TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.
TTI-2-18-74-164-2

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
COMPUTATION OF DYNAMIC LOADS AT GRADE CROSSINGS; A USER'S MANUAL OF THE COMPUTER PROGRAM

5. Report Date
January, 1976

6. Performing Organization Code

7. Author(s)
Aziz Ahmad and Robert L. Lytton

8. Performing Organization Report No.
Research Report No. 164-2

9. Performing Organization Name and Address
Texas Transportation Institute
Texas A&M University
College Station, Texas 77843

10. Work Unit No.

11. Contract or Grant No.
Study No. 2-18-74-164

12. Sponsoring Agency Name and Address
Texas State Department of Highways and Public Transportation; Transportation Planning Division
P. O. Box 5051
Austin, Texas 78763

13. Type of Report and Period Covered
Interim - September, 1973
January, 1976

14. Sponsoring Agency Code

15. Supplementary Notes
Research performed in cooperation with DOT, FHWA.
Study Title: "Structural and Geometric Design of Highway-Railroad Grade Crossings".

16. Abstract

This report gives the theoretical background and a description of a computer program, DYMOL, along with its revisions. This program was originally written to calculate the dynamic forces applied normal to a rigid surface by moving traffic. Revisions are made in the program to include the flexibility (stiffness, damping and inertia effect) of the riding surface, and a special subroutine is added to generate typical grade crossing profiles. Input formats, program listing and a glossary of variables are given for the use of the program. Also included with the report are the descriptions of the program's subroutines and functions and method of calculation of dynamic loads along with Maysmeter readings.

17. Key Words
Dynamic Loads, Highway-Railroad Grade Crossings, Users Manual, Computer Program.

18. Distribution Statement
No Restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
65

22. Price

COMPUTATION OF DYNAMIC LOADS AT GRADE CROSSINGS;

A USER'S MANUAL OF THE COMPUTER PROGRAM

by

Aziz Ahmad
Robert L. Lytton

Research Report Number 164-2

Structural and Geometric Design of
Highway-Railroad Grade Crossings

Research Project 2-18-74-164

conducted for

The State Department of Highways and Public Transportation
in cooperation with the
U.S. Department of Transportation

by the
Texas Transportation Institute
Texas A&M University
College Station, Texas

January, 1976

PREFACE

This report is the second in a series issued under Research Study 2-18-74-164, "Structural and Geometric Design of Highway-Railroad Grade Crossings." This research is sponsored by the State Department of Highways and Public Transportation in cooperation with the Federal Highway Administration to study the problems encountered at highway-railroad grade crossings and to recommend improvements in analysis and design procedure.

ACKNOWLEDGEMENT

The authors sincerely appreciate the assistance and helpful suggestions given by Mr. Dale L. Schafer in computer programming.

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

Abstract

This report gives the theoretical background and a description of a computer program, DYMOL, along with its revisions. This program was originally written to calculate the dynamic forces applied normal to a rigid surface by moving traffic. Revisions are made in the program to include the flexibility (stiffness, damping and inertia effect) of the riding surface, and a special subroutine is added to generate typical grade crossing profiles. Input formats, program listing and a glossary of variables are given for the use of the program. Also included with the report are the descriptions of the program's subroutines and functions and method of calculation of dynamic loads along with Maysmeter readings.

Summary

This report is a user's manual for a revised computer program DYMOL. Input formats, program listing and a glossary of variables are included along with an explanation of the function of each of the program's subroutines.

Highway pavements and structures are always subjected to dynamic wheel loadings due to the traffic moving over them, but their present structural design procedures are mostly based on static loading criteria for stress analysis and for materials evaluation. A reliable technique to measure the locations and magnitudes of these dynamic loadings is desired to achieve an improved and realistic design procedure.

Dynamic loads mainly depend on characteristics of the vehicle, surface roughness and vehicle speed. Computer program DYMOL was written to calculate the dynamic loads applied normal to the surface of highway and other structures by moving traffic. In this version of the program vehicles with two axles are simulated considering them to be damped oscillatory systems with several degrees of freedom. These mathematical simulation models are run over different surface profiles (natural or artificial) with different vehicle speeds. Differential equations of motion are written for each degree of freedom in the model. These equations are solved numerically to obtain dynamic wheel loads over a surface due to moving vehicles. The original program DYMOL was written to calculate dynamic loads applied normal to a rigid surface. The following revisions are made in the program to increase its usefulness to the design of railroad grade crossings:

1. Flexibility of the riding surface (stiffness, damping and inertia effect) is included in the program.
2. A special subroutine is added to generate the profiles of typical grade crossings.

Implementation Statement

The program DYMOL was originally written and later revised in the expectation that design engineers would be able to predict the actual dynamic wheel loading pattern created on a surface of highway pavement or a structure due to the interaction of any surface roughness and vehicle characteristics with different vehicle speeds. This would certainly help arriving at an improved and a realistic design procedure. This program may be used to evaluate an existing pavement or a structure and maintenance operation can be carried out to minimize the surface roughness which causes excessive dynamic loads. Due to the required geometrics, the differential settlements caused by highway and railway traffic, and other practical construction complexities the railroad grade crossings are always a source of higher dynamic loads than on a typical pavement. DYMOL is used in this project to study the influence of different grade crossing profiles upon dynamic tire forces acting normal to the surface. Some typical results are shown in Research Report 164-1. This computer program requires approximately 100,000 bytes of memory .

Table of Contents

List of Figures

List of Tables

List of Reports

Report No. 164-1, "Structural and Geometric Characteristics of Highway-Railroad Grade Crossings," by Thomas M. Newton, Robert L. Lytton, and Robert M. Olson, describes the crossing distribution and geometric characteristics, crossing appraisals, drainage, dynamic loading, stabilization fabrics, and structural details for improved life and rideability.

Report No. 164-2, "Computation of Dynamic Loads at Grade Crossings; A User's Manual of the Computer Program," by Aziz Ahmad and Robert L. Lytton describes the revisions to the computer program DYMOL, including input formats of the program.

INTRODUCTION

This report is written as a user's guide to the revised computer program DYMOL. Program input data and its format are shown here. Remarks and units of input variables are included to help reduce the time for setting up the data.

DYMOL was originally developed by Nasser I. Al-Rashid (1) at the University of Texas at Austin. This was written to calculate the dynamic forces applied normal to the surface of highway pavements and other structures by moving traffic. However, certain revisions were made in the program to increase its flexibility and usefulness. A complete description of the original program DYMOL along with its revisions are documented in this report. Organization of this report is as follows.

a) Description of the original program DYMOL

b) Revisions of the program DYMOL

c) Description of Subroutines and Functions

d) Input description

e) Description of FORTRAN variables that are used in the program

f) Calculation of dynamic loads on each wheel along with Maysmeter reading

g) FORTRAN Listing for Program DYMOL.

DESCRIPTION OF THE COMPUTER PROGRAM

Description of the Original Program DYMOL

This program was originally written for five classes of vehicles. However, vehicles with two axles are simulated in this version of the program by considering them to be damped oscillatory systems with several degrees of freedom. Figure 1 shows the two-axle vehicle model. In this model the vehicle is represented by three distinct masses: (1) the main body, (2) the front axle, and (3) the rear axle. The main body is considered to be rigid. It rests on two axles through four springs, and a shock absorber is connected in parallel with each spring. Again, the two axles rest on at least four tires which are simulated by springs and dashpots. These springs and shock absorbers (or dashpots) may be different for different wheels. The following movements of these three masses are used to calculate the dynamic loads for each wheel:

1. Main body translation in the vertical plane;
2. Main body pitching (rotation about a lateral axis of the body);
3. Main body rolling (rotation about a longitudinal axis of the body);
4. Front axle translation in the vertical plane;
5. Rear axle translation in the vertical plane;
6. Rolling of the front axle;
7. Rolling of the rear axle.

The last four motions of the axle may be accounted for by considering vertical translation of each of the individual wheels. The masses (main body, front axle and rear axle) are excited by surface profiles, which causes vibration in them. Differential equations of motion are set up for each individual mass. These equations are solved by numerical method, resulting in the total dynamic loads for each wheel. General Motors profilometer data from natural surfaces which have been digitized on magnetic tape can be used as input in the program.

These data are averaged for each contact length between the wheel and the ground to calculate the wheel path excitation. Artificial profiles can also be generated internally in place of or in addition to natural surface profile input.

Revisions of the Program DYMOL

The original version of DYMOL considers the surface over which the vehicle rides to be rigid. In reality, a pavement acts more like a viscoelastic material. A definite quantity of pavement and soil mass also vibrates with the vibrating wheel while being resisted by the inertia of the pavement. Consideration of the stiffness, damping and inertia of the pavement was incorporated into the program by rewriting the basic differential equations of motion. These equations were solved by the same numerical method as before. Figure 2 shows the original and revised models of the program.

The revised simulation resulted in four additional degrees of freedom in the model one for each mass of pavement in contact with a tire. Table 1 shows the summary of the degrees of freedom of the revised DYMOL model.

Two special subroutines which generate the profiles of a typical grade-crossing and of sinusoidal curves respectively have been written and added to the DYMOL program. Figure 3 shows a typical grade-crossing profile.

Finally a set of FORTRAN statements were added to the program to accumulate the relative vertical movements of the rear axle with respect to the vehicle body. This movement is recorded by the Mays Road Meter reading in an actual vehicle as an indication of pavement roughness. The Maysmeter simulator is intended for comparison with the data from actual Maysmeter runs on any surface profile. Figure 4 shows a general flowchart of the revised computer program DYMOL.

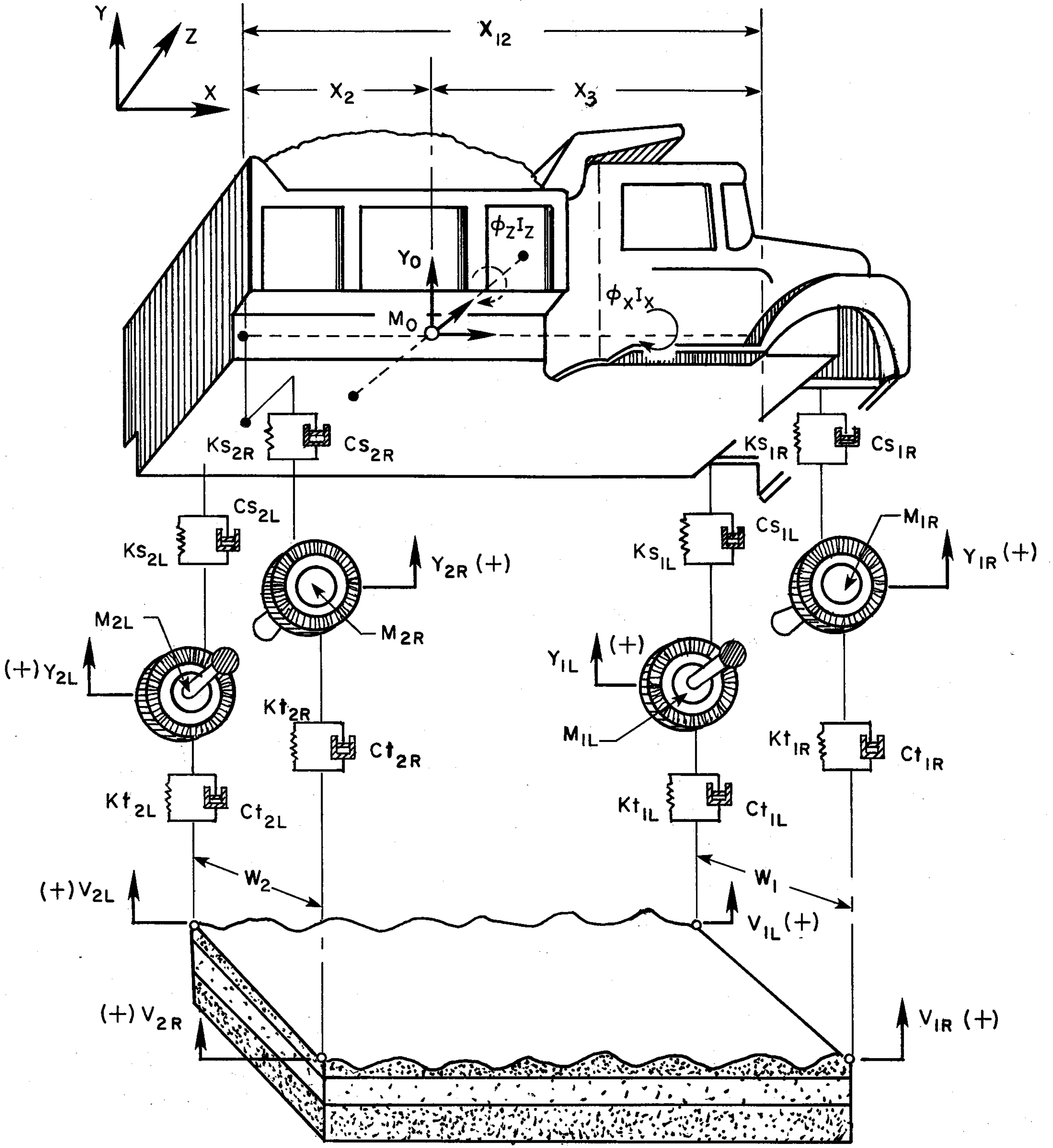
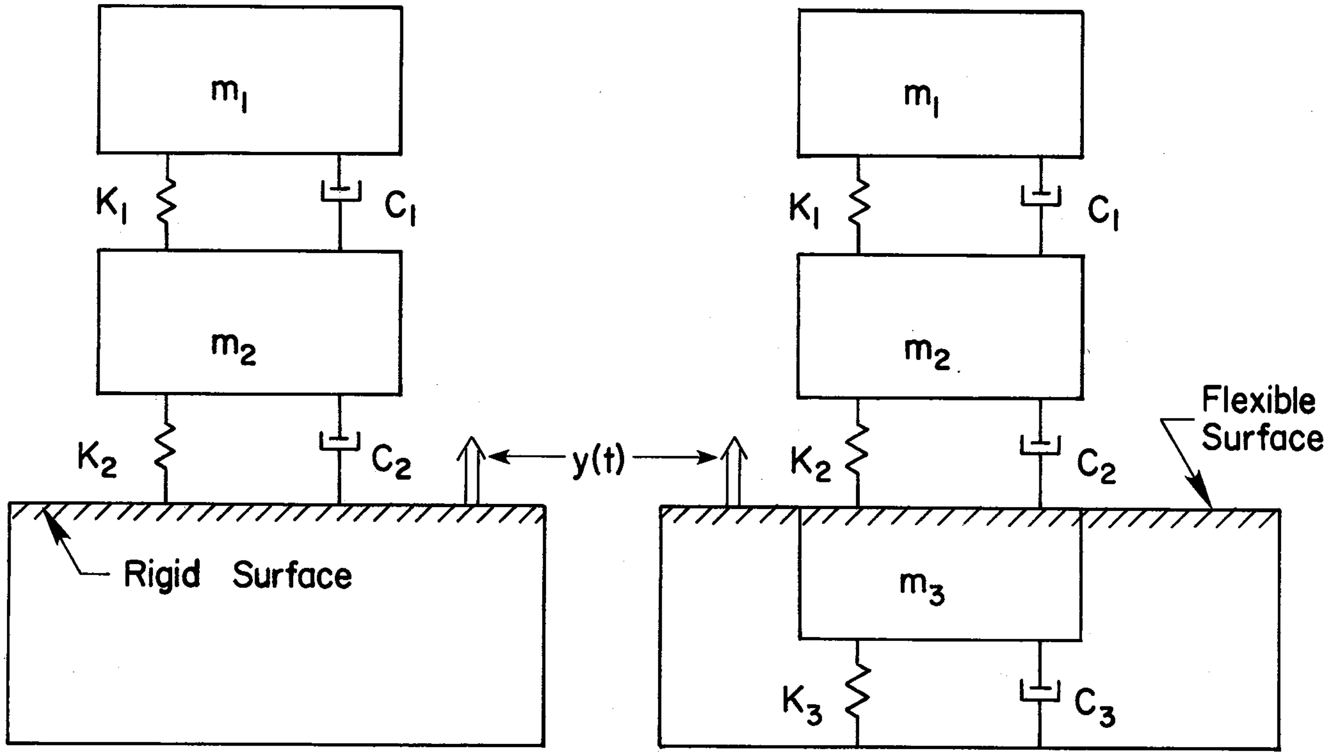


