Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
TX-93+1944-1F		

4. Title and Subtitle	5. Report Date
SENSITIVITY ANALYSIS AND EVALUATION OF THE LOADRATE LOAD ZONE PROCEDURE FOR IMPLEMENTATION BY TEXAS DEPARTMENT OF TRANSPORTATION	January 1993
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Michael T. McNerney and Dennis S. Collier	Research Report 1944-1F

9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Center for Transportation Research The University of Texas at Austin Austin, Texas 78712-1075	11. Contract or Grant No. Research Study 3-18-92-1944
	13. Type of Report and Period Covered Final
12. Sponsoring Agency Name and Address Texas Department of Transportation Transportation Planning Division P. O. Box 5051 Austin, Texas 78763-5051	14. Sponsoring Agency Code

15. Supplementary Notes

Study conducted in cooperation with the Texas Department of Transportation
Research Study Title: "Improved Performance and Operation of the LOADRATE Load Zone Evaluation Procedure"

16. Abstract

A previous research study developed the LOADRATE program to replace current procedures that required three different computer programs for analyzing load-zoned roads in Texas. The Texas Department of Transportation was not satisfied with the results of the LOADRATE analyses of load-zoned roads when compared with analyses using current procedures. Differences between the two methods resulted in the need for an independent analysis of the LOADRATE program and procedures. The purpose of this study was to evaluate the code and analysis procedures of LOADRATE and make recommendations as to whether the Department should abandon the current analysis procedures and implement the LOADRATE program. This report summarizes the findings of a study of the LOADRATE program code and analysis procedures.

The study conducted parallel evaluations of several load-zoned roads using both the LOADRATE and current TxDOT procedures. The results verified previous TxDOT findings that indicated that LOADRATE was very liberal in lifting load restrictions on roads that current analyses showed should remain restricted in gross vehicle weight. This study reviewed the coding of the LOADRATE and LDATA programs and made many corrections and improvements to the existing code. The study also conducted sensitivity analyses of the variables used as input to the programs and compared the results of back-calculation of base and subgrade modulus with the *Modulus* computer program. The study specifically reviewed the method of predicting rutting using traffic and FWD deflections in the LOADRATE program.

The results of the study found significant differences in the results of analyses with LOADRATE versus current procedures that could not be corrected in this limited study. The rutting model unique to this program appears to be the primary cause for the vast difference in the analyses of load-zoned roads. There are significant differences in the calculation of base and subgrade modulus, but this error should have resulted in LOADRATE being more conservative in its results. This study recommended that LOADRATE procedures and computer program not be implemented by TxDOT for statewide analysis of load-zoned roads.

17. Key Words	18. Distribution Statement
load-zoned roads, analyses, procedures, load restrictions, rutting, subgrade modulus, computer program, code, variables, sensitivity	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	84	

SENSITIVITY ANALYSIS AND EVALUATION OF THE LOADRATE LOAD ZONE PROCEDURE FOR IMPLEMENTATION BY TEXAS DEPARTMENT OF TRANSPORTATION

by

Michael T. McNerney
Dennis S. Collier

Research Report 1944-1F

Research Project 3-18-92-1944

Improved Performance and Operation of the LOADRATE Load Zone Evaluation Procedure

conducted for the

Texas Department of Transportation

by the

CENTER FOR TRANSPORTATION RESEARCH
Bureau of Engineering Research
THE UNIVERSITY OF TEXAS AT AUSTIN

January 1993

IMPLEMENTATION STATEMENT

The purpose of this study is to ascertain whether the LOADRATE program is a reasonable method for determining the load-zoning of Farm-to-Market roads in Texas. The study provides a fair and rational basis for evaluating the usability of the LOADRATE program. Based on the results of this study, there are serious problems. The authors recommend that LOADRATE not be implemented.

Prepared in cooperation with the Texas Department of Transportation.

DISCLAIMERS

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

There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine, manufacture, design or composition of matter, or any new and useful improvement thereof, or any variety of plant which is or may be patentable under the patent laws of the United States of America or any foreign country.

NOT INTENDED FOR CONSTRUCTION,
PERMIT, OR BIDDING PURPOSES

Michael T. McNerney, P. E. (Texas No. 70176)
Dennis S. Collier

Study Supervisors

PREFACE

The authors wish to thank the Texas Department of Transportation Pavement Management Section, who sponsored this research and provided data and assistance in the analysis of and procedures for load zoning techniques. The authors are also grateful for the assistance of Paul DiGiovanni, specifically for his work on the computer coding and for running the many computer simulations required for this analysis.

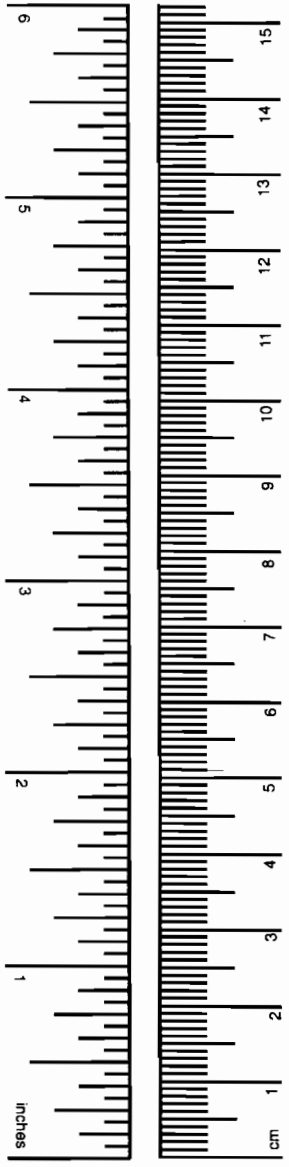
METRIC (SI*) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
in	inches	2.54	centimeters	cm
ft	feet	0.3048	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
AREA				
in^2	square inches	645.2	millimeters squared	mm^2
ft^2	square feet	0.0929	meters squared	m^2
yd^2	square yards	0.836	meters squared	m^2
mi^2	square miles	2.59	kilometers squared	km^2
ac	acres	0.395	hectares	ha
MASS (weight)				
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2,000 lb)	0.907	megagrams	Mg
VOLUME				
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.0328	meters cubed	m^3
yd^3	cubic yards	0.0765	meters cubed	m^3

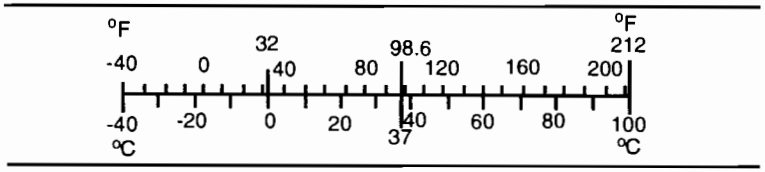
NOTE: Volumes greater than 1,000 L shall be shown in m^3.

TEMPERATURE (exact)				
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C



APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
AREA				
mm^2	millimeters squared	0.0016	square inches	in^2
m^2	meters squared	10.764	square feet	ft^2
m^2	meters squared	1.20	square yards	yd^2
km^2	kilometers squared	0.39	square miles	mi^2
ha	hectares (10,000 m^2)	2.53	acres	ac
MASS (weight)				
g	grams	0.0353	ounces	oz
kg	kilograms	2.205	pounds	lb
Mg	megagrams (1,000 kg)	1.103	short tons	T
VOLUME				
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	meters cubed	35.315	cubic feet	ft^3
m^3	meters cubed	1.308	cubic yards	yd^3
TEMPERATURE (exact)				
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



These factors conform to the requirement of FHWA Order 5190.1A.

* SI is the symbol for the International System of Measurements

TABLE OF CONTENTS

SUMMARY

The study conducted parallel analyses of several load-zoned roads using the LOADRATE program and current TxDOT procedures. The findings verified a recent paper by members of the TxDOT pavement management section that the majority of the roads released from load-zoning by LOADRATE would continue to be restricted if analyzed by current procedures. The findings of the study were that the LOADRATE program gave significantly lower modulus values with higher variability than the *Modulus* program currently used by the Department. Even with the lower modulus, the LOADRATE program permitted significantly greater vehicle loadings than would the current procedures. No major errors in coding were discovered, but the coding was poorly written and several improvements were made in the revised code. The difference in results between the LOADRATE program and the current procedures is primarily contained in the analysis portion of the program that uses a rutting prediction model. Validation of the rutting prediction model was not found to be adequate to justify using this new procedure. This study recommends that the LOADRATE program not be implemented by Texas Department of Transportation for evaluation of load-zoned roads. Implementation of the LOADRATE program without more detailed analysis and field testing of the rutting model could lead to premature failure of 75-90 percent of the 17,000 lane-miles of load-zoned roads.

CHAPTER 1. INTRODUCTION

The Texas Department of Transportation (TxDOT) is responsible for one of the nation's largest road systems with over 78,000 center-line miles of state-maintained highway network. Of this network, approximately 17,250 miles (22 percent) are light-duty pavement that are load-zoned for less than the normal legal limit truck traffic of 80,000 pounds gross vehicle weight and 20,000 pounds per single axle. Load-zoning is accomplished by limiting the maximum gross vehicle weight to 58,420 pounds, which was the weight limitation when many of these older light-duty pavements were first constructed.

The load-zoned roads are typically Farm-to-Market (FM) roads which have very thin-wearing surfaces of asphalt over a compacted base material of various thickness. Some roads may have stabilized based materials.

Frequently, the Pavement Management section of D-18 (now D-8) is requested to review the load-zoning of certain sections of the system to determine if the lifting of the load restrictions is feasible and appropriate. TxDOT sponsored research Project 473 with the Texas Transportation Institute (TTI) to develop a computer program and procedure for an improved method of analyzing load-zoned roadways using nondestructive testing (Ref 1). The results of that research project were the LDATA and LOADRATE computer programs.

When D-18 attempted to use the computer programs, thereby replacing the current procedures, the results were not as expected. In a test of 40 load-zoned road sections in Ellis County, the Department found that the computer programs predicted no rutting problems on 39 out of 40 road sections and recommended lifting load restrictions (Ref 2). In sharp contrast, the current method, though more time-consuming and cumbersome, predicted that the load-zoning restrictions could be lifted on only four of the 40 sections.

Clearly, since the results of these two procedures are at such extreme odds with one another, the Department could not blindly follow the LOADRATE computer program and evaluation procedure without additional information. The risk of premature failure on a majority of the load-zoned roads was too great a risk to take without further exploration of this computer program and the procedures used to evaluate it. The Department sponsored a small research effort with the Center for Transportation Research to evaluate and correct any errors in the LDATA and LOADRATE computer programs so that the Department might use them for evaluating roads for load-zoning.

PROJECT SCOPE

This study (Research Project 1944) was limited in time and budget and was not an effort to replace LOADRATE or write totally new procedures. The objectives of the project were to: (1) correct errors and omissions in the coding and execution of the LDATA and LOADRATE microcomputer programs, and (2) validate the operation of the programs to perform their intended functions as a reasonable tool for analyzing nondestructive test data for farm-to-market roads to permit evaluation of allowable loading.

RESEARCH APPROACH

To achieve the objectives of this study, the research effort was divided into three tasks. The first was to review the coding

and execution of the programs. LDATA and LOADRATE were searched for coding errors and extraneous and/or unnecessary code was eliminated. Each module was tested to see if it accomplished what was expected. In some instances, modules were broken out separately and run independently as separate programs to determine sensitivity to input variables and effects on the analyses.

The second task was to validate the operation of the programs as a tool for analyzing load-zoned roads. The LDATA program, which prepares the falling-weight deflectometer (FWD) data and traffic data for the LOADRATE program, was evaluated. It was compared to current mechanistic analysis method results and to the TTI *Modulus* program, which is more widely accepted for back-calculation of modulus of elasticity (Ref 3). It was believed that the LDATA program and LOADRATE programs would be sensitive to field conditions such as moisture, temperature, and seasonal variation of FWD data. The LOADRATE program was analyzed in terms of the way it calculates the allowable repetitions of legal load until rutting failure and the sensitivity of the model to the input variables.

If the LOADRATE program was found to be suitable for evaluating load-zoned roads, then the documentation of the program and procedures were to be improved as the third task. The Department found that the program was not very user-friendly nor well-documented. The documentation of the program was improved within the program code, but no additional manuals were developed. A report documenting the findings of the research results was also prepared.

CHAPTER 2. BACKGROUND

THE CURRENT LOAD-ZONED ROAD ANALYSIS

The original method for analysis of load-zoned roads was the Texas Triaxial Method. It is similar to the procedure by which many of the roads were designed. Over the years an empirical chart has been developed that relates pavement thickness to allowable design wheel load. The Texas Triaxial test is a laboratory test of saturated soil specimens that determines resistance to shearing forces (Ref 4). The soil is classified from 1 to 6 with 1 being very good base material. In most cases, the Texas Triaxial test values are known from laboratory tests taken during construction. The saturated soil condition is conservative and represents a worst-case scenario in the field (usually found only during the springtime rainy season or more frequently in East Texas near coastal areas). The method has been demonstrated to produce good results, and the data are available or can be estimated from soil samples or soil maps.

With the addition of falling-weight deflectometer equipment, a mechanistic analysis has replaced the Texas Triaxial test as the primary method of load-zone analysis by the Department. This method of testing uses the FWD data and the *Modulus* program to back-calculate the modulus of elasticity of the base and subgrade layers. The program uses the estimated thickness of the base layer and the deflections from the FWD data. A modulus of the base and subgrade are back-calculated using layered elastic theory, usually neglecting the thickness of the thin asphalt layer. The *Modulus* program was developed by TTI and is widely used for this back-calculation procedure. Both the LDATA program and the *Modulus* program back-calculate the modulus of the base and subgrade from the same FWD data.

In the current evaluation procedure, the Department takes the moduli generated from the FWD data using the *Modulus* program and imports the data into a *Lotus 1-2-3*™ spreadsheet. The data are then analyzed with the computer program BISAR, which uses layered elastic theory to determine the stresses and strains in the base and subgrade layers from 18-KIP single-wheel loads. The stresses and strains are then used with Asphalt Institute procedures to determine a limiting wheel load to achieve a ten-year life. If the allowable wheel load is below 20,000 pounds per axle (10,000 pounds per half-axle), the road remains restricted in gross weight. The several different programs that must be used separately to achieve the final answer make the procedure somewhat cumbersome, but the method has proved reliable in the past. Using this current procedure as well as the Texas Triaxial method, the Department obtains a markedly different result than that from the LOADRATE program.

LOADRATE AND LDATA PROGRAMS

The LDATA program is written in FORTRAN and basically takes the FWD data and back-calculates a modulus of elasticity of the base and subgrade based upon inputs of soil type and assumed base thickness. The program also calculates the number of ESALs for a 20-year period based upon input parameters. The LOADRATE program is written in BASIC and runs on a PC-compatible computer. The LOADRATE program takes the output file from LDATA and uses a rutting prediction model to esti-

mate the number of ESAL applications necessary to reach a specified failure criterion. The normal TxDOT criterion is 0.5 inches of rutting.

CHAPTER 3. CODING COMMENTS

This chapter cites examples of coding problems, particularly with the LOADRATE program. The deficiencies of both programs are discussed from a coding perspective. This chapter is not intended to be a criticism of each program's ability to provide and/or predict load-zone restrictions.

LDATA

One minor problem in following the instructions of the LDATA program was noticed. In the step that calculates the 20-year life of traffic, three possible method options are coded. One optional method asks for the current traffic year and a growth rate. If that method is used, the program will calculate only 19 years of traffic if the instructions are followed exactly. The reason this problem is not significant is the way the Department usually analyzes each roadway. The traffic division provides traffic for the first year and 20th year ESAL rate. Therefore the Department does not use that particular option.

PROGRAMMING DEFICIENCIES IN LOADRATE

Variables

- Using variables that are never changed
- Assigning variables/arrays twice in two consecutive lines

Unimplemented code

Code optimization

- Moving code into sequential order instead of using ***GOTO/GOSUB*** routines
- Removing unnecessary and blank ***REM*** statements
- Making multiple comparisons within an ***IF...THEN*** statement explicit
- Combining related/short lines of code
- Specifying variable for ***NEXT*** statement
- Removing ***": GOTO"*** statements within ***IF...THEN*** statements
- Breaking long/complex lines of code
- Restructuring lines of code for simplicity/explicitness
- Removing parentheses
- Using decimal equivalents instead of executing a division
- Removing ***GOTO*** statement referring to next line
- Removing lines of code that have been ***REM***-ed out

Documentation

- No listing of variable descriptions
- Dividing the program into distinct sections

Besides the problems presented by the rutting module, the LOADRATE program is flawed in many other ways. For example, variables and arrays are incorrectly assigned or used, some sections of the code are never executed, very little documentation is provided within the program itself, and the program is not very efficient. Even though some of the comments made below refer to lines of code that are syntactically correct and probably have no effect on the results generated by the program, the code itself may not be the original intent of the programmer, nor may it be understood in the same way by other programmers.

Unless otherwise indicated, all comments made below apply to the original program listed in Appendix A. Excerpts from the code are denoted by bold and italics lettering (i.e., ***Code***).

VARIABLES AND ARRAYS

Variables and arrays throughout the LOADRATE program are often misused, mis-assigned, or simply never changed. Here are a few examples:

1) Using variables/ arrays that are never assigned.

The variable ***TBC*** is never assigned within the program nor by the user; however, it is called several times.

Line 131 ***IF TBC = 1 THEN....***
Line 177 ***IF TBC = 0 THEN....***

Other examples are lines 178, 205, and 341.

2) Using variables/arrays that are never changed.

The variable ***DESIGN*** is referred to only twice within the program.

Line 186 ***IF DESIGN = 1 THEN : GOTO 4340***

Line 342 ***DESIGN = 0***

The variable NH is assigned only once but is called several times.

Line 343 ***NH = 0***

Line 131 ***IF TBC = 1 THEN : BA(I) = NH***

Line 137 ***IF NH = 0 THEN : PPR = 0***

Lines 308-315 ***IF (EQ = 1) AND (NH <> 0) THEN : PRINT #1, NS(I)....***

Similarly, other variables have been identified, such as ***RUT1*** (assigned in line 940), ***SX*** (line 631), and ***BEND*** (line 1110).

3) Assigning a single variable different values in two consecutive lines.

The array ***DE(#,#)***

Line 200 ***IF EQ = 0 THEN : DE(I, 7) = 3.38075 * D(I, 5) ^ 639462***

Line 201 ***IF EQ = 0 THEN : DE(I, 7) = 4.5688721# * D(I, 5) ^ .578444***

CODE

The code, besides being written in BASIC, is not very efficient. Here are a few reasons why.

1) Since the program is relatively small, there is no reason to place the introductory material near the end of the code as a subroutine. This is the case with lines 461 - 486, the introduction to LOADRATE .
2) Even though LOADRATE is relatively small, unnecessary and blank ***REM*** statements simply require more memory and more CPU time. If a line can be left out, then it should be left out. Some ***REM*** statements that describe variables or arrays should be included in the introductory material describing the program to other programmers.
3) Multiple comparison within an ***IF...THEN*** statement should be made explicit (for the programmer and the computer).

Line 444 ***IF (CHO4 <> 1 AND CHO4 <> 0) THEN....***

should read as

IF (CHO4 <> 1) AND (CHO4 <> 0) THEN....

4) Short or related lines should be combined. This not only reduces the size of the program, but can reduce confusion and the possibility of errors within the program.

Line 551 *X = X(II): Y = D(I, II)*
Line 552 *X = LOG(X): Y = LOG(Y)*

should read as

X = LOG (X(II)) : Y = LOG (D(I,II))

In this example, the original code is syntactically correct. However, as shown with the variable *DE(#,#)* above, this may become a problem and may not be what the original programmer wanted.

5) With multiple *FOR...NEXT* loops, especially with nested loops, it becomes necessary to specify the variable intended by the *NEXT* statement. This also makes reading the program easier.

