflection on hard surface were predicted using the Hertz contact theory for two elastic bodies, one of which representing the hard surface has a very high value of Young's modulus. To account for the tangential stress presented on the contact surface the Hertz theory was modified. By matching the load-deflection characteristics on hard ground, the Young's modulus of the tire, which reflects the behaviour of the whole tire assembly, including the hub, tire carcass stiffness, and inflation pressure, was obtained. The value of Poisson's ratio was assumed to be 0.5 for the reason that the volume change of the tire under load is usually insignificant. The normal and shear stress distributions on the contact surface were taken from experimental data and used as inputs to the finite element analysis. Again, this method is only suitable for studying the strain and stress fields in the soil mass. It should be pointed out that most unprepared terrain under load does not behave

Figure 5 Finite element mesh of Yong's model

Figure 6 Tire-terrain configuration according to Yong's model

elastically, therefore, the underlying assumption for this model that the terrain is piecewise elastic appears to be unrealistic. Compared with the model proposed by J.V.Perumpral et al (6), the model developed by Yong et al provides improvements in the prediction of contact geometry.

In 1984, A.Odia (8) developed an advanced finite element model for predicting the deformation of soil under tire loading without requiring the stress boundary conditions as input. In the model, the soil was discretized and considered as a rheological material, the characteristics of which are time-dependent, including creep and stress relaxation. Constitutive equations in both mono-axial and two-dimensional stress state were formulated. Time was a variable in his model. This implies that this model is non-linear. Constant strain or stress boundary conditions were assumed for a very short time in each calculation. The new boundary conditions for the subsequent calculation were from the previous FEM solution.

In comparison with other finite element methods developed so far, this model does not require the stress distributions on tire-terrain interface as inputs. It was claimed that this method could be applied to other analysis such as the prediction of soil deformation under a moving tire. However, the possibility is not obvious because the contact geometry of a pneumatic tire on unprepared terrain can not be defined by this method.

In most of the methods described above, the normal pressure across the contact width of a tire is implicitly assumed to be

uniformly distributed. This is not always consistent with the available experimental data. The elastic rebound of the terrain at the trailing edge of the contact patch due to the unloading by the tire is not taken into consideration.

Generally speaking, tire-terrain interaction is a contact problem between two media, an elastic tire and a mainly plastic terrain. The object is to determine the contact geometry and the pressure distributions on the interface between the two bodies. This class of problem is generally known to be strongly nonlinear, and hence difficult to analyze since the contact geometry is not known a priori and is changing in size and shape as the load is being applied, and the pressure distribution is changing as well (9).

It is generally accepted that due to the nonlinearity of this type of problem, the applicability of the Hertz contact theory is very limited. Hence, finite element methods are widely used. Papers on the mechanics of contact of two bodies with rigid-elastic, elastic-elastic and even elastic-plastic characteristics and having simple contact geometry have been published. The problem is classified into two major groups, with friction and without friction on the contact surface. The methods presented in these papers (9 to 12) have been mainly based on the variational principles and finite element techniques. While these methods could provide a fundamental understanding of the contact problem, they are based on a number of simplifying assumptions. As a result, they have not provided a realistic

engineering solution to the tire-terrain interaction problem, with complicated contact geometry and complex response of the terrain to the normal and shear loadings.

An example of this kind of study was conducted by Oden and Lin (10) in 1986. A cylindrical deformable body which is rolling on a rough, flat hard surface was studied. The material of the cylinder was assumed to be a compressible or incompressible hyperelastic material or a hyperelastic / viscoelastic material characterized by constitutive equations valid for finite deformation. The friction on the contact surface, as well as the inertia of the cylinder, was taken into account. Bifurcations of the solutions of the model at certain rolling velocity were studied, which were manifested as the emergence of standing waves in the cylinder.

Another example was shown by Chen and Tsai (11) in 1985. The paper presented a general and effective finite element technique for analyzing elastodynamic sliding contact problems with friction. The deformed contact area was obtained using the constraint conditions developed by quadratic mathematic programming technique. Lagrangian multipliers were introduced to evaluate the contact pressure due to friction and to determine the adhesion or release of contact surface. Based on these, the sliding process between two contact surfaces was accurately modelled. Their work also provided the correction formulae for modeling the transient response of velocities, and accelerations on contact surface when initial impact or release of the surface

occurred. Several examples were given in the paper.

In addition to the application of the finite element method to the analysis of contact problems mentioned above, FEM has also been employed for the analysis of tire characteristics.

In 1980 and 1981, the researchers of General Motor Laboratory (13 to 15) constructed a large scale, comprehensive, three-dimensional, thermal-mechanical finite element model for analyzing the power loss in pneumatic tire rolling on hard ground. The model consists of three major analysis modules-- deformation, dissipation, and thermal-- which are used in an iterative and relaxation solution scheme.

In 1985, Ridha et al (16) constructed a three-dimensional finite element model of a homogeneous tire to study tire deformation subject to a normal load on hard surface. The contact area was predicted using an incremental-iterative approach. For a given deflection of the hub, the program computes the nodes that come into contact with a predefined contact plane, and the magnitude of the contact forces at the nodes. It was found that the computed shape and size of the contact area at different levels of rim deflection were in good agreement with experimental results. The computed results included the distribution of stress, strain, and strain energy within the tire and on the contact surface.

Most of these research efforts (16 to 22) were concentrated on the analysis of the tire under inflation pressure loading or under vertical loading on hard ground for a given inflation

pressure. The analysis of pneumatic tire performance over deformable terrain was not included in their studies.

1.2 Scope of the Present Work

This thesis is intended to develop a new approach to the analysis of tire-terrain interaction based on FEM. Using this approach the tractive performance of a tire over unprepared terrain, as well as the overall performance of a wheeled vehicle, will be predicted. To achieve this, the analysis of the normal pressure and shear stress distributions on the contact patch and the contact geometry are performed. This has not been carried out in previous studies based on FEM.

From the literature review given previously, it appears that the finite element method should be properly applied to the study of tire-terrain interaction in order to obtain realistic and meaningful results. Theoretically, both tire and terrain can be modelled by finite elements. The possibility is shown by Chen and Tsai (11). This may be an ideal model to the analysis of tire-terrain interaction.

In this thesis, however, the problem is simplified. The complex structure of a tire is first idealized as two-dimensional problem and modelled by finite elements. The case of plane stress is considered. The terrain is represented by a group of semi-empirical equations describing the pressure-sinkage relationship, shear stress-shear displacement relationship, and the response of terrain to repetitive loadings.

Based on the principle of elasticity, the relationship between the deflection and contact length of the tire model on hard ground is first studied. The corresponding contact width is then determined by assuming the contact area of the model on hard ground is identical to the measured area of the tire under the same normal load and published by the tire manufacturer. The relationship between the deflection and contact width of the tire model is subsequently established for the application to the analysis of tire-terrain interaction.

The elastic properties of the tire model, namely Young's modulus and Poisson's ratio, are studied and discussed making use of the "Generalized Deflection Chart" produced by the tire manufacturer. The equilibrium condition of vertical forces acting on the tire model is used in deriving the value of Young's modulus by a single variable search method.

The analysis of the tire-terrain interaction is achieved by simulating the process of tire loading and employing an incremental approach. The process is divided into two phases: the static loading phase, in which the shear stress is absent and the geometry of tire and terrain is symmetric about the vertical axis; and the rolling phase, in which the shear stress is presented and the geometry is asymmetric. Again, the equilibrium of vertical forces serves as the condition in determining the contact geometry and stress distributions. The front tire and the rear tire are modeled separately since they are in contact with different terrain having different terrain properties. The

front tires are in contact with the unprepared terrain (virgin), whereas the rear tires are in the ruts produced by the front tires.

A vehicle model is developed based on the analyses of both the front and rear tires. The tractive performance of a wheeled vehicle on unprepared terrain is predicted on the basis of the performance of individual tires, with dynamic load transfer between the front and rear tires being taken into account. An example is given to show the applicability of this newly developed approach.

Finally, conclusions of the study and recommendations for future work are presented.

CHAPTER II MODELLING TIRE CHARACTERISTICS USING FINITE ELEMENT METHOD

2.1 Description of the Tire Model

The tire-wheel assembly, which performs the functions of supporting weight, transmitting power, cushioning shock, etc., consists of the shaft, hub, tire carcass and pressurized air inside the tire. The pneumatic tire is a highly complex composite structure, consisting primarily of a number of layers of flexible cords of high modulus of elasticity embedded in a rubber matrix. The carcass as a whole is non-homogeneous and anisotropic. Furthermore, the mechanical properties of the tire are typically nonlinear and may be viscoelastic. The deformations under load are large enough to require the consideration of large-displacement, geometric nonlinearities.

In this thesis, to simplify the analysis the pneumatic tire is modelled as a homogeneous, isotropic, elastic, hollow cylinder with inside and outside diameters equal to those of the tire, as shown in Fig.7. The treads are assumed to be smooth. The elastic properties of this tire model will reflect the entire structure of the tire, including the carcass and pressurized air inside the tire.

It should be pointed out that while from the physical viewpoint the tire model presented is different from an actual pneumatic tire, the elastic properties of the tire model, such as its Young's modulus, will be derived from the measured

Figure 7 Representation of the tire model

characteristics of the tire, provided by the tire manufacturer. Thus, it ensures that the elastic behaviour of the tire model is similar to that of the corresponding pneumatic tire.

In the analysis the inner circumference is held fixed to represent the connection between the tire and the hub, which has much higher rigidity than that of the pneumatic tire. This means no matter how the tire deforms, the rim remains round. The length of the cylinder will represent the tire contact width. Since the contact width of the tire varies with the normal load applied, the contact width of the cylinder varies accordingly. Because the lateral forces acting perpendicular to the tire plane are ignored in this model, and the contact width is small in comparison with the outside diameter of the tire, it is treated as a plane stress problem.

The derivation of relationships between the contact length, contact width, contact area and deflection and the values of elastic properties will be described in the following section.

To perform finite element analysis, the solution domain (the object body to be analyzed) must be discretized properly. Figure 8 shows three different types of meshes. Meshes 1 and 2 are 8-node element meshes. Mesh 2 contains a transition area between dense and loose areas. Using mesh 2 could reduce the number of elements in comparison with using mesh 1. But mesh 2 may be severely distorted after several consecutive executions of the FEM program. Furthermore, numbering the elements and nodes of mesh 2 is not as easy as mesh 1, especially when the number of

Mesh 1 Mesh 2 Mesh 3

Figure 8 Three types of meshes of quadrilateral element

elements changes. Mesh 3 is different from the other two types in formulation. It is a variable node element mesh. Using mesh 3 can reduce the number of elements without risking the distortion. But the formulation of mesh 3 is sophisticated. In this thesis mesh 1 is used for simplicity and convenience. Figure 9 illustrates the actual mesh of the tire model.

2.2 Deriving the Geometric Relationships and Elastic Properties of the Tire Model

The geometric relationships of a tire refer to those between the deflection and contact length, contact width and contact area.

Since the tire model presented is an approximation of a real tire, its elastic properties, such as Young's modulus, have to be derived according to the "Generalized Deflection Chart" for the tire, published by the tire manufacturer (see Fig.10) to ensure that the elastic behaviour of the tire model is as close to that of a real tire as possible. Before deriving its elastic properties, the geometric relationships, such as the relationship between the contact length and deflection of the tire model, must be first established. Because this model is a two-dimensional approximation of a three-dimensional tire. The geometric relationships of the tire model will not be exactly the same as those of the tire as described in the "Generalized Deflection Chart".

Figure 9 Finite element mesh of the tire model

Figure 10 Generalized Deflection Chart

2.2.1 Deriving the Geometric Relationships of the Tire Model on Undeformable Flat Surface

It is well known in elasticity that if the boundary condition applied to an elastic body is displacement, the deformation of this elastic body is independent of its Young's modulus, but it is dependent on its Poisson's ratio. This is because Young's modulus is a parameter describing the relation between stress and strain while Poisson's ratio is a geometric constant. According to this principle, the relationship between the contact length and deflection of the tire model can be derived without knowing the Young's modulus.

The relationship is derived using an incremental-iterative approach based on FEM. The procedure is outlined below:

A) A coordinate system is first set up, in which the vertical axis is through the tire centre (or the centre of the cylinder for the tire model) and the horizontal axis is tangent to the bottom of the undeformed cylinder. The origin of the coordinate system is at the bottom of the cylinder, as shown in Fig 9. The inner circumference of the tire model is fixed with respect to the coordinate system. In the computer program the hard ground is moved up to simulate a given deflection. Taking advantage of the symmetry of the configuration, only half of the tire model is discretized. In deriving the relationship between contact length and deflection, the contact width and Young's modulus are chosen arbitrarily. The Poisson's ratio is assumed to be 0.5.

B) A vertical displacement is applied at the nodal point of an element at the bottom of the model. The amount of the displacement is equal to the difference in the vertical coordinate between the bottom point A and the first adjacent point B, as shown in Fig.11. This amount of displacement is the input to the linear finite element analysis procedure to calculate the deformation of the tire model.

C) After the calculation in step B), the first adjacent point B' may be still above the bottom point A' due to the deformation of the entire tire structure. But the vertical distance between these two points is reduced.

D) Steps B) and C) are repeated until the vertical distance between A" and B" is smaller than a prescribed value, so that they can be considered essentially on a horizontal line. The line between A" and B" is considered to form half of the contact length. The deflection is simply the vertical coordinate of the contact length. Thus the first datum point on the contact length-deflection curve for the tire model is obtained.

E) For the second increment, the previous contact length is given a vertical displacement and the linear finite element analysis procedure is repeated until the adjacent point and the contact length determined at the previous increment are on a new level. The vertical coordinate of the new contact length is the new deflection. This determines the second datum point on the contact length-deflection curve for the

Figure 11 Illustration of the incremental-iterative approach for deriving the relationship between contact length and deflection

tire model.

F) Step E) is repeated until the required maximum increment (or deflection) is reached, which should correspond to the maximum deflection of the tire specified by the tire manufacturer.

This incremental-iterative procedure is illustrated in Flow Chart I.

Figure 12 illustrates an example of the calculated contact length of the tire model at the deflection of 10.21 cm.

Because of the simplification made for the tire model, the calculated relationship between contact length and deflection does not perfectly match the measured one, as shown in Fig.13. The calculated contact length is longer for a given deflection. The reason for this may be due to the ignorance of the resistance on the contact surface to the sliding of the tire model and due to the simplification of the tire model. The calculated contact length is shorter than the chord, which has been used as contact length in some previous tire models. The improvement in using FEM in the calculation of contact length is obvious.

It should be mentioned that the above incremental-iterative procedure for deriving the relationship between the contact length and deflection of the tire model is necessary because of the mechanical behaviour of the tire model. As a matter of fact, it is a tire-rigid surface contact problem.

In view of the fact that the volume change of an actual tire

Flow Chart I

Figure 12 Illustration of the calculated contact length at the deflection of 10.21 cm.

Goodyear 29.5-25 SHRL DL L-3 tire

Figure 13 Comparison of the contact lengths by FEM and the measured one

under load is usually not very significant, it is decided that in the subsequent analysis the Poisson's ratio for the tire model is taken as to be 0.5.

2.2.2 Deriving the Young's Modulus of the Tire Model

The Young's modulus derived for the tire model represents the overall property of the entire structure of the tire assembly. It should, therefore, be understood that the Young's modulus for the tire model referred to in this thesis is in fact the equivalent or apparent Young's modulus and is not that for the real tire.

To derive the Young's modulus, the contact width should be calculated first. In this thesis, it is assumed that the contact area of the tire model on hard surface for a given deflection is identical to that of the real tire described in the "Generalized Deflection Chart". Therefore, the contact width for the tire model is determined by dividing the contact area by the calculated contact length for a given deflection. Since the calculated contact length is longer than the actual one, the contact width for the tire model is shorter than the measured one, as shown in Fig.14.

Also, a specific inflation pressure of the tire should be selected. It is obvious that for different inflation pressures, the derived Young's modulus will be different under a given load. Generally speaking, the higher the inflation pressure, the larger the Young's modulus will be. The procedure for deriving the

Goodyear 29.5-25 SHRL DL L-3 tire

Figure 14 Comparison of contact widths predicted by FEM and the measured, one

Young's modulus is described below:

- A) After the Poisson's ratio is selected, the relationship between contact length and deflection for that ratio is recalculated. At the final iteration in each increment, the contact length, deflection, and the coordinates of the nodal points on the outer circumference of the deformed tire model are stored.
- B) The contact width is calculated for a deflection and contact length from step A). The stored coordinates corresponding to that deflection are used as prescribed displacement inputs to the linear finite element analysis procedure to ensure that the deformation of the tire model for this deflection is exactly the same as before.
- C) An initial value of Young's modulus is selected and the linear finite element analysis procedure is executed to calculate the reactions at the prescribed nodal points for the inputs from step B). The resultant normal reaction from the ground is performed by summing the reactions at nodal points on the outer circumference in the vertical direction.
- D) A single variable search method is employed to obtain the Young's modulus by which the difference between the resultant normal reaction from the ground and the normal load at a given deflection, specified in the "Generalized Deflection Chart", is less than a prescribed value.
- E) Steps C) and D) are repeated for a number of normal loads and the relationships between Young's modulus and the load

is then established, as shown in Fig. 15.

The above procedure is illustrated in Flow Chart II.

For different tire inflation pressure, a corresponding curve can be obtained by the above method. It can be seen from Fig.15 that the value of Young's modulus is not a constant over the range of normal load shown. However, in the normal operating range, the variation of the value of Young's modulus is small.

As mentioned earlier, because all major parameters of the tire model, such as Young's modulus, are derived from the "Generalized Deflection Chart" of the corresponding tire published by the tire manufacturer, this ensures that the tire model developed is a reasonable representation of the mechanical behaviour of the real tire.

Goodyear 29.5-25 SHRL DL L-3 tire

Inflation Pressure : 206.8 kPa

Figure 15 Young's modulus' vs. normal load on the model

Flow Chart II

CHAPTER III ANALYSIS OF TIRE-TERRAIN INTERACTION USING THE TIRE MODEL

3.1 Terrain Model

Unprepared terrain is usually a much more complex structure than a pneumatic tire. In cross-country operations, a variety of terrains, ranging from desert sand through deep mud to snow, may be encountered, and their properties may vary with time and from one location to the other. In this thesis, the behaviour of the terrain is modelled using a group of semi-empirical equations.

Over the years, many methods have been proposed for characterizing terrain behaviour. Among them, the bevameter technique proposed by Bekker (2) is one of the widely used. It is based on the premise that the terrain properties pertinent to vehicle mobility can best be measured under loading conditions similar to that exerted by an off-road vehicle. A vehicle exerts normal and shear loads on the terrain surface. To simulate these, the bevameter technique comprises two separate sets of tests: one is the pressure-sinkage test and the other is the shear test. The results of the pressure-sinkage test would allow vehicle sinkage and motion resistance due to terrain compaction to be predicted, while those of the shear test would provide a basis for predicting vehicle traction. The pressure-sinkage relationship proposed by Bekker (1) is adopted in this thesis, which is,

(1)

$$P = \left(\frac{k_c}{b} + k_\phi \right) Z^n$$

where P is the pressure beneath a rectangular or circular plate, Z is the sinkage under continuous loading without interruption, b is the smaller dimension of the loading area, or the width of a rectangular plate, n is the exponent of deformation, and k_c and k_f are cohesive and frictional moduli of deformation, respectively. This relationship is only valid for homogeneous terrain. Implicitly the terrain modelled in this thesis is assumed to be homogeneous.

Figure 16 shows a representative pressure-sinkage relationship for a natural terrain. When an element of the terrain is subject to the loading of an oncoming tire, the pressure-sinkage relationship is characterized by the curve OA. As the tire rolls past the element, the load applied to it is released, and the pressure-sinkage relationship follows the line AB. The inclination of the unloading line indicates that there is a very small elastic rebound during unloading. When a second tire rolls over the element, the load is reapplied at B and the pressure-sinkage relationship follows, more or less, the same path as that during unloading (i.e., BA). When the reapplied load exceeds that exerted by the preceding tire, additional sinkage results (such as that along the path AC). When the second tire rolls past the element, the load applied to it is released again and the pressure-sinkage relationship follows line CD. The equation characterizing the unloading and loading of the terrain proposed by Wong et al (23) is expressed by:

$$P = P_u + K_u(Z_u - Z) \quad (2)$$

Figure 16 Representative of pressure-sinkage relationship and response to repetitive loading

where P_u and Z_u are the pressure and sinkage, respectively, when the unloading begins; K_u is the average slope for the unloading-reloading line and may be expressed by (23):

$$K_u = K_o + A_u Z_u \quad (3)$$

where K_o and A_u are parameters, the values of which are obtained from experimental data.

The shear stress-shear displacement relationship obtained from the shear test is used to calculate the shear stress distribution on the contact surface. Shear stress-shear displacement relations may take different forms dependent upon the type of terrain. Only one of the commonly observed forms (1) is used here:

$$S = (c + P \tan \phi)(1 - e^{-j/K}) \quad (4)$$

where S is the shear stress, and P is normal pressure from Eqn. (1) or (2); c and ϕ are the apparent cohesion and angle of shearing resistance of the terrain, respectively; K is the shear deformation modulus; and j is the shear displacement.

The shear displacement j at a point on the contact surface of a deformed tire, defined by the angle θ , may be described as:

$$j = \int_{\theta_0}^{\theta} [r(\theta) \cos \alpha - R(1 - i) \cos(\theta - \alpha)] d\theta \quad (5)$$

where θ_0 is the front contact angle of the tire measured from the vertical; i is the slip of the tire; r is the distance between the tire centre and the point on tire defined by θ ; and

R is the undeformed radius of the tire. Other parameters in Eqn.(5) are shown in Fig.17.

3.2 Tire Performance Prediction

Based on the tire model and the terrain characteristics described above, finite element codes have been developed for predicting the deformed shape and sinkage of the tire, and the normal and shear stress distributions on the tire-terrain interface under a given operating condition.

The normal load on the tire is determined by:

$$W = b \int_{\theta_1}^{\theta_0} r(\theta) [P(\theta) \cos(\theta - \alpha) + S(\theta) \sin(\theta - \alpha)] d\theta \quad (6)$$

where θ_0 and θ_1 are the front and rear contact angles of the tire measured from the vertical, b is the width of the tire, r is the distance from the tire centre to a point on tire circumference.

The tractive effort or thrust can be calculated by integrating the horizontal component of the shear stress acting on the tire in contact with the terrain:

$$F = b \int_{\theta_1}^{\theta_0} r(\theta) S(\theta) \cos(\theta - \alpha) d\theta \quad (7)$$

The external motion resistance (R_m) can be determined by integrating the horizontal component of the normal stress acting on the tire in contact with the terrain:

$$R_m = b \int_{\theta_1}^{\theta_0} r(\theta) P(\theta) \sin(\theta - \alpha) d\theta \quad (8)$$

Figure.17 Illustration of the derivation of shear displacement

The drawbar pull (DBP) is taken as the difference between the tractive effort F and the external motion resistance R_m :

$$DBP = F - R_m \quad (9)$$

In order to evaluate Eqns (6) to (9), b must be determined in advance. In Section 2.2.2 the contact width on hard surface is obtained by dividing the measured contact area by the calculated contact length. Unfortunately, when a pneumatic tire operates on a deformable terrain the contact area will be different from that on hard surface under the same load. In the following analysis, the relationship between the contact width and deflection of the tire model on a deformable terrain is assumed to be identical to that on hard surface. This relationship is shown previously in Fig.14. This relationship is independent of tire inflation pressure, similar to the that between the contact length and deflection of the tire model.

In the following sections, the tire model is used to analyze the performance of both the front and rear tires on deformable terrain. It should be pointed out that front tires run on undisturbed (or virgin) terrain while the rear tires run in the ruts produced by the front tires, if the vehicle is travelling straight and sets of tires have the same track. The analyses of the performance of the front and rear tires are, therefore, treated separately.

In analyzing the performance of tires, FEM solution technique is used. There are many finite element codes available

on market. These codes can perform very sophisticated analysis. The original finite element code adopted in this thesis (24) is capable of performing the analysis of linear plane problems. It was modified and used in Chapter II in deriving the relationship between contact length and deflection. The approach to analyzing tire-terrain interaction is different from the previous incremental-iterative one used in analyzing the geometric relations for the tire model. Therefore, further modifications to the original finite element code are necessary to perform the analysis of the tire-terrain interaction.

3.3 Front Tire Analysis

The contact patch of the front tire is composed of two parts: in the front contact region of the tire the terrain undergoes plastic deformation while in the rear contact region the terrain exhibits elastic rebound. The prediction of steady-state performance of the tire is achieved by dividing the analysis into two phases: symmetric loading and asymmetric loading. For symmetric loading the tire model is only subject to the normal load sinking into the deformable ground without shear stresses acting on the contact patch. In the asymmetric loading phase, the tire model rolls forward with the presence of shear stresses on the interface.

In the analysis the variable used to update the Young's modulus in each step is the resultant normal reaction (Eqn.(6)).

As the tire-terrain interaction is assumed to be a plane

problem, the normal pressure and shear stress on contact patch across the contact width are considered to be uniformly distributed.