FIGURE 1.– TWO AXLE VEHICLE USED IN COMPUTER PROGRAM DYMOL (After Reference 1)



m_1 = Mass of Vehicle Body
m_2 = Mass of Tire and 1/2 Axle
m_3 = Mass of Soil That Vibrates in Phase With the Wheel
y(t) = Excitation

K_1 = Suspension Stiffness
C_1 = Suspension Damping Constant
K_2 = Tire Stiffness
C_2 = Tire Damping Constant
K_3 = Soil Spring Constant
C_3 = Soil Damping Constant

2(a) Original Simulation Model

2(b) Revised Simulation Model

FIGURE 2 — SIMULATION MODEL

	Degree of Freedom	Mass System	Variables	Differential Eq. of Motion
1.	Vertical Translation of the Main body.	BMO	YO	1*
2.	Rolling of the Main Body	B1X	PHIX	2
3.	Pitching of the Main Body	B1Z	PHIZ	3
4.	Vertical Translation of Front Right Wheel	BM1R	Y1R	4
5.	Vertical Translation of Front Left Wheel	BM1L	Y1L	5
6.	Vertical Translation of Rear Right Wheel	BM2R	Y2R	6
7.	Vertical Translation of Rear Left Wheel	BM2L	Y2L	7
8.	Vertical Translation of Soil Mass with Front Right Wheel	BMPV1R	PX1R	8
9.	Vertical Translation of Soil Mass with Front Left Wheel	BMPV1L	PX1L	9
10.	Vertical Translation of Soil Mass with Rear Right Wheel	BMPV2R	PX2R	10
11.	Vertical Translation of Soil Mass With Rear Left Wheel	BMPV2L	PX2L	11

TABLE 1. Summary of the Degrees of Freedom of the Revised DYMOL Model

* See the differential equations in Appendix B

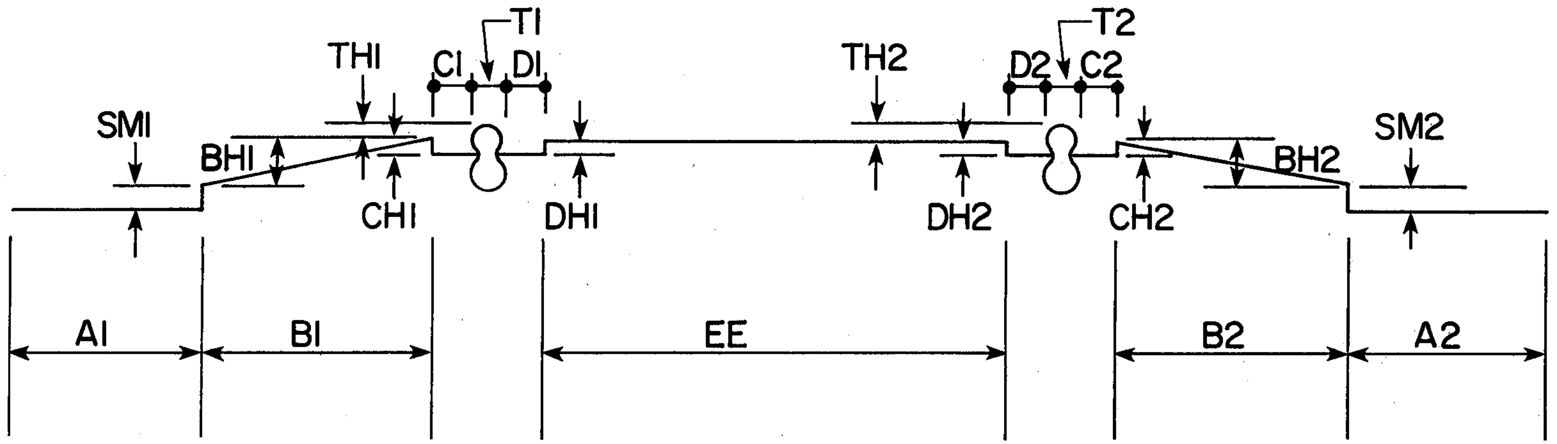


FIGURE 3 – TYPICAL GRADE CROSSING

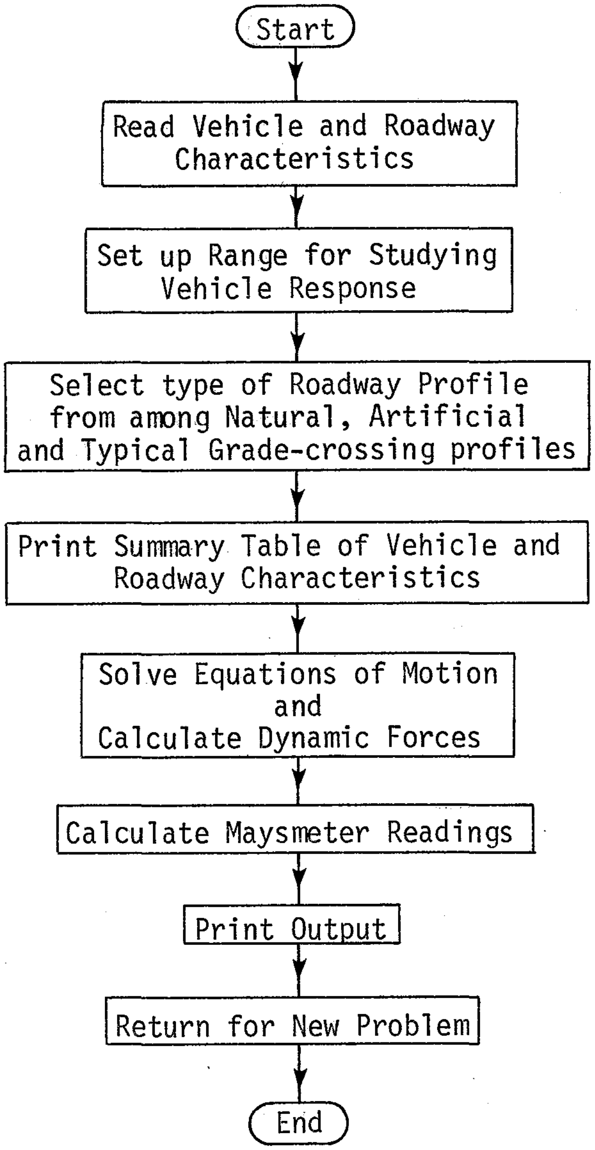

Start
Read Vehicle and Roadway Characteristics
Set up Range for Studying Vehicle Response
Select type of Roadway Profile from among Natural, Artificial and Typical Grade-crossing profiles
Print Summary Table of Vehicle and Roadway Characteristics
Solve Equations of Motion and Calculate Dynamic Forces
Calculate Maysmeter Readings
Print Output
Return for New Problem
End

Fig. 4 General Flowchart of the Computer Program DYMOL

DESCRIPTION OF SUBROUTINES AND FUNCTIONS

This program comprises of the main part, seven subroutines: CLASS1, DRIVER, INDAP, NATPRO, PRFAVG, READIN, and TAPEWR and two Functions: FBUMP, XBUMP. Table 2 gives a cross-reference listing of the main program, subroutines and functions.

CALLED	CALLING PROGRAM NAME									
PROGRAM	MAIN	CLASS1	DRIVER	FUNCTION FBUMP	FUNCTION XBUMP	INDAP	NATPRO	PRAVG	READIN	TAPEWR
CLASS1			X							
DRIVER	X									
FUNCTION FBUMP								X		
FUNCTION XBUMP								X		
INDAP		X								
NATPRO		X								
PRAVG	X									
READIN	X									
TAPEWR	X									

TABLE 2: SUBPROGRAM AND MAIN CROSS-REFERENCE TABLE

A brief description of all the subroutines and functions is given below:

Subroutine TAPEWR

This subroutine reads the natural roadway profiles (profilometer data) from tape no. 10 and writes them on the temporary tape no. 1, which is used later in the program. When artificial bumps are used to calculate dynamic forces, this subroutine writes zeroes for natural profiles (profilometer data) on the temporary tape no. 1.

Subroutine READIN

This subroutine reads all the input-variables. A complete description is shown in the chapter "Input Description".

Subroutine PRAVG

In this subroutine, natural profiles are read from tape no. 1, which is already written in subroutine TAPEWR. Then these values are averaged to calculate the wheel path excitation. The contact length between the wheel and the ground is considered to be eight inches and the profilometer data are assumed to be at a spacing of 2.027 inch. Therefore, four profilometer data points are averaged each time to calculate the wheel path excitation for each contact between the wheel and the ground. When artificial bumps or railroad grade crossing profiles are used, the profilometer data are zeroed (already written on the tape 1) and the subprogram FBUMP or XBUMP generates the heights of the profile. As in the case of natural profile, here also, four data points are averaged to calculate the wheel path excitation for each contact between the wheel and the ground. These averaged wheel path excitation from natural, artificial or grade crossing profiles are written on tape no. 2, for further use in the program.

Subroutine DRIVER

This subroutine calculates V (Velocity, inch per second), KSTEP, IFREQ, H (time step). Then all the variables such as linear and angular displacements, velocities, accelerations of vehicle body, tires and soil masses are initialized. At the end of this subroutine, ITIME (number of time steps to be studied) is fixed; and then subroutine CLASS1 is called to calculate dynamic loads and Maysmeter readings.

Subroutine NATPRO

This subroutine reads the values of wheel path excitation for each time interval from tape no. 2 (on which these data have been written in subroutine PRFAVG). These values are used in subroutine CLASS1 to calculate dynamic loads and Maysmeter readings.

Subroutine INDAP

This subroutine prints out the input data along with vehicle characteristics.

Function FBUMP

This function generates the profile height at each time step, due to sinusoidal curves.

Function XBUMP

This function generates the profile height at each time step, due to railroad grade crossing.

Subroutine CLASS1

This subroutine solves the equations of motion and calculates the total dynamic loads for each wheel, due to the wheel path excitation created by natural profile, grade crossings or artificial bumps. This subroutine also accumulates the vertical movement of vehicle body with respect to the rear axle. These accumulated movements are printed as the Maysmeter reading corresponding to each time interval.

At the beginning of this subroutine, input data are set up corresponding to each variable. Then the center of gravity of the vehicle body is located and body masses and mass moment of inertia are calculated. Subroutine INDAP is called to print out the input data. After the variables I and X3 are defined and XMAYSA and XMAYSM are initialized by zeros, the time do-loop is set up to calculate the dynamic loads on each wheel and to accumulate the Maysmeter reading. The range of this do-loop goes from 1 to ISTOP, covering the whole time limit for running the model, incremented by each time step. Subroutine NATPRO is called within the do-loop on each time step to read its corresponding values of wheelpath excitations from the tape. A detailed description of calculations of dynamic loads on each wheel and the corresponding value of Maysmeter reading is shown in Appendix B.

INPUT DESCRIPTION

Card No.	Column No.	Program Variable	Units	Remarks
1	1-10	BMPWDT	inch	Leave blank when natural profile or grade-crossing profile is used.
	11-20	NREC		Integer number, leave blank when natural profile or grade-crossing profile is used.
	21-30	DSPACE	inch	
	31-40	NLCROS		Integer number, input any positive integer such as 1 if grade-crossing profile is used. Otherwise use 0 or leave blank.
	41-50	NRCRDI		Integer number, this is the record number of the profile from which the data input starts. If all records are used in the input, use NRCRDI = 1.
	51-60	NRCRDF		Integer number, this is the record number of the profile where data input ends. If all the records are used in the input, use NRCRDF = number of the last record.

(If NLCROS = 0, omit this card set)

2nd set (i)	1-10	A1	inch
	11-20	A2	"
	21-30	B1	"
	31-40	B2	"
	41-50	C1	"
	51-60	C2	"
	61-70	D1	"
	71-80	D2	"

Card No.	Column No.	Program Variable	Units	Remarks
(If NLCROS = 0, omit this card set)				
2nd set (ii)	1-10	TI	inch	
	11-20	T2	"	
	21-30	EE	"	
(If NLCROS = 0, omit this card)				
3	1-5	SM1	inch	
	6-10	SM2	"	
	11-15	BH1	"	
	16-20	BH2	"	
	21-25	CH1	"	
	26-30	CH2	"	
	31-35	TH1	"	
	36-40	TH2	"	
	41-45	DH1	"	
	46-50	DH2	"	
4	1-10	NAXL		
	11-20	IEXCIT		Input 3: This is a fixed point number in the program.
5	1-10	DSTR	inch	Leave blank when natural profile or grade-crossing profile is used.
	11-20	DSTL	"	"
	21-30	AMPR	"	"
	31-40	AMPL	"	"

Card No.	Column No.	Program Variable	Units	Remarks
5	41-50	SPACER	inch	Leave blank when natural profile or grade-crossing profile is used.
	51-60	SPACEL	"	"
	61-70	WLR	"	"
	71-80	WLL	"	"
6	1-10	V	inch/sec	
7	1-10	CLOTOL		
	11-20	TLIM	seconds	
	21-30	IFREQ		Leave blank; this value is set up in the program.
	31-40	NBR		Leave blank when natural profile or grade-crossing profile is used.
	41-50	NBL		"
8	1-10	X1(2)	inch	
9	1-10	W(1)	inch	
	11-20	W(2)	"	
10	1-10	AXAWT(1)	lbs.	
	11-20	AXAWT(2)	"	
11	1-10	SKR(1)	lbs/in.	
	11-20	SKR(2)	"	
12	1-10	SKL(1)	lbs/in.	
	11-20	SKL(2)	"	

Card No.	Column No.	Program Variable	Units	Remarks
13	1-10	CSR(1)	lb-sec/in.	
	11-20	CSR(2)	"	
14	1-10	CSL(1)	lb-sec/in.	
	11-20	CSL(2)	"	
15	1-10	TKR(1)	lbs/in.	
	11-20	TKR(2)	"	
16	1-10	TKL(1)	lbs/in.	
	11-20	TKL(2)	"	
17	1-10	CTR(1)	lb-sec/in.	
	11-20	CTR(2)	"	
18	1-10	CTL(1)	lb-sec/in.	
	11-20	CTL(2)	"	
19	1-10	AXWTR(1)	lbs.	
	11-20	AXWTR(2)	"	
20	1-10	AXWTL(1)	lbs.	
	11-20	AXWTL(2)	"	
21	1-10	PKR(1)	lb/in.	
	11-20	PKR(2)	"	
22	1-10	PKL(1)	lb/in.	
	11-20	PKL(2)	"	

Card No.	Column No.	Program Variable	Units	Remarks
23	1-10	CPR(1)	lb-sec/in.	
	11-20	CPR(2)	"	
24	1-10	CPL(1)	lb-sec/in.	
	11-20	CPL(2)	"	
25	1-10	PVWTR(1)	lbs.	
	11-20	PVWTR(2)	"	
26	1-10	PVWTL(1)	lbs.	
	11-20	PVWTL(2)	"	
27	1-64	ITITLE		Alphanumeric; Problem Title
28	1-32	ICLASS		Alphanumeric; Name, type, etc. of the vehicle used.
29	1-10	SPEED	Miles/hour	
	11-20	STRBMP		Leave blank if natural profile or grade-crossing profile is used.
	21-30	BMPHT		"
	31-40	BMPWDT		"
	41-50	BMPSA		"
	51-60	NPTTS		Integer number.