Line 74 *NEXT I*
Line 75 *NEXT*

Line 75 should specify the variable *NTT* (just as line 74 specified the variable *I*).

6) In BASIC, it is not necessary in *IF...THEN* statements to specify *GOTO*. This reduces the code by one less command.

Line 113 *IF ITE > 1 THEN : GOTO 3010*

should read as

IF ITE > 1 THEN 3010

7) *GOTO* statements should be used sparingly. Using *GOTO* statements to refer to the next line is a sign of bad programming and ineffective quality control by the programmer.

Line 56 *GOTO 2290*
Line 57 *'*
Line 58 *2290 'To read from file*

8) LOADRATE has few, if any, error checking routines that should be used whenever user input is accepted. For example, when inputting the unit weight of a material, the programmer can expect that the unit weight of a material material will be higher than water (UW = 64.2 pcf) and less than three times water (concrete weighs approximately 135 pcf). This helps prevent errors in user inputs.

Line 814 *INPUT "UNIT WEIGHT OF MATERIAL (pcf) : ", UW*

The next line should be

IF (UW < 100) OR (UW > 170) THEN {input *UW* again}

9) Some sections of code are not even implemented.

Since, in line 591, there is a *GOTO* statement and since line 592 does not have a line number (meaning that there is no other way of getting to it), line 592 is never reached.

Line 591 *GOTO 7800*
Line 592 *END*

10) Nested *IF...THEN* statements should not be used.

Line 35 *IF SLECT1 = 3 THEN : IF RUT1 <> 1 THEN : INIOPT ....*

should read as

IF (SLECT1 = 3) AND (RUT1 <> 1) THEN INIOPT ....

Also, note the removal of the colon after the second *THEN* statement.

11) Some lines of the code simply do not make mathematical sense. For example, in BASIC *LOG* refers to the base 10 of the logarithmic function. So, *LOG (10)* equals 1.

Line 853 ***THETA = 10 ^ (LOG(EI / K1) / LOG(10) / K2)***

Other times the mathematics performs no function at all. For example, the equation below first divides the negative of ***RHO*** by ***ZZ1***. Then it raises the result to the first power (any number raised to the first power equals that number), and then it multiplies that result by a negative sign, which only negates the negative of ***RHO***. For example, consider the next line of code.

Line 509 ***ER1 = ALPHA * EXP(-((-RHO / ZZ1) ^ 1)) + 5.9***

should read as

ER2 = ALPHA * EXP(RHO / ZZ2) + 5.9

DOCUMENTATION

Most programs written in BASIC are intended to be read by other programmers. Without documentation within the program, reading the program becomes difficult. There are two simple methods that every programmer should use to document a program. The first method is to give a list of variables used in the program and the purpose behind each variable. The second method is to indicate where the program performs different functions by breaking the program up into modules.

CHAPTER 4. PROGRAM PERFORMANCE

COMPARISON OF LDATA AND MODULUS PROGRAMS

In order to determine why there was such a difference in performance between the LOADRATE procedure and the current procedures, several of the individual modules were tested separately. The first module tested was the LDATA program.

This program takes the FWD data file and calculates the modulus of elasticity of the base and subgrade. Input parameters include the estimated thickness of the base layer and the type of subgrade soil. A sensitivity analysis was made of the effects of assumed base thickness on calculated modulus. Normally the thickness of the base layer is known from construction records, but if it is not, then this input parameter has an effect on the base and subgrade modulus calculated by the program. Figure 1 shows the effect of the LDATA program on calculation of base and subgrade modulus based upon an assumed thickness of the base layer. In this particular example, the actual base layer thickness was 11.5 inches. As was expected, the assumed base thickness has a moderate and nearly linear effect on the calculated subgrade modulus but a greater effect on the calculated base modulus. In this example the minimum value of the base modulus is near the actual base layer thickness.

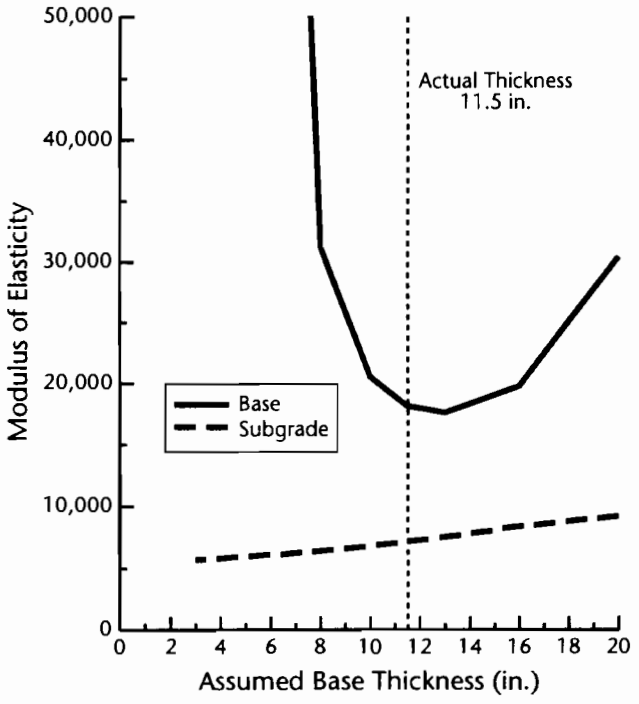


Figure 1. Effect of Assumed Base on LOADRATE Computed Modulus.

Figure 2 shows the effect of assumed base thickness on the number of ESALs to achieve 0.5-inch rutting for all three different soil types used in the program. As Figure 2 shows, the effect of soil type can have a larger impact on the number of ESALs until rutting failure in the LOADRATE model than assumed base thickness.

The LDATA program was also compared to the *Modulus* program, which is the primary FWD data-reduction program used by the Department. Several FM roads in Ellis County were selected as comparison cases. Generally LDATA back-calculates a lower modulus for both the subgrade and base materials using the same FWD and input thickness and soil type. As a typical example, Figures 3 and 4 show that in the weaker sections of the roadway of FM 983, the moduli calculated by LOADRATE and *Modulus* can differ by a factor of two.

Using FM 2377 as another example, Figure 5 for base material and Figure 6 for subgrade material show that the LOADRATE program generally calculates a lower modulus than the *Modulus* program and has a higher variation as well. What is most surprising in the results of the FM 2377 data is that the LDATA program shows a large section of pavement that has a higher back-calculated modulus in the

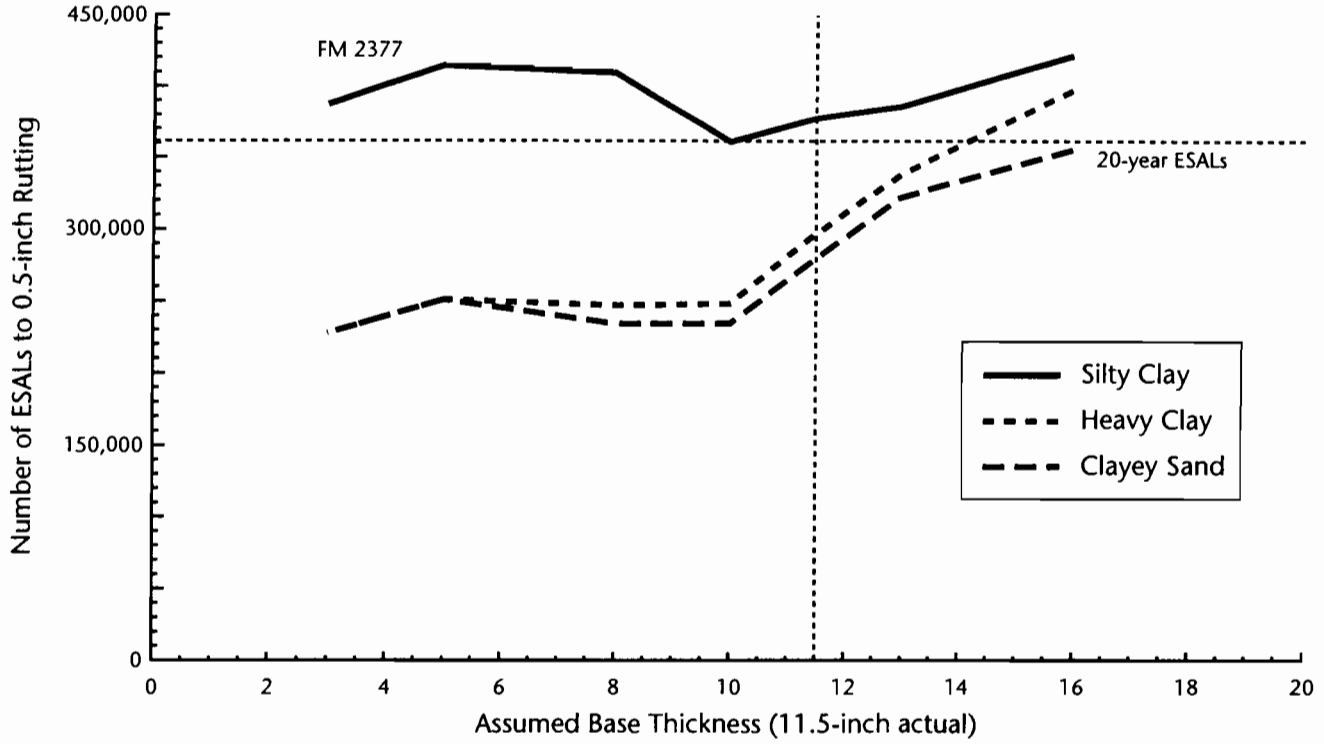


Figure 2. Effect of Assumed Base Thickness on LOADRATE Calculated ESAL to 0.5-inch Rutting.

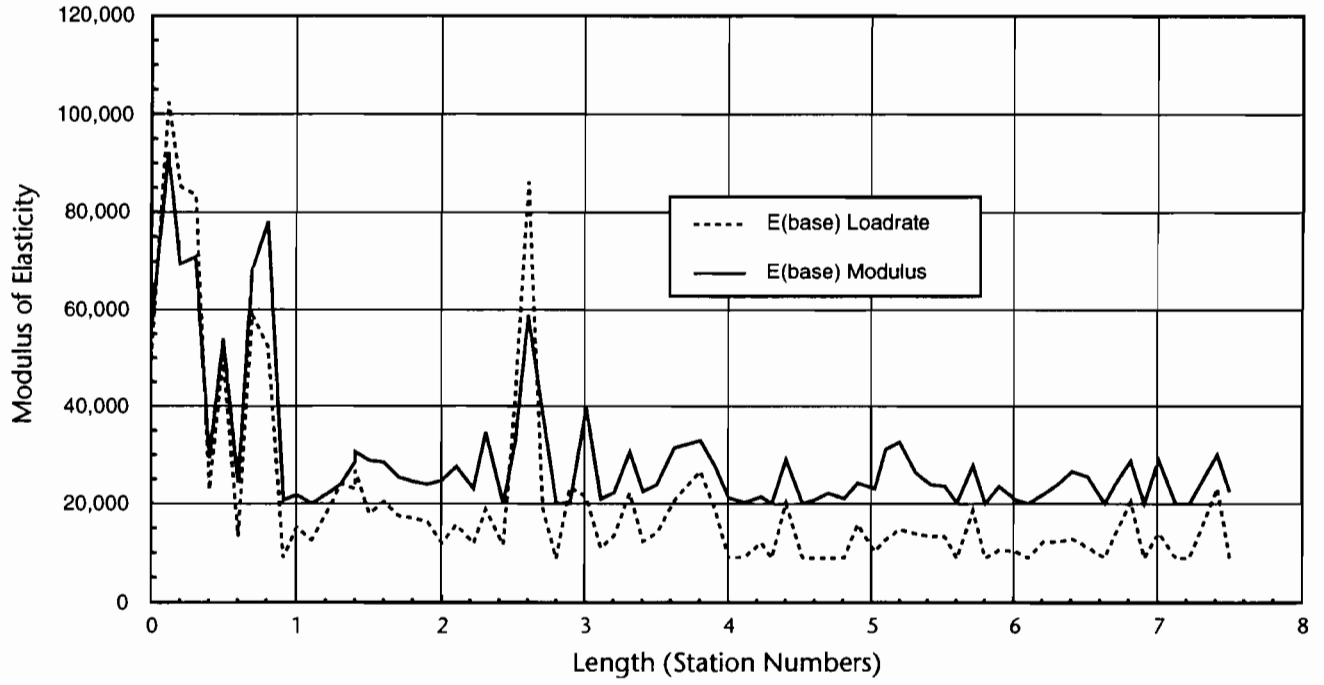


Figure 3. Comparison of Calculated Base Moduli for FM 983.

subgrade layer than in the base layer (the shaded area in Figure 7).

The ratio of the modulus of the base layer to the subgrade layer can be very important in predicting pavement behavior and performance. For good performance of base materials, it is expected that the ratio of base to subgrade stiffness would be 4

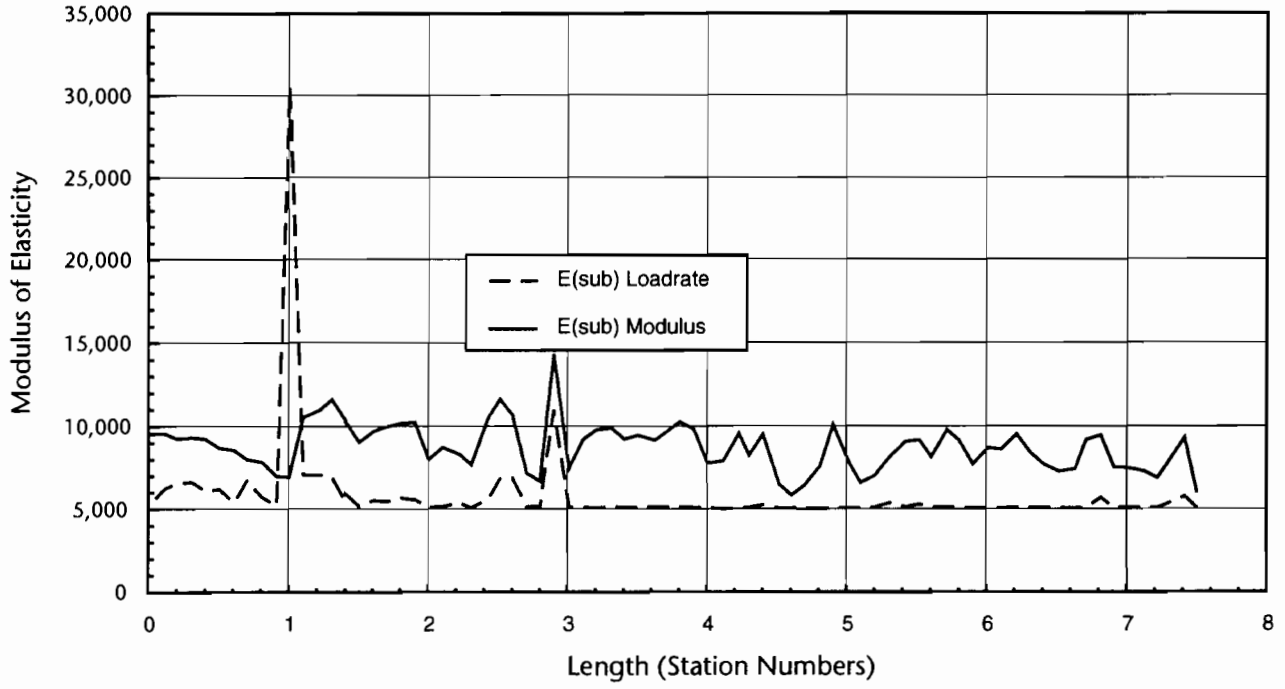


Figure 4. Comparison of Calculated Subgrade Moduli for FM 983.

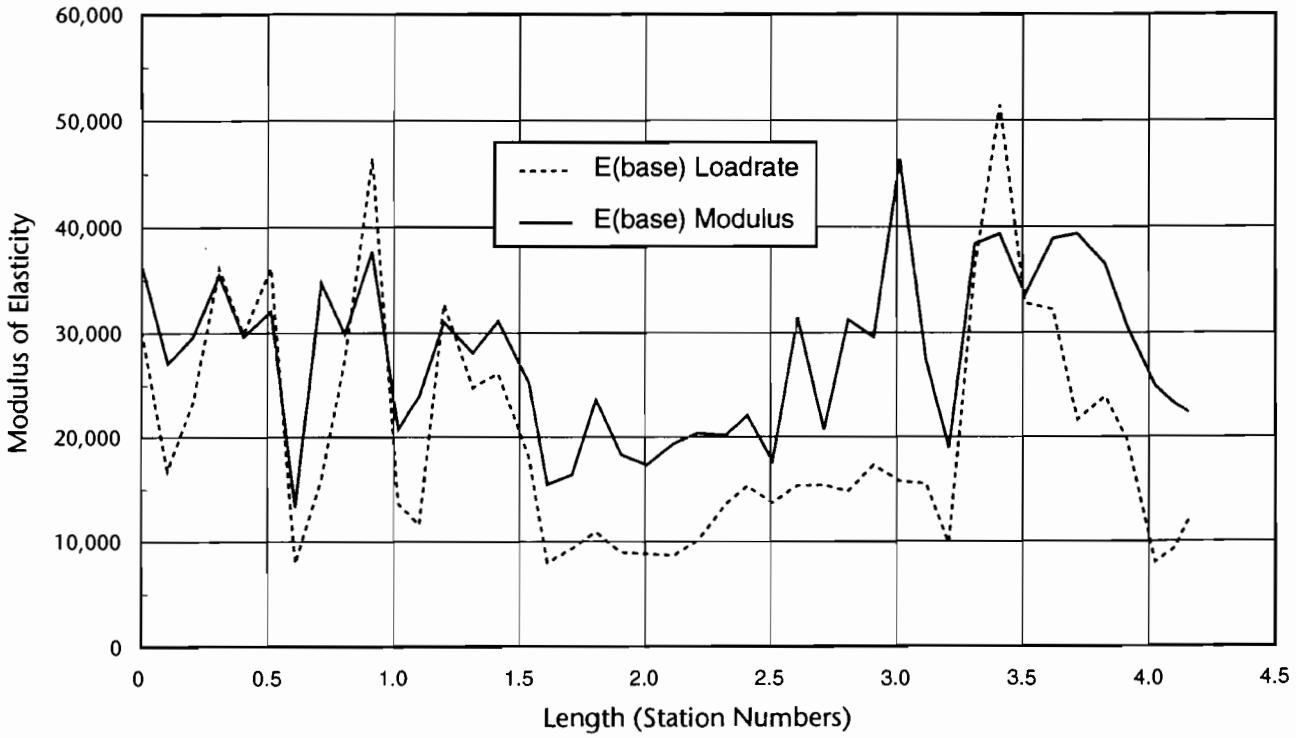


Figure 5. Comparison of Calculated Base Moduli for FM 2377.

to 1. The comparison for FM 2377 of the stiffness ratio of base modulus to subgrade modulus for the two back-calculation programs is shown in Figure 8. Notice the significant area where the stiffness ratio is less than one. The means of the stiffness ratios

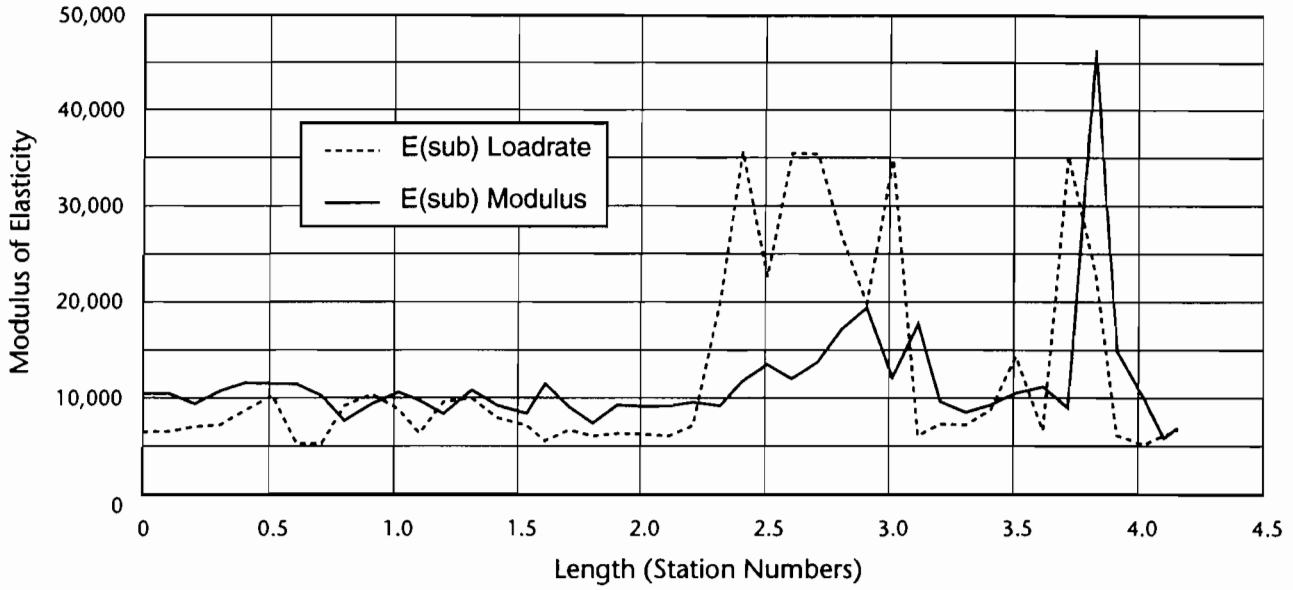


Figure 6. Comparison of Calculated Subgrade Moduli for FM 2377.