3.3.1 Symmetric Loading Condition

Under the symmetric loading condition, only half of the domain is modelled. A step-by-step incremental approach has to be used. The procedure is described as follows:

- A) The coordinate system is set up as described in Section 2.2.1 A).
- B) The terrain upon which the tire model is resting is moved up until one element on the bottom part of the tire model comes into contact with the terrain. Using Bekker's pressure-sinkage relationship, the normal pressure is then calculated and taken as input for this step. The initial contact width is given a very small value, since the contact area and contact length are very small at the beginning. The initial Young's modulus is interpolated according to the current resultant normal reaction using Eqn.(6) where the shear stress term drops off in this case, and is applied to the entire domain. The deformation of the tire model is evaluated for the given inputs using the linear finite element analysis procedure.
- C) After the first calculation the shape of the tire model will be changed. These displacements of the coordinates are updated to form a new configuration for the next step

calculation. The new contact width is then interpolated according to the current deflection and the deflection-contact width relationship obtained previously. By moving the terrain up until two elements are in contact, the normal pressure can be calculated again according to the current configuration. The new value of Young's modulus is interpolated according to the current resultant normal reaction. But the input pressure starting from the second step is not the pressure calculated from Bekker's equation directly. It will be the difference between the current normal pressure and the previous one, because the deformation by the previous pressure has been held.

D) In each step or increment the resultant normal reaction exerted by the terrain is checked with a given normal load. The step C) is repeated until the difference between the resultant normal reaction and the given normal load is less than a prescribed value.

The procedure is illustrated in Flow Chart III.

3.3.2 Asymmetric Loading Condition

Once the tire model is allowed to roll forward, the symmetry in loading condition no longer exists. Shear stress will be present and the elastic rebound of the terrain at the trailing edge of the tire, where unloading occurs, will come into the picture.

Flow Chart III

The transition from the symmetric condition to the asymmetric one is found to be difficult to model. The resultant normal reaction from the terrain will drop suddenly during the transition. This problem will be discussed in detail later.

Basically, the procedure for analyzing the asymmetric condition is more or less the same as that for symmetric one except for the presence of shear stress. The shear stress is calculated by Eqns (4) and (5).

A problem arises from the radius R whether the radius R should be the free-rolling radius of the tire or the deformed radius, which is the distance between the tire centre and the contact point at the bottom.

As it is well known, the slip of a pneumatic tire on hard ground is defined as (1):

$$i = (1 - \frac{V}{R\omega})100\% \quad (10)$$

where V is the translatory speed of the tire centre, ω is the angular speed of the tire, R is the rolling radius of a free-rolling tire.

By looking into the definition of R , it is found the free-rolling radius is not very well defined. It is defined here as the ratio of the circumference of the free-rolling tire to 2π (or 6.28385 to be more specific), which may be neither the undeformed radius nor the deformed radius. It appears that the definition requires further investigation. In the following analysis, R is taken as the radius by the above definition.

From Eqn.(5), it can be seen that r is a function of angle θ . When slip i is small, there could be negative j . Consequently, there will be negative shear stress and possibly negative thrust.

The elastic rebound of the terrain occurs at the unloading zone of the contact patch. The maximum normal pressure from Eqn.(1) should be identical to that from Eqn.(2). While the maximum normal pressure must be the value when the sinkage Z is maximum, its location is not necessarily right under the tire centre. It could be either in front or behind.

The procedure for this part of analysis is as follow:

- A) The inner circumference of the model is still fixed with respect to the reference coordinate system, which is the same as before.
- B) The contact width and Young's modulus are interpolated according to the method described in Section 3.3.1. Normal load, thrust, resistance and DBP are evaluated by Eqns (6) to (9), respectively, and the resultant normal reaction from the terrain is compared with the given load.
- C) The terrain is moved up again until one more element comes into contact with the terrain. The normal pressure and shear stress are calculated according to the current deformed configuration and Eqns (1) to (5). Again, the input stresses are not the stresses themselves, but the differences in stresses, including the difference in shear stress.

D) The linear finite element analysis procedure is used to analyze the tire - terrain interaction for the inputs in step C), and then the deformed configuration is updated.

E) Steps B), C) and D) are repeated until the difference in the resultant normal reaction and the given normal load is less than a prescribed value.

The procedure is illustrated in Flow Chart IV.

It should be noted that there is a tangential deformation of the tire model along the circumference due to the applied shear stress. This is different from the symmetric loading situation.

In the method described above, when the tire model rolls forward, the initial configuration used is from symmetric loading condition. Because the elastic rebound occurs, the contact length at the trailing edge of the tire model is reduced considerably as compared with that when the tire model is in a symmetric condition. Therefore, the transition from a symmetric loading condition to an asymmetric one causes a sudden drop in the resultant normal reaction.

The problem of this lower resultant normal reaction and its inconsistent configuration seems unsolvable as long as this kind of approach is used. To resolve this problem, one of the following methods may be used:

- 1) Starting from the symmetric loading condition, let the tire model sink into the terrain without changing the

Flow Chart IV

configuration. In other words, let the tire model behave like a rigid wheel until the sinkage is deep enough so that the resultant normal reaction reaches the given normal load. Furthermore, the shear stress is only applied at the final step. The shear stress will cause the tire model to deform.

2) After the transition from the symmetric loading condition to the asymmetric one, only the shear stress difference between the current step and the previous one is applied. The calculation is carried out step by step until the resultant normal reaction balances the given normal load. The normal pressure will be used to calculate the resultant normal reaction from the terrain each time.

3.4 Rear Tire Analysis

The procedure for the analysis of the rear tire is similar to that for the front tire. The most difficult situation in this analysis is that there are two layers of elastic rebound, if there is additional sinkage: one is produced by the front tire at the front contact area of the rear tire; the other is due to the rear tire at trailing edge. The rear tire performance is affected by the sinkage and elastic rebound under the front tire.

Since the elastic rebound is usually very small, but the pressure variation during unloading is not small, The mesh has to be very fine to achieve sufficient accuracy of prediction. There are some difficulties in improving the modeling of elastic rebound by reducing the size of the element if the program is to

be run on a microcomputer. On the other hand, the elastic rebounds of the front and rear tires are important to the prediction of the performance of the rear tire, they can not be ignored or simplified. Furthermore, there could be additional sinkage for the rear tire for a given normal load. Thus, although the analysis is similar to that of front tire, it is much more involved.

The amount of the elastic rebound in the rut formed by the front tire is related to its normal load. According to Eqns (1) and (2), the deeper the sinkage or the larger the load, the thinner the elastic rebound layer will be, if the exponent of deformation n is less than 1. If n is larger than 1, the thickness increases with the increase of the sinkage.

Knowing the elastic rebound and sinkage under the front tire, the analysis of rear tire can be carried out in one of the two possible ways:

- I. Similar to the method for analyzing the front tire, the inner circumference of the rear tire model is fixed with respect to the reference coordinate system. The terrain is moved up. The tire model first contacts the elastic layer, and then may penetrate the elastic layer into the plastic terrain without the presence of the shear stress. After having reached an equilibrium of vertical forces, the rear tire model rolls forward with the presence of the shear stress.

However, even with the finest mesh possible on a micro-

computer that was used, numerical instability in the solution process was still encountered, primarily because the terrain stiffness in the unloading and reloading cycle is very high. This instability may become more and more serious as the deflection increases.

II. If we assume that prior to the start of forward motion of the vehicle, both the front and rear tires are in contact with the undisturbed (or virgin) terrain, then the procedure of analyzing the front tire described previously is suitable for the rear tire as well. As soon as the rear tire travels a distance equal to the wheelbase of the vehicle, the rear tire comes into contact with the rut produced by the front tire. The contact length of the rear tire in the front contact region is shortened. Consequently, the resultant normal reaction which is obtained by integrating the normal and shear stresses over the contact patch in the vertical direction (Eqn.(6)) decreases, and additional sinkage may occur.

Comparing the two methods, Method II can avoid the difficulties of Method I described previously, and is, therefore, adopted in this thesis.

3.5 Vehicle Performance Prediction

In predicting the performance of a wheeled vehicle under steady-state operating condition, the dynamic load transfer from the front tire to the rear tire must be taken into account. Load

transfer is dependent on the drawbar pull (DBP), which in turn is related to the thrust and motion resistance developed by the front and rear tires. As described earlier, the thrust and motion resistance are dependent upon the normal load applied on the tire.

Since all the forces mentioned above are inter-dependent, a single variable search method has to be used to obtain a solution. One of these forces is chosen as variable, the value of which is determined based on the condition that it minimizes an objective function. The function selected is the resultant moment of all forces about a given axis. The optimum value of the objective function is zero. In other words, the moments of all the forces acting on the vehicle must be in equilibrium.

To achieve this, the program for front tire analysis was run at different normal loads to obtain corresponding sets of values of the thrust, motion resistance, sinkage and elastic rebound as functions of normal load. The rear tire was treated in the same way.

Assuming that these functions are:

$$\begin{aligned} T_1 &= T_1(N_1); \quad T_2 = T_2(N_2, S_1, E_1) \\ R_1 &= R_1(N_1); \quad R_2 = R_2(N_2, S_1, E_1) \\ S_1 &= S_1(N_1); \quad S_2 = S_2(N_2, S_1, E_1) \\ E_1 &= E_1(N_1); \quad E_2 = E_2(N_2, S_1, E_1) \end{aligned} \quad (11)$$

where 1 and 2 denote front tire and rear tire, respectively. T, R, S, E and N represent thrust, resistance, sinkage, elastic rebound and normal load, respectively. If the vehicle dimensions

are as shown in Fig.18, taking counter-clockwise as positive direction, the moment equilibrium about origin O will be,

$$N1 \times XN1 + T1 \times YT1 - R1 \times YR1 - W \times XL1 + N2 \times XN2 \\ + T2 \times YT2 + R2 \times YR2 + DBP \times YD + \frac{W}{g} \times a \times YL = 0 \quad (12)$$

$$DBP = T1 + T2 - R1 - R2 - \frac{W}{g} \times a \quad (13)$$

Where- $W = N1 + N2$, and W is half of the total vehicle weight; a is the acceleration of the vehicle, g is the acceleration due to gravity. Other parameters used in the Eqn.(12) are illustrated in Fig.18. Since the vehicle is considered travelling straight, only half of the vehicle needs to be modelled. Since only the steady-state performance of the vehicle is of interest, the acceleration of the vehicle a is taken to be zero in the analysis.

For a given $N1$, all other variables can be determined by the equations described above.

Equation (12), with Eqn.(13), may not equal zero with an initial trial value of $N1$. A single variable search method is used to find the value of $N1'$ that minimizes the error to satisfy Eqns (12) and (13).

The procedure for predicting the steady-state performance of a two-axle wheeled vehicle is illustrated in Flow Chart V.

Flow Chart V

**CHAPTER IV. APPLICATIONS OF THE TIRE MODEL TO
THE PREDICTION OF THE PERFORMANCE
OF THE FRONT AND REAR TIRES AND
THE OVERALL PERFORMANCE OF A WHEELED VEHICLE**

4.1 Introduction

Various computer programs have been developed for performing the analyses described in previous chapters. The whole package of finite element codes developed is based on a widely used FEM program capable of solving linear plane strain and plane stress problems (24). Various boundary conditions, such as displacement, distributed stress, point force, temperature and gravity loads, can be applied. The outputs are the displacements of nodes, the reactions on prescribed nodes, stress and strain fields. The element type implemented is eight-node quadrilateral isoparametric element. Because this program can only solve linear elastic plane problems, many modifications have been made to analyze the nonlinear elastic system of the tire-rigid surface contact and the tire-terrain interaction. An outline of the formulation for two dimensional finite element stress analysis using isoparametric elements is given in Appendix A (25).

Because of the limited capability of microcomputers, six major programs for each stage of analysis and many short auxiliary programs are compiled. These programs are:

Program FEM-R is to calculate the relationship between the deflection and corresponding contact length when the tire model

is in contact with rigid flat surface. This program is based on an incremental iterative approach.

Program FEM-M is to determine Young's modulus for a specific tire with a given normal load and a particular inflation pressure. This program includes a single variable search program, a Constrained Single-Variable Golden Section Search Method given in Appendix B. The variable is the Young's modulus. The objective function is the relative error of the resultant normal reaction with respect to the given load.

Programs FEM-S and FEM-F are for the analyses of the front tire under symmetric and asymmetric loading conditions, respectively. The incremental approach is followed in both programs.

Program FEM-FR is for the analysis of the rear tire under asymmetric loading condition.

Program VL is for analyzing the vehicle performance. Subroutine INTERP (Appendix C) is responsible for all the interpolations used in the above programs.

The "Generalized Deflection Chart" provided by the tire manufacturer is usually inaccurate when the normal load is close to zero. This is because when the normal load is small, the deflection, contact length, width and area are consequently small and difficult to measure. Since the application of the tire model on terrain starts at zero load in this approach, the inaccuracy of the initial data will be reflected in the final results.

The example shown in this chapter is for a Goodyear 29.5-25

SHRL DL L-3 tire. The "Generalized Deflection Chart" for this tire produced by **THE GOODYEAR TIRE & RUBBER COMPANY** is shown in Fig.10. The undeflected tire radius is 0.920 m, and the section height is 0.513 m. With these two basic dimensions a finite element mesh is generated as shown in Fig.9. It is quite obvious that the mesh around the contact region should be much finer than the other part since the tire material in contact region undergoes relatively large deformation.

The soil data used are listed in Table I. In the analysis, soft soil is taken as an example.

4.2 Front Tire Performance

The analysis described in Section 3.3 was applied to the prediction of an individual front tire, Goodyear 29.5-25 SHRL DL L-3, with a normal load of 119.5 kN at an inflation pressure of 206.8 kPa, on the soft soil described in Table I.

Figures 19 to 21 show the normal pressure and shear stress distributions against the contact length under the tire at slips 1%, 20% and 40%, respectively. The origin on the figures denotes the contact point right under the wheel axle. The contact length is defined as the length of the contact area. As expected, the rear part of the tire contact patch is shorter than the front part of the tire contact patch. It can be seen that the peak pressure is around the dead centre. It is noticed that when slip is low, for example, at 1%, the shear stress distribution at rear contact region appears negative, which is due to the negative

Table I Soft Soil Data

n	0.8	
K _C	16.45	kN/m ⁿ⁺¹
K _ø	911.4	kN/m ⁿ⁺²
K _O	0	kPa/m
A _u	86000	kPa/m ³
c	3.71	kPa
ø	25.6	Deg.
K	2.12	cm

Figure 19 Normal pressure and shear stress distributions of front tire at 1% slip

Figure 20 Normal pressure and shear stress distributions of front tire at 20% slip

Figure 21 Normal pressure and shear stress distributions of front tire at 40% slip

shear displacement from Eqn. (5).

Figure 22 shows the tire deformation, sinkage, and elastic rebound at 20% slip predicted using the FEM solution technique.

Figure 23 shows the predicted tire performance on soft soil over a range of tire slip. At low slip ranging from 0% to 5%, the thrust and drawbar pull increase quickly with the increase of slip, and then approach to, more or less, constant values at slip higher than 10%. The resistance, on the other hand, varies little over the entire slip range. A complete set of results for the front tire is tabulated in Table II.

4.3 Rear Tire Performance

The analysis described in Section 3.4 was applied to the prediction of an individual rear tire running in the rut formed by the front tire described in the previous section. The rear tire is the same as the front tire, and with the same inflation pressure. However, the normal load it carries, 118.03 kN, is slightly lower than that of the front tire. The normal pressure and shear stress distributions under the rear tire at different slips are shown in Figs. 24 to 26. From these figures, it can be seen that stress distributions are more concentrated than those under the front tire. The stresses change sharply in both the rear and front parts of the tire contact patch. The tire contact length is shorter because the rear tire is travelling in the rut compacted by the front tire.

If additional sinkage occurs, the rear contact patch will be

Figure 22 Illustration of front tire deformation, sinkage and elastic rebound of the terrain

Goodyear 29.5-25 SHRL DL L-3 tire

Figure 23 Front tire performance at a particular tire inflation pressure, a specific slip and a given load

Table II Front Tire Performance for

Goodyear 29.5-25 SHRL DL L-3 Tire

with Inflation Pressure at 206.8 kPa

and Normal Load at 119.5 kN

Unit	Slip					
	1%	3%	5%	10%	20%	40%
Calculated Load (kN)	112.25	114.13	114.72	115.28	115.70	116.05
Resistance (kN)	20.75	21.05	21.11	21.11	21.09	21.10
Resistance/Load (%)	18.49	18.44	18.40	18.31	18.23	18.18
Thrust (kN)	19.22	35.35	43.05	49.45	51.42	52.00
Thrust/Load (%)	17.12	30.97	37.53	42.90	44.44	44.84
DBP (kN)	-1.53	14.30	21.94	28.34	30.33	30.90
DBP/Load (%)	-1.363	12.53	19.12	24.58	26.21	26.63
Sinkage (m)	0.2610	0.2629	0.2633	0.2632	0.2631	0.2631
Deflection (m)	0.1130	0.1149	0.1158	0.1164	0.1165	0.1165
Contact Length (m)	1.006	1.025	1.023	1.022	1.022	1.022
Contact Width (m)	0.5025	0.5052	0.5065	0.5073	0.5075	0.5075
Elastic Rebound (m)	0.0155	0.0150	0.0139	0.0133	0.0132	0.0132

Figure 24 Normal pressure and shear stress distributions of rear tire at 7% slip

Figure 25 Normal pressure and shear stress distributions of rear tire at 20% slip

Figure 26 Normal pressure and shear stress distributions of rear tire at 60% slip

composed of three parts: the portion of the elastic rebound formed by the front tire; the portion formed by the additional sinkage and the part of the elastic rebound by the rear tire. Otherwise, the rear contact patch will be covered only by the elastic layer formed by the front tire.

Figure 27 illustrates the rear tire deformation, additional sinkage, elastic rebound at 20% slip using the FEM solution technique. A complete set of results for the rear tire is tabulated in Table III.

It can be seen in Fig.28 that the thrust becomes positive at slip about 7%, which is much higher than for the front tire. It looks that the entire performance curves have been shifted to the right in comparison with that of the front tire.

4.4 Vehicle Performance

After the calculations of individual tire performance for a certain range of normal loads, an overall performance of a wheeled vehicle can be predicted. A relatively short program VL is compiled, which will predict the sinkages, deflections, and stress distributions at each tire and the overall performance of the vehicle under steady-state operating condition.

For a four-wheel-drive vehicle, the vehicle will operate under optimum condition if both front and rear tires operate at the same slip. This program, however, can predict the performance even when the front and rear tires operate at different slips. Figures 29 and 30 show the deformations of both front tire and

Figure 27 Illustration of rear tire deformation, sinkage and elastic rebound of the terrain

Table III Rear Tire Performance for

Goodyear 29.5-25 SHRL DL L-3 Tire

with Inflation Pressure at 206.8 kPa

and Normal Load at 118.03 kN

	Unit	Slip					
		7%	10%	15%	20%	40%	60%
Calculated Load	(kN)	114.62	116.07	117.26	118.03	119.32	119.76
Resistance	(kN)	15.89	16.11	16.10	16.09	16.09	16.09
Resistance/Load	(%)	13.86	13.88	13.73	13.63	13.48	13.43
Thrust	(kN)	11.14	28.93	41.91	47.08	52.41	53.67
Thrust/Load	(%)	9.72	24.92	35.74	39.89	43.95	44.81
DBP	(kN)	-4.75	12.82	25.81	30.99	36.32	37.58
DBP/Load	(%)	-4.144	11.05	22.01	26.26	30.44	31.38
Sinkage	(m)	0.3396	0.3396	0.3396	0.3396	0.3397	0.3397
Deflection	(m)	0.1093	0.1093	0.1093	0.1093	0.1093	0.1093
Contact Length	(m)	0.7096	0.7096	0.7096	0.7096	0.7096	0.7096
Contact Width	(m)	0.4972	0.4972	0.4972	0.4972	0.4972	0.4972
Elastic Rebound	(m)	0.0133	0.0133	0.0133	0.0133	0.0133	0.0133

Figure 28 Rear tire performance at a particular tire inflation pressure, a specific slip and a given load

Goodyear 29.5-25 SHAL DL L-3 tire

Inflation pressure = 206.8 kPa

Normal load = 222 kN

Load transfer = 12.28 kN

Slip = 10 %

Figure 29 Deformations, sinkages and max. normal pressures and shear stresses of the front and rear tires of a vehicle at 10% slip

Goodyear 29.5-25 SHAL DL L-3 tire

Inflation pressure = 206.8 kPa

Normal load = 222 kN

Load transfer = 18.66 kN

Slip = 40 %

Figure 30 Deformations, sinkages and max. normal pressures and shear stresses of the front and rear tires of a vehicle at 40% slip

rear tires as well as the stress distributions under optimum condition at slips 10% and 40%, respectively. It can be seen that rear tire stresses are narrower in range and higher in amplitude than those of front tire, which are mainly due to the dynamic load transfer. Additional sinkage occurs at rear tire. The percentage of load transfer increases with the increase of slip.

In the calculation the moment residual (FUNMI in the program) of Eqn.(11) is less than 10^{-4} N m. A complete set of results of the vehicle performance on soft soil at different slips is given in Table IV - VI.

Since no pertinent experiments were carried out to validate this model, a comparison with the results obtained using another simulation model was made.

In 1986, Wong et al developed a computer simulation model to predict the wheeled vehicle performance on unprepared terrain (3). The model is a modification and improvement of Bekker's model.

The Wong model included the analyses of both front and rear tires as well as the vehicle. The effects of tire lugs on the performance were examined. Repetitive loading and unloading were taken into consideration. The criterion of determining whether the tire behaves like a rigid or elastic wheel in Wong's model are different from that proposed by Bekker (2).

In Bekker's model the sinkages for both rigid operating mode and elastic mode are calculated according to tire inflation pressure, the dimensions of the tire and soil parameters. Then

2 of/de 2

Microlin
MICROLIN
MICROLIN

Table IV Vehicle Performance at 10% Slip

Goodyear 29.5-25 SHRL DL L-3 Tires

Inflation Pressure: 206.8 kPa

Normal Load: 222 kN

	Unit	Front Tire	Rear Tire
Normal load	(N)	104997.2690	117187.7310
Thrust	(N)	42198.6162	29674.8492
Resistance	(N)	18955.7129	16515.1508
Contact length	(m)	0.99227	0.71528
Deflection	(m)	0.10882	0.11007
Contact width	(m)	0.49658	0.49828
Sinkage	(m)	0.24091	0.33937
Elastic rebound	(m)	0.01605	0.01334

Table V Vehicle Performance at 20% Slip

Goodyear 29.5-25 SHRL DL L-3 Tires

Inflation Pressure: 206.8 kPa

Normal Load: 222 kN

	Unit	Front Tire	Rear Tire
Normal load	(N)	98196.6494	123988.3506
Thrust	(N)	43868.5490	49818.2023
Resistance	(N)	16773.4411	18254.7940
Contact length	(m)	0.95685	0.74072
Deflection	(m)	0.10288	0.11414
Contact width	(m)	0.48895	0.50412
Sinkage	(m)	0.22764	0.33883
Elastic rebound	(m)	0.01529	0.01348

Table VI Vehicle Performance at 40% Slip

Goodyear*29.5-25 SHRL DL L-3 Tires

Inflation Pressure: 206.8 kPa

Normal Load: 222 kN

	Unit	Front Tire	Rear Tire
Normal load	(N)	97358.0543	124826.9457
Thrust	(N)	44004.1457	54657.9743
Resistance	(N)	16377.9286	18063.2795
Contact length	(m)	0.94997	0.73700
Deflection	(m)	0.10184	0.11356
Contact width	(m)	0.48766	0.50329
Sinkage	(m)	0.22572	0.33884
Elastic rebound	(m)	0.01490	0.01345

the two sinkages are compared, the lower one is used to determine the operating mode. Also, in Bekker's model there is no smooth transition between the two operating modes.

In Wong's model the pressure at the maximum sinkage of the undeformed tire is compared with the critical pressure determined by the inflation pressure and carcass stiffness. If the pressure is less than that critical pressure, the tire behaves like a rigid wheel; if the pressure exceeds the critical pressure, the tire will deform. It is obvious that there is a transition point when maximum pressure is equal to that critical pressure. Therefore, the change of operating mode is smooth.

The effect of shear stress on the tire performance was taken into account in Wong's model. A path-dependent shear stress formula was developed in order to obtain more accurate prediction of shear stress on the contact patch. Furthermore, the effects of tire lugs on the tractive performance have been considered.

An iterative procedure was used in Wong's model to obtain a solution for steady-state operation. The equilibrium of vertical forces serves as a condition for achieving a solution for individual tire. When analyzing vehicle performance, dynamic load transfer between the front and rear tires has been taken into consideration. The moment equilibrium is an additional condition that has to be satisfied to arrive at a solution. The tractive performance of the vehicle is obtained by integrating the normal pressure and shear stress over the contact areas of the tires.

Wong's simulation model has been used to predict the

performance of certain tires and wheeled vehicles. The predicted performance has been compared with limited available experimental data, and reasonable agreement has been achieved for the specified cases.

The performance of a four-wheel-drive off-road vehicle with a gross weight of 444 kN and equipped with Goodyear 29.5-25 SHRL DL L-3 tires, was predicted by FEM solution technique described previously and by the Wong's model. The results are shown in Fig.31. It can be seen that the motion resistance predicted by both models are very close and almost undistinguishable, and that the thrusts are also very close at slips higher than about 16%. In the low slip range there is noticeable difference in the thrust predicted by the two methods.

Experimental work is planned to evaluate the FEM solution technique described in this thesis and to provide a firm basis for further improvement and development.

Goodyear 29.5-25 SHRL DL L-3 tire

Figure 31 Comparison of vehicle performances by FEM and Wong's model

CHAPTER V CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The following conclusions can be drawn as a result of this study:

1. The finite element method is applicable in the study of tire-terrain interaction, but it must be applied in an appropriate manner. For predicting the tractive performance, the central issue is to establish a mathematical model with which the normal and shear stress distributions on the tire-terrain interface can be realistically predicted in terms of its design parameters and terrain characteristics.