References

1. Al-Rashid, N. I., Lee, C. E., Dawkins, W. P., "A Theoretical and Experimental Study of Dynamic Highway Loading," Center for Highway Research, The University of Texas at Austin, May 1972.

2. Newmark, N. M., "A Method of Computation for Structural Dynamics," Transactions, Vol. 127, Proceedings of the American Society of Civil Engineers, 1962, pp. 1406-1435.

APPENDIX A

FORTRAN VARIABLE DESCRIPTION

FORTRAN VARIABLE	DESCRIPTION
A1, A2	Lengths of horizontal pavements before and after the grade-crossing
AMPL, AMPR	Amplitude of the left and right wheel path excitation respectively.
AXAWT(J)	Weight of axle assembly, differential, brakes, tires for axle J.
AXWTL(J), AXWTR(J)	Vehicle static weights of left and right side of axle J respectively.
B1, B2	Lengths of the horizontal projection of initial and final ramp of a grade-crossing respectively.
BH1, BH2	Vertical rise of initial and final ramps respectively.
BIX, BIZ	Body mass moment of inertias - Roll and pitch respectively.
BM1L, BM1R	Mass of axle assembly, differential, brakes, tires, etc. of front left and front right axle respectively.
BM2L, BM2R	Mass of axle assembly, differential, brakes, tires, etc. of rear left and rear right axle respectively.
BMPV1L, BMPV1R, BMPV2L, BMPV2R	Mass of pavement and soil that vibrates with (front left, front right, rear left and rear right wheel respectively).
BMO	Mass of the vehicle body.
BMPHT	Bump height.
BMPSA	Bump spacing.
BMPWDT	Bump width.
C1, C2	Gap widths between 1st ramp and 1st rail and between 2nd rail and 2nd ramp respectively of a grade-crossing.
CH1, CH2	Depths of the gaps C1 and C2 respectively.
CLOTOL	Closure tolerance

FORTRAN VARIABLE	DESCRIPTION
CPL(J), CPR(J)	Damping rate of pavement or soil under left and right wheel of the axle, J, respectively.
CP1L, CP1R, CP2L, CP2R	Damping rate of pavement or soil under (front left, front right, rear left and rear right wheel respectively).
CSL(J), CSR(J)	Damping rate for left and right suspension of axle J respectively.
CS1L, CS1R	Damping rate for left and right suspension of front axle.
CS2L, CS2R	Damping rate for left and right suspension of rear axle.
CTL(J), CTR(J)	Damping rate of left and right tire of axle J respectively.
CT1L, CT1R, CT2L, CT2R	Damping rate of (front left, front right, rear left and rear right tire respectively).
D1, D2	Width of the gaps between first rail and horizontal mid-portion and between 2nd rail and horizontal mid-portion respectively of grade-crossing.
DH1, DH2	Depth of the gaps D1 and D2 respectively.
D1L, D1R, D2L, D2R	Displacement of vehicle body (front left, front right, rear left and rear right portion respectively).
DD1L, DD1R, DD2L, DD2R	Velocity of vehicle body (front left, front right, rear left and rear right portion respectively).
DYF1L, DYF1R	Calculated dynamic forces at left and right side of front axle respectively.
DYF2L, DYF2R	Calculated dynamic forces at left and right side of rear axle respectively.
DSTL, DSTR	Distance from initial point to start of left and right wheel path excitation respectively.
DSPACE	Spacing between two data points in the natural road profile (or artificially generated profile). It is measured in inches. For GM Profilometer data, DSPACE = 2.027 inches.

FORTRAN VARIABLE	DESCRIPTION
EE	Straight middle portion of pavement between two rails of a grade-crossing.
FBUMP	Wheel path profile for artificial bump.
G	Acceleration due to gravity in inches/sec^2.
H	Time step interval.
HT2	2H (Twice the time step).
HO2	H/2 (Half the time step).
HE2	H^2 (Time step interval squared).
HHO2	$H^2/2$ (Half time step interval squared).
I	An integer used in subroutine CLASS 1, the value is fixed and equal to 1 in the subroutine.
IEXCITE	Type of excitation. This is a fixed point number. In the program it is 3.
ID	Variable name for identification and initial description of the profilometer data.
IFREQ	Frequency of output.
KSTEP	This integer value is used to calculate the time step length (H). KSTEP is larger for higher speed of the vehicle.
NLCROS	This integer value is any positive number when grade-crossing profile is used; otherwise zero.
NREC	Number of records.
NMBMPS	Number of bumps.
NPTTS	Number of spaces in profilometer data for each contact between the wheel and pavement.
NRCRDI	This integer number is the number of a particular record of data profile from which the input starts in the program.
NRCRDF	This integer number is the number of a particular record of data profile at which the input ends in the program.

FORTRAN VARIABLE	DESCRIPTION
NTA	Number of profilometer data points averaged for each contact between the wheel and the pavement.
NAXL	Number of axles.
NBL, NBR	Number of bumps in left and right wheel path respectively.
P1, P2	Averaged wheel path excitation for right and left wheel respectively.
PX1L, PX1R, PX2L, PX2R	Displacements of soil masses that vibrate in phase with front left wheel, front right wheel, rear left wheel and rear right wheel respectively.
PKL(J), PKR(J)	Stiffness of soil or pavement (subgrade reactions) under left and right wheel respectively of axle J.
PK1L, PK1R, PK2L, PK2R	Stiffness of soil or pavement under (front left, front right, rear left and rear right wheel respectively).
PVWTL(J), PVWTR(J)	Weight of soil that vibrates in phase with left and right wheel respectively of axle J.
PHIX, PHIZ	Vehicle body rotation about x-axis (Roll) and z-axis (Pitch) respectively.
DPHIX, DPHIZ	Angular velocity of the vehicle body about x-axis and z-axis respectively.
DDPHIX, DDPHIZ	Angular acceleration of the vehicle body about x-axis and z-axis respectively.
SM1, SM2	Elevation difference between A1 and B1 and between A2 and B2 respectively of a grade-crossing.
SPEED	Speed of the vehicle (miles/hour).
SPACEL, SPACER	Spacings of left and right wheel-path bumps respectively.
SKL(J), SKR(J)	Stiffness of left and right suspension respectively of axle J.

FORTRAN VARIABLE	DESCRIPTION
SK1L, SK1R	Stiffness of left and right suspension respectively of front axle.
SK2L, SK2R	Stiffness of left and right suspension respectively of rear axle.
STRBMP	Straight distance to the bump.
T1, T2	Width of 1st and 2nd rail-top respectively.
TH1, TH2	Elevation difference between T1 and EE and between T2 and EE respectively.
TD1FL, TD1FR	Total forces at left and right side respectively of front axle.
TD2FL, TD2FR	Total forces at left and right side respectively of rear axle.
TVR	Profilometer data for natural profile in inches.
TLIM	Time limit in seconds for running the vehicle.
TKL(J), TKR(J)	Stiffness of left and right tire respectively of axle J.
TK1L, TK1R, TK2L, TK2R	Stiffness of tire (front left, front right, rear left and rear right tire respectively).
TAPE1, TAPE2	Profilometer data in inches.
V	Velocity of the vehicle in inches/sec.
VFPS	Velocity of the vehicle in feet/sec.
VMPH	Velocity of the vehicle in miles/hour.
VL, VR	Wheel path excitation.
WLL, WLR	Wave lengths of the left and right bumps respectively.
W(J)	Width of axle J.
X1(2)	Distance between axle 1 and axle 2.
XLI, XRI	Initial left and right wheel-path excitation.
X2, XX2	Distance between C.G. and the rear axle of the vehicle.

FORTRAN VARIABLE	DESCRIPTION
X3, XX1	Distance between C.G. and front axle of the vehicle.
XBUMP	Wheel path profile for grade-crossing.
XMAYSA, XMAYS, XMAYSM	Maysmeter simulations calculated in the program.
YO	Displacement of C.G. of the vehicle.
DYO	Velocity of C.G. of the vehicle.
DDYO, ADDYO	Acceleration of C.G. of the vehicle.
Y1L, Y1R, Y2L, Y2R	Displacements of axle (front left, front right, rear left and rear right portion respectively).
DY1L, DY1R, DY2L, DY2R	Velocity of axle (front left, front right, rear left, rear right portion respectively).
DDY1L, DDY1R, DDY2L, DDY2R	Acceleration of axle (front left, front right, rear left, rear right respectively).

APPENDIX B

CALCULATION OF DYNAMIC LOADS ON EACH WHEEL ALONG WITH MAYSMETER READINGS

The following are the differential equations of motion due to different degrees of freedom in the revised model of DYMOL as shown in Table 1.

$$\begin{aligned}&\mathrm{SK1R}\ (\mathrm{D1R} - \mathrm{Y1R}) + \mathrm{CS1R}\ (\frac{\partial \mathrm{D1R}}{\partial t} - \frac{\partial \mathrm{Y1R}}{\partial t}) + \mathrm{SK1L}\ (\mathrm{D1L}\text{-}\mathrm{Y1L})\\ &+ \mathrm{CS1L}\ (\frac{\partial \mathrm{D1L}}{\partial t} - \frac{\partial \mathrm{Y1L}}{\partial t}) + \mathrm{SK2R}\ (\mathrm{D2R} - \mathrm{Y2R}) + \mathrm{CS2R}\ (\frac{\partial \mathrm{D2R}}{\partial t} - \frac{\partial \mathrm{Y2R}}{\partial t})\\ &+ \mathrm{SK2L}\ (\mathrm{D2L} - \mathrm{Y2L}) + \mathrm{CS2L}\ (\frac{\partial \mathrm{D2L}}{\partial t} - \frac{\partial \mathrm{Y2L}}{\partial t}) + \mathrm{BMO} \cdot \frac{\partial^2 \mathrm{Y0}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (1)}\end{aligned}$$

$$\begin{aligned}&(\frac{\mathrm{W1}}{2})\ [\mathrm{SK1R}\ (\mathrm{D1R} - \mathrm{Y1R}) + \mathrm{CS1R}\ (\frac{\partial \mathrm{D1R}}{\partial t} - \frac{\partial \mathrm{Y1R}}{\partial t}) - \mathrm{SK1L}\ (\mathrm{D1L} - \mathrm{Y1L})\\ &- \mathrm{CS1L}\ (\frac{\partial \mathrm{D1L}}{\partial t} - \frac{\partial \mathrm{Y1L}}{\partial t})] + (\frac{\mathrm{W2}}{2})\ [\mathrm{SK2R}\ (\mathrm{D2R} - \mathrm{Y2R}) + \mathrm{CS2R}\ (\frac{\partial \mathrm{D2R}}{\partial t} - \frac{\partial \mathrm{Y2R}}{\partial t})\\ &-\mathrm{SK2L}\ (\mathrm{D2L} - \mathrm{Y2L}) - \mathrm{CS2L}\ (\frac{\partial \mathrm{D2L}}{\partial t} - \frac{\partial \mathrm{Y2L}}{\partial t})] - \mathrm{BIX} \cdot \frac{\partial^2 \mathrm{PHIX}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (2)}\end{aligned}$$

$$\begin{aligned}&\mathrm{X3}\ [\mathrm{SK1R}\ (\mathrm{D1R} - \mathrm{Y1R}) + \mathrm{CS1R}\ (\frac{\partial \mathrm{D1R}}{\partial t} - \frac{\partial \mathrm{Y1R}}{\partial t}) + \mathrm{SK1L}\ (\mathrm{D1L} - \mathrm{Y1L})\\ &+ \mathrm{CS1L}\ (\frac{\partial \mathrm{D1L}}{\partial t} - \frac{\partial \mathrm{Y1L}}{\partial t})] - \mathrm{X2}\ [\mathrm{SK2R}\ (\mathrm{D2R} - \mathrm{Y2R}) + \mathrm{CS2R}\ (\frac{\partial \mathrm{D2R}}{\partial t} - \frac{\partial \mathrm{Y2R}}{\partial t})\\ &+ \mathrm{SK2L}\ (\mathrm{D2L} - \mathrm{Y2L}) + \mathrm{CS2L}\ (\frac{\partial \mathrm{D2L}}{\partial t} - \frac{\partial \mathrm{Y2L}}{\partial t})] - \mathrm{BIZ} \cdot \frac{\partial^2 \mathrm{PHIZ}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (3)}\end{aligned}$$

$$\begin{aligned}&\mathrm{SK1R}\ (\mathrm{D1R} - \mathrm{Y1R}) + \mathrm{CS1R}\ (\frac{\partial \mathrm{D1R}}{\partial t} - \frac{\partial \mathrm{Y1R}}{\partial t}) + \mathrm{TK1R}\ (\mathrm{PX1R} - \mathrm{Y1R})\\ &+ \mathrm{CT1R}\ (\frac{\partial \mathrm{PX1R}}{\partial t} - \frac{\partial \mathrm{Y1R}}{\partial t}) - \mathrm{BM1R} \cdot \frac{\partial^2 \mathrm{Y1R}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (4)}\end{aligned}$$

$$\begin{aligned}&\mathrm{SK1L}\ (\mathrm{D1L} - \mathrm{Y1L}) + \mathrm{CS1L}\ (\frac{\partial \mathrm{D1L}}{\partial t} - \frac{\partial \mathrm{Y1L}}{\partial t}) + \mathrm{TK1L}\ (\mathrm{PX1L} - \mathrm{Y1L})\\ &+ \mathrm{CT1L}\ (\frac{\partial \mathrm{PX1L}}{\partial t} - \frac{\partial \mathrm{Y1L}}{\partial t}) - \mathrm{BM1L} \cdot \frac{\partial^2 \mathrm{Y1L}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (5)}\end{aligned}$$