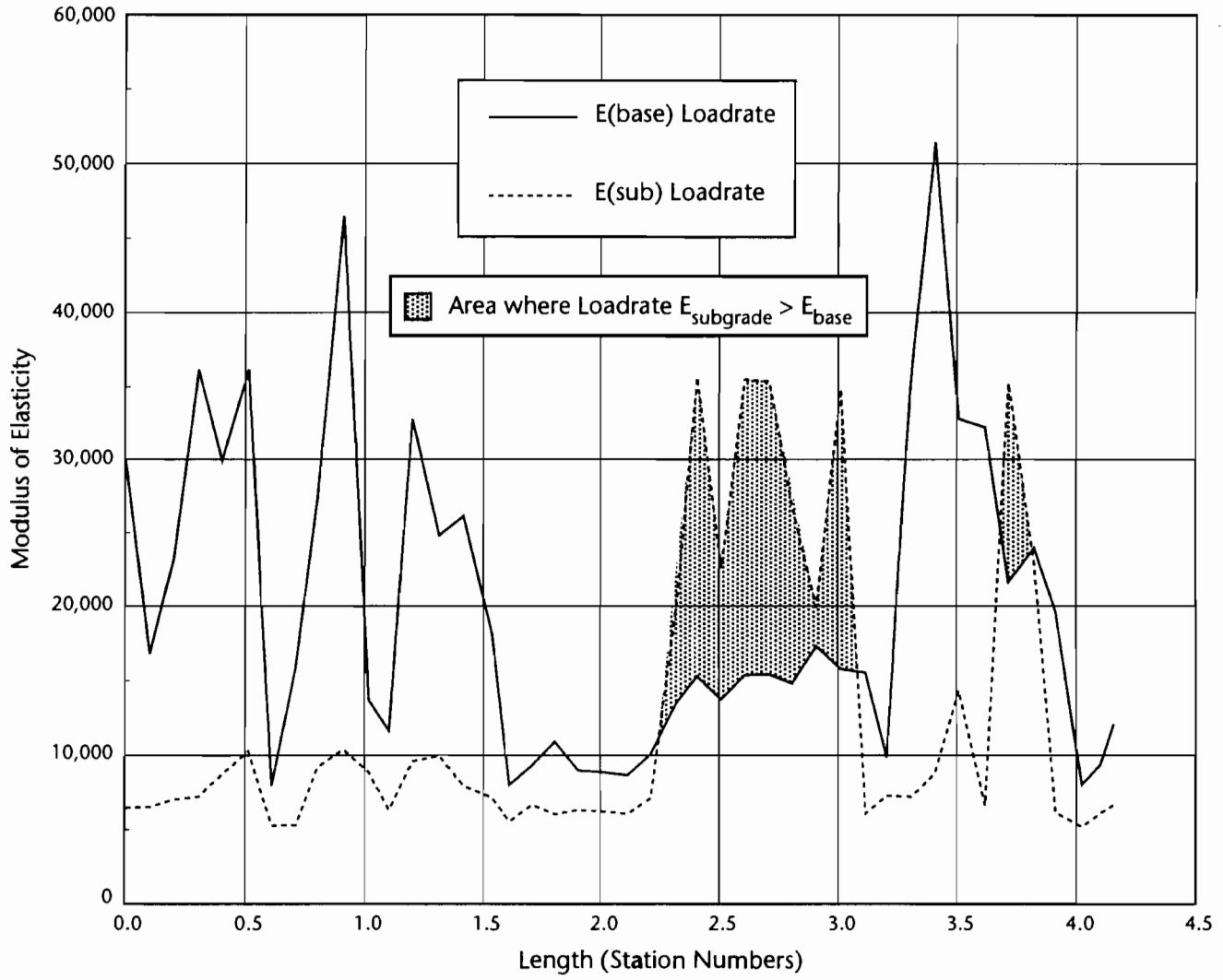


Figure 7. Moduli Calculated by LOADRATE for FM 2377.

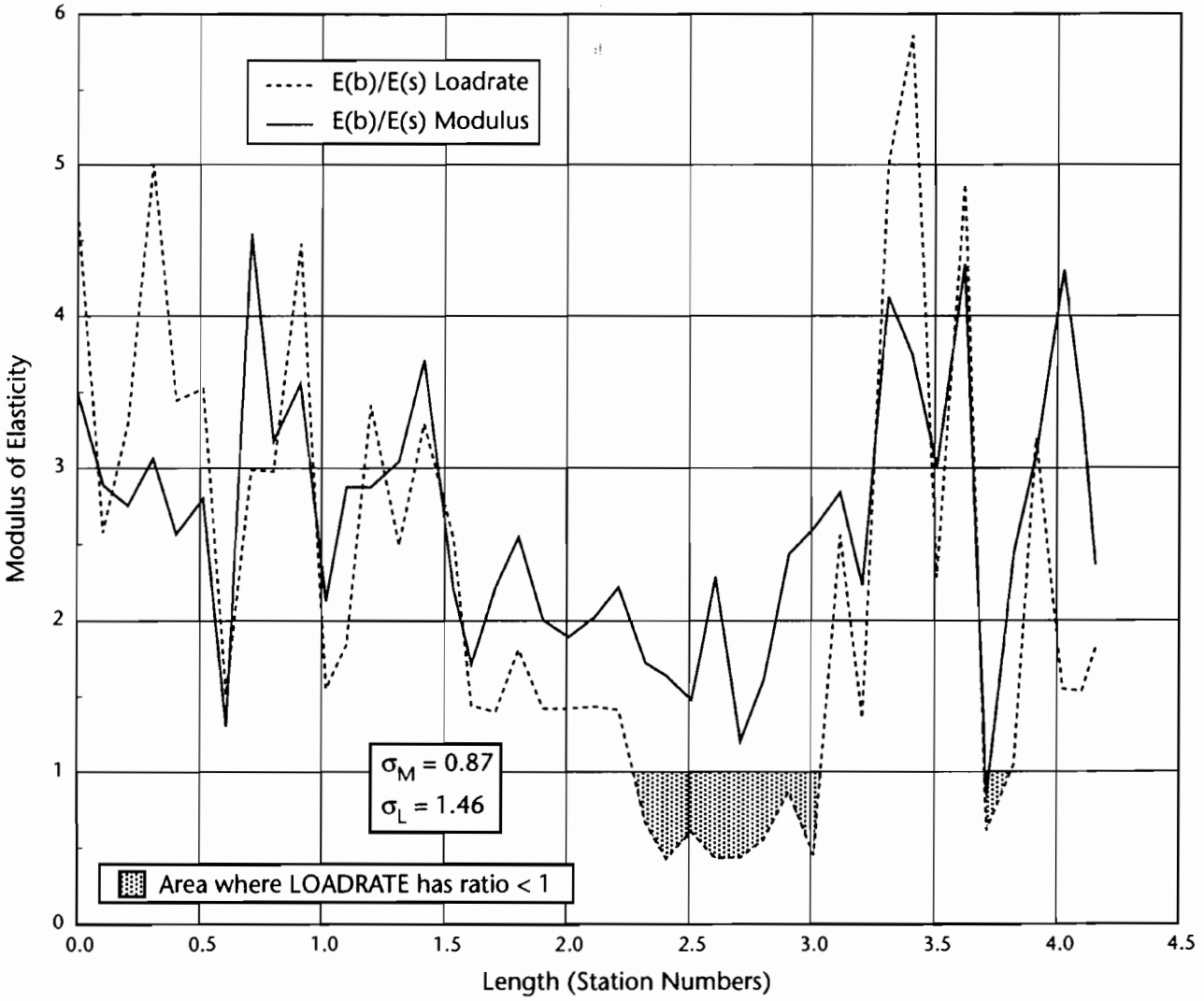


Figure 8. Comparison of Base/Subgrade Modulus for FM 2377

are very nearly the same: 2.63 for the *Modulus* program and 2.24 for the LOADRATE program. It is the difference in variation that is most interesting. The standard deviation of the ratio for the *Modulus* program was 0.87, but it was 1.46 for the LOADRATE program.

The data of the computed stiffness ratios of FM 2377 shown in Figure 8 were statistically analyzed, and a frequency distribution analysis of those data is shown in Figure 9. From the histograms of the frequency distributions, one can see that the frequency distribution of the computed ratio of relative stiffness very nicely approximates a normal distribution for the *Modulus* program. However, the frequency distribution of the LOADRATE-computed ratio of relative stiffness is

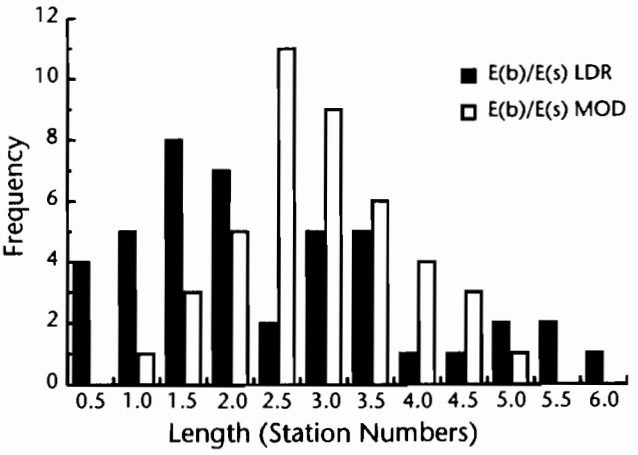


Figure 9. Frequency Distributions of Ratios of E (base) to E (subgrade) for FM 2377.

skewed right and does not approximate a normal distribution pattern.

The distribution pattern of the LOADRATE data could be looked at as bimodal, but what is important is the significant difference in relative stiffness that these two back-calculation programs yield from analysis of the same FWD data. From the frequency distributions it is evident that the base-to-subgrade modulus ratio is better-defined and has less variation in the *Modulus* back-calculation than in the LDATA program. From a brief literature review, it appears that the LDATA program was tested to a limited extent with laboratory evaluation during its development, but it was not compared to the *Modulus* program (Ref 5).

The *Modulus* program has received much wider acceptance than LDATA and has less variation in the back-calculated modulus and modulus ratio. Since the LDATA program generally calculates a lower modulus of elasticity for both the base and subgrade, it would be logical to assume that the LOADRATE program would predict more rutting and more load-zone restrictions than the Department's method of evaluation using the *Modulus* program and layered elastic theory. However, as was demonstrated earlier, the reverse is true; LOADRATE predicts less rutting and nearly no load-zone restrictions. The differences in modulus calculated by LOADRATE and *Modulus* are sufficiently different as to cause a difference in analysis.

MOISTURE CORRECTION

The LDATA program as written appears to have input screens that allow the user to correct the moisture and temperature parameters in calculating a final modulus. However, there is no code that actually applies any corrections to the FWD data or modulus output values based upon either temperature or moisture. This is unfortunate, as both the seasonal variation of moisture and the temperature of the asphalt surface can have a significant impact upon rutting.

Recent testing by Dr. Thomas White at Purdue University has shown that surface rutting increases significantly in thin asphalt overlays at elevated temperatures (unpublished data). Analysis reported by Marshal Thompson at the University of Illinois shows that the rutting variation during spring, when moisture contents are greatest, is one of the most statistically significant variables in rutting prediction (Ref 6). It is possible that nearly all subgrade rutting of a pavement can occur during the Spring when moisture contents are high, and strength and stiffness of the subgrade can be one-half to one-third the normal summertime values.

EVALUATION OF THE RUTTING MODEL

The study had neither the time nor the budget to do a detailed analysis of the rutting model used in LOADRATE. However, the module was removed from the program and run with various base and subgrade moduli and three different soil types. Figures 10, 11, and 12 give the results of the sensitivity analysis.

The sensitivity analysis shows that there is little effect on predicting rutting by the base modulus. One would disagree with this premise unless all the rutting is expected to take place in the subgrade and not in the base or surface treatment. The three figures also show that any subgrade modulus above 10,000 psi will not result in a 0.5-inch rut for the 20-year traffic on FM 2377. Above 10,000 psi subgrade modulus, there is barely any difference in the type of soil. Only with subgrades at 5,000 psi and below is there a difference in soil types and excessive rutting in the 20-year traffic.

A look at the coding of the rutting model and a check of Research Report 473-1 reveals that the data used in the LOADRATE rutting model are computer-generated (Ref 5). The empirical data from

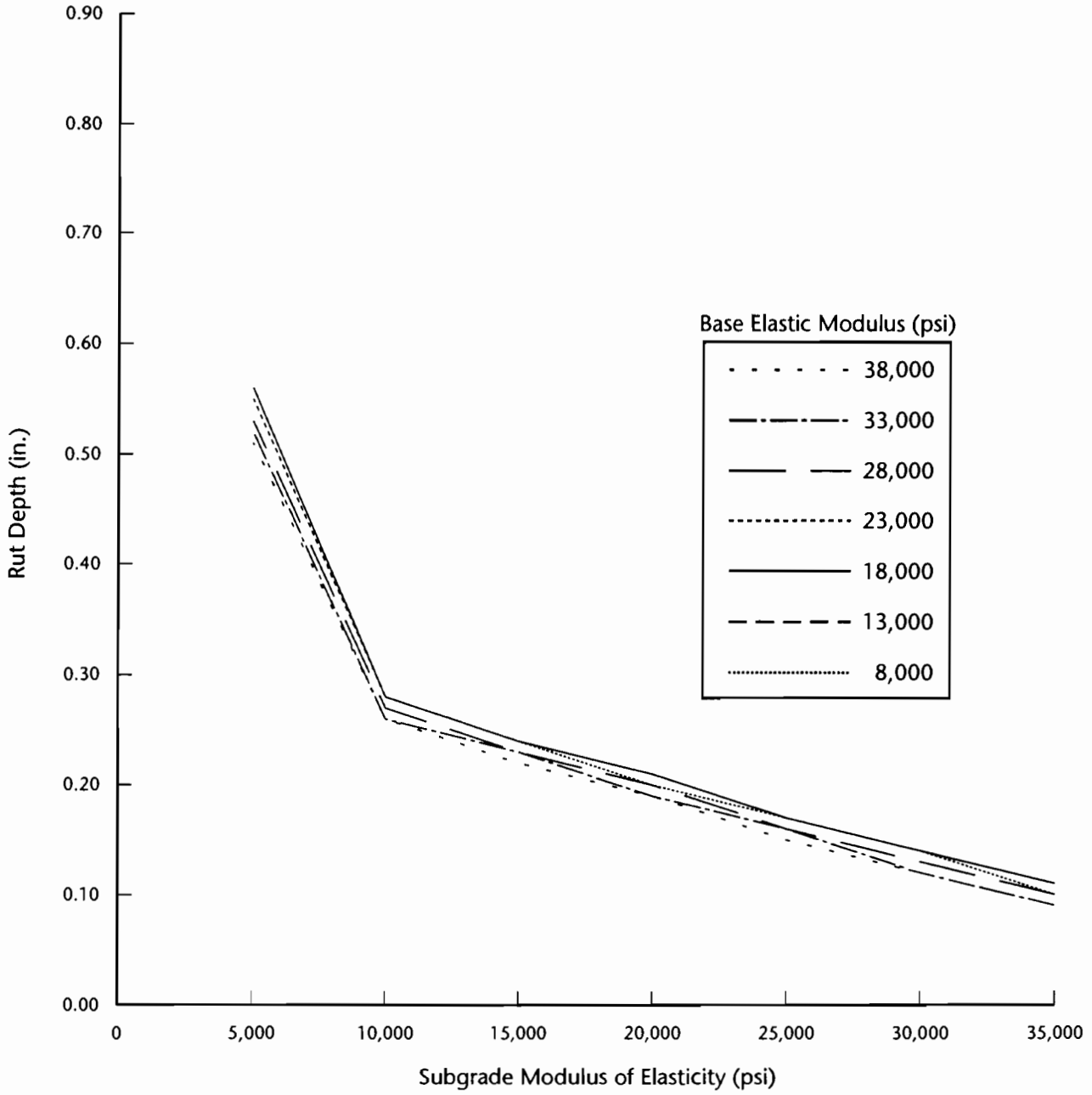


Figure 10. LOADRATE Sensitivity of Rut Depth to Modulus of Elasticity for Silty Clay for FM 2377.

which the computer-generated data are derived may not be applicable to much of the real world scenarios found on load-zoned roads. More information is needed on the rutting model to determine its applicability to predicting performance of currently load-zoned roads. Using regression analysis of field conditions to predict rutting should be limited to very similar conditions of traffic loading, stress conditions, soil type, base material, surface material, and thickness and environmental conditions. There are so many variables that affect rutting that regression analysis of the aggregated variables will not provide predictive results.