2. Using the finite element method and the measured tire characteristics described in the "Generalized Deflection Chart" provided by the tire manufacturer, a realistic tire model for the study of tire-terrain interaction can be established. It is shown that this tire model can provide a more accurate prediction of contact length-deflection relationship than existing methods.

3. It is shown that using the tire model developed and the terrain characteristics defined by the bevameter technique, the geometry of the contact patch and the stress distributions on the tire-terrain interface can be more realistically predicted than using other methods developed so far.

4. It is believed that the method of approach to the analysis of tire-terrain interaction described in this thesis can provide a firm basis upon which a comprehensive and accurate

model for tire performance can be evolved.

5.2 Recommendations

1. While the method of approach to the analysis of tire-terrain interaction described in the thesis is shown to contain a number of improvement over existing methods, its experimental validation is required. This is currently underway.

2. The tire-terrain interaction is currently treated as a two-dimensional problem. As mentioned previously, the contact width of the tire in reality varies with deflection. This indicates that the problem should be treated as a three-dimensional one.

3. Certain numerical procedures should be improved to expedite the solution process and to minimize the possibility of numerical instability. Particular attention should be paid to the numerical procedure used in the analysis of the rear tire which is in contact with the rut having a high value of terrain stiffness.

REFERENCES

1. Wong, J.Y., "Theory of Ground Vehicles", John Wiley, New York, 1978.
2. Bekker, M.G., "Introduction to Terrain-Vehicle Systems", University of Michigan, 1969.
3. Wong, J.Y. and Preston-Thomas, J., "Développement of Vehicle Performance Predicting Software", Report Prepared for the National Research Council of Canada, under Contract Serial No. OS285-00054, March, 1986.
4. Fujimoto, Y., "Performance of Elastic Wheel on Yielding Cohesive Soils", Journal of Terramechanics, Vol.14, No.4, pp.191 to 210, 1977.
5. Baladi, G.Y., and Rohani, B., "Development of a Soil-wheel Interaction Model", Proceedings of the Eighth International Conference of ISTVS, Cambridge, U.K., 1984.
6. Perumpral, J.V., Liljedahl, J.B. and Perloff, W.H., "A Numerical Method for Predicting the Stress Distribution and Soil Deformation under a Tractor Wheel", Journal of Terramechanics, Vol.8, No.1, pp.9 to 20, 1971.
7. Yong, R.N., Fattah, E.A. and Bounsinsuk, P., "Analysis and Prediction of Tire-Soil Interaction and Performance Using Finite Elements", Journal of Terramechanics, Vol.15, No.1, pp.43 to 63, 1978.

8. Oida, A., "Analysis of Rheological Deformation of Soil by Means of Finite Element Method", Journal of Terramechanics, Vol.21, No.3, pp.237 to 251, 1984.
9. Chang, T.Y., Saleeb, A.F. and Shyu, S.C., "Finite Element Solutions of Two-dimensional Contact Problems Based on a Consistent Mixed Formulation", Computers and Structures, Vol.27, No.4, pp.445 to 466, 1987.
10. Oden, J.T., and Lin, T.L., "On the General Rolling Contact Problem for Finite Deformation of A Viscoelastic Cylinder," Computer Method in Applied Mechanics and Engineering 57, pp.297 to 367, 1986.
11. Chen, W.H., and Tsai, P., "Finite Element Analysis of Elastodynamics Sliding Contact Problems with Friction", Computers and Structures, Vol.22, No.6, pp.925 to 936, 1986.
12. Whicker, D. and Rohde, S.M., "Modeling Tire Deformation for Power Loss Calculation", SAE Trans. 810161, 1981.
13. Segalman, D.J., "Modeling Tire Energy Dissipation for Power Loss Calculations", SAE Trans. 810162, 1981.
14. Browne, A.L., and Arambages, A., "Modeling Tire Thermal State of Tires for Power Loss Calculations", SAE Trans. 810163, 1981.
15. Chaudhary, A.B. and Bath, K.J., "A Solution Method for Static and Dynamic Analysis of Three-dimensional Contact Problems with Friction", Computers and Structures, Vol.24, No.6, pp.855 to 873, 1986.

16. Ridha, R.A., Satyamurthy, K., and Hirschfelt, L.R., "Finite Element Modelling of A Homogeneous Pneumatic Tire Subjected to Footprint Loadings", Tire Science and Technology, Vol.13, No.2, pp.91 to 110, April-June 1985.
17. Sarkar, K. Kvyon, Y.D., and Prevorsek, D.C., "A New Approach for the Thermomechanical Analysis of Tires by Finite Element Method", Tire Science and Technology, TSTCA, Vol.15, No.4, pp.161 to 275, October-December 1987.
18. Rothert, H., and Gall, R., "On the Three Dimensional Computation of Steel-Belted Tires", Tire Science and Technology, TSTCA, Vol.14, No.2, pp.116 to 124, April-June 1986.
19. Nakajima, Y. and Padovan, J., "Numerical Simulation of Tire Sliding Events Involving Impacts with Holes and Bumps", Tire Science and Technology, TSTCA, Vol.14, No.2, pp.125 to 136 April-June 1986.
20. Miller, C., Popper, P., Gilmour, P.W., and Schaffers, W.J., "Textile Mechanics Method of a Pneumatic Tire", Tire Science and Technology, TSTCA, Vol.13, No.4, pp.187 to 226, October-December, 1985.
21. Tseng, Nien-Tsu, "Finite Element Analysis of Freely Rolling Tires", Tire Science and Technology, TSTCA, Vol.15, No.2, pp.134 to 158, April-June, 1987.
22. Stechschute, R.A. and Luchini, J.R., "A Laminated Composite Solid Element and Its Application to Tire Analysis", Tire Science and Technology, TSTCA, Vol.15, No.1, pp.42 to 57,

January-March 1987.

23. Wong, J.Y., Garber, M. and Preston-Thomas, J., "Theoretical Prediction and Experimental Substantiation of the Ground Pressure Distribution and Tractive Performance of Tracked Vehicles", Proceedings of the Institution of Mechanical Engineers, Part D, No.4, 1984.
24. Hinton, E., and Owen, D.R.J., "Finite Element Programming", Department of Civil Engineering, Swansea, U.K., Academic Press Inc. (London) LTD., 1977.
25. Tan, C.L., "Lecture Notes for Course 88.565: Finite Element Analysis I", Department of Mechanical and Aeronautical Engineering, Carleton University, Ottawa, Canada, 1986.
26. Bekker, M.G., "Theory of Load Locomotion", University of Michigan, 1956.
27. Wong, J.Y., "On the Study of Wheel-Soil Interaction", Journal of Terramechanics, Vol.221, No.2, pp.117 to 131, 1984.
28. Yong, R.N., Boonsinsuk, P., and Fattah, E.A., "Prediction of Tire Performance on Soils Related to Carcass Stiffness and Contact Areas", Proceedings of the Sixth International Conference of ISTVS, Vol. II, Vienna, Austria, 1978.
29. Krick, G., "Radial and shear Stress Distribution under Rigid Wheels and Pneumatic Tires Operating on Yielding Soils with Consideration of Tire Deformation", Journal of Terramechanics, Vol.6, No.3, pp.73-98, 1969.

30. Klaus-Jurgen Bathe, "Finite Element Procedures in Engineering Analysis", Prentice-Hall, Inc., 1982.
 31. Bhatt, G., Touzot, G. and Cantin, G., "The Finite Element Method Displayed", John Wiley & Sons, 1984.
 32. Timoshenko, S.P. and Goodier, J.N., "Theory of Elasticity", McGraw-Hill Book Company, 3rd edition 1970.
-

**Appendix A. OUTLINE OF THE FORMULATION OF TWO-DIMENSIONAL
FINITE ELEMENT STRESS ANALYSIS USING 8-NODE
QUADRILATERAL ISOPARAMETRIC ELEMENTS**

In many cases, three-dimensional body can be treated as two-dimensional problem. Although it is approximate, it provides a meaningful understanding of the problem without losing the generality.

There are two forms of two-dimensional approximations commonly used, plane stress and plane strain.

Definition of plane stress; $\sigma_{xx} = \sigma_{yy} = \sigma_{zz} = 0$

Definition of plane strain; $\epsilon_{xx} = \epsilon_{yy} = \epsilon_{zz} = 0$

Hence plane analysis directly involve ϵ_{xx} , ϵ_{yy} and ϵ_{xy} only.

Procedure:

STEP I. Identify the problem in question, decide whether plane stress or plane strain should be used, discretize the solution domain into elements, define nodes and elements.

If an 8-node quadrilateral isoparametric element is used, and displacement is the primary variable, the nodal displacement vector for each element,

$$[\delta]_e = \begin{pmatrix} [\delta_1] \\ [\delta_2] \\ \vdots \\ [\delta_8] \end{pmatrix} \quad (A-1)$$

$$[\delta_i] = \begin{pmatrix} u_i \\ v_i \end{pmatrix}$$

where u is X-displacement, v is Y-displacement.

The element nodal force vector,

$$[R]_e = \begin{pmatrix} [R_1] \\ [R_2] \\ \vdots \\ [R_8] \end{pmatrix} \quad (A-2)$$

$$[R_i] = \begin{pmatrix} U_i \\ V_i \end{pmatrix}$$

STEP II Define the variation of displacements in terms of shape functions N_i and the nodal values u_i and v_i ,

$$u(\xi, \eta) = \sum_{i=1}^8 N_i(\xi, \eta) u_i$$

$$v(\xi, \eta) = \sum_{i=1}^8 N_i(\xi, \eta) v_i \quad (A-3)$$

where ξ , η are variables in the reference system.

N_i are listed below;

$$N_1(\xi, \eta) = -\frac{1}{4}(1 - \xi)(1 - \eta)(1 + \xi + \eta)$$

$$N_2(\xi, \eta) = \frac{1}{2}(1 - \xi^2)(1 - \eta)$$

$$N_3(\xi, \eta) = -\frac{1}{4}(1 + \xi)(1 - \eta)(1 - \xi + \eta)$$

$$N_4(\xi, \eta) = \frac{1}{2}(1 + \xi)(1 - \eta^2)$$

$$N_5(\xi, \eta) = -\frac{1}{4}(1 + \xi)(1 + \eta)(1 - \xi - \eta)$$

$$N_6(\xi, \eta) = \frac{1}{2}(1 - \xi^2)(1 + \eta)$$

$$N_7(\xi, \eta) = -\frac{1}{4}(1 - \xi)(1 + \eta)(1 + \xi - \eta)$$

$$N_8(\xi, \eta) = \frac{1}{2}(1 - \xi)(1 - \eta^2)$$

For function values,

$$[\delta(\xi, \eta)] = \sum_{i=1}^8 [N(\xi, \eta) \{\delta_i\}] \quad (\text{A-4})$$

STEP II. Strain-displacement relations

$$[c] = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} = \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \end{pmatrix} \quad (\text{A-5})$$

For derivative transformation,

$$\begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} = [J] \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} \quad (\text{A-6})$$

where $[J] = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{pmatrix}$ is Jacobian matrix.

From Eqn. (A-6), we can find,

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{pmatrix} = [J]^{-1} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{pmatrix} \quad (\text{A-7})$$

Therefore,

$$[B_i] = \begin{pmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{pmatrix} = \begin{pmatrix} \frac{\partial N_i}{\partial \xi} & 0 \\ 0 & \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \eta} & \frac{\partial N_i}{\partial \xi} \end{pmatrix} [J]^{-1} \quad (\text{A-8})$$

$$[c] = [B][\delta] = \sum_{i=1}^8 [B_i][\delta_i]$$

STEP III. Stress-strain relations (Hooke's law)

$$\begin{aligned}
\epsilon_{xx} &= \frac{1}{E}(\sigma_{xx} - \gamma(\sigma_{yy} + \sigma_{zz})) + \alpha \Delta T \\
\epsilon_{yy} &= \frac{1}{E}(\sigma_{yy} - \gamma(\sigma_{xx} + \sigma_{zz})) + \alpha \Delta T \\
\epsilon_{zz} &= \frac{1}{E}(\sigma_{zz} - \gamma(\sigma_{yy} + \sigma_{xx})) + \alpha \Delta T \\
\epsilon_{xy} &= \frac{2(1+\gamma)}{E} \sigma_{xy} \alpha
\end{aligned} \tag{A-9}$$

where E is Young's modulus, γ is Poisson's ratio, ΔT is the change in temperature, α is the coefficient of thermal expansion.

(i) plane stress; $\sigma_{zz} = 0$

(ii) plane strain; $\epsilon_{zz} = 0$

therefore $\sigma_{zz} = \gamma(\sigma_{xx} + \sigma_{yy}) - E\alpha\Delta T$

Hence,

$$[e] = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \end{pmatrix} = \frac{1}{E^*} \begin{pmatrix} 1 & -\gamma^* & 0 \\ -\gamma^* & 1 & 0 \\ 0 & 0 & 2(1+\gamma^*) \end{pmatrix} \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{pmatrix} + \alpha^* \Delta T \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \tag{A-10}$$

where (i) plane stress; $E^* = E$; $\gamma^* = \gamma$; $\alpha^* = \alpha$

where (ii) plane strain; $E^* = \frac{E}{1-\gamma^2}$; $\gamma^* = \frac{\gamma}{1-\gamma}$; $\alpha^* = (1+\gamma)\alpha$

Define

$$[e_T] = \alpha^* \Delta T \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \quad [\sigma] = \begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{pmatrix}$$

Inserting Eqn. (A-10)

$$[\sigma] = [D]\{[e] - [e_T]\} \tag{A-11}$$

$$[D] = \frac{E^*}{1-\gamma^{*2}} \begin{pmatrix} 1 & \gamma^* & 0 \\ \gamma^* & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\gamma^*) \end{pmatrix}$$

therefore

$$[\sigma] = [D][B][\delta]_e - [D][\epsilon_T] \quad (A-12)$$

In a special case, when $\gamma = 0.5$, i.e., the volume of the

body does not change, $\frac{E^*}{1-\gamma^2} = \frac{E(1-\gamma)}{(1+\gamma)(1-2\gamma)} \rightarrow \infty$

To overcome this problem,

- (i) employing γ slightly less than 0.5, e.g., 0.499
- (ii) alternative formulation of FEM

STEP IV. Replace internal stress $[\sigma]$ in each element by equivalent nodal forces $[R]_e$ and thus relate $[R]_e$ to $[\delta]_e$.

By applying virtual work principle, consider an element acted upon by nodal loads $[R]_e$ and body forces $[G^*]$ which result in an equivalent stress field $[\sigma]$; let the element be subjected to arbitrary virtual nodal displacement $[\delta^*]_e$ which result in compatible internal displacement $[\delta]_e$ and strain distribution $[\epsilon]_e$,

The virtual work equation

$$[\delta^*]_e^T [R]_e + \int_{\Delta} [\delta^*]_e^T [G^*] dx dy = \int_{\Delta} [\epsilon^*]_e^T [\sigma] dx dy \quad (A-13)$$

Because $[\delta]_e^T = [[N][\delta^*]_e]^T = [\delta^*]_e^T [N]^T$; $[\epsilon]_e^T = [\delta^*]_e^T [B]^T$

therefore, Eqn. (A-13) becomes,

$$[\delta^*]_e^T ([R]_e + \int_{\Delta} [N]^T [G^*] dx dy) = [\delta^*]_e^T \int_{\Delta} [B]^T [\sigma] dx dy \quad (A-14)$$

Cancel out $\{\delta\}_e^T$ at both sides of Eqn.(A-14),

$$[R]_e + \int_{\Delta} [N]^T [G^e] dx dy = \int_{\Delta} [B]^T [\sigma] dx dy \quad (A-15)$$

Substitute Eqn.(A-12) into Eqn.(A-15)

$$[R]_e + \int_{\Delta} [N]^T [G^e] dx dy = \int_{\Delta} [B]^T [D] [B] dx dy \{\delta\}_e - \int_{\Delta} [B]^T [D] \{e_T\} dx dy \quad (A-16)$$

Define element stiffness matrix

$$[K]_e = \int_{\Delta} [B]^T [D] [B] dx dy$$

body forces

$$[G]_e = \int_{\Delta} [N]^T [G^e] dx dy$$

thermal forces

$$[\theta]_e = \int_{\Delta} [B]^T [D] \{e_T\} dx dy$$

Therefore

$$[K]_e \{\delta\}_e = [R]_e + [\theta]_e + [G]_e \quad (A-17)$$

The above matrices can also be expressed as,

$$[K]_e = \int_{-1}^{+1} \int_{-1}^{+1} [B]^T [D] [B] \det J d\xi d\eta$$

$$[G]_e = \int_{-1}^{+1} \int_{-1}^{+1} [N]^T [G^e] \det J d\xi d\eta$$

$$[\theta]_e = \int_{-1}^{+1} \int_{-1}^{+1} [B]^T [D] \{e_T\} \det J d\xi d\eta$$

Now numerical integrations are necessary, e.g., using Gaussian quadrature,

$$\int_{-1}^{+1} \int_{-1}^{+1} f(\xi, \eta) = \sum_{j=1}^n \sum_{k=1}^n w_j w_k f(\xi_j, \eta_k) \quad (A-18)$$

STEP V. Assemble overall stiffness matrices

$$[K][\delta] = [F] + [G] + [\Theta] \quad (\text{A-19})$$

where

$$[F] = \sum_e [R]_e; [G] = \sum_e [G]_e; [\Theta] = \sum_e [\theta]_e \quad (\text{A-20})$$

STEP VI. Boundary conditions

A) Constraints on displacements

The coefficients of row and column in the overall stiffness matrix corresponding to the prescribed degree of freedom are set to zero, except the diagonal term which is set to unity, and the second member is reduced according to

$$F^* = F_i - k_{ij}\delta_j \quad (\text{A-21})$$

B) Loading

i) point loading: applied explicitly in equation set.

ii) distributed loading: convert the distributed loads acting on the element sides to equivalent nodal forces in the global co-ordinates system.

In terms of shape function and nodal point values, the load distributed on element side can be expressed as:

$$\begin{pmatrix} P_n \\ P_t \end{pmatrix} = \sum_{i=1}^3 N_i \begin{pmatrix} (P_n)_i \\ (P_t)_i \end{pmatrix} \quad (\text{A-22})$$

Consider distributed loading

$$dP_x = P_t \cos \beta ds + P_n \sin \beta ds = P_t dx + P_n dy$$

$$dP_y = P_t \sin \beta ds - P_n \cos \beta ds = P_t dy - P_n dx$$

while

$$dP_x = (P_t \frac{\partial x}{\partial \eta} + P_n \frac{\partial y}{\partial \eta}) d\eta$$

$$dP_y = (P_t \frac{\partial y}{\partial \eta} - P_n \frac{\partial x}{\partial \eta}) d\eta$$

by using the virtual work principle,

$$(P_x)_i = \int_S N_i (P_t \frac{\partial x}{\partial \eta} + P_n \frac{\partial y}{\partial \eta}) d\eta$$

$$(P_y)_i = \int_S N_i (P_t \frac{\partial y}{\partial \eta} - P_n \frac{\partial x}{\partial \eta}) d\eta$$

(A-23)

For loads on other sides, similar expressions can be obtained.

iii) Gravity loading

g - gravitational acceleration; ρ - density; gravity force acting on the element volume

$$\rho g dv = dG \quad (A-24)$$

Therefore, components in the X and Y directions are,

$$dG_x = \rho g dv \sin \beta$$

$$dG_y = \rho g dv \cos \beta$$

(A-25)

Applying the virtual work principle again to convert these

forces to equivalent nodal forces,

$$(P_x)_i = \int_{\Delta} N_i \rho g \sin \beta dx dy$$

(A-26)

$$(P_y)_i = - \int_{\Delta} N_i \rho g \cos \beta dx dy$$

iv) Thermal loading

It is needed to consider the case of plane stress and that of plane strain separately,

plane stress; $[e_T] = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} = \begin{pmatrix} \alpha T \\ \alpha T \\ 0 \end{pmatrix} \quad [\sigma_{xx}]_T = 0$

(A-27)

plane strain; $[e_T] = \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{pmatrix} = \begin{pmatrix} (1+\gamma)\alpha T \\ (1+\gamma)\alpha T \\ 0 \end{pmatrix}$

$$[\epsilon_{xx}]_T = 0$$

$$[\sigma_{xx}] = -E\alpha T$$

Appendix B CONSTRAINED SINGLE-VARIABLE GOLDEN SECTION SEARCH METHOD

Assume function $F(x)$ in (x_1, x_2) is unimodal,

The points are chosen such that

$$\checkmark \quad \frac{x_4 - x_1}{x_2 - x_1} = \frac{x_2 - x_3}{x_2 - x_1} = \lambda \quad \frac{x_3 - x_1}{x_4 - x_1} = \frac{x_4 - x_3}{x_4 - x_1} = \lambda$$

So, after functional evaluations at x_1, x_2, x_3 and x_4 , the x_2 (shown in the figure) is discarded. The region is reduced, but the relation remains, x_3 is the point needed in (x_1, x_4) ..

From the above two equations,

$$x_4 - x_1 = \lambda(x_2 - x_1) \quad x_3 - x_1 = \lambda(x_4 - x_1)$$

Therefore $x_3 - x_1 = \lambda^2(x_2 - x_1)$

meanwhile $x_3 - x_1 = (1 - \lambda)(x_2 - x_1)$

then

$$1 - \lambda - \lambda^2 = 0 \quad \lambda = \frac{\sqrt{5} - 1}{2}$$

Appendix C THREE-POINT LAGRANGIAN INTERPOLATION

In this program, for the given N points, the nearest three-points to the interpolating value are used. The method can be expressed as follows,

$$y(x) = \sum_{k=1}^{i+1} \left(\prod_{\substack{j=i \\ j \neq k}}^{i+2} \frac{x - x_j}{x_k - x_j} \right) y_k$$

where $x_1, x_2, \dots, x_n$ are the given N points, and in the order of $x_1 < x_2 < \dots < x_n$. $y_1, y_2, \dots, y_n$ are corresponding function values.

And,

$$i = \begin{cases} 1 & \text{if } x \leq x_2 \\ s & \text{if } x_s < x \leq x_{s+1}, x - x_s \geq x_{s+1} - x, s = 2, 3, \dots, n-2 \\ s-1 & \text{if } x_s < x \leq x_{s+1}, x - x_s < x_{s+1} - x, s = 2, 3, \dots, n-2 \\ n-2 & \text{if } x > x_{n-1} \end{cases}$$

Appendix D COMPUTER CODES

Computer programs listed in this appendix are:

FEM-R: This program is designed to derive the relationship between the contact length and the deflection of the tire model.

FEM-M: This program is designed to derive the apparent Young' modulus of the tire model.

FEM-S: This program is designed to do the analysis of front and rear tires on terrain under symmetric loading.

FEM-F: This program is designed to do the analysis of tire model on terrain under asymmetric loading.

VL : This program is designed to predict the overall performance of a wheeled vehicle.