$$\begin{aligned}&\mathrm{SK2R}\ (\mathrm{D2R} - \mathrm{Y2R}) + \mathrm{CS2R}\ (\frac{\partial \mathrm{D2R}}{\partial t} - \frac{\partial \mathrm{Y2R}}{\partial t}) + \mathrm{TK2R}\ (\mathrm{PX2R} - \mathrm{Y2R})\\ &+ \mathrm{CT2R}\ (\frac{\partial \mathrm{PX2R}}{\partial t} - \frac{\partial \mathrm{Y2R}}{\partial t}) - \mathrm{BM2R}\ \frac{\partial^2 \mathrm{Y2R}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (6)}\end{aligned}$$

$$\begin{aligned}&\mathrm{SK2L}\ (\mathrm{D2L} - \mathrm{Y2L}) + \mathrm{CS2L}\ (\frac{\partial \mathrm{D2L}}{\partial t} - \frac{\partial \mathrm{Y2L}}{\partial t}) + \mathrm{TK2L}\ (\mathrm{PX2L} - \mathrm{Y2L})\\ &+ \mathrm{CT2L}\ (\frac{\partial \mathrm{PX2L}}{\partial t} - \frac{\partial \mathrm{Y2L}}{\partial t}) - \mathrm{BM2L}\ \frac{\partial^2 \mathrm{Y2L}}{\partial t^2} = 0 \qquad ...\ \text{Eq. (7)}\end{aligned}$$

$$\mathrm{PK1R}\ (\mathrm{V1R} - \mathrm{PX1R}) + \mathrm{CP1R}\ \left(\frac{\partial \mathrm{V1R}}{\partial t} - \frac{\partial \mathrm{PX1R}}{\partial t}\right) - \mathrm{TK1R}\ (\mathrm{PX1R} - \mathrm{Y1R})$$

$$- \mathrm{CT1R}\ \left(\frac{\partial \mathrm{PX1R}}{\partial t} - \frac{\partial \mathrm{Y1R}}{\partial t}\right) - \mathrm{BMPV1R}\ \frac{\partial^2 \mathrm{PX1R}}{\partial t^2} = 0 \qquad \text{. . . Eq. (8)}$$

$$\mathrm{PK1L}\ (\mathrm{V1L} - \mathrm{PX1L}) + \mathrm{CP1L}\ \left(\frac{\partial \mathrm{V1L}}{\partial t} - \frac{\partial \mathrm{PX1L}}{\partial t}\right) - \mathrm{TK1L}\ (\mathrm{PX1L} - \mathrm{Y1L})$$

$$- \mathrm{CT1L}\ \left(\frac{\partial \mathrm{PX1L}}{\partial t} - \frac{\partial \mathrm{Y1L}}{\partial t}\right) - \mathrm{BMPV1L}\ \frac{\partial^2 \mathrm{PX1L}}{\partial t^2} = 0 \qquad \text{. . . Eq. (9)}$$

$$\mathrm{PK2R}\ (\mathrm{V2R} - \mathrm{PX2R}) + \mathrm{CP2R}\ \left(\frac{\partial \mathrm{V2R}}{\partial t} - \frac{\partial \mathrm{PX2R}}{\partial t}\right) - \mathrm{TK2R}\ (\mathrm{PX2R} - \mathrm{Y2R})$$

$$- \mathrm{CT2R}\ \left(\frac{\partial \mathrm{PX2R}}{\partial t} - \frac{\partial \mathrm{Y2R}}{\partial t}\right) - \mathrm{BMPV2R}\ \frac{\partial^2 \mathrm{PX2R}}{\partial t^2} = 0 \qquad \text{. . . Eq. (10)}$$

$$\mathrm{PK2L}\ (\mathrm{V2L} - \mathrm{PX2L}) + \mathrm{CP2L}\ \left(\frac{\partial \mathrm{V2L}}{\partial t} - \frac{\partial \mathrm{PX2L}}{\partial t}\right) - \mathrm{TK2L}\ (\mathrm{PX2L} - \mathrm{Y2L})$$

$$- \mathrm{CT2L}\ \left(\frac{\partial \mathrm{PX2L}}{\partial t} - \frac{\partial \mathrm{Y2L}}{\partial t}\right) - \mathrm{BMPV2L}\ \frac{\partial^2 \mathrm{PX2L}}{\partial t^2} = 0 \qquad \text{. . . Eq. (11)}$$

Solution of Differential Equations of Motion

These differential equations of motion are solved by a numerical technique described by N.M. Newmark (2) in his paper entitled, "A Method of Computation for Structural Dynamics". The procedure is based on the assumption that the displacements, the velocities and the accelerations of the system are known at any particular time, t_i. The values of these variables at t_{i+1} are determined from the following relationships.

$$(Y)_{i+1} = (Y)_i + h(\frac{\partial Y}{\partial t})_i + h^2(\frac{1}{2} - \beta)\ (\frac{\partial^2 Y}{\partial t})_i + h^2\beta\ (\frac{\partial^2 Y}{\partial t^2})_{i+1} \qquad . . . \text{ Eq. (A)}$$

$$(\frac{\partial Y}{\partial t})_{i+1} = (\frac{\partial Y}{\partial t})_i + h\ (1 - \nu)\ (\frac{\partial^2 Y}{\partial t^2})_i + h\nu\ (\frac{\partial^2 Y}{\partial t^2})_{i+1} \qquad . . . \text{ Eq. (B)}$$

Where: Y = displacement of a mass

$\frac{\partial Y}{\partial t}$ = velocity of a mass

$\frac{\partial^2 Y}{\partial t^2}$ = acceleration of a mass

$h = t_{i+1} - t_i$

The values of ν and β are used as 1/2 and 1/6 respectively which make the solution converge quickly. Figure 5 shows the flowchart of the procedure used in the numerical technique.

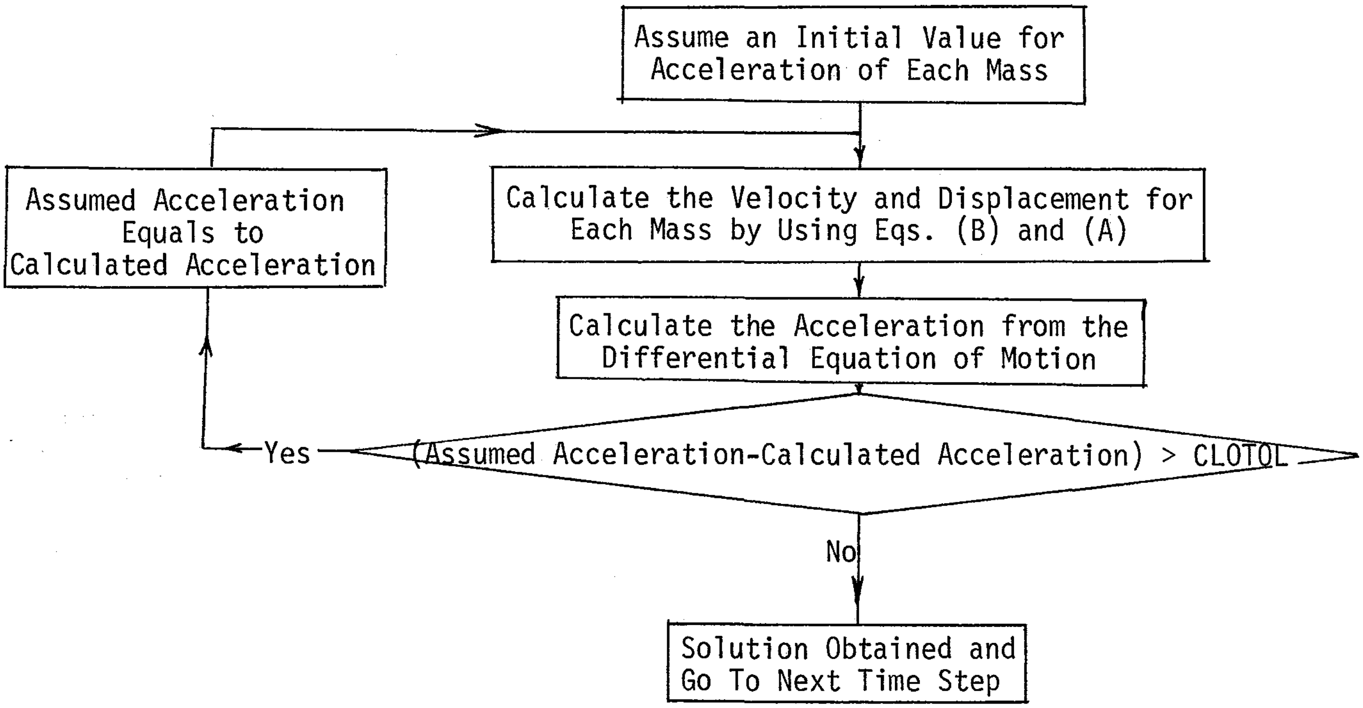

Assume an Initial Value for Acceleration of Each Mass
Calculate the Velocity and Displacement for Each Mass by Using Eqs. (B) and (A)
Calculate the Acceleration from the Differential Equation of Motion
(Assumed Acceleration-Calculated Acceleration) > CLOTOL
Yes
Assumed Acceleration Equals to Calculated Acceleration
No
Solution Obtained and Go To Next Time Step

Figure 5. Flowchart Showing the Steps of Numerical Technique used in DYMOL

Calculation of Total Dynamic Load on a Wheel

The vertical translations of the vehicle body, tires and soil masses are obtained by solving the differential equations of motion for a particular time step. Dynamic forces are calculated in each wheel for the same time step from the following expressions:

$$DYF1R = TK1R\ (PX1R - Y1R) + CT1R\ \left(\frac{\partial PX1R}{\partial t} - \frac{\partial Y1R}{\partial t}\right)$$

$$DYF1L = TK1L\ (PX1L - Y1L) + CT1L\ \left(\frac{\partial PX1L}{\partial t} - \frac{\partial Y1L}{\partial t}\right)$$

$$DYF2R = TK2R\ (PX2R - Y2R) + CT2R\ \left(\frac{\partial PX2R}{\partial t} - \frac{\partial Y2R}{\partial t}\right)$$

$$DYF2L = TK2L\ (PX2L - Y2L) + CT2L\ \left(\frac{\partial PX2L}{\partial t} - \frac{\partial Y2L}{\partial t}\right)$$

After calculating the dynamic force component on a wheel, the total dynamic wheel load may be determined by simply adding the static and the dynamic force components as follows:

$$TDF1R = DYF1R + AXWTR(1)$$

$$TDF1L = DYF1L + AXWTL(1)$$

$$TDF2R = DYF2R + AXWTR(2)$$

$$TDF2L = DYF2L + AXWTL(2)$$

Excessive roughness may cause loss of contact between the wheel and the pavement, making the total dynamic force a negative quantity. Provision is kept in the program to make the total dynamic force as zero when it becomes negative. This is done to avoid any tension force acting on the pavement since it is not possible in reality.

Maysmeter Simulation

Figure 6 shows the flowchart for Maysmeter simulation. XMAYS is the calculated difference between the vertical translation of the rear axle and rear end of the vehicle for a particular time step, t_i. XMAYSM is the value of XMAYS in the previous time step, t_{i-1}. The absolute value of the difference between XMAYS and XMAYSM is accumulated for every time step. This accumulated value up to any time step (XMAYSA) is the Maysmeter reading at that time.

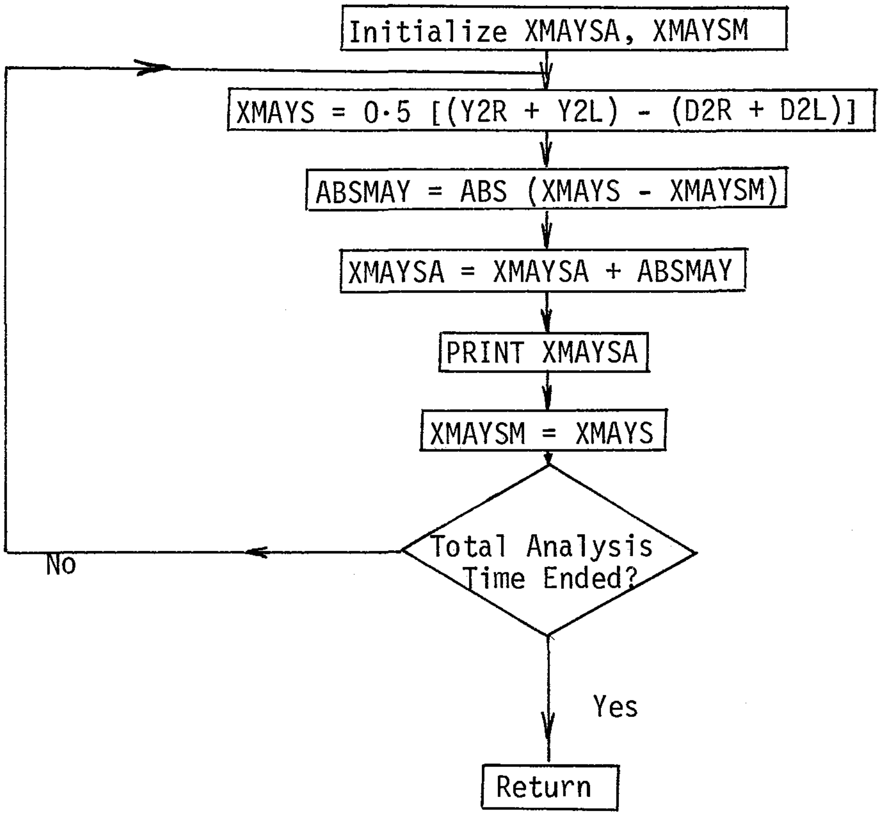

Initialize XMAYSA, XMAYSM
XMAYS = 0·5 [(Y2R + Y2L) - (D2R + D2L)]
ABSMAY = ABS (XMAYS - XMAYSM)
XMAYSA = XMAYSA + ABSMAY
PRINT XMAYSA
XMAYSM = XMAYS
Total Analysis Time Ended?
No
Yes
Return