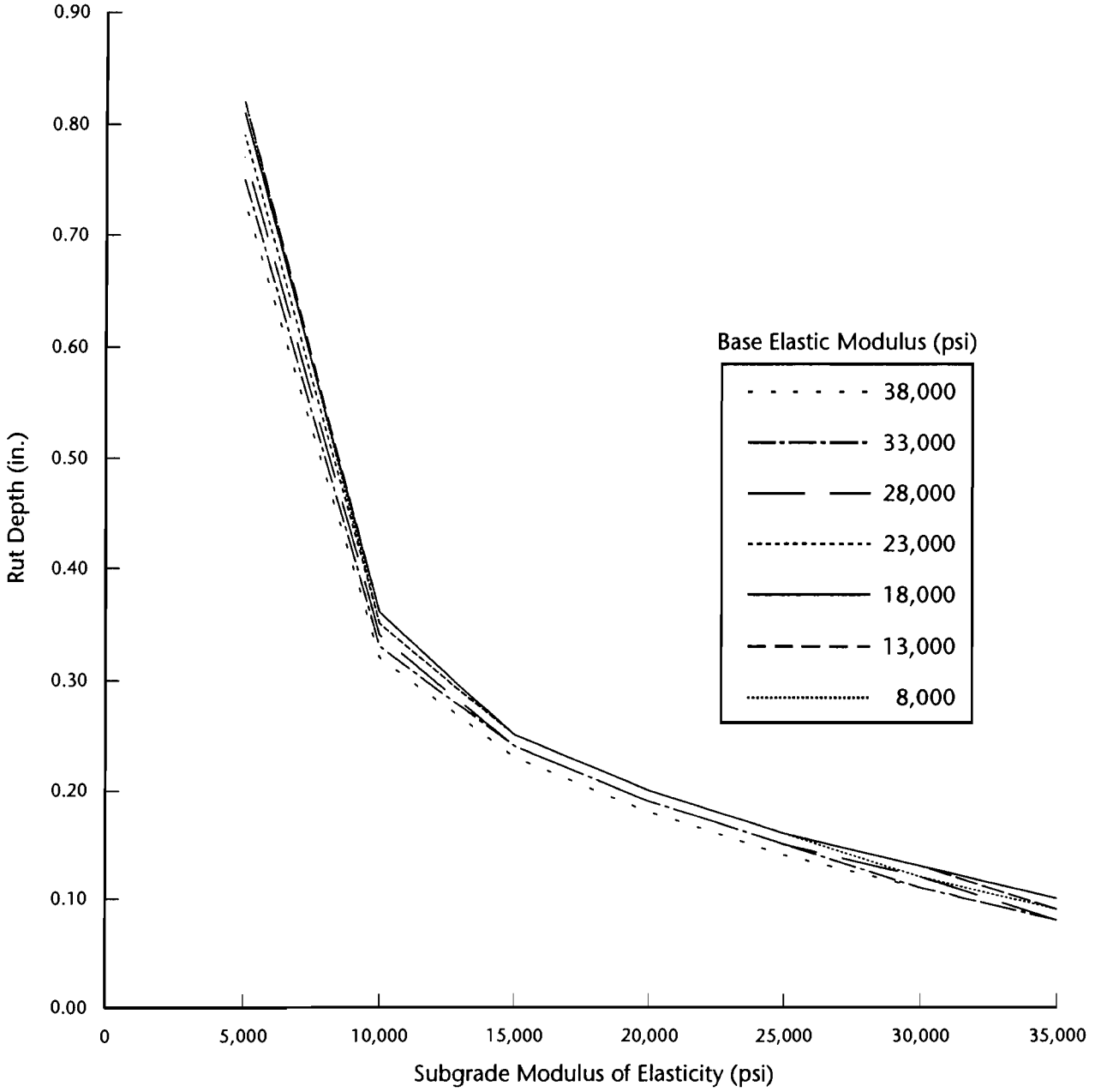


Figure 11. LOADRATE Sensitivity of Rut Depth to Modulus of Elasticity for Clayey Sand for FM 2377

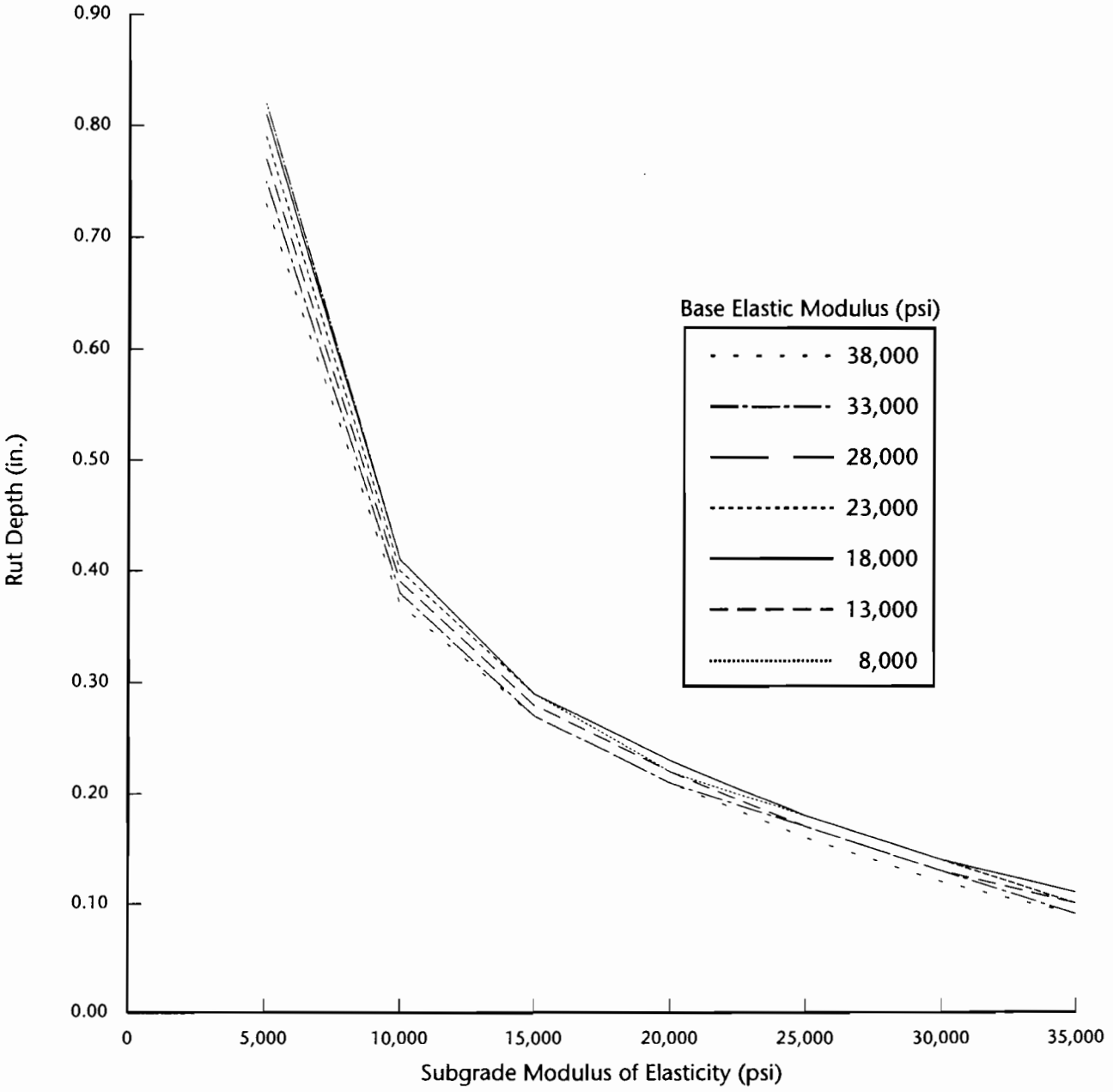


Figure 12. LOADRATE Sensitivity of Rut Depth to Modulus of Elasticity for Heavy Clay for FM 2377

CHAPTER 5. CONCLUSIONS AND RECOMMENDATIONS

FACTORS AFFECTING PAVEMENT LIFE

There are many factors that influence the longevity of low-volume roads. The principal factors affecting the distress and ultimately the performance of asphalt-surfaced pavements are rutting, fatigue, and low-temperature cracking. Of these factors both fatigue and rutting are load-related. If one is to determine whether a particular section of roadway can withstand a certain level of traffic without restriction, several factors should be considered.

The Strategic Highway Research Program (SHRP) has proved that there was a significant amount of knowledge we simply did not possess about asphalt binders and their effects on asphalt concrete pavement performance. The variations in asphalt concrete performance with the current viscosity- or penetration-graded binder specifications is quite significant. However, on low-volume roads with very thin layers of asphalt as a wearing course, the asphalt characteristics may be minor except during high- or low-temperature conditions.

The properties of the aggregate are highly significant in the strength of the asphalt concrete. However, in the typical FM-type road construction, the strength of the asphalt is not the major factor if the asphalt meets minimum State specifications. The quality of the aggregate in the base material can be very significant and can lend considerable strength to the pavement system.

The effect of stress and strain on loading is very important. When traffic volumes are light, a few instances of heavy loads that produce high stress and strains could cause more damage than 20 years of normal loading. Truck loadings are highly variable. Typical weigh-in-motion (WIM) sites report large variations in truck loading with significant amounts of traffic above legal loadings. Some WIM sites have reported as high as 20-30 percent overloads in truck traffic. It is possible that the rutting or fatigue damage in light-duty pavements caused by occasional very heavy loads can be much greater than that caused by normal traffic.

The effects of moisture on the subgrade, base, and pavement surface cannot be over-emphasized. It is widely known that high moisture conditions in the springtime environment result in highly damaged pavements. It has been suggested that the wear to some pavements during the three months of spring, accumulated over the pavement lifetime, could account for a very high percentage of all the load damage to the pavement. Some states have a load-zoning policy that rates roads only during the three months of spring. The theory is that soil support and strength will be at least two or three times weaker during the springtime, so that the greatest damage is likely during this period.

PERFORMANCE OF LOADRATE

The LOADRATE program does not do an adequate job of evaluating roads for removal from load-zoned status. The data available for load-zone evaluation are FWD deflections, soil type and/or lab analysis, base-type and thickness. The calculation of modulus values by LOADRATE is significantly lower than that by the *Modulus* program. The lower modulus could be intentional since there is no correction for season, moisture, or temperature of the FWD deflections, but there is no documentation which might indicate that this is the case. The large variation in base-to-subgrade modulus ratio calculated by

LOADRATE versus *Modulus* leaves one uncomfortable about the results of the LOADRATE-calculated moduli.

The coding of the program is poorly done and not well documented. The code was not written for clarity nor written to be user-friendly. Even with corrections made to the code, the program is not very efficient or useful. The LDATA portion was written in FORTRAN and is case-sensitive requiring all capital letters. The program also has very limited error trapping, requiring a restart of the program if certain common errors are made in the input screens.

The rutting model portion of the program provides results that are extremely different from those using current procedures. If there is any hope of using this program, the rutting model must be studied for its applicability for predicting rutting on low-volume roads. The effect of using FWD data gathered in the spring season versus the summer could provide significantly different results. It may be possible that LOADRATE could predict results similar to those obtained with current procedures if all FWD data were taken during springtime or adjustments were made to account for seasonal variations.

However, the bottom line is that the method of calculation of predicted rutting and allowable loading is empirically based on regression models and compared to computer-generated data. There is not sufficient documentation provided to the authors that this rutting is a better predictor than the current procedures. LOADRATE does not provide results that agree with current procedures or with the more conservative Texas Triaxial method. The implementation of the LOADRATE program and procedures could lead to the premature failure of 75-90 percent of the 17,000 lane-miles of load-zoned roads.

RECOMMENDATIONS

1. LOADRATE should not be adopted as replacement for current TxDOT procedures for evaluating load-zoned roads.
2. LOADRATE should not be used as a replacement for *Modulus* for back-calculating modulus of the base and subgrade from FWD data.
3. The LOADRATE program should not be revised for the purpose of evaluating load-zoned roads.
4. If TxDOT requires integrated computer programs to evaluate load-zoned roads, new programs should be developed using the *Modulus* program as a kernel together with current procedures and the present method of evaluation.
5. Research and field-testing are required to validate other possible subgrade-rutting models. Surface-rutting models should also be incorporated into any program used for load-zone analysis..
6. Monitoring the performance of surface- and subgrade-rutting models as well as the surface-cracking models over the various geographical locations in Texas should be performed and documented in a database with a geographical interface for calibration of these models as well as management of the load-zoned network.

REFERENCES

1. Santha, B. Lanka, Weishih Yang, and , Robert L. Lytton, "A Microcomputer Based Procedure for Load-Zoning Determination on Low-Volume Roads," Report FHWA/TX-90/473-3F, Texas Transportation Institute, The Texas A&M System and Texas Department of Transportation, November 1989.
2. Jackson, J., and M. Murphy, "Implementation of a Falling Weight Deflectometer Load-Zoning Procedure in Texas," paper submitted to the Symposium on Nondestructive Deflection Testing and Backcalculation for Pavements, August 19-21, 1991.
3. Scullion, T., and C. Michalak, "Modulus 4.0, User's Manual," Report FHWA/TX-91/1123-4, Texas Transportation Institute, The Texas A&M System and Texas Department of Transportation, January 1991.
4. "Test Method Tex-117-E Rev: December 1982 Triaxial Compression Tests for Disturbed Soils and Base Materials," Division of Materials and Tests, Texas Department of Transportation, 1982.
5. Yapa, K. A. S., and R. L. Lytton, "A Simplified Mechanistic Rut Depth Prediction Procedure for Low-Volume Roads," Report FHWA/TX-89/473-1, Texas Transportation Institute, The Texas A&M System and Texas Department of Transportation, December 1988.
6. Thompson, M. R., and D. Nauman, "Rutting Rate Analyses of the AASHO Road Test Flexible Pavements," paper 930170 presented to 73rd Annual Meeting of the Transportation Research Board, Washington, D.C., January 1993.