Only program FEM-R is complete. Subroutines, which are used in FEM-R, are not repeated in other programs for simplicity.

```

C *****
C *   PROGRAM FEM-R                                     *
C *   This program is to derive the relationship between the contact *   108
C *   length and the deflection of the tire model                *
C *   SUBROUTINES included:                                     *
C *       INPUT: CHECK1: ECHO;                                  *
C *               NODEXY, GAUSSQ,                              *
C *               CHECK2: ECHO;                                  *
C *       LOAD;                                                 *
C *       STIFPS: MODPS, SFR2, JACOB2, BMATPS, DBE;             *
C *       LOADPS: MODPS, SFR2, JACOB2;                          *
C *       FRONT;                                                *
C *       REINP;                                                *
C *****
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TITLE(12)
      COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&      NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&      NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&      LNODS(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&      STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&      DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&      SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&      NEROR(24)
      COMMON /STORE/ NSTOR(99),NNN,NEE,ACCU,ADD,CLEN(99),DEFL(99),DMAX

C
      WRITE(*,800)
800  FORMAT(/'   INPUT FILE NAME - '/')
      OPEN(5,FILE='ORIGIN.DAT',ACCESS='SEQUENTIAL',STATUS='OLD')
      READ (5,900) NPROB
900  FORMAT(I5)

C
C   Open output file as unit 4
C
      WRITE(*,802)
802  FORMAT(/'   OUTPUT FILE NAME - '/')
      OPEN(4,FILE='OUTPUT',ACCESS='SEQUENTIAL',STATUS='NEW')

C
      WRITE(4,905) NPROB
905  FORMAT(1H0,5X,'TOTAL NUMBER OF PROBLEMS = ',I5)
      DO 20 IPOB=1,NPROB
      READ (5,910) TITLE
910  FORMAT (12A4)
      WRITE(4,915) IPOB,TITLE
915  FORMAT (//////,6X,'PROBLEM NO. ',I3,10X,12A4)
      WRITE(*,820)
820  FORMAT(/'   READING INPUT DATA   '/')
      CALL INPUT
      OPEN(12,FILE='INPUT0',STATUS='OLD')
      READ(12,*) NITER,NEE,ACCU,PROPS(1,2),ADD,DMAX
      CLOSE(12,STATUS='KEEP')
      ITER=0
      KEDGE=NVFIX+NEE-1
      KOCK=0
1111 ITER=ITER+1
      CALL LOAD(ITER)
      OPEN(7,FILE='SCRTH7.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',

```

```

OPEN(8,FILE='SCRTH8.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
OPEN(9,FILE='SCRTH9.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
OPEN(10,FILE='SCRTH1.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
WRITE(*,822)
822 FORMAT(/' CALCULATING ELEMENT STIFFNESS MATRICES '/')
CALL STIFPS
ICASE=1
WRITE(*,824)
824 FORMAT(/' CALCULATING LOADING VECTOR '/')
CALL LOADPS(ITER)
WRITE(*,826)
826 FORMAT(/' ASSEMBLING AND SOLVING EQUATIONS BY FRONTAL SOLVER/')
CALL FRONT
CALL REINP

C
C Close files
C
CLOSE(7,STATUS='DELETE')
CLOSE(8,STATUS='DELETE')
CLOSE(9,STATUS='DELETE')
CLOSE(10,STATUS='DELETE')
IF((NVFIX-KEDGE).EQ.1) THEN
KOOK=KOOK+1
ACCU=ACCU+ADD
IF(ACCU.GT.1.D-3) ACCU=1.D-3
CLEN(KOOK)=2.D0*COORD(NSTOR(NVFIX-MNN+1),1)
DEFL(KOOK)=COORD(9,2)
KEDGE=NVFIX
NEEE=NVFIX-MNN+1
WRITE(*,91) KOOK,ITER
91 FORMAT(3X,'No. of POINT',I5,5X,'No. of ITERATION',I5)
IF(DEFL(KOOK).GT.DMAX) GO TO 2001
ELSE
GO TO 1115
ENDIF
1115 CONTINUE
IF (KOOK.EQ.NITER) GO TO 2001
CLOSE(4,STATUS='DELETE')
OPEN(4,FILE='OUTPUT',ACCESS='SEQUENTIAL',STATUS='NEW')
GO TO 1111
2001 CONTINUE
OPEN(14,FILE='OUTC',STATUS='NEW')
DO 1107 I=1,KOOK
WRITE(14,1108) DEFL(I),CLEN(I)
1107 CONTINUE
1108 FORMAT(1X,2F11.8)
WRITE(14,*) 'No. of FIXED POINT',NEEE
WRITE(14,*) 'No. of ITERATION',ITER
CLOSE(14,STATUS='KEEP')

C
CLOSE(4,STATUS='DELETE')
CLOSE(5,STATUS='KEEP')
20 CONTINUE
STOP
END

```

```

C *****
C * SUBROUTINE INPUT
C * This subroutine retrieves the input data stored on the disk
C * SUBROUTINES included:
C * CHECK1: ECHO;
C * NODEXY, GAUSSQ
C * CHECK2: ECHO;
C *****

SUBROUTINE INPUT
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
&   NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
READ (5,900) NPOIN,NELEM,NVFIX,NCASE,NTYPE,
&   NNODE,NDOFN,NMATS,NPROP,NGAUS,NDIME,NSTRE
900 FORMAT(12I5)
NEVAB=NDOFN*NNODE
NTOTV=NPOIN*NDIME
NGAST=NGAUS*NGAUS*NELEM
WRITE(4,905) NPOIN,NELEM,NVFIX,NCASE,NTYPE,
&   NNODE,NDOFN,NMATS,NPROP,NGAUS,NDIME,NSTRE,NEVAB
905 FORMAT (//3X,' NPOIN =',I5,3X,' NELEM =',I5,3X,' NVFIX=',I5,3X,
&   ' NCASE =',I5,3X,' NTYPE =',I5,3X,' NNODE =',I5,3X,
&   ' NDOFN =',I5//,3X,' NMATS =',I5,3X,' NPROP =',I5,3X,
&   ' NGAUS =',I5,3X,' NDIME =',I5,3X,' NSTRE =',I5,3X,
&   ' NEVAB =',I5,3X/)
CALL CHECK1
WRITE(4,910)
910 FORMAT (//' ELEMENT PROPERTY NODE NUMBERS')
DO 10 IELEM=1,NELEM
READ (5,900) NUMEL,MATNO(NUMEL),
&   (LNODS(NUMEL,INODE),INODE=1,NNODE)
WRITE(4,915) NUMEL,MATNO(NUMEL),
&   (LNODS(NUMEL,INODE),INODE=1,NNODE)
10 CONTINUE
915 FORMAT (1X,I5,I9,6X,8I5)
DO 20 IPOIN=1,NPOIN
DO 20 IDIME=1,NDIME
20 COORD(IPOIN,IDIME)=0.D0
WRITE(4,920)
920 FORMAT (//' NODAL POINT COORDINATES')
WRITE(4,925)
925 FORMAT (' NODE',7X,'X',9X,'Y')
30 READ (5,*) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
IF (IPOIN.NE.NPOIN) GO TO 30
IF (NDIME.EQ.1) GO TO 40
CALL NODEXY
40 CONTINUE
DO 50 IPOIN=1,NPOIN
50 WRITE(4,930) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
930 FORMAT (I5,5F10.5)

```

```

      WRITE(4,940)
940  FORMAT (// ' RESTRAINED NODES' )
      WRITE(4,945)
945  FORMAT (1X, ' NODE', 1X, ' CODE', 6X, ' FIXED VALUES' )
      IF (NDOFN .NE. 2) GO TO 70
      DO 60 IVFIX=1,NVFIX
        READ (5,950) NOFIX(IVFIX), (IFPRE(IVFIX, IDOFN), IDOFN=1, NDOFN)
        &      , (PRESC(IVFIX, IDOFN), IDOFN=1, NDOFN)
60  WRITE(4,950) NOFIX(IVFIX), (IFPRE(IVFIX, IDOFN),
        &      IDOFN=1, NDOFN), (PRESC(IVFIX, IDOFN), IDOFN=1, NDOFN)
950  FORMAT (1X, I4, 3X, 2I1, 2F10.6)
      GO TO 90
70  DO 80 IVFIX=1,NVFIX
      READ (5,955) NOFIX(IVFIX), (IFPRE(IVFIX, IDOFN), IDOFN=1,
        &      NDOFN), (PRESC(IVFIX, IDOFN), IDOFN=1, NDOFN)
80  WRITE(4,955) NOFIX(IVFIX), (IFPRE(IVFIX, IDOFN), IDOFN=1,
        &      NDOFN), (PRESC(IVFIX, IDOFN), IDOFN=1, NDOFN)
955  FORMAT (1X, I4, 2X, 3I1, 3F10.6)
90  CONTINUE
      WRITE(4,960)
960  FORMAT (// 1X, ' MATERIAL PROPERTIES' )
      WRITE(4,965)
965  FORMAT (1X, ' NUMBER', 7X, ' PROPERTIES' )
      DO 100 IMATS=1,NMATS
        READ (5,*) NUMAT, (PROPS(NUMAT, IPROP), IPROP=1, NPROP)
100  WRITE(4,970) NUMAT, (PROPS(NUMAT, IPROP), IPROP=1, NPROP)
970  FORMAT (1X, I5, 5E12.5)
      CALL GAUSSQ
      CALL CHECK2
      RETURN
      END

```



```

C *****
C * SUBROUTINE CHECK1
C * This subroutine checks the input data to ensure the data are as 112
C * required by FEM and prints the error message, if any.
C * SUBROUTINES included:
C * ECHO.
C *****
SUBROUTINE CHECK1
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&   NPROB
  COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
  COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
  COMMON /WORK/ ELCOO(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
  DO 10 IEROR=1,24
10 NEROR(IEROR)=0
  IF (NPOIN .LE. 0) NEROR(1)=1
  IF (NELEM*NNODE .LT. NPOIN) NEROR(2)=2
  IF (NVFIX .LT. 1 .OR. NVFIX .GT. NPOIN) NEROR(3)=1
  IF (NCASE .LE. 0) NEROR(4)=1
  IF (NTYPE .LT. 0 .OR. NTYPE .GT. 2) NEROR(5)=1
  IF (NNODE .LT. 3 .OR. NNODE .GT. 8) NEROR(6)=1
  IF (NDOFN .LT. 2 .OR. NDOFN .GT. 3) NEROR(7)=1
  IF (NMATS .LT. 0 .OR. NMATS .GT. NELEM) NEROR(8)=1
  IF (NPROP .LT. 3 .OR. NPROP .GT. 7) NEROR(9)=1
  IF (NGAUS .LT. 2 .OR. NGAUS .GT. 3) NEROR(10)=1
  IF (NDIME .LT. 1 .OR. NDIME .GT. 2) NEROR(11)=1
  IF (NSTRE .LT. 2 .OR. NSTRE .GT. 5) NEROR(12)=1
  KEROR=0
  DO 20 IEROR=1,12
  IF (NEROR(IEROR) .EQ. 0) GO TO 20
  KEROR=1
  WRITE(4,900) IEROR
900 FORMAT (//' *** DIAGNOSIS BY CHECK1 ERROR ',I3)
  20 CONTINUE
  IF (KEROR .EQ. 0) RETURN
  CALL ECHO
  END

```

```

C *****
C * SUBROUTINE ECHO *
C * This subroutine returns an error message caused by user in * 113
C * inputing the data inproperly *
C *****

```

```

SUBROUTINE ECHO
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION NTITL(12)
COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&   NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELCOO(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
WRITE(4,900)
900 FORMAT (// ' NOW FOLLOWS A LISTING OF POST DISASTER CARDS'//)
10 READ (5,905) NTITL
905 FORMAT (12A4)
WRITE(4,910) NTITL
910 FORMAT (20X,12A4)
GO TO 10
END

```

```

C *****
C * SUBROUTINE NODEXY *
C * This subroutine calculates the coordinates of central nodes *
C *****
      SUBROUTINE NODEXY
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&      NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&      NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&      LNODS(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&      STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/ ELCO(2,8),SHAPE(8),DERIV(2,8),
&      DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&      SMATX(3,16,9),POSGP(3),WEIGP(3),GPOD(2,9),
&      NEROR(24)
      DO 30 IELEM=1,NELEM
      DO 20 INODE=1,NNODE,2
      NODST=LNODS(IELEM,INODE)
      IGASH=INODE+2
      IF (IGASH.GT. NNODE) IGASH=1
      NODFN=LNODS(IELEM,IGASH)
      MIDPT=INODE+1
      NODMD=LNODS(IELEM,MIDPT)
      TOTAL=DABS(COORD(NODMD,1))+DABS(COORD(NODMD,2))
      IF (TOTAL.GT. 0.D0) GO TO 20
      KOUNT=1
10  COORD(NODMD,KOUNT)=(COORD(NODST,KOUNT)+COORD(NODFN,
&      KOUNT))/2.D0
      KOUNT=KOUNT+1
      IF (KOUNT.EQ. 2) GO TO 10
20  CONTINUE
30  CONTINUE
      RETURN
      END

```

```

C *****
C * SUBROUTINE GAUSSQ *
C * This subroutine stores the weighting points of Gaussian *
C * interpolation *
C *****

```

115

```

SUBROUTINE GAUSSQ
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CONTO/NPOIN,NELEM,MNODE,NDOFN,NDIME,NSTRE,NTYPE,
& NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
& NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
& LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
& STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELOD(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
& NEROR(24)
GO TO (10,10,20),NGAUS
10 POSGP(1)=-0.577350269189626D0
WEIGP(1)=1.D0
GO TO 50
20 POSGP(1)=-0.774596669241483D0
POSGP(2)=0.0
WEIGP(1)=0.555555555555555D0
WEIGP(2)=0.888888888888888D0
50 KGAUS=NGAUS/2
DO 60 IGASH=1,KGAUS
JGASH=NGAUS+1-IGASH
POSGP(JGASH)=-POSGP(IGASH)
WEIGP(JGASH)=WEIGP(IGASH)
60 CONTINUE
RETURN
END

```

```

C *****
C * SUBROUTINE CHECK2
C * This subroutine checks the nodal coordinates and returns
C * an error message, if any.
C * SUBROUTINES included:
C * ECHO.
C *****

```

116

```

SUBROUTINE CHECK2
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION NDFRO(120)
COMMON /CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&   NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
MFRON=60
DO 10 IELEM=1,NELEM
10 NDFRO(IELEM)=0
DO 40 IPOIN=2,NPOIN
KPOIN=IPOIN-1
DO 30 JPOIN=1,KPOIN
DO 20 IDIME=1,NDIME
IF (COORD(IPOIN,IDIME) .NE. COORD(JPOIN,IDIME)) GO TO 30
20 CONTINUE
NEROR(13)=NEROR(13)+1
30 CONTINUE
40 CONTINUE
DO 50 IELEM=1,NELEM
50 IF (MATNO(IELEM) .LE. 0 .OR. MATNO(IELEM) .GT. NMATS)
&   NEROR(14)=NEROR(14)+1
DO 70 IELEM=1,NELEM
DO 60 INODE=1,NNODE
IF (LNODS(IELEM,INODE) .EQ. 0) NEROR(15)=NEROR(15)+1
60 IF (LNODS(IELEM,INODE) .LT. 0 .OR. LNODS(IELEM,INODE) .GT.
&   NPOIN) NEROR(16)=NEROR(16)+1
70 CONTINUE
DO 140 IPOIN=1,NPOIN
KSTAR=0
DO 100 IELEM=1,NELEM
KZERO=0
DO 90 INODE=1,NNODE
IF (LNODS(IELEM,INODE) .NE. IPOIN) GO TO 90
KZERO=KZERO+1
IF (KZERO .GT. 1) NEROR(17)=NEROR(17)+1
IF (KSTAR .NE. 0) GO TO 80
KSTAR=IELEM
NDFRO(IELEM)=NDFRO(IELEM)+NDOFN
80 CONTINUE
KLAST=IELEM
NLASt=INODE
90 CONTINUE
100 CONTINUE
IF (KSTAR .EQ. 0) GO TO 110

```

```

      IF (KLAST .LT. NELEM) NDFRO(KLAST+1)=NDFRO(KLAST+1)-NDOFN
      LNODS(KLAST,NLAST)=-IPOIN
      GO TO 140
110  WRITE(4,900) IPOIN
900  FORMAT (/ ' CHECK WHY NODE',I4,' NEVER APPEARS' )
      NEROR(18)=NEROR(18)+1
      SIGMA=0.D0
      DO 120 IDIME=1,NDIME
120  SIGMA=SIGMA+DABS(COORD(IPOIN,IDIME))
      IF(SIGMA .NE. 0.D0) NEROR(19)=NEROR(19)+1
      DO 130 IVFIX=1,NVFIX
130  IF (NOFIX(IVFIX) .EQ. IPOIN) NEROR(20)=NEROR(20)+1
140  CONTINUE
      NFRON=0
      KFRON=0
      DO 150 IELEM=1,NELEM
      NFRON=NFRON+NDFRO(IELEM)
150  IF (NFRON .GT. KFRON) KFRON=NFRON
      WRITE(4,905) KFRON
905  FORMAT (// ' MAXIMUM FRONTWIDTH ENCOUNTERED =',I5)
      IF (KFRON .GT. MFRON) NEROR(21)=1
      DO 170 IVFIX=1,NVFIX
      IF (NOFIX(IVFIX) .LE. 0 .OR. NOFIX(IVFIX) .GT. NPOIN)
&      NEROR(22)=NEROR(22)+1
      KOUNT=0
      DO 160 IDOFN=1,NDOFN
160  IF (IFPRE(IVFIX,IDOFN) .GT. 0) KOUNT=KOUNT+1
      IF (KOUNT .EQ. 0) NEROR(23)=NEROR(23)+1
      KVFIX=IVFIX-1
      DO 170 JVFIX=1,KVFIX
170  IF (IVFIX .NE. 1 .AND. NOFIX(IVFIX) .EQ. NOFIX(JVFIX))
&      NEROR(24)=NEROR(24)+1
      KEROR=0
      DO 180 IEROR=13,24
      IF (NEROR(IEROR) .EQ. 0) GO TO 180
      KEROR=1
      WRITE(4,910) IEROR,NEROR(IEROR)
910  FORMAT (// ' *** DIAGNOSIS BY CHECK2, ERROR',I3,6X,
&      ' ASSOCIATED NUMBER',I5)
180  CONTINUE
      IF(KEROR .NE. 0) GO TO 200
      DO 190 IELEM=1,NELEM
      DO 190 INODE=1,NNODE
190  LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
      RETURN
200  CALL ECHO
      END

```

```

C *****
C * SUBROUTINE LOAD(ITER) *
C * This subroutine checks if the contact length and deflection is *
C * formed and inputs the new displacement input. *
C *****
      SUBROUTINE LOAD(ITER)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NITYPE,
&      NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPEOB,
&      NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&      LNODS(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&      STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&      DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
&      SMATX(3,16,9),POSGP(3),WEIGP(3),GPOOD(2,9),
&      NEROR(24)
      COMMON /STORE/ NSTOR(99),NNN,NEE,ACCU,ADD,CLEN(99),DEFL(99),DMAX
      IF(ITER.GT.1) GO TO 111
      NNN=NVFIX
      NVFIX=NVFIX+NEE-1
      KJ=0
      DO 100 ISTORE=1,99
      KM=(-1)**ISTORE
      KP=9
      IF(KM.GT.0) KP=5
      KJ=KJ+KP
100 NSTOR(ISTORE)=KJ
111 CONTINUE
      DIFF=COORD(NSTOR(NVFIX-NNN+2),2)-COORD(NSTOR(NVFIX-NNN+1),2)
      IF(DIFF.LE.ACCU) NVFIX=NVFIX+1
      DO 101 IVFIX=NNN,NVFIX
      NOFIX(IVFIX)=NSTOR(IVFIX-NNN+1)
      IFPRE(IVFIX,1)=0
      IFPRE(IVFIX,2)=1
      PRESC(IVFIX,1)=0.D0
      PRESC(IVFIX,2)=COORD(NSTOR(NVFIX-NNN+2),2)
      & -COORD(NSTOR(IVFIX-NNN+1),2)
101 CONTINUE
      IFPRE(NNN,1)=1
      PRESC(NNN,1)=0.D0
      RETURN
      END

```

```

C *****
C *   SUBROUTINE STIFPS                                     *
C *   This subroutine computes the element stiffness and stores it   *
C *   onto a disk.                                                 *
C *   SUBROUTINES included:                                         *
C *   MODPS, SFR2, JACOB2, BMATPS, DBE.                             *
C *****
SUBROUTINE STIFPS
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION ESTIF(16,16)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&    NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
&    NPRQB
  COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&    LNODS(416,8),MATNO(416)
  COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&    STRIN(3,3744),NOFIX(208),IFPRE(208,2)
  COMMON /WORK/  ELCOD(2,8),SHAPE(8),DERIV(2,8),
&    DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&    SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&    NEROR(24)
  DO 70 IELEM=1,NELEM
    LPROP=MATNO(IELEM)
    DO 10 INODE=1,NNODE
      LNODE=LNODS(IELEM,INODE)
      DO 10 IDIME=1,NDIME
10    ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
      CALL MODPS(LPROP)
      THICK=PROPS(LPROP,3)
      DO 20 IEVAB=1,NEVAB
        DO 20 JEVAB=1,NEVAB
20    ESTIF(IEVAB,JEVAB)=0.DO
      KGASP=0
      DO 50 IGAUS=1,NGAUS
        DO 50 JGAUS=1,NGAUS
          KGASP=KGASP+1
          EXISP=POSGP(IGAUS)
          ETASP=POSGP(JGAUS)
          CALL SFR2(EXISP,ETASP)
          CALL JACOB2(IELEM,DJACB,KGASP)
          DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
          IF (THICK .NE. 0.DO) DVOLU=DVOLU*THICK
          CALL BMATPS
          CALL DBE
          DO 30 IEVAB=1,NEVAB
            DO 30 JEVAB=1,NEVAB
            DO 30 ISTRE=1,NSTRE
30    ESTIF(IEVAB,JEVAB)=ESTIF(IEVAB,JEVAB)+BMATX(ISTRE,IEVAB)*
&      DBMAT(ISTRE,JEVAB)*DVOLU
            DO 40 ISTRE=1,NSTRE
              DO 40 IEVAB=1,NEVAB
40    SMATX(ISTRE,IEVAB,KGASP)=DBMAT(ISTRE,IEVAB)
50    CONTINUE
            DO 60 IEVAB=1,NEVAB
              DO 60 JEVAB=1,NEVAB
60    ESTIF(JEVAB,IEVAB)=ESTIF(IEVAB,JEVAB)
            WRITE (7) ESTIF
            WRITE (9) SMATX,GPCOD
70    CONTINUE
      RETURN
    END

```



```

C *****
C * SUBROUTINE MODPS(LPROP) *
C * This subroutine computes the D-matrix for *
C * Plane Stress OR Plane Strain *
C *****

SUBROUTINE MODPS (LPROP)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
& NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
& NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
& LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
& STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELCO(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
& NEROR(24)
YM=PROPS(LPROP,1)
PR=PROPS(LPROP,2)
DO 10 ISTRE=1,NSTRE
DO 10 JSTRE=1,NSTRE
DMATX(ISTRE,JSTRE)=0.D0
10 CONTINUE
IF (NTYPE .EQ. 2) GO TO 20
IF (NTYPE .EQ. 1) GO TO 15.
WRITE(4,905)
905 FORMAT(1X'ERROR IN NTYPE - MODPS')
STOP
15 C=YM/(1.D0-PR*PR)
DMATX(1,1)=C
DMATX(2,2)=C
DMATX(1,2)=C*PR
DMATX(2,1)=C*PR
DMATX(3,3)=(1.D0-PR)*C/2.D0
GO TO 30
20 CONTINUE
C=YM*(1.D0-PR)/((1.D0+PR)*(1.D0-2.D0*PR))
C2=C*PR/(1.D0-PR)
DMATX(1,1)=C
DMATX(2,2)=C
DMATX(1,2)=C2
DMATX(2,1)=C2
DMATX(3,3)=C*(1.D0-2.D0*PR)/(2.D0*(1.D0-PR))
30 CONTINUE
RETURN
END

```

```

C *****
C * SUBROUTINE SFR2(S,T) *
C * This subroutine computes the shape functions and the derivatives *
C *****
      SUBROUTINE SFR2(S,T)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&      NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&      NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&      LNODES(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&      STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&      DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&      SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&      NEROR(24)
      S2=S*2.D0
      T2=T*2.D0
      SS=S*S
      TT=T*T
      ST=S*T
      SST=SS*T
      STT=S*TT
      ST2=ST*2.D0
      SHAPE(1)=(-1.D0+ST+SS+TT-SST-STT)/4.D0
      SHAPE(2)=(1.D0-T-SS+SST)/2.D0
      SHAPE(3)=(-1.D0-ST+SS+TT-SST+STT)/4.D0
      SHAPE(4)=(1.D0+S-TT-STT)/2.D0
      SHAPE(5)=(-1.D0+ST+SS+TT+SST+STT)/4.D0
      SHAPE(6)=(1.D0+T-SS-SST)/2.D0
      SHAPE(7)=(-1.D0-ST+SS+TT+SST-STT)/4.D0
      SHAPE(8)=(1.D0-S-TT+STT)/2.D0
      DERIV(1,1)=(T+S2-ST2-TT)/4.D0
      DERIV(1,2)=-S+ST
      DERIV(1,3)=(-T+S2-ST2+TT)/4.D0
      DERIV(1,4)=(1.D0-TT)/2.D0
      DERIV(1,5)=(T+S2+ST2+TT)/4.D0
      DERIV(1,6)=-S-ST
      DERIV(1,7)=(-T+S2+ST2-TT)/4.D0
      DERIV(1,8)=(-1.D0+TT)/2.D0
      DERIV(2,1)=(S+T2-SS-ST2)/4.D0
      DERIV(2,2)=(-1.D0+SS)/2.D0
      DERIV(2,3)=(-S+T2-SS+ST2)/4.D0
      DERIV(2,4)=-T-ST
      DERIV(2,5)=(S+T2+SS+ST2)/4.D0
      DERIV(2,6)=(1.D0-SS)/2.D0
      DERIV(2,7)=(-S+T2+SS-ST2)/4.D0
      DERIV(2,8)=-T+ST
      RETURN
      END

```

```

C *****
C * SUBROUTINE JACOB2(IELEM,DJACB,KGASP) *
C * This subroutine computes the Jacobian matrix *
C *****
      SUBROUTINE JACOB2(IELEM,DJACB,KGASP)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&          NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&          NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&          LNODS(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&          STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&          DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&          SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&          NEROR(24)
      DIMENSION XJACM(2,2),XJACI(2,2)
      DO 10 IDIME=1,NDIME
      GPCOD(IDIME,KGASP)=0.D0
      DO 10 INODE=1,NNODE
      GPCOD(IDIME,KGASP)=GPCOD(IDIME,KGASP)+ELCOD(IDIME,INODE)*
& SHAPE(INODE)
10 CONTINUE
      DO 20 IDIME=1,NDIME
      DO 20 JDIME=1,NDIME
      XJACM(IDIME,JDIME)=0.D0
      DO 20 INODE=1,NNODE
      XJACM(IDIME,JDIME)=XJACM(IDIME,JDIME)+DERIV(IDIME,INODE)*
& ELCOD(JDIME,INODE)
20 CONTINUE
      DJACB=XJACM(1,1)*XJACM(2,2)-XJACM(1,2)*XJACM(2,1)
      IF (DJACB .GT. 0.D0) GO TO 30
      WRITE(4,900) IELEM
      STOP
30 XJACI(1,1)=XJACM(2,2)/DJACB
      XJACI(2,2)=XJACM(1,1)/DJACB
      XJACI(1,2)=-XJACM(1,2)/DJACB
      XJACI(2,1)=-XJACM(2,1)/DJACB
      DO 40 IDIME=1,NDIME
      DO 40 INODE=1,NNODE
      CARTD(IDIME,INODE)=0.D0
      DO 40 JDIME=1,NDIME
      CARTD(IDIME,INODE)=CARTD(IDIME,INODE)+XJACI(IDIME,JDIME)*
& DERIV(JDIME,INODE)
40 CONTINUE
900 FORMAT (// 'PROGRAM HALTED IN JACOB2',11X, 'ZERO OR NEGATIVE
& AREA',10X, 'ELEMENT NUMBER ',I5)
      RETURN
      END