Figure 6. Flowchart for Maysmeter Simulation

APPENDIX C

JCL AND FORTRAN LISTING FOR DYMOL

Supporting IBM Job Control Language (JCL) for the Program DYMOL is as follows:

```
//JØB

/*PASSWØRD

//DYMØL EXEC FØRTGCG,REGIØN.GØ=100K

//FØRT.SYSIN DD *

SØURCE DECK

//GØ.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DISP=(NEW,PASS)

//GØ.FT01F002 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DISP=(NEW,PASS)

//GØ.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DISP=(NEW,PASS)

//GØ.FT02F002 DD UNIT=SYSDA,SPACE=(CYL,(2,2)),DISP=(NEW,PASS)

//GØ.FT10F001 DD UNIT=(TAPE9,,DEFER),VØL=SER=XXXXXX,DISP=(ØLD,KEEP),

// DSNAME=XXXXXX

//GØ.SYSIN DD *

INPUT DATA

/*END
```

```
      COMMON/AZ1/ PX1R,PX1L,PX2R,PX2L,DPX1R,DPX1L,DPX2R,DPX2L,
     1  DDPX1R,DDPX1L,DDPX2R,DDPX2L,ADPX1R,ADPX1L,ADPX2R,ADPX2L,IEND
      COMMON /AZ2/A1,A2,B1,B2,C1,C2,D1,D2,T1,T2,EE
     1 ,NLCROS,SM1,SM2,BH1,BH2,CH1,CH2,TH1,TH2,DH1,DH2
      COMMON / AZ7 / XDIST,DSPACE
       COMMON /AZ9/ NRCRDI,NRCRDF
      COMMON / NAME / NNAME(20),XNAME(20),NEND
      DIMENSION XNAME1(20)
      DATA XNAME1/60.0,1.0,1.0,0.0,0.0,2.0,0.0,10.,3.0,2.42,120.,000.0,
     1                    1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0/
      EQUIVALENCE (XNAME(1),SPEED),(XNAME(2),PRTOUT) ,
     *              (XNAME(6),AXLNUM),(XNAME(7),STRDSP),
     *              (XNAME(8),STRBMP),(XNAME(9),NMBMPS),
     *              (XNAME(10),BMPHT),(XNAME(11),BMPWDT),
     *              (XNAME(12),BMPSPA),(XNAME(3),PRTTEL)
      REAL NMBMPS
      WRITE(6,555)
555   FORMAT(1H1)
 2    CONTINUE
      READ(5,800,END=190)BMPWDT,NREC ,DSPACE,NLCROS, NRCRDI,NRCRDF
      XDIST=DSPACE*750
 800  FORMAT(G10.2,I10,F10.2,3I10)
      IF( NLCROS.EQ.0 ) GO TO 111
      READ(5,1010)A1,A2,B1,B2,C1,C2,D1,D2,T1,T2,EE
1010  FORMAT(8F10.2)
      READ(5,2020)SM1,SM2,BH1,BH2,CH1,CH2,TH1,TH2,DH1,DH2
2020  FORMAT(10F5.3,I5)
      X=A1+A2+B1+B2+T1+T2+C1+C2+D1+D2+EE
      NREC=X/XDIST +1
111   CONTINUE
      DO 1 I=1,20
      XNAME(I)=XNAME1(I)
  1   CONTINUE
      CALL TAPEWR(NREC)
      CALL READIN
      READ(5,500) SPEED,STRBMP,        BMPHT,BMPWDT,BMPSPA,NPTTS
500   FORMAT(5G10.2,I10)
      IF(NREC.EQ.0) GO TO 3
      NMBMPS=4.
 3    NPTTS= 8.0/DSPACE +1
      CALL PREAVG(NPTTS)
      CALL DRIVER
      IF (NLCROS.NE.0) GO TO 4
      IF(NREC.EQ.0) GO TO 4
      WRITE(6,600) (XNAME(L),L=1,20)
      WRITE(6,600)SPEED,STRBMP,NMBMPS,BMPHT,BMPWDT,BMPSPA,NPTTS
600   FORMAT(1X,10G13.5)
 4    GO TO 2
190   STOP
      END
      SUBROUTINE TAPEWR(NREC)
      COMMON / AZ7 / XDIST,DSPACE
       COMMON /AZ9/ NRCRDI,NRCRDF
      DIMENSION  CARD(20)
      DIMENSION TAPE1(750) ,TAPE2(750)
      DIMENSION ID(5)
      REWIND 1
      IF(NREC.LT.0) GO TO 46
      IF(NREC.EQ.0) GO TO 5
      IN=NREC*XDIST
      WRITE(6,1235)NREC, IN
 1235 FORMAT(10X,'SUBROUTINE TAPEWR=',I3,'RECORDS=',10X,I6,
     *' INCHES OF NATURAL PROFILE')
      DO 100 I=1,750
      TAPE1(I)=0.0
 100  TAPE2(I)=0.0
      ID(1)=0
      ID(2)=NREC
      ID(3)=0
      ID(4)=0
      ID(5)=0
      WRITE(1) ID
      DO 200 I=1,NREC
  200 WRITE(1) TAPE1,TAPE2
   6  ENDFILE 1
      ENDFILE 1
      REWIND 1
      RETURN
 5    CONTINUE
      DO 1 I=1,3
      READ(10,111) CARD
 111  FORMAT(20A4)
      WRITE(6,112) CARD
 112  FORMAT(1X,20A4,//)
 1    CONTINUE
      READ(10,17) ID
  17  FORMAT(5I10)
      NREC=ID(2)
      IN=NREC*XDIST
      WRITE(6,1234)NREC,IN
 1234 FORMAT(10X,'SUBROUTINE TAPEWR  =',I3,'RECORDS',10X,I6,'INCHES OF S
     *MOOTH PROFILE')
      WRITE(1) ID
      DO 201  I=1,NREC
      READ(10,18) TAPE1,TAPE2
       IF (I.LT.NRCRDI.OR.I.GT.NRCRDF) GO TO  201
      WRITE(1) TAPE1,TAPE2
 201  CONTINUE
C     *******************************************************************
      GO TO 6
  46  REWIND 1
      READ(1) ID
      NREC=ID(2)
      IN=NREC*XDIST
      WRITE(6,1235)NREC,IN
      REWIND 1
  18  FORMAT(20F5.2)
      RETURN
      END
      SUBROUTINE READIN
      COMMON/AZ1/ PX1R,PX1L,PX2R,PX2L,DPX1R,DPX1L,DPX2R,DPX2L,
     1  DDPX1R,DDPX1L,DDPX2R,DDPX2L,ADPX1R,ADPX1L,ADPX2R,ADPX2L,IEND
      COMMON /AZ2/A1,A2,B1,B2,C1,C2,D1,D2,T1,T2,EE
     1 ,NLCRCS,SM1,SM2,EH1,EH2,CH1,CH2,TH1,TH2,DH1,DH2
      COMMON /AZ4/ PKR(2),PKL(2),CPR(2),CPL(2),PVWTR(2),PVWTL(2)
      COMMON/SET1/TVR(3750),TVL(3750),I77,AXWTR(5),AXWTL(5),X1(5),
     1 ITITLE(16),ICLASS(8),NDF(5),W(5),AXAWT(5),VL(5,2),VR(5,2)
      COMMON / SET2 / IEXCIT,IUNIT,H,NAXL,ITIME
     1,V,SPACEL,SPACER,WLL,WLR,IFREQ,BM0,BM1R,BM1L,BM2R,BM2L,
     2                     EIX,EIZ,ETIZ,VFPS      ,M,KONVER,BETA,HT2,HO2,HE
     32,HHO2,GRAV,ISTOP,NITIME,KR,XX1,XX2,XX3,XX4,XX5,X12,X13,X14,X15,
     4PHIX1,DPHIX1,DDPX1,   ADDX1,   PHIX2,DPHIX2,DDPX2,ADDX2,PHIZ1,
     5DPHIZ1,DDPZ1,ADDZ1,PHIZ2,DPHIZ2,DDPZ2,ADDZ2,Y01,DY01
      COMMON / SEP2 / DDY01,ADDY01,Y02,DY02,DDY02,ADDY02,TRL,A,XC1,XC2,
     1XTR2,XTRT,XTR23,XTR45,EIZ1,EIZ2,BIX1,BIX2,BTIZ1,BTIZ2,BM01,BM02,
     2NREAD,XTR3,XC23
      COMMON/SET3/Y0,DY0,DDY0,ADDY0,Y1R,DY1R,DDY1R,ADDY1R,Y1L,DY1L,DDY1L
     1,ADDY1L,Y2R,DY2R,DDY2R,ADDY2R,Y2L,DY2L,DDY2L,ADDY2L
      COMMON / SET4     / PHIX,DPHIX,DDPHIX,ADDIX,PHIZ,DPHIZ,DDPHIZ,
     *                     ADDIZ,THTZR,DTHTZR,DDTZR,ADDZR,THTZL,
     *                     DTHTZL,DDTZL,ADDZL,PV1R,PV1L,PV2R,PV2L,
     *                                                  CLOTOL,IOUT,TLIM
      COMMON / SET5     / KSTEP,ID(5), I2,XLI,XRI
      COMMON/SET6/AXAWT1,AXAWT2,AXWT1,AXWT2,ECCTCT,ECDWT1,BODWT2,CS1L,
     *CS2L,CS1R,CS2R,CT1L,CT2L,CT1R,CT2R,SK1L,SK2L,SK1R,SK2R,TK1L,TK2L
      COMMON/SEF6/TK1R,TK2R,W1,W2,X2,DSTL,AMPR,AMPL,NBR,NBL
      COMMON / NAME     / NNAME(20),VMPH,XPRT,YPRT,BRLGTH,DSTR,XFAKE(15),
     *                     NEND
      COMMON/AZ5/CTR(5),CTL(5),CSR(5),CSL(5),TKR(5),TKL(5),SKL(5),SKR(5
C     INPUT
C     THE SPEED SHOULD BE LIMITED BETWEEN 6.0 AND 68.0  M.P.H.
      READ  (5,105,END=190) NAXL,IEXCIT
      READ(5,110)DSTR,DSTL,AMPR,AMPL,SPACER,SPACEL,WLR,WLL
      READ(5,110)V
      READ(5,115) CLOTOL,TLIM,IFREQ,NBR,NBL
      READ(5,110) X1(2)
      READ(5,110)(W(J)  , J = 1, 2)
      READ(5,110)(AXAWT(J), J =  1,2)
      READ(5,110)(SKR(J)  , J =  1,2)
      READ(5,110) (SKL(J)  , J = 1 ,2)
      READ(5,110)(CSR(J)  , J =  1 , 2)
      READ(5,110)(CSL(J)  , J = 1 ,    2)
      READ(5,110)(TKR(J)  , J =  1,2)
      READ(5,110)(TKL(J)  , J =  1 ,2)
      READ(5,110)(CTR(J)  , J =  1 ,2)
      READ(5,110)(CTL(J)  , J =  1 ,2)
      READ(5,110) (AXWTR(J), J =  1 , 2)
      READ(5,110) (AXWTL(J), J =  1 , 2)
C     READ PAVEMENT PROPERTIES
      READ(5,110)(PKR(J), J=1,2)
      READ(5,110)(PKL(J), J=1,2)
      READ(5,110)(CPR(J), J=1,2)
      READ(5,110)(CPL(J), J=1,2)
      READ(5,110)(PVWTR(J), J=1,2)
      READ(5,110)(PVWTL(J), J=1,2)
      READ(5,100)(ITITLE(I),I=1,16)
      READ(5,100)(ICLASS(I),I=1,8)
```

```
100   FORMAT(20A4)
105   FORMAT(3I10)
110   FORMAT(8E10.3)
115   FORMAT(2E10.3,4I10)
      READ(1)   ID
      NRECRD=ID(2)
      XRI=ID(4)/10000.
      XLI=ID(5)/10000.
      VFPS  =  V/12.0
      VMPH  =  VFPS*15.0/22.0
      RETURN
 190  STOP
      END
      SUBROUTINE PRFAVG(NTA)
      COMMON /AZ2/A1,A2,B1,B2,C1,C2,D1,D2,T1,T2,EE
     1 ,NLCRCS,SM1,SM2,BH1,BH2,CH1,CH2,TH1,TH2,DH1,DH2
      COMMON / AZ7 / XDIST,DSPACE
      COMMON/SET1/TVR(3750),TVL(3750),I77,AXWTR(5),AXWTL(5),X1(5),
     1 ITITLE(16),ICLASS(8),NDF(5),W(5),AXAWT(5),VL(5,2),VR(5,2)
      COMMON / NAME / NNAME(20),VMPH,XPRT,YPRT,BRLGTH,DSTR,XFAKE(15),NE
      DIMENSION TAPE1(1500),P1(750),P2(750),TEMP(100)
      EQUIVALENCE (TVR,TAPE1),(TVR(1501),P1),(TVR(2251),P2),
     *(TVR(3001),TEMP)
      IF(NTA.EQ.0) NTA=8.0/DSPACE +1
      IF(NTA.LT.1.OR.NTA.GT.50) RETURN
      NT2=NTA/2
      NREC=0
      NT2=NT2*2
      IF(NT2.EQ.NTA)NTA=NTA+1
      IF(NTA.NE.1)GO TO 1
      REWIND 2
2     READ(1,END=3) TVR
      DO 13 J=1,1500
      KJ=(J-1)/2
   13 TVR(J)=TVR(J)*XFAKE(8)+FBUMP(KJ+1+NREC*750)
     1                          +XBUMP(KJ+1+NREC*750)
      NREC=NREC+1
      WRITE(2) TVR
      GO TO 2
    3 ENDFILE 2
      REWIND 2
      DO 15 LL=1,NREC
   15 BACKSPACE 1
      GO TO 500
    1 NT2=NTA/2
      REWIND 2
      NT3=NTA*2
      READ(1) TAPE1
      DO 10 J=1,NT2
      P1(J)=TAPE1(2*J-1)*XFAKE(8)+FBUMP(J)
     1                                 +XBUMP(J)
   10 P2(J)=TAPE1(2*J)*XFAKE(8)+FBUMP(J)
     1                             +XBUMP(J)
      SUM1 = 0.0
      SUM 2 =0.0
      DO 12 K=1,NTA
      SUM1=SUM1+TAPE1(2*K-1)*XFAKE(8)+FBUMP(K)
     1                                       +XBUMP(K)
   12 SUM2=SUM2+TAPE1(2*K)*XFAKE(8)+FBUMP(K)
     1                                    +XBUMP(K)
      P1(NT2+1)=SUM1/NTA
      P2(NT2+1)=SUM2/NTA
   30 K=NT2+2