APPENDIX A

The following code reflects both the corrected listing for LOADRATE and the original code for the Texas Transportation Institute program. The sequential line numbers are for reference only. They are not part of either program.

```
1          '*****************$$$ PROGRAM - LOADRATE - $$$*******************
2          '*                      Version  AUGUST 1989                    *
3          '*           Load Rating of Light Pavement Structures           *
4          '*                     WRITTEN BY K. M. CHUA                    *
5          '*                          REVISED BY                          *
6          '*                       B. LANKA SANTHA                        *
7  "       '*         Pavement Systems, Texas Transportation Institute     *"
8  "       '*             TTI Building, Texas A&M University,              *"
9  "       '*                College Station, Texas 77843                  *"
10         '*                       (409)-845-5982                         *
11         '*                    Update #1 20th August 1989                *
12         '************************************************************
13         '* Update  and Revision 29th January 1993                       *
14         '*                                                              *
15         '* At the Center for Transportation Research                    *
16         '* The University of Texas at Austin                            *
17 "       '* Austin, TX 78705-2650                                        *"
18         '* (512) 472-8875                                               *
19         '************************************************************
20
21         '=============================================================
22         'VARIABLE       DESCRIPTION
23         '=============================================================
24
25         'CK$(#)
26
27         'AC(#)
28         'AM(#)          Rut depth (in.)
29         'BA(#)          Base thickness (in.)
30         'BBA(#)
31         'BC(#)
32         'C(#)
33 "        'D(#,#)         Deflection (mil) [index, sensor #]"
34         'D7(#)
35         'DD(#)
36 "        'DE(#,#)"
37         'DES(#)         Modulus change due to suction(psi);temp.correc.data
38         'DET(#)         Modulus change due to temp(psi);temp. correc. data
39         'DF(#)
40         'DSUM(#)
41         'DYR(#)         Cumulative % sections
42         'E1(#)          Base elastic modulus (psi)
43         'E2(#)          Subgrade elastic modulus (psi)
44         'FSUM(#)
45         'FW(#)
46         'H(#)
47         'JSUM(#)
48         'K1(#)
49         'K1S(#)
50         'NE(#)          Wanted cond. modulus (psi);temp. correction data;new E
51         'NS(#)          Section number
```

```
52          'NX(#)
53          'NY(#)              Remaining life (years)
54          'OE(#)              Input modulus (psi);temp. correction data;old E
55          'passes(1)          Allowable wheel passes
56          'PFW(#)             FWD load (lbs)
57          'PRESS(#)
58          'PS(#)              Number of passes
59          'R(#)
60          'RFINAL(1)          Rut depth caused [in.]
61          'SI(#)
62          'SL(#)
63 "         'SO(#,#)"
64          'SS(#)
65          'SSO(#)
66          'TOTDE(#)           Total change in E due to temp. & suction
67          'W(#)
68 "         'WL(#,#)            Single wheel/ESWL (lbs) [truck#,axle#]"
69          'WLOAD(#)
70          'WOS(#)
71          'WP(#)
72          'WW(#)
73          'X(#)               Forward sensor spacing(in.); radial distance
74          'X1(#)              (Poly.Interpol) Parameter values used in the data base
75          'XI(#)              Base rutting potential input parameter for data base
76          'XJ(#)              Base resilient modulus input parameter for data base
77          'XK(#)              Subgrade resilient modulus input parameter for data base
78          'XL(#)              Subgrade rutting potential input parameter for data base
79          'XM(#)              Base thickness input parameter for data base
80          'Y(#)
81           'Y1(#)              (Poly.Interpol) Rut depths from data base corresponding
to parameter
82          'YITEMP(#)
83          'YJTEMP(#)
84          'YKTEMP(#)
85          'YLTEMP(#)
86
87
88 "         'a                  first cycle strain (ABA,ACH,ACL,ASM)"
89 "         'aa                 Coeff. (AABA,AACH,AACL,AASM)"
90          'AABA               A coeff for ABA
91          'AACH               A coeff for ASG for MSG=1
92          'AACL               A coeff for ASG for MSG=2
93          'AASM               A coeff for ASG for MSG=3
94 "         'ab                 Exponent (ABBA,ABCH,ABCL,ABSM)"
95          'ABA                A coeff for base for XII
96          'ABBA               B coeff for ABA
97          'ABCH               B coeff for ASG for MSG=1
98          'ABCL               B coeff for ASG for MSG=2
99          'ABSM               B coeff for ASG for MSG=3
100            'AL1                Allowable legal axle load single
101            'AL2                Allowable legal axle load tandem
102            'AL3                Allowable legal axle load triple
103            'AL                 # of years for GR
104            'ALP                Alpha; Linear thermal expansion
105            '                   Note: Cubic thermal expansion=Alpha * 3
106            'ASG                A coeff for subgrade for XLL
107            'AV                 Ave. number of passes to cause specified rut depth
108   "         'b                  Residual deformation rate (BBA,BCH,BCL,BSM)"
```

```
109           'BBA            b Base course material;
110           '               B coeff for base for XII
111           'BSG            b Subgrade material;
112           '               B coeff for subgrade for XLL
113           'CHO1-4         Menu selection
114           'COL            Printer column number
115           'DELV           Delta Volume; Change in volume
116           'DT             ABS(DTEMP)
117           'DTEMP          Delta(RTEMP-TTEMP)
118           'E              Elastic modulus of soil grains
119           'EBA            Elastic resilient modulus- base course (psi)
120           'Ei             Elastic modulus
121           'EQ             EQ=0:Dynaflec
122           '               EQ=1:Falling Weight Deflectometer (FWD)
123           'EQPASS         # of equivalent standard wheel (9000 lbs) passes
124           'ESAL1          First year traffic;Equiv.Standard Axle Load
125           'ESG            Elastic resilient modulus- subgrade (psi)
126           'GR             Annual traffic growth rate
127           'HI             Input suction (psi) condition;temp. correction data
128           'HL             Wanted suction (psi) condition;temp. correction data
129   "        'I              Index, counter"
130           'II             Remaining life (years)
131           'ID             District
132           'INIOPT
133           'ITE            Iteration counter
134           'K1             Modulus coeff
135           'K2             K2 is the power of the eq. E=K1*Theta^K2
136           'KO1            SIGR/SIGZ ratio for base course
137           'KO2            SIGR/SIGZ ratio for subgrade
138           'LGABA          log a Base course material
139           'LGASG          log a Subgrade material
140           'MINX           Starting section number
141   "        'MINY           Minimun ""Y"" value"
142           'MAXX           Ending section number
143   "        'MAXY           Maximum ""Y"" value"
144           'MSG            Subgrade Material type
145           '               MSG = 1         Heavy clay (CH)
146   "        '                  MSG = 2          Light/silty clay, clayey silt (CL-
ML)"
147           '               MSG = 3         Clayey/silty/uniform sand (SC-SM)
148   "        'Mr             Resilient modulus (ksi) (EBA,ESG)"
149           'MTYPE          Type of base material
150           'N              Number of wheel passes (cycles)
151           'N2             Porosity
152           'NTT            Truck number
153           'NUM            Number of levels of the parameter
154           'NV             New value; temporary value storage
155           'OPP            OP=1 :soil classification
156           '               OP=2 :Lab data
157           'OO             OO=1: indicates all corrections have been done
158           'PA             Total number of passes during ESAL1 period
159           '               Number of passes used for rut depth calculation
160           'PI#            pi=3.1415926536
161           'PL
162           'PR             Hardcopy 0=no 1=yes 2=results only
163           'PRO1           Property of material= 3(1-(poissons ratio)^2)\(4E)
164           'PR1-10         Printer control characters
165           'PS             Change in pressure due to change in suction
```

```
166           'PT             Printer type and associated printer characters
167           '               PT=0: none specified
168           '               PT=1: Epson/Panasonic
169           '               PT=2: Mannesmann Tally
170            'PT1              Change in Hydrostatic pressure due to a change in
temp.
171           'RALLOW          Allowable rut depth (in.)
172           'REXIST          Existing rut depth (in.)
173           'RM              Recorded rut (in.)
174                'RTEMP                Wanted temp. condition; Temp. correction
data;Reference temp
175            'RX               Allowable rut (in.);  Rut depth used for remaining
life
176           '                    and number of passes calculation
177           'SG              Soil specific gravity
178           'SLECT1          Calculation selection
179           'SLF             Selected format
180           'SLO             Optional output selection
181           '                SLO=1:t$=Elastic mod. of base course
182           '                SLO=2:t$=Elastic mod. of subgrade
183           '                SLO=3:t$=Number of passes for rut depth
184           '                SLO=4:t$=Remaining life
185           'TBA             Thickness of base layer (in.)
186           'TEMRUT          Rut temperature correction (TEMRUT=0: not corrected)
187                  'THETA                   Bulk stress; Sum of the principal
stresses;E=K(1)*THETA^K(2)
188           'TTEMP           Input temp condition;Temp. correction data
189           'U               Poisson.s ratio
190           'UW              Unit weight of material (pcf)
191           'VOID            Void ratio
192           'WL              Which line to edit data
193           'Y               Interpolated value
194           'X               Value of the parameter
195           'XII             Base rutting potential
196           'XJJ             Base resilient modulus
197           'XKK             Subgrade resilient modulus
198           'XLL             Subgrade rutting potential
199           'XMM             Base thickness
200           'ZZ1             Elastic mod. for base course depth
201           'ZZ2             Elastic mod. for subgrade depth
202
203           'a$              Job title
204           'AA$             Job title - rut depth subroutine
205           'C$              County
206           'D$              Date
207           'F$              Road
208   "        'NN$             Null; ""Press return to continue"""
209           'X$              Input file name
210           'XX$             Output file name
211
212           '============================================================
213           'Dimensioning
214           '============================================================
215   "           DIM D(160, 7), SO(160), DE(160, 7), AC(160), PFW(160), K1(160),
W(160), BC(160), K1S(160)"
216   "        DIM WL(160, 8), AM(160), PS(160), SS(160), CK$(160), NS(160), H(160),
SI(160), BA(160), NX(160)"
217   "        DIM SSO(160), BBA(160), WW(160), WP(160), WOS(160), E1(160), E2(160),
```

```
SL(160), Y(160), D7(160)"
218   "         DIM DSUM(10), JSUM(10), FSUM(10), DF(10), FW(10), R(10), PRESS(10),
WLOAD(10), x(10), DYR(160)"
219   "          DIM det(160), des(160), NE(160), OE(160), TOTDE(160), RFINAL(160),
passes(160), NY(160)"
220   "           DIM XM(160), XL(160), XK(160), XJ(160), XI(160), Y1(5), X1(5),
YLTEMP(5), YKTEMP(5)"
221   "         DIM YJTEMP(5), YITEMP(5), C(5), DD(5), RUT(2, 3, 3, 3, 3)"
222
223           '=========================================
224           'INITIALIZE SELECTED VARIABLES
225           '=========================================
226           PT = 0
227           x(1) = .01: x(2) = 12: x(3) = 24: x(4) = 36
228           x(5) = 48: x(6) = 60: x(7) = 72
229           SG = 2.67
230           PI# = 3.1415926536#
231
232           '=========================================
233   730     'MAIN MENU OF SELECTIONS
234           '=========================================
235           CLS : KEY OFF
236   "        LOCATE 3, 10"
237   "        PRINT ""LOADRATE - STRUCTURAL ANALYSIS OF LIGHT PAVEMENTS"""
238   "        LOCATE 5, 10"
239   "        PRINT ""                        MAIN MENU"""
240   "        LOCATE 7, 10"
241   "        PRINT ""              SELECT THE OPERATION DESIRED:"""
242   "        LOCATE 9, 10"
243   "        PRINT ""(1)  Structural analysis of pavements"""
244   "        LOCATE 10, 10"
245   "          PRINT ""    ( moduli, # of passes, rut depth, remaining life and
allowable"""
246   "        LOCATE 11, 10"
247   "         PRINT ""      axle load calculations, and analysis of temperature and
"""
248   "        LOCATE 12, 10"
249   "        PRINT ""      moisture effects on base material)"""
250   "        LOCATE 13, 10"
251   "        PRINT ""(2)  Temperature correction of base modulus only"""
252   "        LOCATE 14, 10"
253   "        PRINT ""(3)  Rut depth calculation only"""
254   "        LOCATE 15, 10"
255   "        PRINT ""(4)  Exit program"""
256   "        LOCATE 20, 10"
257   "        INPUT ""ENTER SELECTION : ""; SLECT1"
258           IF (SLECT1 < 1) OR (SLECT1 > 4) THEN 730
259           IF SLECT1 = 2 THEN CHO1 = 0: GOTO 10010
260           IF SLECT1 = 3 THEN INIOPT = 2: GOSUB 12400
261           IF SLECT1 = 4 THEN CLS : END
262
263           '=========================================
264           'Structural Analysis of Pavements
265           '=========================================
266           FOR I = 1 TO 160
267           NY(I) = 0
268           DYR(I) = 0
269           PS(I) = 0
```

```
270           NEXT I
271           OO = O
272   "         CLS : PRINT : PRINT ""     PROGRAM : LOADRATE  Version 2.0 August
1989"""
273           PRINT : PRINT
274   "1790    PRINT ""INPUT FILE NAME:"""
275           INPUT x$
276   "1791    ' x$ = ""c:\loadrate\fm1044a.out"""
277   1792    'PRINT x$
278   "        PRINT ""Please Wait ......"""
279   "        OPEN ""I"", #1, x$"
280   "        INPUT #1, EQ"
281   "        INPUT #1, ID"
282   "        INPUT #1, C$"
283   "        INPUT #1, F$"
284   "        INPUT #1, D$"
285   "        INPUT #1, a$"
286   "        INPUT #1, NC, NT, RX, RM, AL, PA, GR, ESAL1"
287           FOR NTT = 1 TO NT
288   "        INPUT #1, NX(NTT)"
289           FOR I = 1 TO NX(NTT)
290   "        INPUT #1, WL(NTT, I)"
291           NEXT I
292           NEXT NTT
293           FOR I = 1 TO NC
294   "         IF EQ = O THEN INPUT #1, NS(I), BA(I), D(I, 1), D(I, 2), D(I, 3),
D(I, 4), D(I, 5)"
295   "        IF EQ = 1 THEN INPUT #1, NS(I), BA(I), D(I, 1), D(I, 2), D(I, 3)"
296   "        IF EQ = 1 THEN INPUT #1, D(I, 4), D(I, 5), D(I, 6), D(I, 7), PFW(I)"
297           NEXT I
298           CLOSE
299   2290    'To print input card images
300   "        OPEN ""SCRN:"" FOR OUTPUT AS #1"
301   2340    FOR ite = 1 TO NT
302   "        IF ite > 1 THEN PRINT #1, : GOTO 2440"
303   "        CLS : PRINT #1, ""TEXAS TRANSPORTATION INSTITUTE"""
304   "        PRINT #1, ""LOAD RATING OF LIGHT PAVEMENT"""
305   "        PRINT #1,"
306   "        PRINT #1, ""JOB : ""; a$; ""          (INPUT FILE ""; x$; "")"""
307   "        PRINT #1,"
308   "         PRINT #1, ""DISTRICT:""; ID; ""      COUNTY:""; C$; ""      ROAD:"";
F$"
309   "         PRINT #1, ""ALLOWABLE RUT(INS):""; RX; ""      RECORDED RUT(INS):"";
RM"
310   "2400    'INPUT ""press RETURN to continue""; nn$"
311   "        PRINT #1,"
312   2440    NTT = ite
313   "        PRINT #1, ""TRUCK NO. ""; NTT"
314   "        PRINT #1, ""AXLE NUMBER   SINGLE WHEEL/ESWL(LBS)"""
315           'To loop of the number of axles NX and number of trucks NTT
316   "2441    INPUT ""press return to continue(line2441)""; nn$"
317           FOR I = 1 TO NX(NTT)
318   "        PRINT #1, TAB(5); I; TAB(20); WL(NTT, I)"
319           NEXT I
320   "2442    INPUT ""Press RETURN to continue(line2442)""; nn$"
321   "        PRINT #1,"
322           IF ite > 1 THEN 3010
323   "        PRINT #1, ""ANNUAL TRAFFIC GROWTH RATE:""; GR; ""  # OF YEARS:""; AL;
```

```
"" FIRST YEAR TRAFFIC:""; ESAL1"
324   "        PRINT #1, ""TOTAL NUMBER OF PASSES DURING ABOVE PERIOD:""; PA"
325   "        IF EQ = 0 THEN PRINT #1, ""DATE:""; D$; ""     DYNAFLECT"""
326   "           IF EQ = 1 THEN PRINT #1, ""DATE:""; D$; ""    FALLING WEIGHT
DEFLECTOMETER"""
327   "2443    'INPUT ""Press RETURN to continue""; nn$"
328   "        PRINT #1,"
329   "         PRINT #1, ""SECTION    BASE                                DEFLEC-
TION"""
330   "                IF EQ = 0 THEN PRINT #1, ""    NO.       THICKNESS
(MILS) """
331   "                IF EQ = 1 THEN PRINT #1, ""    NO.       THICKNESS
(MILS)                    LOAD"""
332   "         IF EQ = 0 THEN PRINT #1, ""          (INS)      W1       W2       W3
W4      W5     """
333   "         IF EQ = 1 THEN PRINT #1, ""          (INS)      W1       W2       W3
W4      W5      W6      W7   (LBS)"""
334   "        IF EQ = 1 THEN PRINT #1, ""(RADIAL DISTANCE)"";"
335   "          IF EQ = 1 THEN PRINT #1, USING ""  ###.#    ###.#    ###.#    ###.#
###.#   ###.#   ###.#""; x(1); x(2); x(3); x(4); x(5); x(6); x(7)"
336   "        PRINT #1,"
337          PPR = 0
338          FOR I = 1 TO NC
339   "2444    'IF I = 15 THEN LOCATE 25, 1: INPUT ""Press RETURN to continue"";
nn$"
340          IF TBC = 1 THEN BA(I) = 0
341          IF BA(I) <> 0 THEN 2890
342   "        WW = (DE(I, 1) - DE(I, 7)) / DE(I, 7)"
343          BA(I) = (SSO(I) / 1958 * WW) ^ 1.85
344          BA(I) = INT((BA(I) * 10 + 5) / 10)
345          IF PPR = 2 THEN 3000
346   "2890    IF EQ = 0 THEN PRINT #1, NS(I); TAB(12); BA(I); TAB(23);"
347   "          IF EQ = 0 THEN PRINT #1, USING ""####.##""; D(I, 1); D(I, 2); D(I,
3); D(I, 4); D(I, 5)"
348   "        IF EQ = 1 THEN PRINT #1, NS(I); TAB(12); BA(I); TAB(19);"
349   "          IF EQ = 1 THEN PRINT #1, USING ""###.##   ###.##   ###.##   ###.##
###.##   ###.##   ###.## #######""; D(I, 1); D(I, 2); D(I, 3); D(I, 4); D(I, 5); D(I,
6); D(I, 7); PFW(I)"
350   3000    NEXT I
351   "        PRINT #1,"
352   3010    CLOSE
353          IF PR = 1 THEN 3670
354   "3080    LOCATE 25, 1: PRINT SPC(132);"
355   3570    IF ite <> 1 THEN 3640
356   "          IF TBC = 0 THEN LOCATE 25, 1: INPUT ""DO YOU WANT HARDCOPY? 0=NO
1=YES 2=RESULTS ONLY""; PR"
357   "          IF TBC = 1 THEN LOCATE 25, 1: INPUT ""DO YOU WANT HARDCOPY? 0=NO
1=YES ""; PR"
358   "                                LOCATE    25,    1:        PRINT    ""
"""
359          IF PR = 2 THEN PR = 1: PPR = 2: GOTO 3670
360   "3640    IF PR = 1 THEN CLOSE : OPEN ""LPT1:"" FOR RANDOM AS #1: GOTO 2340"
361   3670    IF ite <> 1 THEN 4790
362   "3671    'PRINT ""line after 3670(LINE3071)"": INPUT nn$"
363          IF EQ = 0 THEN PL = 1000
364          IF EQ = 1 THEN PL = 10956.3
365          FOR I = 1 TO NC
366   "        IF EQ = 1 THEN SO(I) = PFW(I) / D(I, 1): DE(I, 1) = PL / SO(I)"
```

```
367   "         'D(I,7) may not be taken at 94.5"" then need to change"
368   "         IF EQ = 1 THEN GOSUB 7290: DE(I, 7) = D7(I) * PL / PFW(I)"
369             'D7(I) IS FROM SUBROUTINE GEOMBASIN
370             'To convert Dynaflect readings to DE
371   "4000     IF EQ = 0 THEN DE(I, 7) = 3.38075 * D(I, 5) ^ .639462"
372   "4001     IF EQ = 0 THEN DE(I, 7) = 4.5688721# * D(I, 5) ^ .578444: DE(I, 1) =
-7.24474 + (29.6906 * D(I, 1))"
373   "         IF EQ = 0 THEN SO(I) = 86.0122 * EXP(.00187211# * PL / D(I, 1))"
374             IF TBC = 0 THEN SSO(I) = SO(I)
375             'SI - is used only when low overall stiffness is encountered
376             SI(I) = -109.663 + 1.31393 * SO(I)
377             IF SO(I) <= 83.46183 THEN SI(I) = SO(I)
378             H(I) = BA(I)
379             NEXT I
380   "         PRINT ""Please Wait ..... """
381             FOR I = 1 TO NC
382   "         x = BA(I): Y = DE(I, 7)"
383             IF x > 18 THEN x = 18
384             K1S(I) = (9555.651 + 370.3322 * x) * Y ^ (-1.21665 + .016349 * x)
385             IF K1S(I) < 5450 THEN K1S(I) = 5450
386             IF K1S(I) > 62200 THEN K1S(I) = 62200
387             LL1 = .4342945 * LOG(K1S(I))
388             A11 = 10 ^ (10.50698 - 1.97986 * LL1)
389             A12 = 361.506 - 2131.05 / (K1S(I) * 10 ^ -3) - 18.305 * K1S(I) * 10 ^
-3 + .36743 * (K1S(I) * 10 ^ -3) ^ 2 - .00256 * (K1S(I) * 10 ^ -3) ^ 3
390             A13 = -13.6825 + 88.26011 / (K1S(I) * 10 ^ -3) + 1.165388 * K1S(I) *
10 ^ -3 - .0236 * (K1S(I) * 10 ^ -3) ^ 2 + .00017 * (K1S(I) * 10 ^ -3) ^ 3
391             A1 = A11 + A12 * BA(I) + A13 * BA(I) * BA(I)
392             B11 = .032285 - .109 / (K1S(I) * 10 ^ -4)
393             B12 = .123403 - .03883 * LL1
394             B13 = -.00426 + .001256 * LL1
395             B1 = B11 + B12 * BA(I) + B13 * BA(I) * BA(I)
396   "         K1(I) = (DE(I, 1) / A1) ^ (1 / B1)"
397             IF K1(I) <= 3000 THEN K1(I) = 3000
398             IF K1(I) > 60000 THEN K1(I) = 60000
399             NEXT I
400
401             '============================================================
402   4790      'Elastic modulus calculations ECAL
403             '============================================================
404             AV = 0
405             KO1 = .6: KO2 = .82
406             AREA = PI# * 5.9 * 5.9
407             FOR I = 1 TO NC
408             IF EQ = 0 THEN PFW(I) = 9000
409             SL(I) = PFW(I) / AREA
410             ZZ1 = 1 + BA(I) / 2: ZZ2 = 7 + BA(I)
411             ALS1 = 2.416966 * SL(I) ^ .477234: ALS2 = .6007171 * K1S(I) ^ .058992
412             ALH = 1.338562 * (1 + BA(I)) ^ -.145887
413             RHB = .59614133# * K1(I) ^ 5.593462E-02
414             RHH = 3.0385722# * (1 + BA(I)) ^ .419624
415             RHS = 272.63462# * K1S(I) ^ -.58427
416             NSG = -1.01866 + 1.239625 / (K1S(I) * 10 ^ -3) + .000013 * (K1S(I) *
10 ^ -3) ^ 3 + .061115 * (K1S(I) * 10 ^ -3) - .00165 * (K1S(I) * 10 ^ -3) ^ 2