```

```

C *****
C * SUBROUTINE BMATPS
C * This subroutine computes the B-matrix of an element
C *****
SUBROUTINE BMATPS
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&    NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&    NPROB
  COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&    LNODS(416,8),MATNO(416)
  COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&    STRIN(3,3744),NOFIX(208),IFPRE(208,2)
  COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&    DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&    SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&    NEROR(24)
  KN=0
  DO 10 INODE=1,NNODE
    KM=KN+1
    KN=KM+1
    BMATX(1,KM)=CARTD(1,INODE)
    BMATX(1,KN)=0.D0
    BMATX(2,KM)=0.D0
    BMATX(2,KN)=CARTD(2,INODE)
    BMATX(3,KM)=CARTD(2,INODE)
    BMATX(3,KN)=CARTD(1,INODE)
10 CONTINUE
  RETURN
  END

```

```

C *****
C * SUBROUTINE DBE *
C * This subroutine calculates the derivatives of the B-matrix *
C *****
SUBROUTINE DBE
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&   NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELCOO(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
DO 10 ISTRE=1,NSTRE
DO 10 IEVAB=1,NEVAB
DBMAT(ISTRE,IEVAB)=0.D0
DO 10 JSTRE=1,NSTRE
DBMAT(ISTRE,IEVAB)=DBMAT(ISTRE,IEVAB)+DMATX(ISTRE,JSTRE)
&   *BMATX(JSTRE,IEVAB)
10 CONTINUE
RETURN
END

```

```

C *****
C * SUBROUTINE LOADPS(ITER) *
C * This subroutine computes the input loads, including point load, *
C * distributed load, gravity load, thermal load. *
C * SUBROUTINES included: *
C * MODPS, SFR2, JACOB2. *
C *****

SUBROUTINE LOADPS(ITER)
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION TITLE(12),POINT(2),PRESS(3,2),PGASH(2),
& DGASH(2),TEMPE(500),STRAN(3),STRES(3),NOPRS(3)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
& NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
& NPROB
  COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
& LNODS(416,8),MATNO(416)
  COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
& STRIN(3,3744),NOFIX(208),IFPRE(208,2)
  COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
& NEROR(24)
  COMMON /STORE/ NSTOR(99),NNN,NEE,ACCU,ADD,CLEN(99),DEFL(99),DMAX
  DO 10 IELEM=1,NELEM
  DO 10 IEVAB=1,NEVAB
10 ELOAD(IELEM,IEVAB)=0.D0
  IF(ITER.NE.1) GO TO 1234
  READ (5,900) TITLE
900 FORMAT (12A4)
  WRITE(4,905) TITLE,ICASE
905 FORMAT ('0',12A4,5X,'LOAD CASE =',I3)
  READ (5,910) IPLOD,IGRAV,IEDGE,ITEMP
  WRITE(4,910) IPLOD,IGRAV,IEDGE,ITEMP
910 FORMAT (4I5)
  IF (IPLOD .EQ. 0) GO TO 500
20 READ(5,915) LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
  WRITE(4,915) LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
915 FORMAT (I5,2F10.3)
  DO 30 IELEM=1,NELEM
  DO 30 INODE=1,NNODE
  NLOCA=LNODS(IELEM,INODE)
  IF (LODPT .EQ. NLOCA) GO TO 40
30 CONTINUE
40 DO 50 IDOFN=1,NDOFN
  NGASH=(INODE-1)*NDOFN+IDOFN
50 ELOAD(IELEM,NGASH)=POINT(IDOFN)
  IF (LODPT .LT. NPOIN) GO TO 20
500 CONTINUE
  IF (IGRAV .EQ.0) GO TO 600
  READ(5,920) THETA,GRAVY
920 FORMAT(2F10.3)
  WRITE(4,925) THETA,GRAVY
925 FORMAT(1H0,16HGRAVITY ANGLE = ,F10.3,
  20H GRAVITY CONSTANT = ,F10.3)
  THETA=THETA/57.2957794D0/514.D0
  DO 90 IELEM=1,NELEM
  LPROP=MATNO(IELEM)
  THICK=PROPS(LPROP,3)
  DENSE=PROPS(LPROP,4)
  IF(DENSE.EQ.0.D0) GO TO 90

```

```

GXCOM=DENSE*GRAVY*DSIN(THETA)
GYCOM=-DENSE*GRAVY*DCOS(THETA)
DO 60 INODE=1,NNODE
  LNODE=LNODS(IELEM,INODE)
  DO 60 IDIME=1,NDIME
60  ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
  DO 80 IGAUS=1,NGAUS
  DO 80 JGAUS=1,NGAUS
  EXISP=POSGP(IGAUS)
  ETASP=POSGP(JGAUS)
  CALL SFR2(EXISP,ETASP)
  KGASP=1
  CALL JACOB2(IELEM,DJACB,KGASP)
  DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
  IF(THICK.NE.0.0) DVOLU=DVOLU*THICK
  DO 70 INODE=1,NNODE
  NGASH=(INODE-1)*NDOFN+1
  MGASH=(INODE-1)*NDOFN+2
  ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)+
&  GXCOM*SHAPE(INODE)*DVOLU
70  ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)+
&  GYCOM*SHAPE(INODE)*DVOLU
80  CONTINUE
90  CONTINUE
600 CONTINUE
1234 CONTINUE
  IF( IEDGE .EQ.0) GO TO 700
  WRITE(4,935) NEDGE
935  FORMAT (1H0,5X,'NO. OF LOADED EDGES = ',I5)
  WRITE(4,940)
940  FORMAT (1H0,5X,'LIST OF LOADED EDGES AND APPLIED LOADS')
  NODEG=3
  NEASS=4*IEDGE
  WRITE(4,950) NEASS,(NOPRS(IODEG),IODEG=1,NODEG)
950  FORMAT (1I0,5X,3I5)
  WRITE(4,956) ((PRESS(IODEG,IDOFN),IODEG=1,NODEG),
&  IDOFN=1,NDOFN)
956  FORMAT (6F10.3)
  ETASP=-1.D0
  DO 100 IODEG=1,NODEG
  LNODE=NOPRS(IODEG)
  DO 100 IDIME=1,NDIME
100  ELCOD(IDIME,IODEG)=COORD(LNODE,IDIME)
  DO 150 IGAUS=1,NGAUS
  EXISP=POSGP(IGAUS)
  CALL SFR2(EXISP,ETASP)
  DO 110 IDOFN=1,NDOFN
  PGASH(IDOFN)=0.D0
  DGASH(IDOFN)=0.D0
  DO 110 IODEG=1,NODEG
  PGASH(IDOFN)=PGASH(IDOFN)+PRESS(IODEG,IDOFN)*SHAPE(IODEG)
110  DGASH(IDOFN)=DGASH(IDOFN)+ELCOD(IDOFN,IODEG)*DERIV(1,IODEG)
  DVOLUM=WEIGP(IGAUS)
  PXCOM=DGASH(1)*PGASH(2)-DGASH(2)*PGASH(1)
  PYCOM=DGASH(1)*PGASH(1)+DGASH(2)*PGASH(2)
  DO 120 INODE=1,NNODE
  NLOCA=LNODS(NEASS,INODE)
  IF (NLOCA .EQ. NOPRS(1)) GO TO 130
120  CONTINUE
130  JNODE=INODE+NODEG-1

```

```

KOUNT=0
DO 140 KNODE=INODE,JNODE
KOUNT=KOUNT+1
NGASH=(KNODE-1)*NDOFN+1
MGASH=(KNODE-1)*NDOFN+2
IF (KNODE .GT. NNODE) NGASH=1
IF (KNODE .GT. NNODE) MGASH=2
ELOAD(NEASS,NGASH)=ELOAD(NEASS,NGASH)+SHAPE(KOUNT)*PXCOM*DVOLUM
140 ELOAD(NEASS,MGASH)=ELOAD(NEASS,MGASH)+SHAPE(KOUNT)*PYCOM*DVOLUM
150 CONTINUE
160 CONTINUE
700 CONTINUE

```

C

```

IF (ITEMP .EQ. 0) GO TO 800
DO 170 IPOIN=1,NPOIN
170 TEMPE(IPOIN)=0.D0
WRITE(4,960)
960 FORMAT (1H0,5X,'PRESCRIBED NODAL TEMPERATURES')
180 READ (5,965) NODPT,TEMPE(NODPT)
WRITE(4,965) NODPT,TEMPE(NODPT)
965 FORMAT (I5,F10.3)
IF (NODPT .LT. NPOIN) GO TO 180
MDIME=NDIME+1
KGAST=0
DO 280 IELEM=1,NELEM
LPROP=MATNO(IELEM)
DO 200 INODE=1,NNODE
LNODE=LNODS(IELEM,INODE)
DO 190 IDIME=1,NDIME
190 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
200 ELCOD(MDIME,INODE)=TEMPE(LNODE)
CALL MODPS (LPROP)
YOUNG=PROPS(LPROP,1)
POISS=PROPS(LPROP,2)
THICK=PROPS(LPROP,3)
ALPHA=PROPS(LPROP,5)
DO 270 IGAUS=1,NGAUS
DO 270 JGAUS=1,NGAUS
KGAST=KGAST+1
EXISP=POSGP(IGAUS)
ETASP=POSGP(JGAUS)
CALL SFR2 (EXISP,ETASP)
KGASP=1
CALL JACOB2 (IELEM,DJACB,KGASP)
THERM=0.D0
DO 210 INODE=1,NNODE
210 THERM=THERM+ELCOD(MDIME,INODE)*SHAPE(INODE)
DVOLUM=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
IF (THICK .NE. 0.D0) DVOLUM=DVOLUM*THICK
EIGEN=THERM*ALPHA
IF (NTYPE .EQ. 2) GO TO 220
STRAN(1)=-EIGEN
STRAN(2)=-EIGEN
STRAN(3)=0.D0
GO TO 230
220 STRAN(1)=-(1.D0+POISS)*EIGEN
STRAN(2)=-(1.D0+POISS)*EIGEN
STRAN(3)=0.D0
230 DO 250 ISTRE=1,NSTRE
STRES(ISTRE)=0.D0

```



```

DO 240 JSTRE=1,NSTRE
240 STRES(ISTRE)=STRES(ISTRE)+DMATX(ISTRE,JSTRE)*STRAN(JSTRE)
250 STRIN(ISTRE,KGAST)=STRES(ISTRE)
  IF (NTYPE.EQ. 2) STRIN(4,KGAST)=-YOUNG*EIGEN
  IF (NTYPE.EQ. 1) STRIN(4,KGAST)=0.D0
DO 260 INODE=1,NNODE
  NGASH=(INODE-1)*NDOFN+1
  MGASH=(INODE-1)*NDOFN+2
  ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)-(CARTD(1,INODE)*
&      STRES(1)+CARTD(2,INODE)*STRES(3))*DVOLUM
260 ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)-(CARTD(1,INODE)*
&      STRES(3)+CARTD(2,INODE)*STRES(2))*DVOLUM
270 CONTINUE
280 CONTINUE
800 CONTINUE
  NNNN=4
  WRITE(4,970)
970 FORMAT ('0',5X,'TOTAL NODAL FORCES FOR EACH ELEMENT')
  DO 290 IELEM=4,NNNN,4
290 WRITE(4,975) IELEM,(ELOAD(IELEM,IEVAB),IEVAB=1,NEVAB)
975 FORMAT (1X,I4,5X,8E12.4/(10X,8E12.4))
  RETURN
END

```

```

C *****
C * SUBROUTINE FRONT *
C * This subroutine solves the linear equations by the front solver *
C * and outputs the displacements and the reactions on constrained *
C * nodes. *
C *****

```

129

```

SUBROUTINE FRONT
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION FIXED(2912),EQUAT(60),VECRV(2912),GLOAD(60),
& GSTIF(1830),ESTIF(16,16),IFFIX(2912),NACVA(60),
& LOCEL(16),NDEST(16)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
& NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
& NPROB
  COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
& LNODS(416,8),MATNO(416)
  COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
& STRIN(3,3744),NOFIX(208),IFPRE(208,2)
  COMMON /WORK/ ELOD(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
& NEROR(24)
  MFRON=60
  MSTIF=1830
  NTOTV=NPOIN*NDOFN
  DO 100 ITOTV=1,NTOTV
    IFFIX(ITOTV)=0
    VECRV(ITOTV)=0.D0
  100 FIXED(ITOTV)=0.D0
    DO 110 IVFIX=1,NVFIX
      NLOCA=(NOFIX(IVFIX)-1)*NDOFN
      DO 110 IDOFN=1,NDOFN
        NGASH=NLOCA+IDOFN
        IFFIX(NGASH)=IFPRE(IVFIX,IDOFN)
  110 FIXED(NGASH)=PRESC(IVFIX,IDOFN)
      DO 140 IPOIN=1,NPOIN
        KLAST=0
        DO 130 IELEM=1,NELEM
          DO 120 INODE=1,NNODE
            IF (LNODS(IELEM,INODE) .NE. IPOIN) GO TO 120
            KLAST=IELEM
            NLAST=INODE
          120 CONTINUE
        130 CONTINUE
        IF (KLAST .NE. 0) LNODS(KLAST,NLAST)=-IPOIN
      140 CONTINUE
      DO 150 ISTIF=1,MSTIF
        GSTIF(ISTIF)=0.D0
        DO 160 IFRON=1,MFRON
          GLOAD(IFRON)=0.D0
          EQUAT(IFRON)=0.D0
        160 NACVA(IFRON)=0
        REWIND 7
        REWIND 8
        REWIND 9
        REWIND 10

```

```

C
C ENTER MAIN ELEMENT ASSEMBLY-REDUCTION LOOP
C

```

```

NFRON=0
KELVA=0
DO 380 IELEM=1,NELEM
KEVAB=0
READ (7) ESTIF
DO 170 INODE=1,NNODE
DO 170 IDOFN=1,NDOFN
NPOSI=(INODE-1)*NDOFN+IDOFN
LOCNO=LNODS(IELEM,INODE)
IF (LOCNO.GT. 0) LOCEL(NPOSI)=(LOCNO-1)*NDOFN+IDOFN
IF (LOCNO.LT. 0) LOCEL(NPOSI)=(LOCNO+1)*NDOFN-IDOFN
170 CONTINUE
DO 210 IEVAB=1,NEVAB
NIKNO=IABS(LOCEL(IEVAB))
KEXIS=0
DO 180 IFRON=1,NFRON
IF (NIKNO.NE. NACVA(IFRON)) GO TO 180
KEVAB=KEVAB+1
KEXIS=1
NDEST(KEVAB)=IFRON
180 CONTINUE
IF (KEXIS.NE. 0) GO TO 210
DO 190 IFRON=1,MFRON
IF (NACVA(IFRON).NE. 0) GO TO 190
NACVA(IFRON)=NIKNO
KEVAB=KEVAB+1
NDEST(KEVAB)=IFRON
GO TO 200
190 CONTINUE
200 IF (NDEST(KEVAB).GT. NFRON) NFRON=NDEST(KEVAB)
210 CONTINUE
DO 240 IEVAB=1,NEVAB
IDEST=NDEST(IEVAB)
GLOAD(IDEST)=GLOAD(IDEST)+ELOAD(IELEM,IEVAB)
IF (ICASE.GT.1) GO TO 230
DO 220 JEVAB=1,JEVAB
JDEST=NDEST(JEVAB)
NGASH=(JDEST*JDEST-JDEST)/2+IDEST
NGISH=(IDEST*IDEST-IDEST)/2+JDEST
IF (JDEST.GE. IDEST) GSTIF(NGASH)=GSTIF(NGASH)+ESTIF(IEVAB,
& JEVAB)
IF (JDEST.LT. IDEST) GSTIF(NGISH)=GSTIF(NGISH)+
& ESTIF(IEVAB,JEVAB)
220 CONTINUE
230 CONTINUE
240 CONTINUE
DO 370 IEVAB=1,NEVAB
NIKNO=LOCEL(IEVAB)
IF (NIKNO.LE. 0) GO TO 370
DO 350 IFRON=1,NFRON
IF (NACVA(IFRON).NE. NIKNO) GO TO 350
IF (ICASE.GT. 1) GO TO 260
DO 250 JFRON=1,MFRON
IF (IFRON.LT. JFRON) NLOCA=(JFRON*JFRON-JFRON)/2+IFRON
IF (IFRON.GE. JFRON) NLOCA=(IFRON*IFRON-IFRON)/2+JFRON
EQUAT(JFRON)=GSTIF(NLOCA)
250 GSTIF(NLOCA)=0.D0
260 CONTINUE
EQRES=GLOAD(IFRON)
GLOAD(IFRON)=0.D0

```

```

      KELVA=KELVA+1
      IF (ICASE .GT. 1) GO TO 270
      WRITE (8) 'EQUAT,EQRHS,IFRON,NIKNO'
      GO TO 280
270  WRITE (10) EQRHS
      READ (8) EQUAT,DUMMY,IDUMM,NIKNO
280  CONTINUE
      PIVOT=EQUAT(IFRON)
      EQUAT(IFRON)=0.DO
      IF (IFFIX(NIKNO) .EQ. 0) GO TO 300
      DO 290 JFRON=1,NFRON
290  GLOAD(JFRON)=GLOAD(JFRON)-FIXED(NIKNO)*EQUAT(JFRON)
      GO TO 340
300  DO 330 JFRON=1,NFRON
      GLOAD(JFRON)=GLOAD(JFRON)-EQUAT(JFRON)*EQRHS/PIVOT
      IF (ICASE .GT. 1) GO TO 320
      IF (EQUAT(JFRON) .EQ. 0.DO) GO TO 330
      NLOCA=(JFRON*JFRON-JFRON)/2
      DO 310 LFRON=1,JFRON
      NGASH=LFRON+NLOCA
310  GSTIF(NGASH)=GSTIF(NGASH)-EQUAT(JFRON)*EQUAT(LFRON)/PIVOT
320  CONTINUE
330  CONTINUE
340  EQUAT(IFRON)=PIVOT
      NACVA(IFRON)=0
      GO TO 360
350  CONTINUE
360  IF (NACVA(NFRON) .NE. 0) GO TO 370
      NFRON=NFRON-1
      IF (NFRON .GT. 0) GO TO 360
370  CONTINUE
380  CONTINUE
      DO 410 IELVA=1,KELVA
      BACKSPACE 8
      READ (8) EQUAT,EQRHS,IFRON,NIKNO
      BACKSPACE 8
      IF (ICASE .EQ. 1) GO TO 390
      BACKSPACE 10
      READ (10) EQRHS
      BACKSPACE 10
390  CONTINUE
      PIVOT=EQUAT(IFRON)
      IF (IFFIX(NIKNO) .EQ. 1) VECRV(IFRON)=FIXED(NIKNO)
      IF (IFFIX(NIKNO) .EQ. 0) EQUAT(IFRON)=0.DO
      DO 400 JFRON=1,NFRON
400  EQRHS=EQRHS-VECRV(JFRON)*EQUAT(JFRON)
      IF (IFFIX(NIKNO) .EQ. 0) VECRV(IFRON)=EQRHS/PIVOT
      IF (IFFIX(NIKNO) .EQ. 1) FIXED(NIKNO)=-EQRHS
      ASDIS(NIKNO)=VECRV(IFRON)
410  CONTINUE
      IF (IPLAST .EQ. 1) GO TO 511
      WRITE(4,900)
900  FORMAT ('0',5X,'DISPLACEMENTS')
      IF (NDOFN .NE. 2) GO TO 430
      IF (NDIME .NE. 1) GO TO 420
      WRITE(4,905)
905  FORMAT ('0',5X,'NODE',6X,'DISP',7X,'ROTATION')
      GO TO 440
420  WRITE(4,910)
910  FORMAT ('0',5X,'NODE',5X,'X-DISP.',7X,'Y-DISP')

```

```

      GO TO 440
430 WRITE(4,915)
915 FORMAT ('0',5X,'NODE',6X,'DISP.',8X,'XZ-ROT.',7X,'YZ-ROT.')
440 CONTINUE
      DO 450 IPOIN=1,NPOIN
      NGASH=IPOIN*NDOFN
      NGISH=NGASH-NDOFN+1
450 WRITE(4,920) IPOIN,(ASDIS(IGASH),IGASH=NGISH,NGASH)
920 FORMAT (I10,3E14.6)
      WRITE(4,925)
925 FORMAT ('0',5X,'REACTIONS')
      IF (NDOFN .NE. 2) GO TO 470
      IF (NDIME .NE. 1) GO TO 460
      WRITE(4,930)
930 FORMAT ('0',5X,'NODE',6X,'FORCE',8X,'MOMENT')
      GO TO 480
460 WRITE(4,935)
935 FORMAT ('0',5X,'NODE',5X,'X-FORCE',7X,'Y-FORCE')
      GO TO 480
470 WRITE(4,940)
940 FORMAT ('0',5X,'NODE',6X,'FORCE',6X,'XZ-MOMENT',5X,'YZ-MOMENT')
480 CONTINUE
      DO 510 IPOIN=1,NPOIN
      NLOCA=(IPOIN-1)*NDOFN
      DO 490 IDOFN=1,NDOFN
      NGUSH=NLOCA+IDOFN
      IF (IFFIX(NGUSH) .GT. 0) GO TO 500
490 CONTINUE
      GO TO 510
500 NGASH=NLOCA+NDOFN
      NGISH=NLOCA+1
      WRITE(4,945) IPOIN,(FIXED(IGASH),IGASH=NGISH,NGASH)
510 CONTINUE
511 CONTINUE
945 FORMAT (I10,3E14.6)
      DO 520 IELEM=1,NELEM
      DO 520 INODE=1,NNODE
520 LNODS(IELEM,INODE)=IABS(LNODS(IELEM,INODE))
      RETURN
      END

```

```

C *****
C * SUBROUTINE REINP
C * This subroutine updates the deformed configuration as the input
C * for the next evaluation.
C *****

```

133

```

SUBROUTINE REINP
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&   NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)

```

```

C
C Re-input the coordinate data
C

```

```

DO 3 IPOIN=1,NPOIN
NGASH=IPOIN*NDOFN
NGISH=NGASH-NDOFN+1
DO 3 IGASH=NGISH,NGASH
IDIME=1
IF(IGASH.GT.NGISH) IDIME=2
3 COORD(IPOIN,IDIME)=ASDIS(IGASH)+COORD(IPOIN,IDIME)
RETURN
END

```

```

C *****
C *   PROGRAM PEM-M
C *   This program is to derive the APPARENT Young's modulus of
C *   the tire model.
C *   SUBROUTINES included:
C *       INPUT: CHECK1: ECHO;
C *             NODEXY, GAUSSQ,
C *             CHECK2: ECHO;
C *       GOLDN: FUNC: STIFPS: MODPS, SFR2, JACOB2, BMATPS, DBE;
C *             LOADPS: MODPS, SFR2, JACOB2;
C *             FRONT;
C *****
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TITLE(12)
      COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&          NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&          NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&          LNODS(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&          STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/  ELCOO(2,8),SHAPE(8),DERIV(2,8),
&          DMATX(3,3),CARTD(2,8),DEMAT(3,16),BMATX(3,16),
&          SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&          NEROR(24)
      COMMON /WT/ XLOAD,YCOMP,NPONE,KLOAD
      CHARACTER*15 FILEA,FILEB

C
C   Open input file
C
      WRITE(*,800)
800  FORMAT(/' INPUT FILE NAME - '/')
      READ(*,801) FILEA
801  FORMAT(A)
      OPEN(5,FILE=FILEA,ACCESS='SEQUENTIAL',STATUS='OLD')
      READ (5,900) NPROB
900  FORMAT(I5)

C
C   Open output file as unit 4
C
      WRITE(*,802)
802  FORMAT(/' OUTPUT FILE NAME - '/')
      READ(*,803) FILEB
803  FORMAT(A)
      OPEN(4,FILE=FILEB,ACCESS='SEQUENTIAL',STATUS='NEW')

C
      WRITE(4,905) NPROB
905  FORMAT(1H0,5X,'TOTAL NUMBER OF PROBLEMS = ',I5)
      DO 20 IPOB=1,NPROB
      READ (5,910) TITLE
910  FORMAT (12A4)
      WRITE(4,915) IPOB,TITLE
915  FORMAT (//////,6X,'PROBLEM NO. ',I3,10X,12A4)
      WRITE(*,820)
820  FORMAT(/' READING INPUT DATA '/')
      CALL INPUT
      READ(5,*) ALPHA,BETA,GAMMA,A,B,XLOAD,NPONE
      KLOAD=0
      CALL GOLDN(ALPHA,BETA,GAMMA,A,B)

```

```
CLOSE(4,STATUS='KEEP')  
CLOSE(5,STATUS='KEEP')  
20 CONTINUE  
STOP  
END
```

135


```

C *****
C * SUBROUTINE GOLDN (ALPHA,BETA,GAMMA,A,B) *
C * This subroutine is a constrained single-variable search program. *
C * SUBROUTINES included: *
C * FUNC: STIPPS: MODPS, SFR2, JACOB2, BMATPS, DBE; *
C * LOADPS: MODPS, SFR2, JACOB2, BMATPS, DBE; *
C * FRONT; *
C *****