   50 SUM1=SUM1+(TAPE1(2*K+NTA-2)-TAPE1(2*K-NTA-2))*
     *       XFAKE(8)+FBUMP(K+NT2+NREC*750)-FBUMP(K-NT2+NREC*750)
     1             +XBUMP(K+NT2+NREC*750)-XBUMP(K-NT2+NREC*750)
      SUM2=SUM2+(TAPE1(2*K+NTA-1)-TAPE1(2*K-NTA-1))*
     *       XFAKE(8)+FBUMP(K+NT2+NREC*750)-FBUMP(K-NT2+NREC*750)
     1             +XBUMP(K+NT2+NREC*750)-XBUMP(K-NT2+NREC*750)
      P1(K)=SUM1/NTA
      P2(K)=SUM2/NTA
      K=K+1
      ITEST=750-NT2
      IF(K.GT.ITEST)GC TC 40
      GO TC 5C
   40 CC 45 L=1,NT3
   45 TEMF(L)=TAPE1(L+15C0-2*NTA)
      FEAC(1,ENC=48)   TAPE1
      CC 46 L=1,NTA
      SLM1=SLM1+(TAPE1(2*L-1)-TEMF(2*L-1))*XFAKE(8)
     *     +FEUMP(L+(NREC+1)*750)-FBUMP(750-(NTA-L)+NREC*750)
     1     +XEUMF(L+(NREC+1)*750)-XEUMP(750-(NTA-L)+NREC*750)
      SUM2=SUM2+(TAPE1(2*L)-TEMP(2*L))*XFAKE(8)
     *     +FEUMF(L+(NREC+1)*75C)-FBUMP(750-(NTA-L)+NREC*750)
     1     +XEUMF(L+(NREC+1)*750)-XEUMF(750-(NTA-L)+NREC*750)
      P1(K)=SLM1/NTA
      F2(K)=SUM2/NTA
      K=K+1
      IF (K.EQ.751 ) GO TO 47
      GC TC 46
   47 K=1
      WRITE(2) (F1(I),F2(I),I=1,75C)
   46 CONTINLE
      NREC=NREC+1
      GC TC 30
   48 CC 7C N=K,75C
      P1(N)=TAPE1(N*2-1)*XFAKE(8)+FBLMP(N+75C*NREC)
     1                             +XEUMF(N+750*NREC)
   7C P2(N)=TAPE1(N*2)*XFAKE(8)+FEUMF(N+750*NREC)
     1                          +XBUMP(N+750*NREC)
      WFITE(2)(F1(I),F2(I),I=1,750)
      ENCFILE 2
      FEWINC 2
      IF (NREC.EC.C ) NREC=1
      CO 16 LL=1,NREC
   16 EACKSPACE 1
  5CC  CONTINLE
      WRITE(6,31) NTA
   31 FCFMAT( I10,' PCINT MCVING AVERAGE OF PROFILE' )
      RETURN
      ENC
      SUBROUTINE DRIVER
      COMMON/AZ1/ PX1R,PX1L,PX2R,PX2L,DPX1R,DPX1L,DPX2R,DPX2L,
     1  DDPX1R,DDPX1L,DDPX2R,DDPX2L,ADPX1R,ADPX1L,ADPX2R,ADPX2L,IEND
      COMMON /AZ2/A1,A2,E1,E2,C1,C2,D1,D2,T1,T2,EE
     1 ,NLCROS,SM1,SM2,BH1,BH2,CH1,CH2,TH1 ,TH2,DH1,DH2
      COMMON / AZ7 / XDIST,DSPACE
      COMMON/SET1/TVR(3750),TVL(3750 ),I77,AXWTR(5),AXWTL(5),X1(5),
     1 ITITLE(16),ICLASS(8),NDP(5),W(5),AXAWT(5),VL(5,2),VR(5,2)
C
      COMMON / SET2 / IEXCIT ,IUNIT,H,NAXL,ITIME
     1,V,SPACEL,SPACER,WLL,WLR,IFREQ,BM0,BM1R,EM1L,EM2R,BM2L,
     2                     EIX,EIZ,ETIZ,VFPS      ,M,KONVER,BETA,HT2,HO2,HE
     32,HHO2,GRAV,ISTOP,NITIME,KR,XX1,XX2,XX3,XX4,XX5,X12,X13,X14,X15,
     4PHIX1,DPHIX1,DDPX1   ,ADDX1   ,PHIX2,DPHIX2,DDPX2,ADDX2,PHIZ1,
     5DPHIZ1,DDPZ1   ,ADDZ1   ,PHIZ2,DPHIZ2,DDPZ2,ADDZ2,Y01,DY01
      COMMON / SEP2 / DDYC1,ADDYC1,YC2,DY02,DDY02,ADDY02,TRL,A,XC1,XC2,
     1XTR2,XTRT,XTR23,XTR45,EIZ1,EIZ2,BIX1,BIX2,BTIZ1,BTIZ2,BM01,BM02,
     2NREAD,XTR3,XC23
      COMMON/SET3/Y0,DY0,DDY0,ADDY0,Y1R,DY1R,DDY1R,ADDY1R,Y1L,DY1L,DDY1L
     1,ADDY1L,Y2R,DY2R,DDY2R,ADDY2R,Y2L,DY2L,DDY2L,ADDY2L
      COMMON/SET4/PHIX,DPHIX,DDPHIX,ADDIX,PHIZ,DPHIZ,DDPHIZ,ADDIZ,
     1THTZR,DTHTZR,DDTZR,ADDZR,THTZL,DTHTZL,DDTZL,ADDZL,PV1R,PV1L ,
     2  PV2R,PV2L,                                   CLOTOL,IOUT,TLIM
C
      COMMON / SET5 / KSTEP, ID(5),I2,XLI,XRI
      COMMON / NAME / NNAME(20),VMPH,XPRT,YPRT,BRLGTH,CSTR,XFAKE(15),NE
      COMMON/AZ5/CTR(5),CTL(5),CSR(5),CSL(5),TKR(5),TKL(5),SKL(5),SKR(5)
      IEND=0
      X1(1)=0.0
      REWIND 2
      VFPS=VMPH*22.0/15.0
      V=VFPS*12.0
      KSTEP=1
      IFREQ=2
      IF(XFAKE(14).NE.0.0)  IFREQ=XFAKE(14)
      H=DSPACE*KSTEP/V
      DO 20 L=1,2
      VR(L,2)=XRI
      VL(L,2)=XLI
   20 NDP(L)=-X1(L)/DSPACE+KSTEP
C
      DY0=0.0
      DDY0=0.0
      ADDY0=0.0
      Y0=(XRI+XLI)/2.0
      Y1R=XRI
      Y1L=XLI
      Y2R=XRI
      Y2L=XLI
      DY1R=0.0
      DY1L=0.0
      DY2R=0.0
      DY2L=0.0
      DDY1R=0.0
      DDY1L=0.0
      DDY2R=0.0
      DDY2L=0.0
      PX1R=XRI
```

```
PX1L=XLI
PX2R=XRI
PX2L=XLI
DPX1R=0.0
DPX1L=0.0
DPX2R=0.0
DPX2L=0.0
DDPX1R=0.0
DDPX1L=0.0
DDPX2R=0.0
DDPX2L=0.0
ADPX1R=0.0
ADPX1L=0.0
ADPX2R=0.0
ADPX2L=0.0
PHIX=0.0
PHIZ=0.0
DPHIX=0.0
DPHIZ=0.0
DDPHIX=0.0
DDPHIZ=0.0
PV1R=XRI
PV1L=XLI
PV2R=XRI
PV2L=XLI
ADDY1R=0.0
ADDY1L=0.0
ADDY2R=0.0
ADDY2L=0.0
ADDIX=0.0
ADDIZ=0.0
Y01=0.0
DY01=0.0
DDY01=0.0
ADDY01=0.0
Y02=0.0
DDY02=0.0
DY02=0.0
ADDY02=0.0
Y01=(XLI+XRI)/2.0
Y02=Y01
PHIX1=0.0
DPHIX1=0.0
DDPX1=0.0
ADDX1=0.0
PHIX2=0.0
DPHIX2=0.0
DDPX2 = 0.0
ADDX2 = 0.0
PHIX1=ATAN((XLI-XRI)/W(1))
PHIX2 = PHIX1
PHIZ1=0.0
DPHIZ1=0.0
DDPZ1 = 0.0
ADDZ1 = 0.0
PHIZ2=0.0
DPHIZ2=0.0
DDPZ2 = 0.0
```

```
      ADDZ2 = 0.0
      N   = 0
      KONVER = 0
               BETA = 1.0 / 6.0
               HT2  = 2.0 * H
               HC2  = 0.5 * H
               HE2  = H * H
               HHC2 = 0.5 * H * H
               GRAV = 386.4
C     SET UP RANGE OVER WHICH RESPONSE IS TO BE STUDIED.
      ITIME = TLIM /H
      ISTOP = ITIME-1
      NITIME=1
280   CALL CLASS 1
  490     CONTINUE
      WRITE(6,99999)
99999 FORMAT(1H1)
      RETURN
          END
      SUBROUTINE CLASS1
      COMMON/AZ1/ PX1R,PX1L,PX2R,PX2L,DPX1R,DPX1L,DPX2R,DPX2L,
     1  DDPX1R,DDPX1L,DDPX2R,DDPX2L,ADPX1R,ADPX1L,ADPX2R,ADPX2L,IEND
      COMMON/AZ3/PK1R,PK1L,PK2R,PK2L,CP1R,CP1L,CP2R,CP2L,PVWT1R,
     1  PVWT1L,PVWT2R,PVWT2L
      COMMON /AZ4/ PKR(2),PKL(2),CPR(2),CPL(2),PVWTR(2),PVWTL(2)
      COMMON/SET1/TVR(3750),TVL(3750),I77,AXWTR(5),AXWTL(5),X1(5),
     1 ITITLE(16),ICLASS(8),NCF(5),W(5),AXAWT(5),VL(5,2),VR(5,2)
C
      COMMON / SET2 / IEXCIT,IUNIT,H,NAXL,ITIME
     1,V,SPACEL,SPACER,WLL,WLR,IFREQ,BM0,BM1R,BM1L,BM2R,BM2L,
     2                    BIX,BIZ,BTIZ,VFPS      ,M,KONVER,BETA,HT2,HO2,HE
     32,RHO2,GRAV,ISTOP,NITIME,KR,XX1,XX2,XX3,XX4,XX5,X12,X13,X14,X15,
     4PHIX1,DPHIX1,DDPX1,ADDX1,PHIX2,DPHIX2,DDPX2,ADDX2,PHIZ1,
     5DPHIZ1,DDPZ1,ADDZ1,PHIZ2,DPHIZ2,DDPZ2,ADDZ2,Y01,DY01
      COMMON / SEF2 / DDY01,ADDY01,Y02,DY02,DDY02,ADDY02,TRL,A,XC1,XC2,
     1XTR2,XTRT,XTR23,XTR45,EIZ1,EIZ2,EIX1,EIX2,BTIZ1,BTIZ2,BM01,BM02,
     2NREAD,XTR3,XC23
      COMMON/SET3/Y0,DY0,DDY0,ADDY0,Y1R,DY1R,DDY1R,ADDY1R,Y1L,DY1L,DDY1L
     1,ADDY1L,Y2R,DY2R,DDY2R,ADDY2R,Y2L,DY2L,DDY2L,ADDY2L
C
      COMMON/SET4/PHIX,DPHIX,DDPHIX,ADDIX,PHIZ,DPHIZ,DDPHIZ,ADDIZ,
     1THTZR,DTHTZR,DDTZR,ADDZR,THTZL,DTHTZL,DDTZL,ADDZL,PV1R,PV1
     2L,PV2R,PV2L,                                 CLOTOL,IOUT,TLIM
C
      COMMON / SET5 / KSTEP,ID(5),I2,XLI,XRI
C
      COMMON/SET6/AXAWT1,AXAWT2,AXWT1,AXWT2,BODTOT,BODWT1,BODWT2,CS1L,
     *CS2L,CS1R,CS2R,CT1L,CT2L,CT1R,CT2R,SK1L,SK2L,SK1R,SK2R,TK1L,TK2L
      COMMON/SEF6/TK1R,TK2R,W1,W2,X2,DSTL,AMPR,AMPL,NBR,NEL
C
      COMMON / NAME / NNAME(20),VMPH,XPRT,YPRT,BRLGTH,DSTR,XFAKE(15),NE
      COMMON / OVRLAY / NCLASS,NFFINF
      COMMON/AZ5/CTR(5),CTL(5),CSR(5),CSL(5),TKR(5),TKL(5),SKL(5),SKR(5)
C     SET INPUT
      PK1R=PKR(1)
      PK2R=PKR(2)
      PK1L=PKL(1)
      PK2L=PKL(2)
      CP1R=CPR(1)
      CP2R=CPR(2)
      CP1L=CPL(1)
      CP2L=CPL(2)
      PVWT1R=PVWTR(1)
      PVWT2R=PVWTR(2)
      PVWT1L=PVWTL(1)
      PVWT2L=PVWTL(2)
      BMPV1R=PVWT1R/386.4
      BMPV1L=PVWT1L/386.4
      BMPV2R=PVWT2R/386.4
      BMPV2L=PVWT2L/386.4
      SK1R=SKR(1)
      SK2R=SKR(2)
      SK1L=SKL(1)
      SK2L=SKL(2)
      TK1R=TKR(1)
      TK2R=TKR(2)
      TK1L=TKL(1)
      TK2L=TKL(2)
      CS1R=CSR(1)
      CS2R=CSR(2)
      CS1L=CSL(1)
      CS2L=CSL(2)
      CT1R=CTR(1)
      CT2R=CTR(2)
      CT1L=CTL(1)
      CT2L=CTL(2)
      W1=W(1)
      W2=W(2)
      X12=X1(2)
      AXAWT1=AXAWT(1)
      AXAWT2=AXAWT(2)
      AXWT1=AXWTL(1) +AXWTR(1)
      AXWT2=AXWTL(2) +AXWTR(2)
C     LOCATE THE CENTER OF GRAVITY OF THE VEHICLE BODY
               BODWT1 = AXWT1 - AXAWT1
               BODWT2 = AXWT2 - AXAWT2
               BODTOT = BODWT1 + BODWT2
                   X2 = BODWT1 * X12 /BODTOT
                      X3 = X12 - X2
                  XX1=X3
                  XX2=X2
C     CALCULATE THE BODY MASSES AND MASS MOMENT OF INERTIA
               BMO = BODTOT / 386.4
               BM1R = (0.5) * AXAWT1 / 386.4
               BM1L = BM1R
               BM2R = (0.5) * AXAWT2 / 386.4
               BM2L = BM2R
               BIX  =(BMO * ( W1*W1 + W2*W2 ) /24.0)*XFAKE(10)
               BIZ =(BMO * X12 * X12 / 12.0)*XFAKE(9)
C     NOW LET US PRINT THE INPUT DATA
      NCLASS=1
      CALL INDAP
C     OUTSIDE OF TIME STEPS SET THE FOLLOWING
      I=1
      XMAYSA = 0.0
      XMAYSM=0.0
      DO 190 I2=1,ISTOP
      ICOUNT=0
      CALL NATPRO
      IF(IEND.NE.0) RETURN
      X3=XX1
               M = M + 1
      PX1R=PX1R+H*DPX1R+HHO2*DDPX1R
      PX1L=PX1L+H*DPX1L+HHO2*DDPX1L
      PX2R=PX2R+H*DPX2R+HHO2*DDPX2R
      PX2L=PX2L+H*DPX2L+HHO2*DDPX2L
      DPX1R=DPX1R+H*DDPX1R
      DPX1L=DPX1L+H*DDPX1L
      DPX2R=DPX2R+H*DDPX2R
      DPX2L=DPX2L+H*DDPX2L
               Y1R = Y1R + H * DY1R + HHO2 * DDY1R
               Y1L = Y1L + H * DY1L + HHO2 * DDY1L
               Y2R = Y2R + H * DY2R + HHO2 * DDY2R