417             ALPHA = ALS1 * ALS2 * ALH
418             RHO = (1 + BA(I)) * LOG(RHH * RHS * RHB / ALPHA) / LOG(2.7182818#)
419             ER1 = ALPHA * EXP(-((-RHO / ZZ1) ^ 1)) + 5.9: ER2 = ALPHA * EXP(-((-
RHO / ZZ2) ^ 1)) + 5.9
```

```
420           SIGZ1 = PFW(I) / (PI# * ER1 * ER1): SIGZ2 = PFW(I) / (PI# * ER2 * ER2)
421           SIGR1 = KO1 * SIGZ1: SIGR2 = KO2 * SIGZ2
422           E2(I) = K1S(I) * (ABS(SIGZ2 - SIGR2)) ^ NSG: E1(I) = K1(I) * (SIGZ1 +
2 * SIGR1) ^ .33
423           NEXT I
424
425           '============================================================
426   5080    'To calculate the rut depth and number of passes
427           '============================================================
428           INIOPT = 3: GOSUB 12400
429           FOR I = 1 TO NC
430           PS(I) = passes(I)
431           AM(I) = RFINAL(I)
432           IF GR < 0 THEN 5097
433           NY(I) = (LOG(PS(I) * GR + ESAL1) - LOG(ESAL1)) / LOG(1 + GR)
434           TOTYR = TOTYR + NY(I)
435           IF NY(I) > NYMAX THEN NYMAX = NY(I)
436   5097    AV = AV + PS(I)
437           NEXT I
438           IF GR < 0 THEN 5100
439           FOR II = 1 TO NYMAX
440           FOR I = 1 TO NC
441           IF NY(I) >= II THEN DYR(II) = DYR(II) + 1
442           NEXT I
443           DYR(II) = (DYR(II) / NC) * 100
444           IF DYR(II) = 100 THEN S100 = II
445           NEXT II
446           AL1 = ((S100 / 10) ^ .2432) * 18
447           AL2 = ((S100 / 10) ^ .2467) * 33.5
448           AL3 = ((S100 / 10) ^ .2534) * 48.25
449           IF AL1 > 18 THEN AL1 = 18
450           IF AL2 > 34 THEN AL2 = 34
451           IF AL3 > 46 THEN AL3 = 46
452           MINX = NS(1)
453           MAXX = NS(NC)
454   "5100       IF PR = 1 THEN OPEN ""LPT1:"" FOR RANDOM AS #1: CLS : PRINT
""Printing ....."""
455   "         IF PR = 0 THEN OPEN ""SCRN:"" FOR OUTPUT AS #1: CLS"
456   "             IF GR < 0 THEN PRINT #1, "" SECTION         LAYER PROPERTIES
NO. OF      RUT"""
457   "           IF GR < 0 THEN PRINT #1, ""     NO.           ELASTIC MODULUS
ALLOWABLE   DEPTH"""
458   "         IF (EQ = 0) AND (GR < 0) THEN PRINT #1, ""           E1-BASE     E2-
SUBGR                  PASSES     (IN)"""
459   "          IF (EQ = 1) AND (GR < 0) THEN PRINT #1, ""             BASE/SUBB
SUBGRADE                  PASSES     (IN)   """
460   "             IF GR >= 0 THEN PRINT #1, "" SECTION         LAYER PROPERTIES
REMAINING       NO. OF      RUT"""
461   "            IF GR >= 0 THEN PRINT #1, ""     NO.           ELASTIC MODULUS
LIFE         ALLOWABLE   DEPTH"""
462   "          IF (EQ = 0) AND (GR >= 0) THEN PRINT #1, ""             E1-BASE
E2-SUBGR      (YEARS)        PASSES     (IN)"""
463   "          IF (EQ = 1) AND (GR >= 0) THEN PRINT #1, ""             BASE/SUBB
SUBGRADE      (YEARS)        PASSES     (IN)   """
464           FOR I = 1 TO NC
465   "         IF (EQ = 1) THEN PRINT #1, TAB(3); NS(I); TAB(13); USING ""########"";
E1(I); TAB(25);"
466   "         IF (EQ = 1) THEN PRINT USING ""######""; E2(1);"
```

```
467   "          IF (EQ = 1) AND (GR >= 0) THEN PRINT #1, TAB(39); USING ""###.#""; NY(I);"
468   "         IF (EQ = 1) THEN PRINT #1, TAB(51); USING ""##########""; PS(I);"
469   "         IF (EQ = 1) THEN PRINT #1, USING ""#######.##""; AM(I)"
470           NEXT I
471   "         PRINT #1,"
472   "5101     'INPUT ""press return to continue(line5101)""; nn$"
473   "         PRINT #1,"
474           IF GR < 0 THEN 5740
475   "         PRINT #1,"
476   "         PRINT #1, ""                          Allowable Axle Load Limits (kips)"""
477   "              PRINT #1, ""                SINGLE                          TANDEM TRIPLE"""
478   "         PRINT #1,"
479   "         IF S100 >= 10 THEN PRINT #1, ""          LEGAL                    LEGAL LEGAL"""
480   "               IF S100 < 10 THEN PRINT #1, USING ""                    ##  ##                ##""; AL1; AL2; AL3"
481   "         PRINT #1,"
482   "         PRINT #1,"
483   5740    AV = AV / NC
484   "         PRINT #1, ""AVERAGE NUMBER OF PASSES TO CAUSE SPECIFIED RUT (""; RX; ""): ""; AV"
485   "         PRINT #1,"
486   "         PRINT #1, ""NUMBER OF PASSES USED FOR RUT DEPTH CALCULATION : ""; PA"
487   "         PRINT #1,"
488   "          PRINT #1, ""RUT DEPTH USED FOR REMAINING LIFE AND NUMBER OF PASSES CALCULATION :""; RX"
489   "5741     'INPUT ""press return to continue(line5741)""; nn$"
490           NEXT ite
491           PR = 0
492           PRINT
493           CLOSE
494           IF TBC = 0 THEN 7580
495           GOTO 2290
496
497           '============================================================
498   6510    ' SUBROUTINE to open an output file OUTFILE
499           '============================================================
500           CLS : KEY OFF
501   "         LOCATE 5, 1: PRINT ""TO CREATE OUTPUT FILE"""
502   "         LOCATE 6, 1: INPUT ""ENTER OUTPUT FILE NAME AND <RET>:""; XX$"
503           CLS : KEY OFF
504           IF TEMRUT = 1 THEN SLF = 1: GOTO 6536
505   "         LOCATE 7, 1: PRINT ""Select the format:"""
506   "         LOCATE 9, 1: PRINT ""(1)"""
507   "            IF GR < 0 THEN LOCATE 10, 1: PRINT ""Sec.    Thick    E1(Base)  E2(Subgr)  Passes  Rut depth  """
508   "           IF GR >= 0 THEN LOCATE 10, 1: PRINT ""Sec.    Thick    E1(Base)  E2(Subgr)  Passes  Rut depth  Remaining life"""
509   "         LOCATE 11, 1: PRINT ""(2)"""
510   "         IF (EQ = 1) AND (GR < 0) THEN LOCATE 12, 1: PRINT ""Sect.  Thick  D1  D7  FWD load   E1(Base)  E2(Subgr)  Passes  """
511   "         IF (EQ = 0) AND (GR < 0) THEN LOCATE 12, 1: PRINT ""Sect.  Thick  D1  D5  FWD load   E1(Base)  E2(Subgr)  Passes  """
512   "         IF (EQ = 1) AND (GR >= 0) THEN LOCATE 12, 1: PRINT ""Sect.  Thick  D1  D7  FWD load   E1(Base)  E2(Subgr)  Passes  Remaining life"""
513   "         IF (EQ = 0) AND (GR >= 0) THEN LOCATE 12, 1: PRINT ""Sect.  Thick  D1
```

```
D5   FWD load   E1(Base)  E2(Subgr)  Passes  Remaining life"""
514   "         LOCATE 20, 1: INPUT ""ENTER SELECTION AND <RET> :""; SLF"
515            IF (SLF < 1) OR (SLF > 2) THEN 6510
516   "6536     OPEN ""O"", #2, XX$: PRINT #2, ""     """
517   "         CLS : PRINT #2, "" TEXAS TRANSPORTATION INSTITUTE """
518   "         PRINT #2, ""LOAD RATING OF LIGHT PAVEMENT"""
519   "         PRINT #2,"
520   "         PRINT #2, ""JOB : ""; a$; ""          (INPUT FILE ""; x$; "")"""
521   "         PRINT #2,"
522   "          PRINT #2, ""DISTRICT:""; ID; ""      COUNTY:""; C$; ""      ROAD:""; 
F$"
523   "          PRINT #2, ""ALLOWABLE RUT(INS):""; RX; ""      RECORDED RUT(INS):""; 
RM"
524   "         PRINT #2,"
525   "         PRINT #2, ""TRUCK NO. 1"""
526   "         PRINT #2, ""AXLE NUMBER   SINGLE WHEEL/ESWL(LBS)"""
527   "         PRINT #2, TAB(5); ""1""; TAB(20); ""9000"""
528   "         PRINT #2,"
529   "         PRINT #2, ""ANNUAL TRAFFIC GROWTH RATE:""; GR; ""  # OF YEARS:""; AL; 
""  FIRST YEAR TRAFFIC:""; ESAL1"
530   "         PRINT #2, ""TOTAL NUMBER OF PASSES DURING ABOVE PERIOD:""; PA"
531   "         PRINT #2,"
532   "         IF EQ = 0 THEN PRINT #2, ""DATE:""; D$; ""      DYNAFLECT"""
533   "            IF EQ = 1 THEN PRINT #2, ""DATE:""; D$; ""     FALLING WEIGHT 
DEFLECTOMETER"""
534            IF GR < 0 THEN 6562
535   "           IF TEMRUT = 1 THEN PRINT #2, ""                  TEMP. CORRECTED
*              **              *"""
536   "                  IF  TEMRUT  =  0  AND  SLF  =  1  THEN  PRINT  #2,  ""
*              **              *"""
537   "         IF TEMRUT = 0 AND SLF = 1 THEN PRINT #2, "" SEC   BASE       E1-BASE
E2-SUBGR   NO. OF       RUT DEPTH    REMAINING LIFE"""
538   "         IF TEMRUT = 1 THEN PRINT #2, "" SEC   BASE       E1-BASE    E2-SUBGR
NO. OF       RUT DEPTH    REMAINING LIFE"""
539   "          IF TEMRUT = 0 AND SLF = 1 THEN PRINT #2, ""  #    (IN)        (PSI)
(PSI)     PASSES         (IN)         (YEARS)"""
540   "         IF TEMRUT = 1 THEN PRINT #2, "" #    (IN)        (PSI)        (PSI)
PASSES         (IN)         (YEARS)"""
541   "           IF TEMRUT = 0 AND SLF = 1 THEN FOR I = 1 TO NC: PRINT #2, USING 
""##.###  ##.#     #########      ########   #########          #.##           ##.#"";
NS(I); BA(I); E1(I); E2(I); PS(I); AM(I); NY(I): NEXT"
542   "          IF TEMRUT = 1 THEN FOR I = 1 TO NC: PRINT #2, USING ""##.###  ##.#
#########     ########   #########          #.##           ##.#""; NS(I); BA(I); E1(I);
E2(I); PS(I); AM(I); NY(I): NEXT"
543   "                              IF   SLF   =   2   THEN   PRINT   #2,   ""
*              *"""
544   "         IF SLF = 2 AND EQ = 1 THEN PRINT #2, "" SEC   BASE       D1        D7
FWD LOAD   E1-BASE     E2-SUBG        NO. OF   REMAINING LIFE"""
545   "         IF SLF = 2 AND EQ = 0 THEN PRINT #2, "" SEC   BASE       D1        D5
FWD LOAD   E1-BASE     E2-SUBG        NO. OF   REMAINING LIFE"""
546   "         IF SLF = 2 THEN PRINT #2, ""  #    (IN)     (MILS)    (MILS)    (LBS)
(PSI)       (PSI)         PASSES     (YEARS) """
547   "         IF SLF = 2 AND EQ = 1 THEN FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#      ###.##      ###.##      #######      #########       #######       ###########
##.#""; NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I); NY(I): NEXT"
548   "         IF SLF = 2 AND EQ = 0 THEN FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#      ###.##      ###.##      #######      #########       #######       ###########
##.#""; NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I); NY(I): NEXT"
```

```
549   "         PRINT #2,"
550   "          IF SLF = 1 THEN PRINT #2, "" *  NUMBER OF PASSES AND REMAINING LIFE
FOR SPECIFIED RUT DEPTH OF ""; RX; ""in"""
551   "          IF SLF = 1 THEN PRINT #2, "" ** RUT DEPTH FOR SPECIFIED NUMBER OF
PASSES OF ""; PA; "" IN ""; AL; ""YEARS"":"
552   "          IF SLF = 2 THEN PRINT #2, "" *  NUMBER OF PASSES AND REMAINING LIFE
FOR SPECIFIED RUT DEPTH OF ""; RX; ""in"":"
553   "         PRINT #2,"
554   "           PRINT #2, ""        Remaining Life(yrs)              Cumulative %
Sections"""
555           FOR II = 1 TO NYMAX
556   "         PRINT #2, USING ""            ##                              ###.#"";
II; DYR(II)"
557           NEXT II
558   "         PRINT #2,"
559   "         PRINT #2,"
560   "         PRINT #2, ""                    Allowable Axle Load Limits (kips)"""
561   "             PRINT #2, ""              SINGLE                          TANDEM
TRIPLE"""
562   "         PRINT #2,"
563   "         IF S100 >= 10 THEN PRINT #2, ""         LEGAL                   LEGAL
LEGAL"""
564   "               IF  S100  <  10  THEN  PRINT  #2,  USING  ""                  ##
##                ##""; AL1; AL2; AL3"
565           GOTO 6563
566
567           '=============================================================
568   6562    'SAVE DATA WITHOUT REMAINING LIFE - GR<0
569           '=============================================================
570   "           IF TEMRUT = 1 THEN PRINT #2, ""                  TEMP. CORRECTED
*              **              """
571   "                   IF  TEMRUT  =  0  AND  SLF  =  1  THEN  PRINT  #2,  ""
*              **              """
572   "         IF TEMRUT = 0 AND SLF = 1 THEN PRINT #2, "" SEC   BASE       E1-BASE
E2-SUBGR   NO. OF       RUT DEPTH    """
573   "         IF TEMRUT = 1 THEN PRINT #2, "" SEC   BASE       E1-BASE    E2-SUBGR
NO. OF       RUT DEPTH    """
574   "          IF TEMRUT = 0 AND SLF = 1 THEN PRINT #2, ""  #    (IN)       (PSI)
(PSI)     PASSES         (IN)          """
575   "          IF TEMRUT = 1 THEN PRINT #2, "" #    (IN)       (PSI)       (PSI)
PASSES         (IN)          """
576   "           IF TEMRUT = 0 AND SLF = 1 THEN FOR I = 1 TO NC: PRINT #2, USING
""##.###  ##.#    #########     ########   #########          #.##          ""; NS(I);
BA(I); E1(I); E2(I); PS(I); AM(I): NEXT"
577   "          IF TEMRUT = 1 THEN FOR I = 1 TO NC: PRINT #2, USING ""##.###  ##.#
#########     ########   #########          #.##          ""; NS(I); BA(I); E1(I); E2(I);
PS(I); AM(I): NEXT"
578   "                              IF   SLF   =   2   THEN   PRINT   #2,   ""
*              *"""
579   "         IF SLF = 2 AND EQ = 1 THEN PRINT #2, "" SEC   BASE       D1         D7
FWD LOAD   E1-BASE     E2-SUBG         NO. OF   REMAINING LIFE"""
580   "         IF SLF = 2 AND EQ = 0 THEN PRINT #2, "" SEC   BASE       D1         D5
FWD LOAD   E1-BASE     E2-SUBG         NO. OF   REMAINING LIFE"""
581   "         IF SLF = 2 THEN PRINT #2, ""  #    (IN)     (MILS)    (MILS)    (LBS)
(PSI)       (PSI)          PASSES     (YEARS) """
582   "         IF SLF = 2 AND EQ = 1 THEN FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#    ###.##     ###.##     ######     #########     ########     ###########      "";
NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I): NEXT"
```

```
583   "         IF SLF = 2 AND EQ = 0 THEN FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#     ###.##     ###.##     ######     ########      #######      ##########      "";
NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I): NEXT"
584   "         PRINT #2,"
585   "          IF SLF = 1 THEN PRINT #2, "" *  NUMBER OF PASSES FOR SPECIFIED RUT
DEPTH OF ""; RX; ""in"""
586   "          IF SLF = 1 THEN PRINT #2, "" ** RUT DEPTH FOR SPECIFIED NUMBER OF
PASSES OF ""; PA; "" IN ""; AL; ""YEARS"":"
587   "          IF SLF = 2 THEN PRINT #2, "" *  NUMBER OF PASSES FOR SPECIFIED RUT
DEPTH OF ""; RX; ""in"":"
588   "6563      IF TEC = 1 THEN INPUT ""    DO YOU WANT TO SAVE TEMPERATURE AND
MOISTURE CORRECTED DATA ? 1=YES, 0=NO ""; CH04"
589           IF (CH04 <> 1) AND (CH04 <> 0) THEN 6563
590           IF CH04 = 0 THEN 6584
591   "         PRINT #2, ""  """
592   "         PRINT #2, ""          TEMPERATURE CORRECTION DATA """
593   "         PRINT #2,"
594   "            PRINT #2, ""      INPUT CONDITIONS:                        WANTED
CONDITIONS:"""
595   "          PRINT #2, USING ""   TEMPERATURE =   ###.# deg. F     TEMPERATURE =
###.# deg. F""; TTEMP; RTEMP"
596   "          PRINT #2, USING ""   SUCTION     = #####.# psi         SUCTION     =
#####.# psi""; HI; HL"
597   "         PRINT #2,"
598   "           PRINT #2, ""      INPUT       WDNTED CONDITION     MODULUS CHANGE
MODULUS CHANGE"""
599   "          PRINT #2, ""   MODULUS          MODULUS         DUE TO TEMPER.     DUE
TO SUCTION"""
600   "           PRINT #2, ""      (psi)               (psi)                  (psi)
(psi)"""
601   "         PRINT #2,"
602           FOR I = 1 TO KK1
603   "            PRINT #2, USING ""     #######          #######           ######
######"; OE(I); NE(I); det(I); des(I)"
604           NEXT I: CLOSE
605   6584    RETURN
606
607           '============================================================
608   6970    'SUBROUTINE  MAINMENU
609           '============================================================
610           'KEY OFF: CLS
611           'Input for geometric regression GEOMINPUT
612           'for non-standard FWD sensor spacing
613           GR = 1: CLS : ' To initialize and activate GEOMBASIN
614           'Standard FWD sensor spacing for Texas SDHPT is as follows
615   "         LOCATE 5, 10: PRINT ""FWD Sensor           #1   #2   #3   #4   #5   #6
#7"""
616   "         LOCATE 6, 10: PRINT ""Dist.(ins) from load""; 0; 12; 24; 36; 48; 60;
72"
617   "         LOCATE 7, 10: PRINT ""Type in new spacing ....."""
618   "         LOCATE 8, 10: INPUT ""Sensor #1 at (ins) ""; x(1)"
619   "         LOCATE 9, 10: INPUT ""Sensor #2 at (ins) ""; x(2)"
620   "         LOCATE 10, 10: INPUT ""Sensor #3 at (ins) ""; x(3)"
621   "         LOCATE 11, 10: INPUT ""Sensor #4 at (ins) ""; x(4)"
622   "         LOCATE 12, 10: INPUT ""Sensor #5 at (ins) ""; x(5)"
623   "         LOCATE 13, 10: INPUT ""Sensor #6 at (ins) ""; x(6)"
624   "         LOCATE 14, 10: INPUT ""Sensor #7 at (ins) ""; x(7)"
625           IF x(1) = 0 THEN x(1) = .01
```

```
626           RETURN
627
628           '============================================================
629    7290   ' SUBROUTINE geometric regression GEOMBASIN
630           '============================================================
631           J = 0: K = 0: L = 0: M = 0: R2 = 0
632           N = 4
633    "       IF D(I, 7) = 0 THEN N = 3"
634           IF x(7) = 0 THEN N = 3
635           FOR II = 4 TO (3 + N)
636    "       x = LOG(x(II)): Y = LOG(D(I, II))"
637           J = J + x: K = K + Y: L = L + x * x: M = M + Y * Y: R2 = R2 + x * Y
638           NEXT II
639           b = (N * R2 - K * J) / (N * L - J * J)
640           a = (K - b * J) / N
641           J = b * (R2 - J * K / N)
642           M = M - K * K / N
643           K = M - J
644           R2 = J / M
645    "       'print ""D7@94.49 =""; exp(a)*94.49^b"
646           D7(I) = EXP(a) * 94.49 ^ b
647           RETURN
648
649           '============================================================
650    7580   ' SUBROUTINE  OUTMENU
651           '============================================================
652    "7581   'PRINT ""press return to continue(line7581)"": INPUT nn$"
653           CLS : KEY OFF
654    "       LOCATE 2, 10: PRINT ""          OUTPUT MENU """
655    "       LOCATE 4, 10: PRINT ""Select Optional Output Format"""
656    "       LOCATE 6, 10: PRINT "" (1) Plot Base Course Elastic Modulus"""
657    "       LOCATE 7, 10: PRINT "" (2) Plot Subgrade Elastic Modulus"""
658    "       LOCATE 8, 10: PRINT "" (3) Plot Number of Vehicle Passes"""
659    "       LOCATE 9, 10: PRINT "" (4) Plot Remaining Life"""
660    "       LOCATE 10, 10: PRINT "" (5) Temperature Correction of Base Modulus"""
661    "         LOCATE 11, 10: PRINT ""       (5.1) Rut depth, Passes and Remaining
Life"""
662    "       LOCATE 12, 10: PRINT ""            for corrected moduli"""
663    "       LOCATE 13, 10: PRINT "" (6) Save the Output as a file"""
664    "       LOCATE 14, 10: PRINT "" (7) Exit to Main Menu"""
665    "       LOCATE 20, 10: INPUT ""ENTER SELECTION AND <RET> : ""; SLO"
666           IF SLO = 7 OR SLO = 0 THEN CLS : SLO = 0: TEMRUT = 0: TEC = 0: GOTO
730
667           IF SLO = 6 THEN GOSUB 6510: GOTO 7580: 'To OUTFILE and back
668           IF SLO = 5 THEN TEC = 1: GOSUB 10010
669           IF SLO = 5.1 THEN FOR I = 1 TO 100: NY(I) = 0: NEXT
670           IF SLO = 5.1 THEN FOR I = 1 TO 100: DYR(I) = 0: NEXT
671           IF SLO = 5.1 THEN FOR I = 1 TO 100: PS(I) = 0: NEXT
672           IF SLO = 5.1 THEN TEMRUT = 1: GOTO 5080
673           PRCHK = 0
674           CLS
675           PRINT : PRINT
676           PRINT
677    "       IF SLO = 1 THEN t$ = ""ELASTIC MODULUS OF BASE COURSE"""
678    "       IF SLO = 2 THEN t$ = ""ELASTIC MODULUS OF SUBGRADE"""
679    "       IF SLO = 3 THEN t$ = ""NUMBER OF PASSES FOR RUT DEPTH"""
680    "       IF SLO = 4 THEN t$ = ""REMAINING LIFE"""
681           MINY = 0
```

```
682   "         IF SLO = 1 THEN INPUT ""Minimum E1  [Default=0]: ""; MINY"
683   "         IF SLO = 1 THEN INPUT ""Maximum E1 [Default=100000]: ""; MAXY"
684             IF (SLO = 1) AND (MAXY = 0) THEN MAXY = 100000
685   "         IF SLO = 2 THEN INPUT ""Minimum E2 [Default= 0   ]: ""; MINY"
686   "         IF SLO = 2 THEN MAXY = 25000: INPUT ""Maximum E2 [Default=25000]: ""; MAXY"
687             IF (SLO = 2) AND (MAXY = 0) THEN MAXY = 25000
688   "         IF SLO = 3 THEN INPUT ""Minimum PASSES [Default=   0 ]: ""; MINY"
689   "         IF SLO = 3 THEN INPUT ""Maximum PASSES [Default=0.5E6]: ""; MAXY"
690             IF (SLO = 3) AND (MAXY = 0) THEN MAXY = 500000
691   "           IF SLO = 4 THEN INPUT ""Minimum REMAINING LIFE [Default=  0]: ""; MINY"
692   "           IF SLO = 4 THEN INPUT ""Maximum REMAINING LIFE [Default= 24]: ""; MAXY"
693             IF (SLO = 4) AND (MAXY = 0) THEN MAXY = 24
694             PR = 1
695             XSCALE = 60 / (MAXX - MINX)
696             YSCALE = 20 / (MAXY - MINY)
697             YORG = 22 + MINY * YSCALE
698             XORG = 10 - MINX * XSCALE
699             KEY OFF
700             CLS
701   "8299      'INPUT ""press return to continue (line8299)""; nn$: CLS"
702   8300      'these are ASCII for the IBM
703              IF PRCHK = 0 THEN PR1 = 196: PR2 = 179: PR3 = 218: PR4 = 191: PR5 = 192
704              IF PRCHK = 0 THEN PR6 = 217: PR7 = 193: PR8 = 195: PR9 = 180: PR10 = 194
705             FOR COL = 11 TO 69
706   "          LOCATE 2, COL: PRINT CHR$(PR1)"
707   "          LOCATE 22, COL: PRINT CHR$(PR1)"
708             NEXT COL
709             FOR ROW = 3 TO 21
710   "          LOCATE ROW, 10: PRINT CHR$(PR2)"
711   "          LOCATE ROW, 70: PRINT CHR$(PR2)"
712             NEXT ROW
713   "          LOCATE 2, 10: PRINT CHR$(PR3)"
714   "          LOCATE 2, 70: PRINT CHR$(PR4)"
715   "          LOCATE 22, 10: PRINT CHR$(PR5)"
716   "          LOCATE 22, 70: PRINT CHR$(PR6)"
717             FOR I = 1 TO 9
718   "          LOCATE 22, (10 + 6 * I): PRINT CHR$(PR7)"
719   "          LOCATE (2 + 2 * I), 10: PRINT CHR$(PR8)"
720   "          LOCATE (2 + 2 * I), 70: PRINT CHR$(PR9)"
721   "          LOCATE 2, (10 + 6 * I): PRINT CHR$(PR10)"
722             NEXT I
723             IF PRCHK = 1 AND PT = 1 THEN GOSUB 9350
724             IF PRCHK = 1 AND PT = 2 THEN GOSUB 9210
725   "          LOCATE 1, 10: PRINT t$"
726   "          LOCATE 1, 55: PRINT x$"
727             XINT = (MAXX - MINX) / 10
728             YINT = (MAXY - MINY) / 10
729   "          LOCATE 23, 8: PRINT (MINX)"
730   "          LOCATE 23, 20: PRINT (MINX + 2 * XINT)"
731   "          LOCATE 23, 32: PRINT (MINX + 4 * XINT)"
732   "          LOCATE 23, 44: PRINT (MINX + 6 * XINT)"
733   "          LOCATE 23, 56: PRINT (MINX + 8 * XINT)"
734   "          LOCATE 23, 69: PRINT (MINX + 10 * XINT)"
```

```
735   "         LOCATE 2, 1: PRINT MAXY"
736   "         LOCATE 6, 1: PRINT INT((MAXY - YINT * 2) * 100) / 100"
737   "         LOCATE 10, 1: PRINT INT((MAXY - YINT * 4) * 100) / 100"
738   "         LOCATE 14, 1: PRINT INT((MAXY - YINT * 6) * 100) / 100"
739   "         LOCATE 18, 1: PRINT INT((MAXY - YINT * 8) * 100) / 100"
740   "         LOCATE 22, 1: PRINT MINY"
741   "         IF SLO = 1 THEN LOCATE 11, 2: PRINT "" E1"""
742   "         IF SLO = 1 THEN LOCATE 12, 2: PRINT ""(psi)"""
743   "         IF SLO = 2 THEN LOCATE 11, 2: PRINT "" E2"""
744   "         IF SLO = 2 THEN LOCATE 12, 2: PRINT ""(psi)"""
745   "         IF SLO = 3 THEN LOCATE 11, 2: PRINT ""No.of"""
746   "         IF SLO = 3 THEN LOCATE 12, 2: PRINT ""Passes"""
747   "         IF SLO = 4 THEN LOCATE 11, 2: PRINT ""Remain."""
748   "         IF SLO = 4 THEN LOCATE 12, 2: PRINT "" Life"""
749   "         IF SLO = 4 THEN LOCATE 13, 2: PRINT ""(years)"""
750   "         IF PRCHK = 1 THEN LOCATE 25, 1: PRINT SPC(70);"
751   "         IF PRCHK = 1 THEN LOCATE 25, 1: PRINT ""RE-PLOTTING ... "";"
752           FOR I = 1 TO NC