```

136

```

SUBROUTINE GOLDN (ALPHA,BETA,GAMMA,A,B)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION VAR(4),FUNV(4)
GLD=0.6180339885D0
VAR(1)=A
CALL FUNC(A,V)
FUNV(1)=V
VAR(4)=B
CALL FUNC(B,V)
FUNV(4)=V
VAR(3)=A+(B-A)*GLD
CALL FUNC(VAR(3),V)
FUNV(3)=V
VAR(2)=B-(B-A)*GLD
CALL FUNC(VAR(2),V)
FUNV(2)=V
10 DEL=VAR(4)-VAR(1)
FUNMA=-1.D30
DO 1 I=1,4
IF(FUNV(I).LE.FUNMA) GO TO 1
FUNMA=FUNV(I)
VARMA=VAR(I)
KMAX=I
1 CONTINUE
IF(KMAX.EQ.2.OR.KMAX.EQ.3) GO TO 100
FUNMI=FUNMA
DO 5 I=1,4
IF (FUNV(I).GE.FUNMI) GO TO 5
FUNMI=FUNV(I)
VARMI=VAR(I)
KMIN=I
5 CONTINUE
IF(DABS(FUNMA-FUNMI).LE.BETA) GO TO 99
IF(DEL.LT.ALPHA) GO TO 99
IF(FUNMI.LE.GAMMA) GO TO 99
IF (KMIN.GT.2) THEN
VAR(1)=VAR(2)
FUNV(1)=FUNV(2)
VAR(2)=VAR(3)
FUNV(2)=FUNV(3)
VAR(3)=VAR(1)+(VAR(4)-VAR(1))*GLD
CALL FUNC(VAR(3),V)
FUNV(3)=V
ELSE
VAR(4)=VAR(3)
FUNV(4)=FUNV(3)
VAR(3)=VAR(2)
FUNV(3)=FUNV(2)
VAR(2)=VAR(4)-(VAR(4)-VAR(1))*GLD
CALL FUNC(VAR(2),V)
FUNV(2)=V

```

```
ENDIF
GO TO 10
99 WRITE(6,122)
WRITE(6,123) FUNMI,ALPHA,BETA,DEL,VARM
122 FORMAT(1X,'MIN. VALUE ', ' ALPHA ', ' BETA ',
& ' DELTA ', 'MIN. VARIBAL')
123 FORMAT(1X,6E12.5)
WRITE(6,123) VARM
GO TO 201
124 FORMAT('MIN. VALUE AT INITIAL OR AT FINAL POINTS')
100 WRITE(6,124)
201 RETURN
END
```

```

C *****
C * SUBROUTINE FUNC (YMODU,Y) *
C * This subroutine calls the linear finite element program to *
C * evaluate the system for a given Young's modulus *
C * SUBROUTINES included: *
C * STIPPS: MODPS, SFR2, JACOB2, BMATPS, DBE; *
C * LOADPS: MODPS, SFR2, JACOB2; *
C * FRONT; *
C *****
      SUBROUTINE FUNC (YMODU,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /CONTO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&      NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&      NPROB
      COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&      LNODS(416,8),MATNO(416)
      COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&      STRIN(3,3744),NOFIX(208),IFPRE(208,2)
      COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&      DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
&      SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&      NEROR(24)
      COMMON /WT/ XLOAD,YCOMP,NPONE,KLOAD
      PROPS(1,1)=YMODU
      OPEN(7,FILE='SCRTH7.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
&      FORM='UNFORMATTED')
      OPEN(8,FILE='SCRTH8.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
&      FORM='UNFORMATTED')
      OPEN(9,FILE='SCRTH9.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
&      FORM='UNFORMATTED')
      OPEN(10,FILE='SCRTH1.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
&      FORM='UNFORMATTED')
      WRITE(*,822)
822 FORMAT(/' CALCULATING ELEMENT STIFFNESS MATRICES '/')
      CALL STIPPS
      DO 10 ICASE=1,NCASE
      WRITE(*,824)
824 FORMAT(/' CALCULATING LOADING VECTOR '/')
      IF (KLOAD.NE.0) GO TO 1001
      CALL LOADPS
1001 CONTINUE
      WRITE(*,826)
826 FORMAT(/' ASSEMBLING AND SOLVING EQUATIONS BY FRONTAL SOLVER/')
      CALL FRONT
      10 CONTINUE

C
C Close files
C
      CLOSE(7,STATUS='DELETE')
      CLOSE(8,STATUS='DELETE')
      CLOSE(9,STATUS='DELETE')
      CLOSE(10,STATUS='DELETE')
      CLOSE(4,STATUS='KEEP')
      CLOSE(5,STATUS='KEEP')
      Y=DABS(YCOMP-XLOAD)/XLOAD
      KLOAD=KLOAD+1
      RETURN
      END

```

```

C *****
C * PROGRAM FEM-S
C * This program is to do the analysis of front and rear tire on
C * terrain under syemmttric loading
C * SUBROUTINES included:
C * INPUT: CHECK1: ECHO;
C * NODEXY, GAUSSQ,
C * CHECK2: ECHO;
C * LOAD: INTERP;
C * STIFPS: MODPS, SFR2, JACOB2, BMATPS, DBE;
C * LOADPS: MODPS, SFR2, JACOB2;
C * FRONT;
C * REINP;
C *****

```

139

```

C IMPLICIT REAL*8 (A-H,O-Z)
C DIMENSION TITLE(12)
C COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
C & NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
C & NPROB
C COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
C & LNODS(416,8),MATNO(416)
C COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
C & STRIN(3,3744),NOFIX(208),IFPRE(208,2)
C COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
C & DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
C & SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
C & NEROR(24)
C COMMON /PARA/ WT(25),FUNC(25,3),DEFL(25),Y(3),PKO,AU,ELARY,ZUO,
C & PU,PKC,PKFI,BB,PN,NITER,NDAT,NESD
C COMMON /STORE/ BSTOR(99),ASTOR(99),PRESN(99),SUMY,YLOAD,
C & NS1(100),NS2(100)
C WRITE(*,800)
C 800 FORMAT(/' INPUT FILE NAME - '/')
C OPEN(5,FILE='INPUT',ACCESS='SEQUENTIAL',STATUS='OLD')
C READ (5,900) NPROB
C 900 FORMAT(I5)
C
C Open output file as unit 4
C
C WRITE(*,802)
C 802 FORMAT(/' OUTPUT FILE NAME - '/')
C OPEN(4,FILE='OUTPUT',ACCESS='SEQUENTIAL',STATUS='NEW')
C WRITE(*,1003)
C 1003 FORMAT (/ ' ITERATION NUMBER - '/')
C READ(5,*) NITER,IJK
C READ(5,*) PKC,PKFI,BB,PN,NDAT,YLOAD,NESD,PKO,AU,ELARY,ZUO
C DO 801 I=1,NDAT
C 801 READ(5,*) DEFL(I),FUNC(I,2),FUNC(I,1),WT(I)
C PU=(AU*ZUO+PKO)*ELARY
C DO 213 IUY=1,99
C ASTOR(IUY)=0.D0
C PRESN(IUY)=0.D0
C 213 BSTOR(IUY)=0.D0
C KP=0.
C KQ=-4
C DO 4 I=1,100
C KK=(-1)**I
C KL=9
C KO=5

```

```

IF(KK.LT.0) KO=9
IF(KK.LT.0) KL=5
KP=KP+KO
KQ=KQ+KL
NS1(I)=KP
NS2(I)=KQ
4 CONTINUE
WRITE(4,905) NPROB
905 FORMAT(1H0,5X,'TOTAL NUMBER OF PROBLEMS = ',I5)
DO 20 IPROB=1,NPROB
READ (5,910) TITLE
910 FORMAT (12A4)
WRITE(4,915) IPROB,TITLE
915 FORMAT (//////,6X,'PROBLEM NO. ',I3,10X,12A4)
WRITE(*,820)
820 FORMAT(/' READING INPUT DATA '/')
CALL INPUT
ITER=0
ITER0=0
1111 ITER=ITER+1
ITER0=ITER0+1
CALL LOAD(ITER)
OPEN(7,FILE='SCRTH7.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
OPEN(8,FILE='SCRTH8.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
OPEN(9,FILE='SCRTH9.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
OPEN(10,FILE='SCRTH1.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
& FORM='UNFORMATTED')
WRITE(*,822)
822 FORMAT(/' CALCULATING ELEMENT STIFFNESS MATRICES '/')
CALL STIFPS
DO 10 ICASE=1,NCASE
WRITE(*,824)
824 FORMAT(/' CALCULATING LOADING VECTOR '/')
CALL LOADPS(ITER)
WRITE(*,826)
826 FORMAT(/' ASSEMBLING AND SOLVING EQUATIONS BY FRONTAL SOLVER/')
CALL FRONT
WRITE(*,830) ITER
830 FORMAT(/' IN ITERATION PROCEDURE NO.',I5,/)
CALL REINP
10 CONTINUE
C
C Close files
C
CLOSE(7,STATUS='DELETE')
CLOSE(8,STATUS='DELETE')
CLOSE(9,STATUS='DELETE')
CLOSE(10,STATUS='DELETE')
IF(ITER.EQ.NITER.OR.SUMY.GE.YLOAD) GO TO 2001
CLOSE(4,STATUS='DELETE')
OPEN(4,FILE='OUTPUT',ACCESS='SEQUENTIAL',STATUS='NEW')
IF(ITER0.GT.IJK) THEN
KJI=ITER0-IJK
GO TO (700,701,702,703,704,705,706,707,708,709),KJI
700 OPEN(30,FILE='A',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 50 IPOIN=1,NPOIN
50 WRITE(30,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(30,*) 'SUMY=',SUMY

```

```

WRITE(30,*) 'WIDTH=',PROPS(1,3)
CLOSE(30,STATUS='KEEP')
GO TO 3000
701 OPEN(31,FILE='B',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 51 IPOIN=1,NPOIN
51 WRITE(31,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(31,*) 'SUMY=',SUMY
WRITE(31,*) 'WIDTH=',PROPS(1,3)
CLOSE(31,STATUS='KEEP')
GO TO 3000
702 OPEN(32,FILE='C',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 52 IPOIN=1,NPOIN
52 WRITE(32,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(32,*) 'SUMY=',SUMY
WRITE(32,*) 'WIDTH=',PROPS(1,3)
CLOSE(32,STATUS='KEEP')
GO TO 3000
703 OPEN(33,FILE='D',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 53 IPOIN=1,NPOIN
53 WRITE(33,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(33,*) 'SUMY=',SUMY
WRITE(33,*) 'WIDTH=',PROPS(1,3)
CLOSE(33,STATUS='KEEP')
GO TO 3000
704 OPEN(34,FILE='E',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 54 IPOIN=1,NPOIN
54 WRITE(34,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(34,*) 'SUMY=',SUMY
WRITE(34,*) 'WIDTH=',PROPS(1,3)
CLOSE(34,STATUS='KEEP')
GO TO 3000
705 OPEN(35,FILE='F',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 55 IPOIN=1,NPOIN
55 WRITE(35,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(35,*) 'SUMY=',SUMY
WRITE(35,*) 'WIDTH=',PROPS(1,3)
CLOSE(35,STATUS='KEEP')
GO TO 3000
706 OPEN(36,FILE='G',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 56 IPOIN=1,NPOIN
56 WRITE(36,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(36,*) 'SUMY=',SUMY
WRITE(36,*) 'WIDTH=',PROPS(1,3)
CLOSE(36,STATUS='KEEP')
GO TO 3000
707 OPEN(37,FILE='H',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 57 IPOIN=1,NPOIN
57 WRITE(37,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(37,*) 'SUMY=',SUMY
WRITE(37,*) 'WIDTH=',PROPS(1,3)
CLOSE(37,STATUS='KEEP')
GO TO 3000
708 OPEN(38,FILE='I',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 58 IPOIN=1,NPOIN
58 WRITE(38,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
WRITE(38,*) 'SUMY=',SUMY
WRITE(38,*) 'WIDTH=',PROPS(1,3)
CLOSE(38,STATUS='KEEP')
GO TO 3000
709 OPEN(39,FILE='J',ACCESS='SEQUENTIAL',STATUS='NEW')
DO 59 IPOIN=1,NPOIN

```

```
59 WRITE(39,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
   WRITE(39,*) 'SUMY=',SUMY
   WRITE(39,*) 'WIDTH=',PROPS(1,3)
   CLOSE(39,STATUS='KEEP')
   GO TO 3000
```

142

```
   ELSE
   CONTINUE
   ENDIF
3000 CONTINUE
   GO TO 1111
2001 CONTINUE
   DO 60 IPOIN=1,NPOIN
   60 WRITE(4,935) IPOIN,(COORD(IPOIN,IDIME),IDIME=1,NDIME)
   935 FORMAT (1X,I5,3F11.6)
```

C

```
   WRITE(4,1002)
   WRITE(4,1001) ITER
1002 FORMAT(1X,'ITERATION NUMBER IS')
1001 FORMAT(1X,I5)
   CLOSE(4,STATUS='KEEP')
   CLOSE(5,STATUS='KEEP')
20 CONTINUE
   STOP
   END
```

```

C *****
C * SUBROUTINE LOADPS(ITER) *
C * This subroutine inputs the distributed loads from Bekker's *
C * equation and stores them in each increment *
C * SUBROUTINES included: *
C * MODPS, SFR2, JACOB2. *
C *****

SUBROUTINE LOADPS(ITER)
  IMPLICIT REAL*8 (A-H,O-Z)
  DIMENSION TITLE(12),POINT(2),PRESS(3,2),PGASH(2),
& DGASH(2),TEMPE(500),STRAN(3),STRES(3),NOPRS(3)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
& NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
& NPROB
  COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
& LNODS(416,8),MATNO(416)
  COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
& STRIN(3,3744),NOFIX(208),IFPRE(208,2)
  COMMON /WORK/ ELOAD(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPOOD(2,9),
& NEROR(24)
  COMMON /PARA/ WT(25),FUNC(25,3),DEFL(25),Y(3),PKO,AU,ELARY,ZU0,
& PU,PKC,PKFI,BB,PN,NITER,NDAT,NESD
  COMMON /STORE/ BSTOR(99),ASTOR(99),PRESN(99),SUMYLOAD,
& NS1(100),NS2(100)
  DO 10 IELEM=1,NELEM
  DO 10 IEVAB=1,NEVAB
  10 ELOAD(IELEM,IEVAB)=0.DO
  IF(ITER.NE.1) GO TO 1234
  READ (5,900) TITLE
  900 FORMAT(12A4)
  WRITE(4,905) TITLE,ICASE
  905 FORMAT('0',12A4,5X,'LOAD CASE =',I3)
  READ (5,910) IPLOD,IGRAV,IEDGE,ITEMP
  WRITE(4,910) IPLOD,IGRAV,IEDGE,ITEMP
  910 FORMAT(4I5)
  IF (IPLOD .EQ. 0) GO TO 500
  20 READ(5,915) LODPT,(POINT(IDOBN),IDOBN=1,NDOBN)
  WRITE(4,915) LODPT,(POINT(IDOBN),IDOBN=1,NDOBN)
  915 FORMAT(I5,2F10.3)
  DO 30 IELEM=1,NELEM
  DO 30 INODE=1,NNODE
  NLOCA=LNODS(IELEM,INODE)
  IF (LODPT .EQ. NLOCA) GO TO 40
  30 CONTINUE
  40 DO 50 IDOBN=1,NDOBN
  NGASH=(INODE-1)*NDOBN+IDOBN
  50 ELOAD(IELEM,NGASH)=POINT(IDOBN)
  IF (LODPT .LT. NPOIN) GO TO 20
  500 CONTINUE
  IF (IGRAV .EQ.0) GO TO 600
  READ(5,920) THETA,GRAVY
  920 FORMAT(2F10.3)
  WRITE(4,925) THETA,GRAVY
  925 FORMAT(1H0,16HGRAVITY ANGLE = ,F10.3,
& 20H GRAVITY CONSTANT = ,F10.3)
  THETA=THETA/57.2957794DO/514.DO
  DO 90 IELEM=1,NELEM

```



```

LPROP=MATNO(IELEM)
THICK=PROPS(LPROP,3)
DENSE=PROPS(LPROP,4)
IF(DENSE.EQ.0.D0) GO TO 90
GXCOM=DENSE*GRAVY*DSIN(THETA)
GYCOM=-DENSE*GRAVY*DCOS(THETA)
DO 60 INODE=1,NNODE
LNODE=LNODS(IELEM,INODE)
DO 60 IDIME=1,NDIME
60 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
DO 80 IGAUS=1,NGAUS
DO 80 JGAUS=1,NGAUS
EXISP=POSGP(IGAUS)
ETASP=POSGP(JGAUS)
CALL SFR2(EXISP,ETASP)
KGASP=1
CALL JACOB2(IELEM,DJACB,KGASP)
DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
IF(THICK.NE.0.D0) DVOLU=DVOLU*THICK
DO 70 INODE=1,NNODE
NGASH=(INODE-1)*NDOFN+1
MGASH=(INODE-1)*NDOFN+2
ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)+
& GXCOM*SHAPE(INODE)*DVOLU
70 ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)+
& GYCOM*SHAPE(INODE)*DVOLU
80 CONTINUE
90 CONTINUE
600 CONTINUE
IF( IEDGE .EQ.0) GO TO 700
1234 CONTINUE
NEDGE=(ITER+1)/2
WRITE(4,935) NEDGE
935 FORMAT (1H0,5X,'NO. OF LOADED EDGES = ',I5)
WRITE(4,940)
940 FORMAT (1H0,5X,'LIST OF LOADED EDGES AND APPLIED LOADS')
NODEG=3
DO 160 IEDGE=1,NEDGE
DO 1 I=1,NODEG
NOPRS(I)=NS1(2*IEDGE+I-2)
PRESS(I,1)=PRESN(2*IEDGE+I-2)-ASTOR(2*IEDGE+I-2)
1 PRESS(I,2)=0.D0
C
NEASS=4*IEDGE
WRITE(4,950) NEASS,(NOPRS(IODEG),IODEG=1,NODEG)
950 FORMAT (I10,5X,3I5)
WRITE(4,956) ((PRESS(IODEG,IDOFN),IODEG=1,NODEG),
& IDOFN=1,NDOFN)
956 FORMAT (6F10.3)
ETASP=-1.D0
DO 100 IODEG=1,NODEG
LNODE=NOPRS(IODEG)
DO 100 IDIME=1,NDIME
100 ELCOD(IDIME,IODEG)=COORD(LNODE,IDIME)
DO 150 IGAUS=1,NGAUS
EXISP=POSGP(IGAUS)
CALL SFR2(EXISP,ETASP)
DO 110 IDOFN=1,NDOFN
PGASH(IDOFN)=0.D0
DGASH(IDOFN)=0.D0

```

```

DO 110 IODEG=1,NODEG
PGASH(IDOFN)=PGASH(IDOFN)+PRESS(IODEG,IDOFN)*SHAPE(IODEG)
110 DGASH(IDOFN)=DGASH(IDOFN)+ELCOD(IDOFN,IODEG)*DERIV(1,IODEG)
DVOLUM=WEIGP(IGAUS)
PXCOM=DGASH(1)*PGASH(2)-DGASH(2)*PGASH(1)
PYCOM=DGASH(1)*PGASH(1)+DGASH(2)*PGASH(2)
DO 120 INODE=1,NNODE
NLOCA=LNODS(NEASS,INODE)
IF (NLOCA.EQ. NOPRS(1)) GO TO 130
120 CONTINUE
130 JNODE=INODE+NODEG-1
KOUNT=0
DO 140 KNODE=INODE,JNODE
KOUNT=KOUNT+1
NGASH=(KNODE-1)*NDOFN+1
MGASH=(KNODE-1)*NDOFN+2
IF (KNODE.GT. NNODE) NGASH=1
IF (KNODE.GT. NNODE) MGASH=2
ELOAD(NEASS,NGASH)=ELOAD(NEASS,NGASH)+SHAPE(KOUNT)*PXCOM*DVOLUM
140 ELOAD(NEASS,MGASH)=ELOAD(NEASS,MGASH)+SHAPE(KOUNT)*PYCOM*DVOLUM
150 CONTINUE
160 CONTINUE
700 CONTINUE
C
DO 701 III=1,ITER
701 ASTOR(III)=PRESN(III)
C
IF (ITEMP.EQ. 0) GO TO 800
DO 170 IPOIN=1,NPOIN
170 TEMPE(IPOIN)=0.D0
WRITE(4,960)
960 FORMAT (1H0,5X,'PRESCRIBED NODAL TEMPERATURES')
180 READ (5,965) NODPT,TEMPE(NODPT)
WRITE(4,965) NODPT,TEMPE(NODPT)
965 FORMAT (I5,F10.3)
IF (NODPT.LT. NPOIN) GO TO 180
MDIME=NDIME+1
KGAST=0
DO 280 IELEM=1,NELEM
LPROP=MATNO(IELEM)
DO 200 INODE=1,NNODE
LNODE=LNODS(IELEM,INODE)
DO 190 IDIME=1,NDIME
190 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
200 ELCOD(MDIME,INODE)=TEMPE(LNODE)
CALL MODPS (LPROP)
YOUNG=PROPS(LPROP,1)
POISS=PROPS(LPROP,2)
THICK=PROPS(LPROP,3)
ALPHA=PROPS(LPROP,5)
DO 270 IGAUS=1,NGAUS
DO 270 JGAUS=1,NGAUS
KGAST=KGAST+1
EXISP=POSGP(IGAUS)
ETASP=POSGP(JGAUS)
CALL SFR2 (EXISP,ETASP)
KGASP=1
CALL JACOB2 (IELEM,DJACB,KGASP)
THERM=0.D0
DO 210 INODE=1,NNODE

```

```

210 THERM=THERM+ELCOD(MDIME,INODE)*SHAPE(INODE)
    DVOLUM=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
    IF (THICK.NE. 0.D0) DVOLUM=DVOLUM*THICK
    EIGEN=THERM*ALPHA
    IF (NTYPE.EQ. 2) GO TO 220
    STRAN(1)=-EIGEN
    STRAN(2)=-EIGEN
    STRAN(3)=0.D0
    GO TO 230
220 STRAN(1)=-(1.D0+POISS)*EIGEN
    STRAN(2)=-(1.D0+POISS)*EIGEN
    STRAN(3)=0.D0
230 DO 250 ISTORE=1,NSTRE
    STRES(ISTRE)=0.D0
    DO 240 JSTORE=1,NSTRE
240 STRES(ISTRE)=STRES(ISTRE)+DMATX(ISTRE,JSTORE)*STRAN(JSTORE)
250 STRIN(ISTRE,KGAST)=STRES(ISTRE)
    IF (NTYPE.EQ. 2) STRIN(4,KGAST)=-YOUNG*EIGEN
    IF (NTYPE.EQ. 1) STRIN(4,KGAST)=0.D0
    DO 260 INODE=1,NNODE
    NGASH=(INODE-1)*NDOFN+1
    MGASH=(INODE-1)*NDOFN+2
    ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)-(CARTD(1,INODE)*
&      STRES(1)+CARTD(2,INODE)*STRES(3))*DVOLUM
260 ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)-(CARTD(1,INODE)*
&      STRES(3)+CARTD(2,INODE)*STRES(2))*DVOLUM
270 CONTINUE
280 CONTINUE
800 CONTINUE
    NNN=(ITER+1)*2
    WRITE(4,970)
970 FORMAT ('0',5X,'TOTAL NODAL FORCES FOR EACH ELEMENT')
    DO 290 IELEM=4,NNN,4
290 WRITE(4,975) IELEM,(ELOAD(IELEM,IEVAB),IEVAB=1,NEVAB)
975 FORMAT (1X,I4,5X,8E12.4/(10X,8E12.4))
    RETURN
    END

```

```

C *****
C * SUBROUTINE LOAD(ITER)
C * This subroutine calculates the normal pressure distribution
C * according to Bekker's pressure-sinkage relationship, and check
C * the normal load and updates the Young's modulus.
C * SUBROUTINES included:
C * INTERP.
C *****

```

147

```

SUBROUTINE LOAD(ITER)
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
&   NPROB
  COMMON /GEOMDA/COORD(740,2),PROPS(10,5),
&   LNODS(210,8),MATNO(210)
  COMMON /LDATA/ PRESC(130,2),ASDIS(1480),ELOAD(210,16),
&   STRIN(3,1890),NOFIX(130),IFPRE(130,2)
  COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
  COMMON /PARA/ WT(25),FUNC(25,3),DEFL(25),Y(3),PKO,AU,ELARY,ZU0,
&   PU,PKC,PKFI,BB,PN,NITER,NDAT,NESD
  COMMON /STORE/ BSTOR(99),ASTOR(99),PRESN(99),SUMY,YLOAD,
&   NS1(100),NS2(100)
  NSMAX=NS1(ITER+1)
  DO 777 IEGP=1,ITER
    KSD=NS1(IEGP)
    777 PRESN(IEGP)=(PKC/BB+PKFI)*(COORD(NSMAX,2)-COORD(KSD,2))*PN
    SUMY=0.D0
    DO 888 IOPS=1,ITER
      KSD1=NS1(IOPS)
      KSD2=NS1(IOPS+1)
      SUMY=SUMY+0.5D0*PROPS(1,3)*(PRESN(IOPS)+PRESN(IOPS+1))*
&   (COORD(KSD2,1)-COORD(KSD1,1))
    888 CONTINUE
    NEDGE=NLODP/2
    CLEN=0.D0
    DO 222 IOPS=1,ITER
      KSD1=NS1(IOPS)
      KSD2=NS1(IOPS+1)
      CLEN=CLEN+DSQRT((COORD(KSD2,1)-COORD(KSD1,1))**2
&   +(COORD(KSD2,2)-COORD(KSD1,2))**2)
    222 CONTINUE
    SUMY=2.D0*SUMY
    CLEN=2.D0*CLEN
    WRITE(4,*) 'SUMY=',SUMY
    WRITE(4,*) 'CLEN',CLEN
    CALL INTERP (NDAT,2,SUMY,WT,FUNC,Y)
    PROPS(1,1)=Y(1)
    NSMIN=2*NESD+1
    DEFM=COORD(NSMIN,2)
    WRITE(4,*) 'DEFM=',COORD(NSMIN,2)
    CALL INTERP (NDAT,2,DEFM,DEFL,FUNC,Y)
    PROPS(1,3)=Y(2)
    BB=DMIN1(PROPS(1,3),CLEN)
    WRITE(4,*) 'WIDTH',PROPS(1,3)
    WRITE(4,*) 'MODU',PROPS(1,1)
    RETURN
  END