               Y2L = Y2L + H * DY2L + HHO2 * DDY2L
               YO  = YO  + H * DYO  + HHO2 * DDYO
```

```
              PHIX = PHIX + H * DPHIX + HHO2 * DDPHIX
              PHIZ = PHIZ + H * DPHIZ + HHO2 * DDPHIZ
              DY1R = DY1R + H * DDY1R
              DY1L = DY1L + H * DDY1L
              DY2R = DY2R + H * DDY2R
              DY2L = DY2L + H * DDY2L
              DYO  = DYO  + H * DDYO
              DPHIX = DPHIX + H * DDPHIX
              DPHIZ = DPHIZ + H * DDPHIZ
  160         D1R = YO - (W1/2.0) * PHIX - X3   *PHIZ
              D1L = YO + (W1/2.0) * PHIX - X3   *PHIZ
              D2R = YO - (W2/2.0) * PHIX + X2 * PHIZ
              D2L = YO + (W2/2.0) * PHIX + X2 * PHIZ
              DD1R =DYO - (W1/2.0) * DPHIX - X3   *DPHIZ
              DD1L =DYO + (W1/2.0) * DPHIX - X3   *DPHIZ
              DD2R =DYO - (W2/2.0) * DPHIX + X2 * DPHIZ
              DD2L = DYO + (W2/2.0)*DPHIX + X2 * DPHIZ
       IF(ICOUNT.EQ.30) KONVER=1
      ICOUNT=ICOUNT+1
C     CALCULATE DYNAMIC FORCE ON THE PAVEMENT
      DYPV1R=CP1R*((VR(1,I+1)-PV1R)/HT2-DPX1R)+PK1R*(VR(1,I)-PX1R)
      DYPV1L=CP1L*((VL(1,I+1)-PV1L)/HT2-DPX1L)+PK1L*(VL(1,I)-PX1L)
      DYPV2R=CP2R*((VR(2,I+1)-PV2R)/HT2-DPX2R)+PK2R*(VR(2,I)-PX2R)
      DYPV2L=CP2L*((VL(2,I+1)-PV2L)/HT2-DPX2L)+PK2L*(VL(2,I)-PX2L)
C     CALCULATE DYNAMIC FORCES ON THE TIRE
      DYF1R=CT1R*(DPX1R-DY1R) +TK1R*(PX1R-Y1R)
      DYF1L=CT1L*(DPX1L-DY1L) +TK1L*(PX1L-Y1L)
      DYF2R=CT2R*(DPX2R-DY2R) +TK2R*(PX2R-Y2R)
      DYF2L=CT2L*(DPX2L-DY2L) +TK2L*(PX2L-Y2L)
         TDF1R = DYF1R + AXWTR(1)
         TDF1L = DYF1L + AXWTL(1)
         TDF2R = DYF2R + AXWTR(2)
         TDF2L = DYF2L + AXWTL(2)
         IF( TDF1R .LT. 0 )  DYF1R =-AXWTR(1)
         IF( TDF1L .LT. 0 )  DYF1L =-AXWTL(1)
         IF( TDF2R .LT. 0 )  DYF2R =-AXWTR(2)
         IF( TDF2L .LT. 0 )  DYF2L =-AXWTL(2)
C     ACCELERATION ESTIMATES FROM DIFFERENTIAL EQUATIONS OF MOTION
           DDY1R = ( SK1R*(D1R-Y1R) + CS1R*(DD1R-DY1R)+DYF1R)/BM1R
           DDY1L = ( SK1L*(D1L-Y1L) + CS1L*(DD1L-DY1L)+DYF1L)/BM1L
           DDY2R = ( SK2R*(D2R-Y2R) + CS2R*(DD2R-DY2R)+DYF2R)/EM2R
           DDY2L = ( SK2L*(D2L-Y2L) + CS2L*(DD2L-DY2L)+DYF2L)/EM2L
           DDPHIX =( (W1/2.0) * SK1R *( D1R-Y1R)-(W1/2.0)*SK1L*(D1L-Y1L)
     1               +(W2/2.0) * SK2R * (D2R-Y2R)-(W2/2.0)*SK2L*(D2L-Y2L)
     2               +(W1/2.0)*CS1R*(DD1R-DY1R)-(W1/2.0)*CS1L*(DD1L-DY1L)
     3               +(W2/2.0)*CS2R*(DD2R-DY2R)-(W2/2.0)*CS2L*(DD2L-DY2L)
     4               ) /BIX
       DDPHIZ=(X3    *SK1R*(D1R-Y1R)+X3    *SK1L*(D1L-Y1L)-X2*SK2R*(D2R-Y2R
     1        )-X2*SK2L*(D2L-Y2L)+X3    *CS1R*(DD1R-DY1R)+X3    *CS1L*(DD1L
     2        -DY1L)-X2*CS2R*(DD2R-DY2R)-X2*CS2L*(DD2L-DY2L))/BIZ
          DDYO =(-SK1R*(D1R-Y1R) - SK1L*(D1L-Y1L) - SK2R*(D2R-Y2R)
     1           -SK2L*(D2L-Y2L) - CS1R*(DD1R-DY1R)-CS1L*(DD1L-DY1L)
     2           -CS2R*(DD2R-DY2R) - CS2L*(DD2L-DY2L) ) / BMO
C     CALCULATE ACCELERATION OF SOIL MASSES
      DDPX1R=(DYPV1R-DYF1R)/EMPV1R
      DDPX1L=(DYPV1L-DYF1L)/EMPV1L
      DDPX2R=(DYPV2R-DYF2R)/EMPV2R
      DDPX2L=(DYPV2L-DYF2L)/BMPV2L
```

```
C **  CHECK FOR CONVERGENCE
C
          IF ( ABS( ADDY1R - DDY1R ) .GT. CLOTOL )  GO TO 170
          IF ( ABS( ADDY1L - DDY1L ) .GT. CLOTOL )  GO TO 170
          IF ( ABS( ADDY2R - DDY2R ) .GT. CLOTOL )  GO TO 170
          IF ( ABS( ADDY2L - DDY2L ) .GT. CLOTOL )  GO TO 170
          IF ( ABS( ADDYC  - DDYC  ) .GT. CLOTOL )  GO TO 170
          IF ( ABS(ADDIX - DDPHIX ) .GT. CLOTOL )  GO TO 170
          IF ( ABS(ADDIZ - DDPHIZ ) .GT. CLOTOL ) GO TO 170
C     CHECK FOR CONVERGENCE OF ACCELERATION OF SOIL MASS
      IF(ABS(ADPX1R-DDPX1R).GT.CLOTOL) GO TO 170
      IF(ABS(ADPX1L-DDPX1L).GT. CLOTOL) GO TO 170
      IF(ABS(ADPX2R-DDPX2R).GT.CLOTOL) GO TO 170
      IF(ABS(ADPX2L-DDPX2L).GT.CLOTOL) GO TO 170
              KONVER = 1
C
C **  REVISE ESTIMATES
C
 170          DY1R = DY1R + HO2 * ( DDY1R - ADDY1R )
              DY1L = DY1L + HO2 * ( DDY1L - ADDY1L )
              DY2R = DY2R + HO2 * ( DDY2R - ADDY2R )
              DY2L = DY2L + HO2 * ( DDY2L - ADDY2L )
              DYO  = DYO  + HO2 * ( DDYO  - ADDYO  )
              DPHIX = DPHIX + HO2 * ( DDPHIX - ADDIX)
              DPHIZ = DPHIZ + HO2 * ( DDPHIZ - ADDIZ)
              Y1R = Y1R + BETA * HE2 * ( DDY1R - ADDY1R )
              Y1L = Y1L + BETA * HE2 * ( DDY1L - ADDY1L )
              Y2R = Y2R + BETA * HE2 * ( DDY2R - ADDY2R )
              Y2L = Y2L + BETA * HE2 * ( DDY2L - ADDY2L )
              YO  = YO  + BETA * HE2 * ( DDYO  - ADDYO  )
              PHIX = PHIX+BETA*HE2* ( DDPHIX - ADDIX )
              PHIZ = PHIZ + BETA*HE2* (DDPHIZ - ADDIZ )
C     REVISE ESTIMATES
      DPX1R=DPX1R+HO2*(DDPX1R-ADPX1R)
      DPX1L=DPX1L+HO2*(DDPX1L-ADPX1L)
      DPX2R=DPX2R+HO2*(DDPX2R-ADPX2R)
      DPX2L=DPX2L+HO2*(DDPX2L-ADPX2L)
      PX1R=PX1R+BETA*HE2*(DDPX1R-ADPX1R)
      PX1L=PX1L+BETA*HE2*(DDPX1L-ADPX1L)
      PX2R=PX2R+BETA*HE2*(DDPX2R-ADPX2R)
      PX2L=PX2L+BETA*HE2*(DDPX2L-ADPX2L)
      ADPX1R=DDPX1R
      ADPX1L=DDPX1L
      ADPX2R=DDPX2R
      ADPX2L=DDPX2L
      ADDY1R=DDY1R
      ADDY1L=DDY1L
      ADDY2R=DDY2R
      ADDY2L=DDY2L
      ADDYC=DDYC
      ADDIX = DDPHIX
      ADDIZ = DDPHIZ
         IF( KONVER .NE. 1 )  GO TO 160
                   KONVER = 0
      IF(M.NE.IFREQ) GO TO 160
      TIME=I2*H
               DIST = TIME * V
        TDF1R = DYF1R + AXWTR(1)
```

```
         TDF1L = DYF1L + AXWTL(1)
         TDF2R = DYF2R + AXWTR(2)
         TDF2L = DYF2L + AXWTL(2)
                M = 0
      X1(1)=0.0
C     MAYS METER SIMULATOR
      XMAYS = 0.5*(Y2R + Y2L - D2R - D2L )
      ABSMAY=ABS(XMAYS-XMAYSM)
      XMAYSA=XMAYSA+ABSMAY
      XMAYSM=XMAYS
 175  WRITE(6,125)    TIME,DIST,VR(1,I),VL(1,I),VR(2,I),VL(2,I),TDF1R,
     1                TDF1L,TDF2R,TDF2L,XMAYSA   ,ICOUNT
 125  FORMAT (F10.3,F8.0,2X,4F8.3,2F8.0,2F9.0,12X,G12.4,I15)
 180  PV1R=VR(1,I)
      PV1L=VL(1,I)
      PV2R=VR(2,I)
      PV2L=VL(2,I)
  190     CONTINUE
          RETURN
          END
```

```
       COMMON/AZ1/ PX1R,PX1L,PX2R,PX2L,DPX1R,DPX1L,DPX2R,DPX2L,
      1  DDPX1R,DDPX1L,DDPX2R,DDPX2L,ADPX1R,ADPX1L,ADPX2R,ADPX2L,IEND
       COMMON/SET1/TVR(3750),TVL(3750),I77,AXWTR(5),AXWTL(5),X1(5),
      1 ITITLE(16),ICLASS(8),NDP(5),W(5),AXAWT(5),VL(5,2),VR(5,2)
       COMMON / SET2 / IEXCIT ,IUNIT,H,NAXL,ITIME
      1,V,SPACEL,SPACER,WLL,WLR,IFREQ,BM0,BM1R,BM1L,BM2R,BM2L,
      2                       BIX,BIZ,BTIZ,VFPS      ,M,KCNVER,BETA,HT2,HO2,HE
      32,HHC2,GRAV,ISTOP,NITIME,KR,XX1,XX2,XX3,XX4,XX5,X12,X13,X14,X15,
      4PHIX1,DPHIX1,DDPX1,ADDX1,PHIX2,DPHIX2,DDPX2,ADDX2,PHIZ1,
      5DPHIZ1,DDPZ1,ADDZ1,PHIZ2,DPHIZ2,DDPZ2,ADDZ2,Y01,DY01
       COMMON / SEP2 / DDY01,ADDY01,Y02,DY02,DDY02,ADDY02,TRL,A,XC1,XC2,
      1XTR2,XTRT,XTR23,XTR45,EIZ1,EIZ2,EIX1,EIX2,ETIZ1,ETIZ2,BM01,BM02,
      2NREAD,XTR3,XC23
       COMMON/SET3/Y0,DY0,DDY0,ADDY0,Y1R,DY1R,DDY1R,ADDY1R,Y1L,DY1L,DDY1L
      1,ADDY1L,Y2R,DY2R,DDY2R,ADDY2R,Y2L,DY2L,DDY2L,ADDY2L
       COMMON / SET4    / PHIX,DPHIX,DDPHIX,ADDIX ,PHIZ,DPHIZ,DDPHIZ,
      *                    ADDIZ,THTZR,DTHTZR,DDTZR,ADDZR,THTZL,
      *                    DTHTZL,DDTZL,ADDZL,PV1R,PV1L,PV2R,PV2L,
      *                                                    CLOTCL,IOUT,TLIM
       COMMON / SET5    / KSTEP,ID(5), I2,XLI,XRI
       COMMON/SET6/AXAWT1,AXAWT2,AXWT1,AXWT2,BODTOT,BODWT1,BODWT2,CS1L,
      *CS2L,CS1R,CS2R,CT1L,CT2L,CT1R,CT2R,SK1L,SK2L,SK1R,SK2R,TK1L,TK2L
       COMMON/SEP6/TK1R,TK2R,W1,W2,X2,DSTL,AMPR,AMPL,NBR,NEL
       COMMON / NAME    / NNAME(20),VMPH,XPRT,YPRT,BRLGTH,DSTR,XFAKE(15),
      *                    NEND
       IF (    I2. NE . 1 )  GO TO 1548
       JREC=1
       JPT=KSTEP
       REWIND 2
       READ(2) (TVR(MX),TVL(MX),MX=1,750)
  1548 JPT=JPT+KSTEP
       DO 138 L=1,2
       NDP(L)=NDP(L)+KSTEP
       IDEL=NDP(L)
       IF(IDEL)130,130,140
   140 IF(IDEL-3750)143,143,142
   142 IDEL=IDEL-3750
       GO TO 140
   143 VR(L,1)=VR(L,2)
       VL(L,1)=VL(L,2)
       VR(L,2)=TVR(IDEL)
       VL(L,2)=TVL(IDEL)
       GO TO 133
   130 VR(L,1)=XRI
       VL(L,1)=XLI
       VR(L,2)=XRI
       VL(L,2)=XLI
   133 IF(L.NE.1) GO TO 138
       IF(JPT-750)138,135,135
   135 JREC=JREC+1
       IF(JREC-6)137,136,137
  136  JREC=1
   137 MX1=(JREC-1)*750+1
    1  MX2=JREC*750
       READ(2,  END=1000   ) (TVR(MX),TVL(MX),MX=MX1,MX2)
       GO TO 1001
  1000 IEND=2
       RETURN
  1001 CONTINUE
       JPT=0
   138 CONTINUE
       RETURN
       END
      FUNCTION FBUMP(IX)
      COMMON / AZ7 / XDIST,DSPACE
      COMMON / NAME / NNAME(20),VMFF,XPRT,YPRT,BRLGTH,DSTR,XFAKE(15),NE
      FBUMP=0.0
      IF(XFAKE(6).EQ.0) RETURN