753           x = NS(I)
754           IF SLO = 1 THEN Y = E1(I)
755           IF SLO = 1 AND Y > MAXY THEN Y = MAXY
756           IF SLO = 2 THEN Y = E2(I)
757           IF SLO = 2 AND Y > MAXY THEN Y = MAXY
758           IF SLO = 3 THEN Y = PS(I)
759           IF SLO = 3 AND Y > MAXY THEN Y = MAXY
760           IF SLO = 4 THEN Y = NY(I)
761           IF SLO = 4 AND Y > MAXY THEN Y = MAXY
762           SCY = YORG - INT(Y * YSCALE): SCX = XORG + INT(x * XSCALE)
763           IF (SCX < 10) OR (SCX > 70) OR (SCY < 2) OR (SCY > 22) THEN 9050
764   "         LOCATE SCY, SCX: PRINT CHR$(42)"
765   9050    NEXT I
766   "         LOCATE 25, 1: PRINT SPC(75);"
767   "          IF PRCHK = 1 THEN LOCATE 25, 1: INPUT ""PrtSc?  <RET> if NO""; nn$:
CLS : GOTO 7580"
768   "         LOCATE 25, 1: PRINT SPC(75);"
769   "9051     LOCATE 25, 1: PRINT ""PRESS ENTER TO CONTINUE (line9051)""; : INPUT ;
nn$"
770           PRCHK = 1
771           GOTO 8300
772
773           '============================================================
774   9210    ' SUBROUTINE MT180 to replot the graph for the mannesmann tally
775           '============================================================
776           PR1 = 157: PR2 = 156: PR3 = 134: PR4 = 149: PR5 = 153
777           PR6 = 154: PR7 = 158: PR8 = 150: PR9 = 151: PR10 = 152
778           PRCHK = 1: RETURN
779
780           '============================================================
781   9350    ' SUBROUTINE EPSON to replot the graph for the epson/panasonic
782           '============================================================
783           PR1 = 45: PR2 = 124: PR3 = 46: PR4 = 46: PR5 = 46
784           PR6 = 46: PR7 = 43:  PR8 = 43: PR9 = 43: PR10 = 43
785           PRCHK = 1: RETURN
786
787           '============================================================
788   10010   'TEMPERATURE CORRECTION OF BASE MODULUS ONLY
789           '============================================================
790           'THIS SUBROUTINE PREDICTS THE MODULUS OF GRANULAR MATERIALS
```

```
791           'AT DIFFERENT TEMPERATURES AND MOISTURES BASED ON A
792           'MICROMECHANICAL APPROACH
793           CLS
794           PRINT : PRINT : PRINT
795   "        PRINT TAB(25); ""TYPE OF BASE MATERIAL:"": PRINT"
796   "        PRINT TAB(25); ""  1 = CRUSHED LIMESTONE """
797   "        PRINT TAB(25); ""       LIME ORE GRAVEL"""
798   "        PRINT TAB(25); ""       IRON ORE GRAVEL"""
799   "        PRINT TAB(25); ""  2 = RIVER GRAVEL"""
800   "        PRINT TAB(25); ""       CALICHE"""
801   "        PRINT TAB(25); ""       CALICHE GRAVEL"""
802   "        PRINT TAB(25); ""  3 = SANDSHELL"""
803   "        PRINT TAB(25); : INPUT ""SELECT YOUR OPTION ""; MTYPE"
804           IF (MTYPE <= 0) OR (MTYPE > 3) THEN 10010
805           IF MTYPE = 1 THEN ALP = 5 * 10 ^ (-6): E = 6.4 * 10 ^ 6: U = .17: K1 =
14000: K2 = .4
806           IF MTYPE = 2 THEN ALP = 6.5 * 10 ^ (-6): E = 7.8 * 10 ^ 6: U = .2: K1
= 24000: K2 = .37
807           IF MTYPE = 3 THEN ALP = 5 * 10 ^ (-6): E = 8.534001 * 10 ^ 6: U = .17:
K1 = 7210: K2 = .45
808   10260   CLS : PRINT : PRINT
809   "        PRINT TAB(20); ""PROPERTIES OF THE MATERIAL SELECTED:"""
810           PRINT
811   "        PRINT TAB(25); ""1. LINEAR THERMAL EXPANSION        = ""; ALP"
812   "        PRINT TAB(25); ""2. ELASTIC MODULUS OF SOIL GRAINS  = ""; E"
813   "        PRINT TAB(25); ""3. POISSON'S RATIO                 = ""; U"
814           IF SLO = 5 THEN 10350
815   "        PRINT TAB(25); ""4. K1                              = ""; K1"
816   "        PRINT TAB(25); ""5. K2                              = ""; K2"
817   "10350   PRINT : PRINT : PRINT TAB(20); : INPUT ""DO YOU WANT TO CHANGE? 0=NO
1=YES ""; CHOICE"
818           IF (CHOICE <> 1) AND (CHOICE <> 0) THEN 10260
819           IF CHOICE = 0 THEN 10460
820   "        PRINT : PRINT TAB(20); : INPUT ""WHICH LINE ""; WL"
821   "        PRINT TAB(20); : INPUT ""INPUT NEW VALUE ""; NV"
822           IF WL = 1 THEN ALP = NV
823           IF WL = 2 THEN E = NV
824           IF WL = 3 THEN U = NV
825           IF WL = 4 THEN K1 = NV
826           IF WL = 5 THEN K2 = NV
827           GOTO 10260
828
829   10460   CLS
830   "        LOCATE 3, 5: INPUT ""UNIT WEIGHT OF MATERIAL (pcf) :   "", UW"
831           IF (UW < 100) OR (UW > 170) THEN 10460
832           IF SLECT1 = 2 THEN 10478
833           CH01 = 1
834           GOTO 10500
835
836   10478   KK1 = 1
837   "        LOCATE 5, 5: INPUT ""MODULUS OF MATERIAL (psi)          : "", EI"
838           IF (EI < 20000) OR (EI > 6000000) THEN 10478
839   "10500   LOCATE 7, 5: PRINT ""CONDITIONS AT WHICH THE MODULUS IS OBTAINED: """
840   "        LOCATE 9, 5: INPUT ""TEMPERATURE, Ti (40 deg.F < Ti < 110 deg.F)   :
"", TTEMP"
841           IF (TTEMP < 40) OR (TTEMP > 110) THEN 10500
842   "10540   LOCATE 11, 5: INPUT ""SUCTION, Hi (-145 psi < Hi < 0 psi)          :
"", HI"
```

```
843           IF (HI < -145) OR (HI > 0) THEN 10540
844   "10560   LOCATE 13, 5: PRINT ""INPUT WANTED CONDITIONS: """
845   "        LOCATE 15, 5: INPUT ""TEMPERATURE, To (40 deg.F < To < 110 deg.F)   :
"", RTEMP"
846           IF (RTEMP < 40) OR (RTEMP > 110) THEN 10560
847   "10590   LOCATE 17, 5: INPUT ""SUCTION, Ho (-145 psi < Ho < 0 psi)          :
"", HL"
848           IF (HL < -145) OR (HL > 0) THEN 10590
849           IF CHO1 = 1 THEN KK1 = NC
850           FOR I = 1 TO KK1
851           IF CHO1 = 1 THEN EI = E1(I): K1 = K1(I): K2 = .33
852           VOID = 1 / (UW / SG / 62.4) - 1
853           N2 = VOID / (1 + VOID)
854           XS = (.4764 - N2) / (.4764 - .2595)
855           IF XS < 0 THEN XS = 0
856           IF XS > 1 THEN XS = 1
857           PRO1 = .75 * (1 - U ^ 2) / E
858           THETA = 10 ^ (LOG(EI / K1) / LOG(10) / K2)
859           DTEMP = RTEMP - TTEMP
860           IF DTEMP = 0 THEN det(I) = 0: GOTO 10800
861           DT = ABS(DTEMP)
862           DELV = DT * ALP * 3
863            P = (XS / SQR(2) / PRO1 + (1 - XS) / 4 / PRO1) * (1 / 3 * DELV) ^
(1.5)
864           PT1 = (DTEMP / DT) * P
865           det(I) = K1 * K2 * THETA ^ (K2 - 1) * PT1
866   10800   PS = -(HL - HI) * .13
867           des(I) = K1 * K2 * THETA ^ (K2 - 1) * PS
868           TOTDE(I) = det(I) + des(I)
869           NE(I) = EI + TOTDE(I)
870           OE(I) = EI
871           NEXT I
872           CLS
873           PRINT : PRINT : PRINT TAB(10);
874   "        PRINT ""INPUT CONDITIONS:                    WANTED CONDITIONS:"""
875           PRINT
876   "        PRINT USING ""  TEMPERATURE =    ###.#   deg. F        TEMPERATURE =
###.#   deg. F""; TTEMP; RTEMP"
877   "        PRINT USING ""  SUCTION     =  #####.##  psi           SUCTION     =
#####.##  psi    ""; HI; HL"
878           PRINT
879   "        PRINT ""      INPUT     WANTED CONDITION     MODULUS CHANGE    MODULUS
CHANGE"""
880   "        PRINT ""     MODULUS        MODULUS          DUE TO TEMPER.    DUE TO
SUCTION"""
881   "           PRINT ""      (psi)              (psi)                    (psi)
(psi)"""
882           PRINT
883           FOR I = 1 TO KK1
884   "             PRINT USING ""##########.##     ##########.##       ##########.##
##########.##""; OE(I); NE(I); det(I); des(I)"
885           NEXT I
886           PRINT : PRINT
887   "        INPUT ""         PRESS RETURN TO CONTINUE....."""; PRESS"
888           CLS
889           IF SLECT1 <> 2 THEN 10988
890   "        PRINT : PRINT : PRINT ""          1. ANOTHER TEMPERATURE CORRECTION
"""
```

```
891   "         PRINT : PRINT ""          2. EXIT TO MAIN MANUAL """
892   "           PRINT : PRINT : INPUT ""        Select the Operation Desired: "";
CH02"
893           IF CH02 = 1 THEN 10010
894           GOTO 730
895
896   "10988    PRINT : PRINT : PRINT ""          1. EXIT TO MAIN MENU """
897   "         PRINT : PRINT ""          2. EXIT TO OUTPUT MENU """
898   "           PRINT : PRINT : INPUT ""        Select the Operation Desired: "";
CH03"
899           IF CH03 = 1 THEN 730
900           GOTO 7580
901
902           '*****************************************************
903   12000   'RUT DEPTH CALCULATION ONLY
904           '*****************************************************
905           'by K.A.S. Yapa
906   "         ' Pavement Systems, Texas Transportation Institute,"
907   "         ' TTI Building, Texas A & M University,"
908   "         ' College Station, Texas 77843."
909           ' (409)-845-9910.
910           ' 24th MARCH 1988.
911
912           FOR I = 1 TO 2
913           FOR J = 1 TO 3
914           FOR K = 1 TO 3
915           FOR L = 1 TO 3
916   "         READ RUT(I, J, K, L, 1), RUT(I, J, K, L, 2), RUT(I, J, K, L, 3)"
917           NEXT L: NEXT K: NEXT J: NEXT I
918   "         FOR I = 1 TO 3: READ XM(I), XL(I), XK(I), XJ(I), XI(I): NEXT I"
919
920           '============================================================
921   12400   ' LOOP TO CALCULATE RUT DEPTHS FOR EACH FWD SECTION
922           '============================================================
923           GOSUB 14240
924           IF INIOPT = 2 THEN NC = 1
925           FOR INC = 1 TO NC
926           IF TEMRUT = 1 THEN E1(INC) = NE(INC)
927           GOSUB 14850
928           'GOTO INPUT1 IF ANY CORRECTIONS ARE NEEDED
929           IF CORR = 1 THEN GOSUB 14240
930           CLS
931   "         PRINT "" Calculating....."""
932           'SELECT PARAMETERS FOR INTERPOLATION
933           FOR I = 1 TO 2
934           FOR J = 1 TO 3
935           FOR K = 1 TO 3
936           FOR L = 1 TO 3
937           FOR M = 1 TO 3
938
939   "         Y1(M) = RUT(I, J, K, L, M): X1(M) = XM(M): NEXT M"
940           NUM = 3: x = XMM: GOSUB 13700
941           YLTEMP(L) = Y: XMFLAG = XFLAG
942           NEXT L
943
944           FOR LL = 1 TO 3: X1(LL) = XL(LL): Y1(LL) = YLTEMP(LL): NEXT LL
945           NUM = 3: x = XLL: GOSUB 13700
946           YKTEMP(K) = Y: XLFLAG = XFLAG
```

```
947           NEXT K
948
949           FOR KK = 1 TO 3: X1(KK) = XK(KK): Y1(KK) = YKTEMP(KK): NEXT KK
950           NUM = 3: x = XKK: GOSUB 13700
951           YJTEMP(J) = Y: XKFLAG = XFLAG
952           NEXT J
953
954           FOR JJ = 1 TO 3: X1(JJ) = XJ(JJ): Y1(JJ) = YJTEMP(JJ): NEXT JJ
955           NUM = 3: x = XJJ: GOSUB 13700
956           YITEMP(I) = Y: XJFLAG = XFLAG
957           NEXT I
958
959           FOR II = 1 TO 2: X1(II) = XI(II): Y1(II) = YITEMP(II): NEXT II
960           NUM = 2: x = XII: GOSUB 13700
961           RUTCAL = Y * CYL / 300000: XIFLAG = XFLAG
962
963           'GOTO SUBROUTINE OUTPUT
964           GOSUB 15680
965           NEXT INC
966           IF INIOPT <> 2 THEN RETURN
967           PRINT : PRINT : PRINT
968   "        PRINT ""            1. ANOTHER RUT DEPTH CALCULATION """
969   "        PRINT : PRINT ""            2. EXIT TO MAIN MANUAL """
970   "          PRINT : PRINT : PRINT : INPUT ""            Select the Operation
Desired: ""; CHO5"
971           IF CHO5 = 1 THEN 12400
972           GOTO 730
973
974           'DATA BASE OF RUT DEPTHS
975   "        DATA 0.2609, 0.2804, 0.3191, 0.2216, 0.2152, 0.2223, 0.1831, 0.1502,
0.1253"
976   "        DATA 0.2705, 0.3030, 0.3431, 0.2451, 0.2491, 0.2556, 0.2168, 0.1951,
0.1680"
977   "        DATA 0.367 , 0.4519, 0.5286, 0.3503, 0.4172, 0.4580, 0.3331, 0.3831,
0.3882"
978   "        DATA 0.2702, 0.2912, 0.3019, 0.2249, 0.2185, 0.2002, 0.1797, 0.1458,
0.0991"
979   "        DATA 0.2804, 0.3099, 0.3549, 0.2473, 0.2490, 0.2600, 0.2137, 0.1882,
0.1652"
980   "        DATA 0.3900, 0.4610, 0.5322, 0.3688, 0.4231, 0.4561, 0.3475, 0.3842,
0.3798"
981   "        DATA 0.2946, 0.3130, 0.3110, 0.2344, 0.2263, 0.2031, 0.1750, 0.1403,
0.0837"
982   "        DATA 0.3013, 0.3356, 0.3755, 0.2548, 0.2600, 0.2687, 0.2087, 0.1842,
0.1618"
983   "        DATA 0.3977, 0.4840, 0.5424, 0.3750, 0.4343, 0.4541, 0.3498, 0.3834,
0.3670"
984   "        DATA 0.1613, 0.2376, 0.3056, 0.1258, 0.1727, 0.2086, 0.0934, 0.1082,
0.1119"
985   "        DATA 0.1901, 0.2528, 0.3247, 0.1588, 0.1993, 0.2374, 0.1280, 0.1457,
0.1500"
986   "        DATA 0.2607, 0.3681, 0.4915, 0.2453, 0.3355, 0.4217, 0.2300, 0.3038,
0.3523"
987   "        DATA 0.1951, 0.2524, 0.2767, 0.1491, 0.1794, 0.1747, 0.1039, 0.1072,
0.0732"
988   "        DATA 0.1957, 0.2643, 0.3400, 0.1628, 0.2037, 0.2453, 0.1296, 0.1433,
0.1505"
989   "        DATA 0.2920, 0.4006, 0.5025, 0.2713, 0.3648, 0.4262, 0.2508, 0.3230,
```

```
0.3502"
990   "         DATA 0.2254, 0.2785, 0.2834, 0.1652, 0.1918, 0.1736, 0.1056, 0.1059,
0.0643"
991   "         DATA 0.2256, 0.2974, 0.3642, 0.1790, 0.2216, 0.2574, 0.1328, 0.1461,
0.1506"
992   "         DATA 0.3322, 0.4287, 0.5197, 0.3034, 0.3786, 0.4327, 0.2749, 0.3287,
0.3456"
993           'INPUT PARAMETERS USED IN CREATING THE DATA BASE
994           '    XM(#) XL(#)  XK(#)   XJ(#)   XI(#)
995   "         DATA 18, 0.0100, 25000, 100000, 0.0075"
996   "         DATA 12, 0.0060, 15000,  70000, 0.0025"
997   "         DATA  6, 0.0020,  5000,  40000, 0.0000"
998           END
999
1000          '============================================================
1001  13700   'SUBROUTINE FOR POLYNOMIAL INTERPOLATION
1002          '============================================================
1003          NS = 1
1004          DIF = ABS(x - X1(1))
1005          FOR a = 1 TO NUM
1006          DIFT = ABS(x - X1(a))
1007          'SELECT THE BEST STARTING POINT
1008          IF DIFT < DIF THEN NS = a: DIF = DIFT
1009          C(a) = Y1(a): DD(a) = Y1(a)
1010          NEXT a
1011          XFLAG = 0
1012           'ENFORCE LIMITS ON EXTRAPOLATION (MAXIMUM = 1.5 * DIFFERENCE BETWEEN
TWO
1013          'CONSECUTIVE PARAMETER LEVELS)
1014          IF DIF <= 1.5 * (ABS(X1(1) - X1(2))) THEN 13980
1015  13940   DIF = 1.5 * (ABS(X1(1) - X1(2)))
1016          IF NS = 1 THEN x = X1(1) + DIF
1017          IF NS = NUM THEN x = X1(NUM) - DIF
1018          XFLAG = x
1019  13980   Y = Y1(NS)
1020          NS = NS - 1
1021          AEND = NUM - 1
1022          FOR a = 1 TO AEND
1023          HO = X1(a) - x
1024          HP = X1(a + 1) - x
1025          W = C(a + 1) - DD(a)
1026          DEN = W / (HO - HP)
1027          DD(a) = HP * DEN
1028          C(a) = HO * DEN
1029          NEXT a
1030          'PICK THE SHORTEST PATH TO MOVE
1031          IF (2 * NS) < AEND THEN DY = C(NS + 1): GOTO 14210
1032  14200   DY = DD(NS): NS = NS - 1
1033  14210   Y = Y + DY
1034          RETURN
1035
1036          '============================================================
1037  14240   ' SUBROUTINE INPUT1
1038          '============================================================
1039          'USE IF ONLY THE RUT LEVELS ARE NEEDED
1040          IF INIOPT <> 2 GOTO 14400
1041  "        CLS : INPUT ""JOB DESCRIPTION :""; AA$"
1042  "        INPUT ""Resilient Modulus - Base Course (psi)""; EBA"
```

```
1043  “         INPUT “”Resilient Modulus - Subgrade (psi)””; ESG”
1044  “         INPUT “”Thickness of Base Layer (in)””; TBA”
1045  “         INPUT “”# of Equivalent Standard Wheel (9000 lbs) Passes””; EQPASS”
1046  “         INPUT “”Allowable Rut Depth (in.)””; RALLOW”
1047  “         INPUT “”Existing Rut Depth (in.)””; REXIST”
1048  14400    CLS
1049  “         LOCATE 4, 10: PRINT “” INPUT DATA OPTIONS FOR RUT DEPTH CALCULATION:
“””
1050  “         LOCATE 6, 10: PRINT “” 1) Require subgrade material”””
1051  “         LOCATE 7, 10: PRINT “”    classification to determine approximate “””
1052  “         LOCATE 8, 10: PRINT “”    rutting potentials.”””
1053  “            LOCATE 10, 10: PRINT “” 2) Require laboratory data on residual
deforma- “””
1054  “         LOCATE 11, 10: PRINT “”    tion behavior of base and subgrade.”””
1055  “         LOCATE 15, 10: INPUT “”OPTION: 1=SOIL CLASS, 2=LAB DATA “”; OPP”
1056  14405    IF OPP = 2 THEN 14590
1057           IF OPP <> 1 THEN 14400
1058           'ALL BASE COURSE MATERIALS ARE CLASSIFIED AS ONE GROUP.
1059  “         CLS : LOCATE 4, 10: PRINT “”Subgrade Material Type :”””
1060  “         LOCATE 7, 10: PRINT “”1) Heavy Clay - (CH)”””
1061  “         LOCATE 9, 10: PRINT “”2) Light/Silty Clay, Clayey Silt - (CL-ML)”””
1062  “         LOCATE 11, 10: PRINT “”3) Clayey/Silty/Uniform Sand - (SC-SM)”””
1063  “         LOCATE 15, 10: INPUT “”ENTER SELECTION & <RET> “”; MSG”
1064           GOTO 14750
1065
1066  “14590    CLS : LOCATE 4, 10: PRINT “”Laboratory Data Input:”””
1067  “         LOCATE 6, 10: PRINT “”Log a - Intercept of the Straight Line Fit on a
“””
1068  “          LOCATE 7, 10: PRINT “”          Log-Log Plot of Accumulated Residual
Strain”””
1069  “         LOCATE 8, 10: PRINT “”         vs. Number of Load Repetitions”””
1070  “         LOCATE 10, 10: PRINT “”b      - Slope of the Straight Line Fit”””
1071           PRINT : PRINT
1072  “         INPUT “”Log a - Base Course Material””; LGABA”
1073  “         INPUT “”b     - Base Course Material””; BBA”
1074  “         INPUT “”Log a - Subgrade Material””; LGASG”
1075  “         INPUT “”b     - Subgrade Material””; BSG”
1076           IF INIOPT = 2 GOTO 14718
1077           CLS
1078           IF OPP <> 1 GOTO 14704
1079  “         LOCATE 16, 10: IF MSG = 1 THEN PRINT “”Subgrade      - CH - Clay”””
1080  “         LOCATE 16, 10: IF MSG = 2 THEN PRINT “”Subgrade      - CL-ML “””
1081  “         LOCATE 16, 10: IF MSG = 3 THEN PRINT “”Subgrade      - SC-SM”””
1082           GOTO 14714
1083
1084  “14704    LOCATE 16, 10: PRINT “”Base Course Material - “””
1085  “         LOCATE 17, 10: PRINT “”            Log a = “”; LGABA; “”,     b = “”;
BBA”
1086  “         LOCATE 19, 10: PRINT “”Subgrade Material    - “””
1087  “         LOCATE 20, 10: PRINT “”            Log a = “”; LGASG; “”,     b = “”;
BSG”
1088  “14714    LOCATE 22, 10: INPUT “”DO YOU WANT ANY CORRECTIONS - 0=NO, 1=YES””;
CORR”
1089           IF CORR = 1 THEN 14405
1090           IF CORR <> 0 GOTO 14714
1091  14718    ABA = 10 ^ LGABA: ASG = 10 ^ LGASG
1092           IF ((ABA * 50000 ^ BBA) < .015) AND ((ASG * 50000 ^ BSG) < .016) GOTO
14750
```

```
1093  "        CLS : BEEP: PRINT ""Input Data are Incompatible. Check & Re-enter"":"
1094  "        INPUT ""Press RETURN to continue""; nn$"
1095          GOTO 14405
1096
1097  14750   'SKIP IF ONLY RUT LEVELS ARE CALCULATED
1098          IF INIOPT = 2 GOTO 14840
1099          'OBTAIN THE TRUCK PASSES AND ALLOWABLE AND MEASURED RUT LEVELS FROM
1100          'THE MAIN PROGRAM
1101          EQPASS = PA
1102          RALLOW = RX
1103          REXIST = RM
1104  14840   RETURN
1105
1106          '============================================================
1107  14850   'SUBROUTINE INPUT2
1108          '============================================================
1109          'SKIP IF ONLY RUT LEVELS ARE CALCULATED
1110          IF INIOPT = 2 GOTO 14940
1111          EBA = E1(INC)
1112          ESG = E2(INC)
1113          TBA = BA(INC)
1114  14940   'APPROXIMATE METHOD TO DETERMINE RUTTING BEHAVIOR
1115          IF OPP <> 1 GOTO 15280
1116          AABA = .0174: AACH = .0933: AACL = .001: AASM = .095
1117          ABBA = -.57: ABCH = -2.64: ABCL = -.73: ABSM = -1.95
1118          BBA = .125: BCH = .236: BCL = .162: BSM = .142
1119          ABA = AABA * (EBA / 1000) ^ ABBA
1120          IF ABA > .0035 THEN ABA = .0035
1121          IF MSG <> 1 GOTO 15190
1122          BSG = BCH
1123          ASG = AACH * (ESG / 1000) ^ ABCH
1124          IF ASG > .004 THEN ASG = .004
1125  15190   IF MSG <> 2 GOTO 15230
1126          BSG = BCL
1127          ASG = AACL * (ESG / 1000) ^ ABCL
1128          IF ASG > .001 THEN ASG = .001
1129  15230   IF MSG <> 3 GOTO 15280
1130          BSG = BSM
1131          ASG = AASM * (ESG / 1000) ^ ABSM
1132          IF ASG > .005 THEN ASG = .005
1133  15280   'USE IF ONLY RUT LEVELS ARE NEEDED
1134          IF INIOPT <> 2 GOTO 15550
1135          CLS
1136  "        LOCATE 2, 10: PRINT ""JOB : ""; AA$"
1137  "        LOCATE 4, 10: PRINT ""Resilient Modulus (psi) - Base     = ""; EBA"
1138  "        LOCATE 6, 10: PRINT ""                        - Subgrade = ""; ESG"
1139  "        LOCATE 8, 10: PRINT ""Thickness of Base Layer             = ""; TBA;
"" in."""
1140  "          LOCATE 10, 10: PRINT ""# of Equivalent Standard Wheel Passes =""; 
EQPASS"
1141  "        LOCATE 12, 10: PRINT ""Allowable Rut Depth =""; RALLOW; "" in."""
1142  "        LOCATE 14, 10: PRINT ""Measured Rut Depth =""; REXIST; "" in."""
1143          IF OPP <> 1 GOTO 15460
1144  "        LOCATE 16, 10: IF MSG = 1 THEN PRINT ""Subgrade      - CH - Clay"""
1145  "        LOCATE 16, 10: IF MSG = 2 THEN PRINT ""Subgrade      - CL-ML """
1146  "        LOCATE 16, 10: IF MSG = 3 THEN PRINT ""Subgrade      - SC-SM"""
1147          GOTO 15510
1148
```

```
1149  "15460   LOCATE 16, 10: PRINT ""Base Course Material - """
1150  "        LOCATE 17, 10: PRINT ""            Log a = ""; LGABA; "",     b = ""; BBA"
1151  "        LOCATE 19, 10: PRINT ""Subgrade Material    - """
1152  "        LOCATE 20, 10: PRINT ""            Log a = ""; LGASG; "",     b = ""; BS"
1153  "15510    LOCATE 22, 10: INPUT ""DO YOU WANT ANY CORRECTIONS - 0=NO, 1=YES""; CORR"
1154          IF CORR = 1 THEN RETURN
1155          IF CORR <> 0 GOTO 15510
1156  15550   'ASSIGN VALUES TO INTERPOLATION PARAMETERS
1157          CYL = 300000
1158          XII = ABA * CYL ^ BBA
1159          XLL = ASG * CYL ^ BSG
1160          IF XII > .015 THEN XII = .015
1161          IF XLL > .016 THEN XLL = .016
1162          IF XLL < .001 AND TBA < 10 THEN XLL = .001
1163          XMM = TBA: XJJ = EBA: XKK = ESG
1164          IF TBA < 10 AND ESG > 20000 AND EBA > 100000 THEN XJJ = 100000
1165          IF TBA < 10 AND EBA < 60000 AND ESG > 30000 THEN XKK = 30000: IF TBA < 6 THEN XMM = 6: IF EBA < 30000 THEN XJJ = 30000
1166          RETURN
1167
1168          '============================================================
1169  15680   'SUBROUTINE OUTPUT
1170          '============================================================
1171          IF RUTCAL < 0 THEN RUTCAL = 0
1172          RFINAL(INC) = RUTCAL * EQPASS / CYL
1173          REXTRA = RALLOW - REXIST
1174          IF RUTCAL = 0 THEN passes(INC) = 0: GOTO 15760
1175          passes(INC) = CYL * REXTRA / RUTCAL
1176  15760   IF INIOPT <> 2 GOTO 15860
1177          CLS
1178          IF EQPASS = 0 GOTO 15810
1179  "        LOCATE 6, 10: PRINT ""EQUIVALENT STANDARD WHEEL PASSES =""; EQPASS"
1180  "         LOCATE 8, 10: PRINT ""RUT DEPTH CAUSED         = ""; RFINAL(1); "" in."""
1181  "15810    LOCATE 10, 10: PRINT ""ALLOWABLE RUT DEPTH       = ""; RALLOW; "" in."""
1182  "          LOCATE 12, 10: PRINT ""EXISTING RUT DEPTH       = ""; REXIST; "" in."""
1183  "        LOCATE 14, 10: PRINT ""ALLOWABLE WHEEL PASSES  = ""; passes(1)"
1184  "           IF passes(1) = 0 THEN LOCATE 16, 5: PRINT ""INPUT DATA MAY BE INCOMPATIBLE. CHECK & RERUN!"""
1185          PRINT : PRINT : PRINT
1186  "15860   IF INIOPT = 2 THEN INPUT ""      PRESS RETURN TO CONTINUE ......."; PRESS"
1187          CLS
1188          RETURN
```