```

```

C *****
C * SUBROUTINE LOAD(ITER) *
C * This subroutine calculates the normal pressure distribution *
C * according to Bekker's pressure-sinkage relationship, and check *
C * the normal load and updates the Young's modulus. *
C * SUBROUTINES included: *
C * INTERP. *
C *****
SUBROUTINE LOAD(ITER)
  IMPLICIT REAL*8 (A-H,O-Z)
  COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NITYPE,
& NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
& NPROB
  COMMON /GEOMDA/COORD(740,2),PROPS(10,5),
& LNODS(210,8),MATNO(210)
  COMMON /LDATA/ PRESC(130,2),ASDIS(1480),ELOAD(210,16),
& STRIN(3,1890),NOFIX(130),IFPRE(130,2)
  COMMON /WORK/ ELOAD(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARID(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
& NEROR(24)
  COMMON /PARA/ WT(25),FUNC(25,3),DEFL(25),Y(3),PKO,AU,ELARY,ZU0,
& PU,PKC,PKFI,BB,PN,NITER,NDAT,NESD
  COMMON /STORE/ BSTOR(99),ASTOR(99),PRESN(99),SUMY,YLOAD,
& NS1(100),NS2(100)
  NSMAX=NS1(ITER+1)
  DQ 777 IEGP=1,ITER
  KSD=NS1(IEGP)
777 PRESN(IEGP)=(PKC/BB+PKFI)*(COORD(NSMAX,2)-COORD(KSD,2))**PN
  SUMY=0.DO
  DO 888 IOPS=1,ITER
  KSD1=NS1(IOPS)
  KSD2=NS1(IOPS+1)
  SUMY=SUMY+0.5D0*PROPS(1,3)*(PRESN(IOPS)+PRESN(IOPS+1))*
& (COORD(KSD2,1)-COORD(KSD1,1))
888 CONTINUE
  NEDGE=NLODP/2
  CLEN=0.DO
  DO 222 IOPS=1,ITER
  KSD1=NS1(IOPS)
  KSD2=NS1(IOPS+1)
  CLEN=CLEN+DSQRT((COORD(KSD2,1)-COORD(KSD1,1))**2
& +(COORD(KSD2,2)-COORD(KSD1,2))**2)
222 CONTINUE
  SUMY=2.DO*SUMY
  CLEN=2.DO*CLEN
  WRITE(4,*) 'SUMY=',SUMY
  WRITE(4,*) 'CLEN',CLEN
  CALL INTERP (NDAT,2,SUMY,WT,FUNC,Y)
  PROPS(1,1)=Y(1)
  NSMIN=2*NESD+1
  DEFM=COORD(NSMIN,2)
  WRITE(4,*) 'DEFM=',COORD(NSMIN,2)
  CALL INTERP (NDAT,2,DEFM,DEFL,FUNC,Y)
  PROPS(1,3)=Y(2)
  BB=DMIN1(PROPS(1,3),CLEN)
  WRITE(4,*) 'WIDTH',PROPS(1,3)
  WRITE(4,*) 'MODU',PROPS(1,1)
  RETURN
END

```

```

C *****
C *   PROGRAM FEM-F
C *   This program is to do the analysis of the tire model on terrain
C *   under asymmetric loading.
C *   SUBROUTINES included:
C *       INPUT: CHECK1: ECHO;
C *             NODEXY, GAUSSQ,
C *             CHECK2: ECHO;
C *       LOAD: INTERP;
C *       STIEPS: MODPS, SFR2, JACOB2, BMATPS, DBE;
C *       LOADPS: MODPS, SFR2, JACOB2;
C *       FRONT;
C *       REINP;
C *****
C   IMPLICIT REAL*8 (A-H,O-Z)
C   DIMENSION TITLE(12)
C   COMMON /CONTROL/ NPOIN, NELEM, NNODE, NDOFN, NDIME, NSTRE, NTYPE,
C   & NGAUS, NPROP, NMATS, NVFIX, NEVAB, ICASE, NCASE, ITEMP, IPROB,
C   & NPROB
C   COMMON /GEOMDA/COORD(1456,2), PROPS(10,5),
C   & LNODS(416,8), MATNO(416)
C   COMMON /LDATA/ PRESC(208,2), ASDIS(2912), ELOAD(416,16),
C   & STRIN(3,3744), NOFIX(208), IFPRE(208,2)
C   COMMON /WORK/ ELOD(2,8), SHAPE(8), DERIV(2,8),
C   & DMATX(3,3), CARID(2,8), DBMAT(3,16), BMATX(3,16),
C   & SMATX(3,16,9), POSGP(3), WEIGP(3), GPCOD(2,9),
C   & NEROR(24)
C   COMMON /PARA/ WT(25), DEFL(25), FUNC(25,3), Y(3), CLEN, SUMXR, RBAR,
C   & PKC, PKFI, BB, PN, AU, PKO, C, PHI, PKJ, SLIP, R, YLOAD, SUMY, SUMXF, DEFM,
C   & ASTOR(160), PRESN(160), BSTOR(160), NSTOR(160), SHEAR(160), DISJ(160)
C   COMMON /NDATA/ NX0, NMAXE, NESD, NEGP, NOTE, NX1, NMINO, NSMIN, NEGP1,
C   & NMIN, NEGE1, NEDGE, ITER, NITER, NLODP, KEEP, NCTRL, NADD, NOON
C
C   WRITE(*,800)
800 FORMAT(/' INPUT FILE NAME - '/')
C   OPEN(5, FILE='INPUT1', ACCESS='SEQUENTIAL', STATUS='OLD')
C   READ (5,900) NPROB
900 FORMAT(I5)
C
C   Open output file as unit 4
C
C   WRITE(*,802)
802 FORMAT(/' OUTPUT FILE NAME - '/')
C   OPEN(4, FILE='OUTPUT', ACCESS='SEQUENTIAL', STATUS='NEW')
C   OPEN(11, FILE='INPUT0', ACCESS='SEQUENTIAL', STATUS='OLD')
C   READ(11,*) PKC, PKFI, BB, PN, AU, PKO, YLOAD, C, R, PHI, PKJ, SLIP, RBAR
C   READ(11,*) NX0, NX1, NMAXE, NESD, NEGP, NITER
C   CLOSE(11, STATUS='KEEP')
C   DO 701 I=1, NX0
701 READ(5,*) DEFL(I), FUNC(I,2), FUNC(I,1), WT(I)
C   DO 703 I=1, NX1
C   BSTOR(I)=0.D0
C   NSTOR(I)=0
703 ASTOR(I)=0.D0
C
C   WRITE(4,905) NPROB
905 FORMAT(1H0,5X,'TOTAL NUMBER OF PROBLEMS = ',I5)
C   DO 20 IPROB=1,NPROB
C   READ (5,910) TITLE

```

```

910 FORMAT (12A4)
    WRITE(4,915) IPROB,TITLE
915 FORMAT (//////,6X,'PROBLEM NO. ',I3,10X,12A4)
    WRITE(*,820)
820 FORMAT(/'  READING INPUT DATA  '/')
    CALL INPUT
    ITER=0
    NCON=0
    SUMY=0.D0
    NOTE=NEGP
1111 ITER=ITER+1
    CALL LOAD
    IF(NCTRL.EQ.1) GO TO 2001
    OPEN(7,FILE='SCRTH7.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
    &    FORM='UNFORMATTED')
    OPEN(8,FILE='SCRTH8.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
    &    FORM='UNFORMATTED')
    OPEN(9,FILE='SCRTH9.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
    &    FORM='UNFORMATTED')
    OPEN(10,FILE='SCRTH1.DAT',ACCESS='SEQUENTIAL',STATUS='NEW',
    &    FORM='UNFORMATTED')
    WRITE(*,822)
822 FORMAT(/'  CALCULATING ELEMENT STIFFNESS MATRICES  '/')
    CALL STIFPS
    DO 10 ICASE=1,NCASE
    WRITE(*,824)
824 FORMAT(/'  CALCULATING LOADING VECTOR  '/')
    CALL LOADPS
    WRITE(*,826)
826 FORMAT(/'  ASSEMBLING AND SOLVING EQUATIONS BY FRONTAL SOLVER  '/')
    CALL FRONT
    WRITE(*,830) ITER
830 FORMAT(/'  IN ITERATION PROCEDURE NO. ',I5/)
    10 CONTINUE
C
C    Close files
C
    CLOSE(7,STATUS='DELETE')
    CLOSE(8,STATUS='DELETE')
    CLOSE(9,STATUS='DELETE')
    CLOSE(10,STATUS='DELETE')
    CALL REINP
    CLOSE(4,STATUS='DELETE')
    OPEN(4,FILE='OUTPUT',ACCESS='SEQUENTIAL',STATUS='NEW')
    GO TO 1111
2001 CONTINUE
    OPEN (14,FILE='OUTF',ACCESS='SEQUENTIAL',STATUS='NEW')
    WRITE(14,*) 'CONTACT LENGTH'
    CLEN=0.D0
    DO 222 I=1,NLODP
    KSD1=NSTOR(NMIN-NEGE1+I)
    KSD2=NSTOR(NMIN-NEGE1+I+1)
    CLEN=CLEN+DSQRT((COORD(KSD2,1)-COORD(KSD1,1))**2
    & +(COORD(KSD2,2)-COORD(KSD1,2))**2)
    IF(COORD(KSD1,1).LE.0.D0.AND.COORD(KSD2,1).GE.0.D0) CLEN0=CLEN
    WRITE(14,*) CLEN
222 CONTINUE
    WRITE(14,*) 'CLEN0=',CLEN0
    WRITE(14,*) 'PRESN(I)      SHEAR(I) '
    DO 8975 I=1,NLODP

```

151

```

      KGO=NLODP+1-I
8975 WRITE(14,217) PRESN(KGO),SHEAR(KGO),
217  FORMAT(1X,2(E12.5,3X))
      IFRON=NSTOR(NMIN+NEGP1)
      IREAR=NSTOR(NMIN+1-NEGE1)
      SINK=COORD(IFRON,2)-COORD(NSMIN,2)
      ELARY=COORD(IREAR,2)-COORD(NSMIN,2)
      WRITE(14,*) '      COORDINATES'
      WRITE(14,141)
141  FORMAT(1X,'IFRON',6X,'NSMIN',6X,'IREAR')
      WRITE(14,142) COORD(IFRON,1),COORD(NSMIN,1),COORD(IREAR,1)
142  FORMAT(1X,'X-COORD:',4F11.6)
      WRITE(14,143) COORD(IFRON,2),COORD(NSMIN,2),COORD(IREAR,2)
143  FORMAT(1X,'Y-COORD:',4F11.6)
      WRITE(14,144) CLEN,PROPS(1,3),NSMIN,NMIN
144  FORMAT(1X,'CLEN=',F9.6,2X,'WIDTH=',F9.6,2X,'NSMIN=',I5,
      & ,2X,'NMIN=',I5)
      WRITE(14,145) SUMY,SUMXF,SUMXR
145  FORMAT(1X,'SUMY=',F12.4,2X,'SUMXF=',F12.4,2X,'SUMXR=',F12.4)
      WRITE(14,146) ELARY,SINK
146  FORMAT(1X,'ELARY=',F10.7,2X,'SINK=',F9.6)
      DO 50 I=1,NPOIN
50  WRITE(14,935) I,(COORD(I,ID),ID=1,NDIME)
935  FORMAT (1X,I5,3F12.6)
C
      CLOSE(14,STATUS='KEEP')
      CLOSE(4,STATUS='KEEP')
      CLOSE(5,STATUS='KEEP')
20  CONTINUE
      STOP
      END

```



```

C *****
C * SUBROUTINE LOADPS *
C * This subroutine inputs the normal pressure distribution *
C * and the shear stress distribution and stores them in each *
C * increment. *
C * SUBROUTINES included: *
C * MODPS, SFR2, JACOB2. *
C *****

```

152

```

SUBROUTINE LOADPS
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION TITLE(12),POINT(2),PRESS(3,2),PGASH(2),
& DGASH(2),TEMPE(500),STRAN(3),STRES(3),NOPRS(3)
COMMON /CONTRO/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
& CAUS,NPROP,NMATS,NVFLX,NEVAB,ICASE,NCASE,ITEMP,IPROB,
& NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
& LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
& STRIN(3,3744),NOFX(208),IFPRE(208,2)
COMMON /WORK/ ELCOD(2,8),SHAPE(8),DERIV(2,8),
& DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
& SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
& NEROR(24)
COMMON /PARA/ WT(25),DEFL(25),FUNC(25,3),Y(3),CLEN,SUMXR,RBAR,
& PKC,PKFI,BB,PN,AU,PKO,C,PHI,PKJ,SLIP,R,YLOAD,SUMY,SUMXF,DEFM,
& ASTOR(160),PRESN(160),BSTOR(160),NSTOR(160),SHEAR(160),DISJ(160)
COMMON /NDATA/ NX0,NMAXE,NESD,NEGP,NOTE,NX1,NMINO,NSMIN,NEGP1,
& NMIN,NEGEL,NEDGE,ITER,NITER,NLODP,KEEP,NCTRL,NADD,NCON
DO 10 IELEM=1,NELEM
DO 10 IEVAB=1,NEVAB
10 ELOAD(IELEM,IEVAB)=0.DO
IF(ITER.GT.1) GO TO 1234
READ (5,900) TITLE
900 FORMAT (12A4)
WRITE(4,905) TITLE,ICASE
905 FORMAT ('0',12A4,5X,'LOAD CASE =',I3)
READ (5,910) IPLOD,IGRAV,IEDGE,ITEMP
WRITE(4,910) IPLOD,IGRAV,IEDGE,ITEMP
910 FORMAT (4I5)
IF (IPLOD.EQ. 0) GO TO 500
20 READ(5,915) LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
WRITE(4,915) LODPT,(POINT(IDOFN),IDOFN=1,NDOFN)
915 FORMAT (I5,2F10.3)
DO 30 IELEM=1,NELEM
DO 30 INODE=1,NNODE
NLOCA=LNODS(IELEM,INODE)
IF (LODPT.EQ. NLOCA) GO TO 40
30 CONTINUE
40 DO 50 IDOFN=1,NDOFN
NGASH=(INODE-1)*NDOFN+IDOFN
50 ELOAD(IELEM,NGASH)=POINT(IDOFN)
IF (LODPT.LT. NPOIN) GO TO 20
500 CONTINUE
IF (IGRAV.EQ.0) GO TO 600
READ(5,920) THETA,GRAVY
920 FORMAT(2F10.3)
WRITE(4,925) THETA,GRAVY
925 FORMAT(1H0,16HGRAVITY ANGLE = ,F10.3,
& 20H GRAVITY CONSTANT = ,F10.3)

```

```

    THETA=THETA/57.2957794D0/514.D0
    DO 90 IELEM=1,NELEM
    LPROP=MATNO(IELEM)
    THICK=PROPS(LPROP,3)
    DENSE=PROPS(LPROP,4)
    IF(DENSE.EQ.0.D0) GO TO 90
    GXCOM=DENSE*GRAVY*DSIN(THETA)
    GYCOM=-DENSE*GRAVY*DCOS(THETA)
    DO 60 INODE=1,NNODE
    LNODE=LNODS(IELEM,INODE)
    DO 60 IDIME=1,NDIME
60  ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
    DO 80 IGAUS=1,NGAUS
    DO 80 JGAUS=1,NGAUS
    EXISP=POSGP(IGAUS)
    ETASP=POSGP(JGAUS)
    CALL SFR2(EXISP,ETASP)
    KGASP=1
    CALL JACOB2(IELEM,DJACB,KGASP)
    DVOLU=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
    IF(THICK.NE.0.D0) DVOLU=DVOLU*THICK
    DO 70 INODE=1,NNODE
    NGASH=(INODE-1)*NDOFN+1
    MGASH=(INODE-1)*NDOFN+2
    ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)+
    & GXCOM*SHAPE(INODE)*DVOLU
70  ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)+
    & GYCOM*SHAPE(INODE)*DVOLU
    80 CONTINUE
    90 CONTINUE
600 CONTINUE
    IF ( IEDGE.EQ.0) GO TO 700
1234 CONTINUE
    NODEG=3
    DO 160 IEDGE=1,NEDGE
    DO 1 I=1,NODEG
    KILL=2*IEDGE-2+I
    KOOL=NODEG-I+1
    NOPRS(KOOL)=NSTOR(NMIN+NEGP1-KILL+1)
    PRESS(KOOL,1)=PRESN(KILL)-ASTOR(KILL)
    PRESS(KOOL,2)=SHEAR(KILL)-BSTOR(KILL)
    IF(PRESS(KOOL,1).GT.0.D0) GO TO 1
    PRESS(KOOL,1)=0.D0
    PRESS(KOOL,2)=0.D0
    1 CONTINUE
    NEASS=NESD*(NOPRS(2)/(3*NESD+2))
C
    WRITE(4,950) NEASS,(NOPRS(IDEG),IDEG=1,NODEG)
950  FORMAT (I10,5X,3I5)
    WRITE(4,956) ((PRESS(IDEG,IDOFN),IDEG=1,NODEG),
    & IDOFN=1,NDOFN)
956  FORMAT (6E12.5)
    ETASP=-1.D0
    DO 100 IDEG=1,NODEG
    LNODE=NOPRS(IDEG)
    DO 100 IDIME=1,NDIME
100  ELCOD(IDIME,IDEG)=COORD(LNODE,IDIME)
    DO 150 IGAUS=1,NGAUS
    EXISP=POSGP(IGAUS)
    CALL SFR2(EXISP,ETASP)

```

```

DO 110 IDOFN=1,NDOFN
PGASH(IDOFN)=0.D0
DGASH(IDOFN)=0.D0
DO 110 IODEG=1,NODEG
PGASH(IDOFN)=PGASH(IDOFN)+PRESS(IODEG,IDOFN)*SHAPE(IODEG)
110 DGASH(IDOFN)=DGASH(IDOFN)+ELCOD(IDOFN,IODEG)*DERIV(1,IODEG)
DVOLUM=WEIGP(IGAUS)
PXCOM=DGASH(1)*PGASH(2)-DGASH(2)*PGASH(1)
PYCOM=DGASH(1)*PGASH(1)+DGASH(2)*PGASH(2)
DO 120 INODE=1,NNODE
NLOCA=LNODS(NEASS,INODE)
IF (NLOCA.EQ. NOPRS(1)) GO TO 130
120 CONTINUE
130 JNODE=INODE+NODEG-1
KOUNT=0
DO 140 KNODE=INODE,JNODE
KOUNT=KOUNT+1
NGASH=(KNODE-1)*NDOFN+1
MGASH=(KNODE-1)*NDOFN+2
IF (KNODE.GT. NNODE) NGASH=1
IF (KNODE.GT. NNODE) MGASH=2
ELOAD(NEASS,NGASH)=ELOAD(NEASS,NGASH)+SHAPE(KOUNT)*PXCOM*
& DVOLUM
140 ELOAD(NEASS,MGASH)=ELOAD(NEASS,MGASH)+
& SHAPE(KOUNT)*PYCOM*DVOLUM
150 CONTINUE
160 CONTINUE
700 CONTINUE

```

C

```

DO 22 IY=1,NX1
ASTOR(IY)=0.D0
22 BSTOR(IY)=0.D0
DO 2 IY=1,NLODP
BSTOR(IY)=SHEAR(IY)
2 ASTOR(IY)=PRESN(IY)
IF (ITEMP.EQ. 0) GO TO 800
DO 170 IPOIN=1,NPOIN
170 TEMPE(IPOIN)=0.D0
WRITE(4,960)
960 FORMAT (1H0,5X,'PRESCRIBED NODAL TEMPERATURES')
180 READ (5,965) NODPT,TEMPE(NODPT)
WRITE(4,965) NODPT,TEMPE(NODPT)
965 FORMAT (15,F10.3)
IF (NODPT.LT. NPOIN) GO TO 180
MDIME=NDIME+1
KGAST=0
DO 280 IELEM=1,NELEM
LPROP=MATNO(IELEM)
DO 200 INODE=1,NNODE
LNODE=LNODS(IELEM,INODE)
DO 190 IDIME=1,NDIME
190 ELCOD(IDIME,INODE)=COORD(LNODE,IDIME)
200 ELCOD(MDIME,INODE)=TEMPE(LNODE)
CALL MODPS (LPROP)
YOUNG=PROPS(LPROP,1)
POISS=PROPS(LPROP,2)
THICK=PROPS(LPROP,3)
ALPHA=PROPS(LPROP,5)
DO 270 IGAUS=1,NGAUS
DO 270 JGAUS=1,NGAUS

```

```

KGAST=KGAST+1
EXISP=POSGP(IGAUS)
ETASP=POSGP(JGAUS)
CALL SFR2 (EXISP,ETASP)
KGASP=1
CALL JACOB2 (IELEM,DJACB,KGASP)
THERM=0.D0
DO 210 INODE=1,NNODE
210 THERM=THERM+ELCOD(MDIME,INODE)*SHAPE(INODE)
    DVOLUM=DJACB*WEIGP(IGAUS)*WEIGP(JGAUS)
    IF (THICK.NE. 0.D0) DVOLUM=DVOLUM*THICK
    EIGEN=THERM*ALPHA
    IF (NTYPE.EQ. 2) GO TO 220
    STRAN(1)=-EIGEN
    STRAN(2)=-EIGEN
    STRAN(3)=0.D0
    GO TO 230
220 STRAN(1)=- (1.D0+POISS)*EIGEN
    STRAN(2)=- (1.D0+POISS)*EIGEN
    STRAN(3)=0.D0
230 DO 250 ISTRE=1,NSTRE
    STRES(ISTRE)=0.D0
    DO 240 JSTRE=1,NSTRE
240 STRES(ISTRE)=STRES(ISTRE)+DMATX(ISTRE,JSTRE)*STRAN(JSTRE)
250 STRIN(ISTRE,KGAST)=STRES(ISTRE)
    IF (NTYPE.EQ. 2) STRIN(4,KGAST)=-YOUNG*EIGEN
    IF (NTYPE.EQ. 1) STRIN(4,KGAST)=0.D0
    DO 260 INODE=1,NNODE
    NGASH=(INODE-1)*NDOFN+1
    MGASH=(INODE-1)*NDOFN+2
    ELOAD(IELEM,NGASH)=ELOAD(IELEM,NGASH)-(CARTD(1,INODE)*
    & STRES(1)+CARTD(2,INODE)*STRES(3))*DVOLUM
260 ELOAD(IELEM,MGASH)=ELOAD(IELEM,MGASH)-(CARTD(1,INODE)*
    & STRES(3)+CARTD(2,INODE)*STRES(2))*DVOLUM
270 CONTINUE
280 CONTINUE
800 CONTINUE
    WRITE(4,970)
970 FORMAT ('0',5X,'TOTAL NODAL FORCES FOR EACH ELEMENT')
    RETURN
END

```

```

C *****
C * SUBROUTINE LOAD *
C * This subroutine calculates the normal pressure distribution and *
C * the shear stress distribution, and check the normal load, and *
C * updates the Young's modulus. *
C * SUBROUTINES included: *
C * INTERP. *
C *****

```

```

SUBROUTINE LOAD
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /CONTR/NPOIN,NELEM,NNODE,NDOFN,NDIME,NSTRE,NTYPE,
&   NGAUS,NPROP,NMATS,NVFIX,NEVAB,ICASE,NCASE,ITEMP,IPOB,
&   NPROB
COMMON /GEOMDA/COORD(1456,2),PROPS(10,5),
&   LNODS(416,8),MATNO(416)
COMMON /LDATA/ PRESC(208,2),ASDIS(2912),ELOAD(416,16),
&   STRIN(3,3744),NOFIX(208),IFPRE(208,2)
COMMON /WORK/ ELOAD(2,8),SHAPE(8),DERIV(2,8),
&   DMATX(3,3),CARTD(2,8),DBMAT(3,16),BMATX(3,16),
&   SMATX(3,16,9),POSGP(3),WEIGP(3),GPCOD(2,9),
&   NEROR(24)
COMMON /PARA/ WT(25),DEFL(25),FUNC(25,3),Y(3),CLEN,SUMXR,RBAR,
&   PKC,PKFI,BB,PN,AU,PKO,C,PHI,PKJ,SLIP,R,YLOAD,SUMY,SUMXF,DEFM,
&   ASTOR(160),PRESN(160),BSTOR(160),NSTOR(160),SHEAR(160),DISJ(160)
COMMON /NDATA/ NX0,NMAXE,NESD,NEGP,NOTE,NX1,NMINO,NSMIN,NEGP1,
&   NMIN,NEGE1,NEDGE,ITER,NITER,NLODP,KEEP,NCTRL,NADD,NCON
DIMENSION COORDS(1456,2)
DO 511 IX=1,NX1
PRESN(IX)=0.D0
SHEAR(IX)=0.D0
511 DISJ(IX)=0.D0
YMIN=2.D0*R
NMAX1=NMAXE+1
NMAX2=NMAXE
DO 555 IMIN=1,NMAX1
KSD=(IMIN-1)*(3*NESD+2)+2*NESD+1
NSTOR(2*IMIN-1)=KSD
IF(COORD(KSD,2).GE.YMIN) GO TO 555
YMIN=COORD(KSD,2)
NMIN=2*IMIN-1
555 CONTINUE

```

```

C
DO 666 IMIN=1,NMAX2
KSD=IMIN*(3*NESD+2)
NSTOR(2*IMIN)=KSD
IF(COORD(KSD,2).GE.YMIN) GO TO 666
YMIN=COORD(KSD,2)
NMIN=2*IMIN
666 CONTINUE