      IF(XFAKE(3).LE.0.0)RETURN
      IF(XFAKE(4).EQ.0.0)RETURN
      DISTE=IX*DSPACE
      IF ( DISTE.LT.  XFAKE(3)-XFAKE(11))RETURN
      DISTL=XFAKE(3)+(XFAKE(4)-1.0)*XFAKE(7)+XFAKE(6)
      IF ( DISTE.GT.DISTL) RETURN
      NBUMPS=XFAKE(4)
C     XFAKE(1)=AXLENUM
C     XFAKE(2)=STRDSP
C     XFAKE(3)=STRBMP
C     XFAKE(4)=NUMBMPS
C     XFAKE(5)=BMPHT
C     XFAKE(6)=BMPWDT
C     XFAKE(7)=BMPSPA
      DO 10 JK=1,NBUMPS
      DISTL=XFAKE(3)+(JK-1)*XFAKE(7)
      DIST2=DISTL+XFAKE(6)
      IF(DISTB.GE.DISTL.AND.DISTE.LE.DIST2)
     1   FBUMP=(XFAKE(5)*SIN(6.28318*(DISTB-DISTL)/(DIST2-DISTL)))
   10 CONTINUE
      RETURN
      END
      FUNCTION XBUMP(IX)
      COMMON /AZ2/A1,A2,B1,B2,C1,C2,D1,D2,T1,T2,EE
     1 ,NLCRCS,SM1,SM2,BH1,BH2,CH1,CH2,TH1,TH2,DH1,DH2
      COMMON / AZ7 / XDIST,DSPACE
      XBUMP=0.0
      IF(NLCRCS.EQ.0) RETURN
      XISTB=IX*DSPACE
      IF(XISTB.LE.A1) RETURN
      XX1=A1
      XX2=XX1+B1
      XX3=XX2+C1
      XX4=XX3+T1
      XX5=XX4+D1
      XX6=XX5+EE
      XX7=XX6+D2
      XX8=XX7+T2
      XX9=XX8+C2
      XX10=XX9+B2
      IF(XISTB.GE.XX1.AND.XISTB.LE.XX2)XBUMP=SM1+BH1/B1*(XISTB-XX1)
      IF(XISTB.GT.XX2.AND.XISTB.LE.XX3)XBUMP=SM1+BH1-CH1
      IF(XISTB.GT.XX3.AND.XISTB.LE.XX4)XBUMP=SM1+BH1+TH1
      IF(XISTB.GT.XX4.AND.XISTB.LE.XX5)XBUMP=SM1+BH1-DH1
      IF(XISTB.GT.XX5.AND.XISTB.LE.XX6)XBUMP=SM1+BH1
      IF(XISTB.GT.XX6.AND.XISTB.LE.XX7)XBUMP=SM1+BH1-DH2
      IF(XISTB.GT.XX7.AND.XISTB.LE.XX8)XBUMP=SM1+BH1+TH2
      IF(XISTB.GT.XX8.AND.XISTB.LE.XX9)XBUMP=SM1+BH1-CH2
      IF(XISTB.GT.XX9.AND.XISTB.LE.XX10)XBUMP=SM1+BH1-BH2/B2*(XISTB-XX9)
      IF(XISTB.GT.XX10)XBUMP=SM1+BH1-(BH2+SM2)
      RETURN
      END
      SUERCUTINE INCAP
       CCMMCN /A29/ NRCRCI,NFCFCF
      COMMCN/AZ1/ PX1R,PX1L,PX2R,PX2L,DPX1R,DFX1L,DFX2F,DFX2L,
     1  CDFX1R,CCPX1L,CCFX2R,DDFX2L,ACPX1R,ACPX1L,ADPX2R,ADPX2L,IEND
      COMMON /AZ2/A1,A2,B1,B2,C1,C2,C1,C2,T1,T2,EE
     1 ,NLCRCS,SM1,SM2,BH1,BH2,CH1,CH2,TH1,TH2,DH1,DH2
      CCMMCN/AZ3/FK1R,FK1L,FK2R,FK2L,CP1R,CP1L,CP2R,CP2L,PVWT1R,
     1  FVWT1L,PVWT2R,FVWT2L
      CCMMCN / AZ7 / XCIST,DSPACE
      CCMMCN/SET1/TVR(3750),TVL(3750),I77,AXWTR(5),AXWTL(5),X1(5),
     1 ITITLE(16),ICLASS(8),NDP(5),W(5),AXAWT(5),VL(5,2),VR(5,2)
C
      COMMON / SET2 / IEXCIT ,ILNIT,H,NAXL,ITIME
     1,V,SPACEL,SPACER,WLL,WLR,IFREG,BMO,BM1R,BM1L,EM2R,BM2L,
     2                     EIX,EIZ,ETIZ,VFPS      ,M,KONVER,BETA,HT2,HO2,HE
     32,HHC2,GRAV,ISTCF,NITIME,KF,XX1,XX2,XX3,XX4,XX5,X12,X13,X14,X15,
     4PHIX1,CPHIX1,CCPX1,ADCX1,PHIX2,DPHIX2,DDPX2,ADDX2,PHIZ1,
     5DPHIZ1,CDFZ1,ACCZ1,FHIZ2,CFHIZ2,CDPZ2,ACCZ2,YO1,CYO1
      COMMON / SEP2 / DDYC1,ADDYC1,YC2,DYO2,DDYO2,ADDYO2,TRL,A,XC1,XC2,
     1XTF2,XTRT,XTR23,XTR45,EIZ1,EIZ2,BIX1,BIX2,BTIZ1,BTIZ2,BMO1,BMO2,
     2NREAD,XTR3,XC23
C
      CCMMCN/SET3/YO,CYO,DDYO,ACCYO,Y1R,DY1R,CCY1R,ADDY1R,Y1L,DY1L,DDY1L
     1,ADDY1L,Y2R,DY2R,DDY2R,ADDY2R,Y2L,DY2L,CCY2L,ACCY2L
C
      CCMMCN / SET4    / FHIX,CFHIX,CCPHIX,ACCIX,PHIZ,DPHIZ,DDPHIZ,ADDIZ,
     1THTZR,DTHTZR,DDTZR,ACCZR,THTZL,DTHTZL,DDTZL,ACCZL,FV1R,PV1
     2L,FV2R,PV2L,                          CLOTCL,ICUT,TLIM
      CCMMCN / SET5     / KSTEF,IC(5), I2,XLI,XRI
      COMMON/SET6/AXAWT1,AXAWT2,AXWT1,AXWT2,BCDTCT,BCDWT1,BCDWT2,CS1L,
     *CS2L,CS1R,CS2R,CT1L,CT2L,CT1R,CT2R,SK1L,SK2L,SK1R,SK2R,TK1L,TK2L
      COMMON/SEP6/TK1R,TK2R,W1,W2,X2,DSTL,AMFR,AMPL,NBR,NEL
      COMMCN / NAME     / NNAME(2C),VMPH,XPRT,YPRT,BRLGTH,DSTR,XFAKE(15),
     *                   NENC
      CCMMCN / CVRLAY / NCLASS,NPRIMF
      CCMMCN/CLAS55/ACC1L,ADC1R,ACC2L,ADD2R,BODT1,BCDT2,BCD11,BOD12,
     *BCC13,ECC2E,EOC245,CE,CC,CCE,CCC,DCT1L,CCZ1R,CDZ2L,CDZ2R,CD1L,
     *CC1R,CC2L,CC2R
      CCMMCN/CLAS65/CIST,CIST77,CTH1L,CTH1R,CTH2L,DTH2R,DYF1L,DYF1R,
     *CYF2L,CYF2R,D1L,C1R,D2L,D2R
      CCMMCN/CLAS75/SK,TCF1L,TDF1R,TCF2L,TDF2R,THE1L,THE1R,THE2L,
     *THE2F,XC3,XTR4,XTR5,X23,X34,X45
      CCMMCN/AZ5/CTR(5),CTL(5),CSR(5),CSL(5),TKR(5),TKL(5),SKL(5),SKR(5)
      CIMENSICN EM(5),EL(5)
      IF(NFCFCI.EG.C.ANC.NRCFCF.EG.0) GC TC 1122
      WRITE(6,5) DSFACE   ,NRCRDI,NRCRDF
    5 FCRMAT(10X,'PRCFILCMETER CATA SPACING=',F5.3,//,'     STARTING
     1RECORD NO.=',I5,//,'      ENC      RECCRD NC.=', I5,//)
      GO TO 1133
 1122 WRITE(6,15) CSFACE
   15 FORMAT(10X,' DATA SFACING= ',F5.3,//)
 1133 CONTINUE
      WRITE(6,110)  ITITLE ,ICLASS
  11C FORMAT(/,48X,'MATHEMATICAL  MCDEL  OF  MOVING  VEHICLE',///,16A4,
     1 //, ' A.  VEHICLE  CLASSIFICATION - ', 8A4,//,
     2     ' E.  VEHICLE  CHARACTERISTICS ',      ///,
     3     10X,'I.  MAIN BODY',35X,'(LB-SEC 2/IN)',11X,
     4     'TREAC WICTH (IN)',11X,'AXLE SPACING (IN)')
```

```
      WRITE(6,120)  BMO,W1,X12,BIZ,W2,BIX
  120 FORMAT(15X,'1. BODY MASS (BMO )',24X,'=',G10.3,13X,'4.  W1     = ',
     1      F6.1,9X,' 6.  X12 = ',F6.1,/,
     2      15X,'2. BODY MASS MOM. OF INERTIA-PITCH-(BIZ )  =',G10.3,12X,
     3      ' 5.  W2     = ',F6.1,/,
     4      15X,'3. BODY MASS MOM. OF INERTIA-ROLL -(BIX )  =',G10.3,/)
      GO TO 2010
 2010 CONTINUE
      WRITE(6,170)
  170 FORMAT(//,10X,'II.  AXLES  INPUT',/,
     1      '   AXLE    ',10X,'STATIC WT.',10X,'   MASS   ',10X,
     2      'SUSP. STIF.',9X,'SUSP. DAMP.',9X,'TIRE STIF.',10X,
     3      'TIRE  DAMP.',/,'      NO.   ',10X,'    (LB.)  ',9X,
     4      '(LB-SEC 2/IN)',8X,'  (LB/IN) ',10X,'(LB/IN/SEC)',10X,
     5      '   (LB/IN) ',9X,'(LB/IN/SEC)')
      BM(1)=BM1R
      BM(2)=BM2R
      BL(1)=BM1L
      BL(2)=BM2L
      DO 2020  I = 1 , 2
      WRITE(6,180)  I,AXWTR(I),BM(I),SKR(I),CSR(I),TKR(I),CTR(I),I,
     1AXWTL(I),BL(I),SKL(I),CSL(I),TKL(I),CTL(I)
  180 FORMAT(I4,' RIGHT',6(10X,G10.3),/,I4,' LEFT  ',6( 10X,G10.3))
 2020 CONTINUE
      WRITE(6,190)   CLCTOL,IEXCIT ,H,IFREQ,TLIM,IOUT
  190 FORMAT(//,' C.  CONTROL  INPUT',//,
     1      10X,'I.  INTEGRATION  PARAMETERS   ',25X,'II.  OPTIONS',/,
     2      15X,'1.  TOLERANCE        = ',G10.3,23X,'1.  EXCITATION TYPE '
     3      ,I5,/,15X,'2.  TIME INCREMENT  = ',G10.3,23X,
     4      '2.  OUTPUT  INTERVAL',I5,/,15X,'3.  TIME LIMIT      = ',
     5      G10.3,23X,'3.  PLOT  OPTION    ',I5)
      WRITE(6,200)  VMPH,DSTR,SPACER
  200 FORMAT(///,' D.  VARIABLE  SPEED AND  ROADWAY  INPUT',///,
     1      10X,'I.  SPEED ',21X,'II.  EXCITATION PARAMETERS',/,
     2      15X,'1.  ',G10.3,'  MPH   ',9X,
     3      '1.  DIST.TO FIRST EXC.RIGHT  = ',G10.3,5X,
     4      '5.  RIGHT BUMP SPACING=',F10.3)
      WRITE(6,210)  VFPS,DSTL,SPACEL,V,WLR,NBR,WLL,NBL
  210 FORMAT(15X,'2.  ',G10.3,'  FT/SEC',9X,
     1      '2.  DIST.TO FIRST EXC. LEFT  = ',G10.3,5X,
     2      '6.  LEFT BUMP SPACING=',F10.3,/,15X,
     3      '3.  ',G10.3,'  IN/SEC',9X,
     4      '3.  RIGHT BUMPS WAVELENGTH   = ',G10.3,5X,
     5      '7.  NUMBER OF RIGHT BUMPS=',I10,/,46X,
     6      '4.  LEFT BUMPS WAVELENGTH    = ',G10.3,5X,
     7      '8.  NUMBER OF LEFT BUMPS=',I10)
      WRITE(6,1616) PK1R,PK1L,PK2R,PK2L,CP1R,CP1L,CP2R,CP2L,
     1  PVWT1R,PVWT1L,PVWT2R,PVWT2L
 1616 FORMAT (///,10X,'PK1R=',F8.2,2X,'PK1L=',F8.2,2X,'PK2R=',F8.2,2X,
     1  'PK2L=',F8.2,//, 10X,'CP1R=',F5.2,2X,'CP1L=',F5.2,2X,'CP2R=',
     2  F5.2,2X, 'CP2L=',F5.2,//,10X,'PVWT1R=',F8.2,2X,'PVWT1L=',F8.2,
     3 2X, 'PVWT2R=',F8.2,2X,'PVWT2L=',F8.2,///)
      IF (NLCROS.EQ.0) GO TO 2021
      WRITE(6,3)A1,A2,B1,B2,C1,C2,D1,D2,T1,T2,EE
    3 FORMAT(1X,'A1=',F10.2,5X,'A2=',F10.2,5X, 'B1=',F10.2,5X, 'B2=',
     1  F10.2,5X,'C1=',F10.2,5X,'C2=',F10.2,5X,//,1X,'D1=',F10.2,5X,
     2  'D2=',F10.2,5X,'T1=',F10.2,5X,'T2=',F10.2,5X,'EE=',F10.2,5X,//)
      WRITE(6,4)SM1,SM2,EH1,EH2,CH1,CH2,TH1,TH2,DH1,DH2
 4    FORMAT(1X,'SM1=',F7.3,5X,'SM2=',F7.3,5X,'BH1=',F7.3,5X,'BH2=',
     1 F7.3,5X,/,1X,'CH1=',F7.3,5X,'CH2=',F7.3,5X,'TH1=',F7.3,5X,'TH2=',
     2  F7.3,5X,'DH1=',F7.3,5X,'DH2=',F7.3,5X,///)
2021 IF ( XPRT . NE . 1.0 )  GO TO 4010
      WRITE(6,230)
 230 FORMAT(///,40X,'RESPONSE OF THE MOVING VEHICLE')
      GO TO 3010
3010 CONTINUE
      WRITE(6,500)
 500 FORMAT(27X,'EXCITATIONS',30X,'TOTAL DYNAMIC FORCES', 8X,'MAYSMETER
     * READING          NO.OF')
      WRITE(6,510)
510  FORMAT(5X,'TIME     DIST.      1-RT.    1-LT.    2-RT.    2-LT.    1-RT.
     1    1-LT.     2-RT.      2-LT.                FOR ROADWAY          ITERATIO
     2N')
      WRITE(6,520)
520  FORMAT(5X,'SEC.       IN.         IN.       IN.      IN.       IN.      LB.
     1     LB.         LB.         LB.                      IN')
      WRITE(6,530)
530  FORMAT(4X,'------    -----       -----     -----    -----     -----     ----
     1-    -----      -----       ----                     ------')
      IF(YPRT.NE.1.0)GO TO 2021
      GO TO 4010
4010 CONTINUE
      RETURN
      END
```