```

```

C
PI=3.141592653589793D0
KS=2*NESD+1
DD=0.D0
DO 21 IX=1,(NVFIX-1)
KK=(-1)**IX
KP=NESD+1
IF(KK.GT.0) KP=2*NESD+1
KS=KS+KP
DD=DD+DSQRT((COORD(KS,1)-COORD(KS-KP,1))**2)

```

& + (COORD(KS,2)-COORD(KS-KP,2))**2)
21 CONTINUE

C

DA=DSQRT((COORD(NPOIN,1)-COORD(2*NESD+1,1))**2
& + (COORD(NPOIN,2)-COORD(2*NESD+1,2))**2)
DF=DD+DA
RBAR=DF/(2.D0*PI)

C

IF(ITER.GT.1) GO TO 611
OPEN(12,FILE='INPUT3',ACCESS='SEQUENTIAL',STATUS='OLD')
DO 22 I=1,NPOIN
22 READ(12,*) IDUMMY,COORDS(I,1),COORDS(I,2)
CLOSE(12,STATUS='KEEP')
611 CONTINUE
IF(ITER.EQ.1) NMINO=NMIN
4321 NOTE=NOTE+1
NMAX=NMINO+NOTE
NSMAX=NSTOR(NMAX)
NSMIN=NSTOR(NMIN)
IF(ITER.GT.1) GO TO 622
NMAX0=NMINO+NOTE-1
NSMAX0=NSTOR(NMAX0)
ZU=COORDS(NSMAX0,2)-COORDS(NSMIN,2)
PU=(PKC/BB+PKFI)*ZU**PN
PKU=AU*ZU+PKO
ZO=ZU-PU/PKU
KAP=0
11 KSD=NSTOR(NMAX0-KAP)
DIFF=COORDS(NSMAX0,2)-COORDS(KSD,2)
IF(KSD.EQ.NSMIN) GO TO 12
KAP=KAP+1
ASTOR(KAP)=(PKC/BB+PKFI)*DIFF**PN
GO TO 11
12 CONTINUE
NEGP=KAP
KAP=0
13 KSD=NSTOR(NMIN-KAP)
DIFF=COORDS(KSD,2)-COORDS(NSMIN,2)
IF(DIFF.GT.(ZU-ZO)) GO TO 14
KAP=KAP+1
ASTOR(NEGP+KAP)=PU-PKU*DIFF
GO TO 13
14 CONTINUE
NEGE=KAP
IF(ASTOR(NEGP+NEGE).GT.0.D0) ASTOR(NEGP+NEGE)=0.D0
IF(ASTOR(NEGP+NEGE).GT.0.D0) NEGP=NEGP+NEGE+1
KEEP=NEGE+NEGP
DJ=0.D0
DO 544 IX=1,(KEEP-1)
KEL=NMINO+NEGP-IX+1
KSD1=NSTOR(KEL-1)
KSD2=NSTOR(KEL)
R01=DSQRT(COORDS(KSD1,1)**2+(COORDS(KSD1,2)-R)**2)
R02=DSQRT(COORDS(KSD2,1)**2+(COORDS(KSD2,2)-R)**2)
R00=0.5D0*(R01+R02)
DSQE=(COORDS(KSD1,1)-COORDS(KSD2,1))**2
& + (COORDS(KSD1,2)-COORDS(KSD2,2))**2
DIHETA=DACOS((R01*R01+R02*R02-DSQE)/(2.D0*R01*R02))
IF(COORDS(KSD1,1).LE.0.D0) THEN
CX=-1.D0

```

ELSE
CX=1.D0
ENDIF
THETA=DACOS((R-0.5D0*(COORDS(KSD1,2)+COORDS(KSD2,2)))/R00)
GAMMA=DATAN((COORDS(KSD2,2)-COORDS(KSD1,2))/
& DABS(COORDS(KSD2,1)-COORDS(KSD1,1)))
ALPHA=THETA-GAMMA*CX
DJ=DJ+(R00*DCOS(ALPHA)-RBAR*(1.D0-SLIP)*DCOS(GAMMA))*DTHETA
544 DISJ(IX+1)=DJ
DO 542 ISHER=1,KEEP
BSTOR(ISHER)=(C+ASTOR(ISHER)*DTAN(PHI))*
& (1.D0-DEXP(-DISJ(ISHER)/PKJ))
542 CONTINUE
C
622 CONTINUE
C
ZU=COORD(NSMAX,2)-COORD(NSMIN,2)
PU=(PKC/BB+PKFI)*ZU**PN
PKU=AU*ZU+PKO
ZO=ZU-PU/PKU
KAP=0
1 KSD=NSTOR(NMAX-KAP)
DIFF=COORD(NSMAX,2)-COORD(KSD,2)
IF(KSD.EQ.NSMIN) GO TO 2
KAP=KAP+1
PRESN(KAP)=(PKC/BB+PKFI)*DIFF**PN
GO TO 1
2 CONTINUE
NEGP=KAP
KAP=0
3 KSD=NSTOR(NMIN-KAP)
DIFF=COORD(KSD,2)-COORD(NSMIN,2)
IF(DIFF.GT.(ZU-ZO)) GO TO 4
KAP=KAP+1
PRESN(NEGP+KAP)=PU-DIFF*PKU
GO TO 3
4 CONTINUE
NEGE=KAP
NOP=NEGE+NEGP
C
LPF=(-1)**(NMIN-NEGE+1)
LPI=(-1)**(NMIN+NEGP)
IF(LPI.GT.0) THEN
DO 999 IX=1,NOP
KEL=NOP-IX+1
999 PRESN(KEL+1)=PRESN(KEL)
PRESN(1)=0.D0
NEGP1=NEGP+1
NADD=2
IF(LPF.GT.0) THEN
NEGE1=NEGE+1
ELSE
NEGE1=NEGE
ENDIF
ELSE
NEGP1=NEGP
NADD=0
IF(LPF.GT.0) THEN
NEGE1=NEGE+1
ELSE

```

```

NEGE1=NEGE
ENDIF
ENDIF
NLODP=NEGE1+NEGP1

```

C

159

```

DO 411 IX=1,NX1
411 DISJ(IX)=0.D0
DJ=0.D0
NLODPL=NLODP-1-NADD/2
DO 534 IX=1,NLODPL
KEL=NMIN+NEGP-IX+1
KSD1=NSTOR(KEL-1)
KSD2=NSTOR(KEL)
R01=DSQRT(COORD(KSD1,1)**2+(COORD(KSD1,2)-R)**2)
R02=DSQRT(COORD(KSD2,1)**2+(COORD(KSD2,2)-R)**2)
R00=0.5D0*(R01+R02)
DSQE=(COORD(KSD1,1)-COORD(KSD2,1))**2
& +(COORD(KSD1,2)-COORD(KSD2,2))**2
DTHETA=DACOS((R01*R01+R02*R02-DSQE)/(2.D0*R01*R02))
IF(COORD(KSD1,1).LE.0.D0) THEN
CX=-1.D0
ELSE
CX=1.D0
ENDIF
THETA=DACOS((R-0.5D0*(COORD(KSD1,2)+COORD(KSD2,2)))/R00)
GAMMA=DATAN((COORD(KSD2,2)-COORD(KSD1,2))/
& DABS(COORD(KSD2,1)-COORD(KSD1,1)))
ALPHA=THETA-GAMMA*CX
DJ=DJ+(R00*DCOS(ALPHA)-RBAR*(1.D0-SLIP)*DCOS(GAMMA))*DTHETA
534 DISJ(IX+1+NADD/2)=DJ
DO 532 ISHER=1,NLODP
SHEAR(ISHER)=(C+PRESN(ISHER)*DTAN(PHI))*
& (1.D0-DEXP(-DISJ(ISHER)/PKJ))
532 CONTINUE
IF((NEGE1-NEGE).EQ.1) SHEAR(1)=0.D0
SUMY=0.D0
SUMXF=0.D0
SUMXR=0.D0
DO 888 IX=1,(NLODP-1)
KSD1=NSTOR(NMIN+NEGP1-IX)
KSD2=NSTOR(NMIN+NEGP1-IX+1)
SUMY=SUMY+0.5D0*PROPS(1,3)*(PRESN(IX)+PRESN(IX+1))*
& (COORD(KSD2,1)-COORD(KSD1,1))
& +0.5D0*PROPS(1,3)*(SHEAR(IX)+SHEAR(IX+1))*
& (COORD(KSD2,2)-COORD(KSD1,2))
SUMXR=SUMXR+0.5D0*PROPS(1,3)*(PRESN(IX)+PRESN(IX+1))*
& (COORD(KSD2,2)-COORD(KSD1,2))
SUMXF=SUMXF+0.5D0*PROPS(1,3)*(SHEAR(IX)+SHEAR(IX+1))*
& (COORD(KSD2,1)-COORD(KSD1,1))
888 CONTINUE
CLEN=0.D0
DO 222 IX=1,NLODPL
KSD1=NSTOR(NMIN-NEGE+IX)
KSD2=NSTOR(NMIN-NEGE+IX+1)
CLEN=CLEN+DSQRT((COORD(KSD2,1)-COORD(KSD1,1))**2
& +(COORD(KSD2,2)-COORD(KSD1,2))**2)
222 CONTINUE
CALL INTERP (NX0,2,SUMY,WT,FUNC,Y)
PROPS(1,1)=Y(1)
DEFM=COORD(NSMIN,2)
CALL INTERP (NX0,2,DEFM,DEFL,FUNC,Y)

```


PROPS(1,3)=Y(2)
BB=DMIN1(PROPS(1,3),CLEN)

C

WRITE(*,*) 'NMIN=',NMIN
WRITE(*,*) 'NSMIN=',NSMIN
WRITE(*,*) 'YMIN=',YMIN
WRITE(*,*) 'SUMY=',SUMY
WRITE(*,*) 'SUMXF=',SUMXF
WRITE(*,*) 'SUMXR=',SUMXR
WRITE(*,*) 'NADD=',NADD
WRITE(*,*) 'DIFF=',DIFF
IF(NCON.EQ.1) GO TO 414
IF(SUMY.LT.YLOAD) THEN
NCON=0
GO TO 4321
ELSE
NOTE=NOTE-3
NCON=1
GO TO 4321
ENDIF
414 CONTINUE
IF(NADD.EQ.0) GO TO 423
DO 791 IX=1,KEEP
KEL=KEEP-IX+1
BSTOR(KEL+NADD)=BSTOR(KEL)
791 ASTOR(KEL+NADD)=ASTOR(KEL)
DO 781 IX=1,NADD
BSTOR(IX)=0.D0
781 ASTOR(IX)=0.D0
423 CONTINUE
NEDGE=(NLODP+NADD)/2
KEEP=NLODP
WRITE(4,935) NEDGE
935 FORMAT (1H0,5X,'NO. OF LOADED EDGES = ',I5)
WRITE(4,940)
940 FORMAT (1H0,5X,'LIST OF LOADED EDGES AND APPLIED LOADS')
NCTRL=0
IF(ITER.EQ.NITER.OR.SUMY.GE.YLOAD) NCTRL=1
RETURN
END

```

C *****
C * PROGRAM VL *
C * This program is to predict the overall performance of a vehicle *
C * SUBROUTINES included: *
C * INPUT; *
C * GOLDN: FUNC: INTERP; *
C * INTERP; *
C *****

```

161

```

C *****
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /DATA1/ COORD(208,2,12)
      COMMON /DATA2/ DEFLN(6,2),SINKG(6,2),ELARY(6,2),
      & COLTH(6,2),WIDTH(6,2),YLOAD(6,2),THRUS(6,2),RESIS(6,2),
      & YT(6,2),YR(6,2),XN(6,2),CLENR(6,2),PRESN(100),SHEAR(100),
      & ASTOR(100,2,6),BSTOR(100,2,6),X1(6),X2(6)
      COMMON /DATA3/ XMFRO(6,2),XFRON(6,2),XDMIN(6,2),XREAR(6,2)
      COMMON /NDATA/ NSTOR(208),NFIX,NPOIN,NX1,NMAX1,NMAX2
      COMMON /CALCU/ XL1,XL2,XD,YD,W,A,B,ALPHA,BETA
      COMMON /GOLDEN/ FUN(4),VAR(4),XX(6)
      DIMENSION COINT(6,2)
      CHARACTER*15 FILEN
      READ(1,*) NFIX,NPOIN,NESD
      READ(1,*) W,XD,YD,XL1,XL2
      READ(1,*) A,B,ALPHA,BETA
      DO 21 I=1,6
      READ(1,*) XX(I)
21 CONTINUE
      NMAX1=0
      NMAX2=0
      DO 22 I=1,6
      DO 22 J=1,2
      DEFLN(I,J)=0.D0
      ELARY(I,J)=0.D0
      SINKG(I,J)=0.D0
      COLTH(I,J)=0.D0
      WIDTH(I,J)=0.D0
      YLOAD(I,J)=0.D0
      THRUS(I,J)=0.D0
      RESIS(I,J)=0.D0
      CLENR(I,J)=0.D0
      YT(I,J)=0.D0
      YR(I,J)=0.D0
      XN(I,J)=0.D0
      XMFRO(I,J)=0.D0
      XFRON(I,J)=0.D0
      XREAR(I,J)=0.D0
22 XDMIN(I,J)=0.D0
      DO 23 I=1,100
      DO 23 J=1,2
      DO 23 K=1,6
      ASTOR(I,J,K)=0.D0
23 BSTOR(I,J,K)=0.D0
      NLONG=2*NESD+1
      NSHOT=NESD+1
      NTOIL=3*NESD+2
      KK=0
      DO 1 I=1,NFIX
      IK=(-1)**I
      KP=NLONG
      IF(IK.GT.0) KP=NSHOT

```



```

114 FORMAT(1X,'FRONT',6(F8.5,2X),/)
WRITE(6,115) XFRON2,XDMIN2,XRVAR2,XN2,YT2,YR2
115 FORMAT(1X,'REAR ',6(F8.5,2X),/)
WRITE(6,116) FUNMI,VARM1
116 FORMAT(1X,'FUNMI=',F8.5,4X,'VARM1=',F12.3/)
ADSIN=SINKG2+DEFLN2-(SINKG1+DEFLN1)
WRITE(6,117) ADSIN,XMFRO2
117 FORMAT(1X,'ADSIN=',F8.5,4X,'XMFRO=',F8.5,/)
OPEN(20,FILE='FPCL.DAT',STATUS='NEW')
OPEN(21,FILE='FPRE.DAT',STATUS='NEW')
OPEN(22,FILE='FSHR.DAT',STATUS='NEW')
DO 11 I=1,NMAX1
J=NMAX1-I+1
DO 12 K=1,6
COINT(K,1)=ASTOR(J,1,K)
12 COINT(K,2)=BSTOR(J,1,K)
CALL INTERP(6,1,WNORM1,X1,COINT,P)
CALL INTERP(6,2,WNORM1,X1,COINT,S)
P=P/1000.DO
S=S/1000.DO
WRITE(21,108) P
WRITE(22,108) S
11 CONTINUE
108 FORMAT(1X,F13.5,)
CLOSE(21,STATUS='KEEP')
CLOSE(22,STATUS='KEEP')
DO 901 I=1,NMAX1
C1=DBLE(I-1)*COLTH1/DBLE(NMAX1-1)+CLENR1-COLTH1
WRITE(20,108) C1
901 CONTINUE
CLOSE(20,STATUS='KEEP')
OPEN(20,FILE='RPCL.DAT',STATUS='NEW')
OPEN(21,FILE='RPRE.DAT',STATUS='NEW')
OPEN(22,FILE='RSHR.DAT',STATUS='NEW')
DO 13 I=1,NMAX2
J=NMAX2-I+1
DO 14 K=1,6
COINT(K,1)=ASTOR(J,2,K)
14 COINT(K,2)=BSTOR(J,2,K)
CALL INTERP(6,1,WNORM2,X2,COINT,P)
CALL INTERP(6,2,WNORM2,X2,COINT,S)
P=P/1000.DO
S=S/1000.DO
WRITE(21,108) P
WRITE(22,108) S
13 CONTINUE
CLOSE(21,STATUS='KEEP')
CLOSE(22,STATUS='KEEP')
XLD=DSQRT((XL1+XL2)*(XL1+XL2)-ADSIN*ADSIN)
DO 902 I=1,NMAX2
C2=DBLE(I-1)*COLTH2/DBLE(NMAX2-1)+XLD+CLENR2-COLTH2
WRITE(20,108) C2
902 CONTINUE
CLOSE(20,STATUS='KEEP')
OPEN(21,FILE='FCDX.DAT',STATUS='NEW')
OPEN(22,FILE='FCDY.DAT',STATUS='NEW')
DO 31 I=1,NFIX
DO 32 J=1,2
DO 32 K=1,6
32 COINT(K,J)=COORD(I,J,K)

```

```
CALL INTERP(6,1,WNORM1,X1,COINT,X)
CALL INTERP(6,2,WNORM1,X1,COINT,Y)
IF(I.EQ.1) XS=X
IF(I.EQ.1) YS=Y
WRITE(21,138) X
WRITE(22,138) Y
138 FORMAT(1X,F11.6)
31 CONTINUE
WRITE(21,138) XS
WRITE(22,138) YS
CLOSE(21,STATUS='KEEP')
CLOSE(22,STATUS='KEEP')
OPEN(21,FILE='RCDX.DAT',STATUS='NEW')
OPEN(22,FILE='RCDY.DAT',STATUS='NEW')
DO 33 I=1,NFIX
DO 34 J=1,2
DO 34 K=7,12
34 COINT(K-6,J)=COORD(I,J,K)
CALL INTERP(6,1,WNORM2,X2,COINT,X)
CALL INTERP(6,2,WNORM2,X2,COINT,Y)
X0=X+XL1+XL2
Y0=Y-ADSIN
IF(I.EQ.1) XS=X0
IF(I.EQ.1) YS=Y0
WRITE(21,138) X0
WRITE(22,138) Y0
33 CONTINUE
WRITE(21,138) XS
WRITE(22,138) YS
CLOSE(21,STATUS='KEEP')
CLOSE(22,STATUS='KEEP')
STOP
END
```

```

C *****
C * SUBROUTINE FUNC
C * This subroutine calculates the overall moment equilibrium
C * equation.
C * SUBROUTINES included:
C * INTERP.
C *****
SUBROUTINE FUNC(WNORM1,OBJ)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /DATA2/ DEFLN(6,2),SINKG(6,2),ELARY(6,2),
% COLTH(6,2),WIDTH(6,2),YLOAD(6,2),THRUS(6,2),RESIS(6,2),
% YT(6,2),YR(6,2),XN(6,2),CLENR(6,2),PRESN(100),SHEAR(100),
% ASTOR(100,2,6),BSTOR(100,2,6),X1(6),X2(6)
COMMON /NDATA/ NSTOR(208),NFI, NPOIN,NX1,NMAX1,NMAX2
COMMON /CALCU/ XL1,XL2,XD,YD,W,A,B,ALPHA,BETA
COMMON /GOLDEN/ FUN(4),VAR(4),XX(6)
WNORM2=W-WNORM1
CALL INTERP(6,1,WNORM1,X1,THRUS,THRUS1)
CALL INTERP(6,2,WNORM2,X2,THRUS,THRUS2)
CALL INTERP(6,1,WNORM1,X1,RESIS,RESIS1)
CALL INTERP(6,2,WNORM2,X2,RESIS,RESIS2)
CALL INTERP(6,1,WNORM1,X1,DEFLN,DEFLN1)
CALL INTERP(6,2,WNORM2,X2,DEFLN,DEFLN2)
CALL INTERP(6,1,WNORM1,X1,SINKG,SINKG1)
CALL INTERP(6,2,WNORM2,X2,SINKG,SINKG2)
CALL INTERP(6,1,WNORM1,X1,YT,YT1)
CALL INTERP(6,2,WNORM2,X2,YT,YT2)
CALL INTERP(6,1,WNORM1,X1,YR,YR1)
CALL INTERP(6,2,WNORM2,X2,YR,YR2)
CALL INTERP(6,1,WNORM1,X1,XN,XN1)
CALL INTERP(6,2,WNORM2,X2,XN,XN2)
DELTA=SINKG2+DEFLN2-(SINKG1+DEFLN1)
YMU1=DCOS(DASIN(DELTA/(XL1+XL2)))
OBJ=THRUS1*YT1+THRUS2*YT2-RESIS1*YR1-RESIS2*YR2-WNORM1*XN1+
% WNORM2*((XL1+XL2)*YMU1-XN2)-YD*(THRUS1+THRUS2-RESIS1-RESIS2)
% -W*XL1*YMU1+(THRUS1-RESIS1)*DELTA
OBJ=DABS(OBJ)
RETURN
END

```

```

C *****
C * SUBROUTINE INPUT
C * This subroutine retrieves the stored data of front and rear
C * performances, and calculates the acting positions of the force
C * components.
C *****

```

166

```

SUBROUTINE INPUT(FILEN,KI,KJ,KK)
IMPLICIT REAL*8 (A-H,O-Z)
COMMON /DATA1/ COORD(208,2,12)
COMMON /DATA2/ DEFLN(6,2),SINKG(6,2),ELARY(6,2),
% COLTH(6,2),WIDTH(6,2),YLOAD(6,2),THRUS(6,2),RESIS(6,2),
% YT(6,2),YR(6,2),XN(6,2),CLENR(6,2),PRESN(100),SHEAR(100),
% ASTOR(100,2,6),BSTOR(100,2,6),X1(6),X2(6)
COMMON /DATA3/ XMFRO(6,2),XFRON(6,2),XDMIN(6,2),XREAR(6,2)
COMMON /NDATA/ NSTOR(208),NFIX,NPOIN,NX1,NMAX1,NMAX2
CHARACTER*15 FILEN
OPEN(12,FILE=FILEN,ACCESS='SEQUENTIAL',STATUS='OLD')
DO 3 I=1,100
PRESN(I)=0.D0
3 SHEAR(I)=0.D0
READ(12,*) NX1
IF(KK.LE.6) THEN
IF(NX1.GT.NMAX1) NMAX1=NX1
ELSE
IF(NX1.GT.NMAX2) NMAX2=NX1
ENDIF
DO 1 I=1,NX1
READ(12,*) PRESN(I),SHEAR(I)
ASTOR(I,KJ,KI)=PRESN(I)
1 BSTOR(I,KJ,KI)=SHEAR(I)
IF(KJ.EQ.1) THEN
READ(12,*) XFRON(KI,KJ),XDMIN(KI,KJ),XREAR(KI,KJ)
READ(12,*) CFRON,CDMIN,CREAR
DEFLN(KI,KJ)=CDMIN
SINKG(KI,KJ)=CFRON-CDMIN
ELARY(KI,KJ)=CREAR-CDMIN
ELSE
READ(12,*) XMFRO(KI,KJ),XFRON(KI,KJ),XDMIN(KI,KJ),XREAR(KI,KJ)
READ(12,*) CMFRO,CFRON,CDMIN,CREAR
DEFLN(KI,KJ)=CDMIN
SINKG(KI,KJ)=CMFRO-CDMIN
ELARY(KI,KJ)=CREAR-CDMIN
ENDIF
READ(12,*) COLTH(KI,KJ),WIDTH(KI,KJ)
READ(12,*) YLOAD(KI,KJ),THRUS(KI,KJ),RESIS(KI,KJ)
NJ=0
KU=1
YMIN=1.D20
DO 2 I=1,NPOIN
READ(12,*) IDUMMY,X,Y
IF(I.NE.NSTOR(KU)) GO TO 2
COORD(KU,1,KK)=X
COORD(KU,2,KK)=Y
IF(Y.GE.YMIN) GO TO 6
YMIN=Y
NMIN=KU
6 CONTINUE
IF(NJ.EQ.1) GO TO 4
IF(DABS(ELARY(KI,KJ)+DEFLN(KI,KJ)-Y).LE.1.D-4) THEN

```

```

IREAR=KU
NJ=1
ELSE
NJ=0
ENDIF
167
4 KU=KU+1
2 CONTINUE
DO 7 I=IREAR,(NMIN-1)
  CLENR(KI,KJ)=CLENR(KI,KJ)+DSQRT((COORD(I+1,1,KK)-COORD(I,1,KK))**2
  & +(COORD(I+1,2,KK)-COORD(I,2,KK))**2)
7 CONTINUE
XCOORD=0.D0
YCOORD=0.D0
YCOOP=0.D0
DO 5 I=1,(NK1-1)
  KSD1=IREAR+I-1
  KSD2=IREAR+I
  XCOORD=XCOORD+((PRESN(I)+PRESN(I+1))*
  & (COORD(KSD2,1,KK)-COORD(KSD1,1,KK))+(SHEAR(I)+
  & SHEAR(I+1))*(COORD(KSD2,2,KK)-COORD(KSD1,2,KK)))*
  & (COORD(KSD2,1,KK)+COORD(KSD1,1,KK))*WIDTH(KI,KJ)*.25D0
  YCOOP=YCOOP+(SHEAR(I)+SHEAR(I+1))*
  & (COORD(KSD2,1,KK)-COORD(KSD1,1,KK))*
  & (COORD(KSD2,2,KK)+COORD(KSD1,2,KK))*WIDTH(KI,KJ)*.25D0
  YCOORD=YCOORD+(PRESN(I)+PRESN(I+1))*
  & (COORD(KSD2,2,KK)-COORD(KSD1,2,KK))*
  & (COORD(KSD2,2,KK)+COORD(KSD1,2,KK))*WIDTH(KI,KJ)*.25D0
5 CONTINUE
XN(KI,KJ)=XCOORD/YLOAD(KI,KJ)
YT(KI,KJ)=YCOOP/THRUS(KI,KJ)
YR(KI,KJ)=YCOORD/RESIS(KI,KJ)
CLOSE(12,STATUS='KEEP')
RETURN
END

```


END

0610789

FIN