Original Loadrate document

```
1       '*****************$$$ PROGRAM - LOADRATE - $$$*******************
2       '*                     Version  AUGUST 1989                     *
3       '*           Load Rating of Light Pavement Structures           *
4       '*                      WRITTEN BY K. M. CHUA                   *
```

```
5       '*                          REVISED BY                              *
6       '*                       B. LANKA SANTHA                            *
7  "    '*          Pavement Systems, Texas Transportation Institute        *"
8  "    '*              TTI Building, Texas A&M University,                 *"
9  "    '*                 College Station, Texas 77843                     *"
10      '*                       (409)-845-5982                             *
11      '*                     Update 20th August 1989                      *
12      '*****************************************************************
13     'Dimensioning ————————————————————————
14     '
15     '
16     '$DYNAMIC
17 "    DIM D(160, 7), SO(160), DE(160, 7)"
18 "    DIM AC(160), PFW(160), K1(160), W(160), BC(160), K1S(160)"
19 "    DIM WL(160, 8), AM(160), PS(160): DIM SS(160), CK$(160)"
20 "    DIM NS(160), H(160), SI(160), BA(160), NX(160)"
21 "    DIM SSO(160), BBA(160), WW(160), WP(160), WOS(160)"
22 "    DIM E1(160), E2(160), SL(160), Y(160), D7(160)"
23 "    DIM DSUM(10), JSUM(10), FSUM(10), DF(10), FW(10), R(10), PRESS(10)"
24 "    DIM WLOAD(10), X(10), DYR(160)"
25 "    DIM DET(160), DES(160), NE(160), OE(160), TOTDE(160)"
26 "    DIM RFINAL(160), passes(160), NY(160)"
27 "    DIM XM(160), XL(160), XK(160), XJ(160), XI(160)"
28 "    DIM Y1(5), X1(5), YLTEMP(5), YKTEMP(5)"
29 "    DIM YJTEMP(5), YITEMP(5), C(5), DD(5)"
30     REM To initialize FWD sensor spacing
31     X(1) = .01: X(2) = 12: X(3) = 24: X(4) = 36: X(5) = 48: X(6) = 60: X(7) = 72
32 730 GOSUB 6585
33     REM SELECT TEMP. CORRECTION OR STRUCTURAL ANALYSIS OR RUT DEPTH CALCULATION
34     IF SLECT1 = 2 THEN : CH01 = 0: GOSUB 10010
35     IF SLECT1 = 3 THEN : IF RUT1 <> 1 THEN : INIOPT = 2: GOSUB 12000
36     IF SLECT1 = 3 THEN : IF RUT1 = 1 THEN : INIOPT = 2: GOSUB 12400
37     IF SLECT1 = 4 THEN : CLS : GOTO 5980
38     'GOSUB 6950: REM SELECTION TABLE - MAIN MENU
39     '
40       FOR I = 1 TO 160
41         NY(I) = 0
42         DYR(I) = 0
43         PS(I) = 0
44       NEXT I
45     REM MAIN PROGRAM ————————————————————————
46     'Initializing
47         CO = 0: OO = 0
48     '
49 "     CLS : PRINT : PRINT ""    PROGRAM : LOADRATE  Version 2.0 August 1989"""
50      PRINT : PRINT
51      GOTO 1790
52 1440 IF OP <> 1 AND OO <> 1 THEN : GOTO 1740
53      IF OO = 1 THEN : OO = 0: GOTO 3570
54      '
55 1740 IF OO <> 0 THEN GOTO 3080
56      GOTO 2290
57      '
58 1790 'To read from file ————————————————————
59 "     PRINT ""     FILE NAME"""
60      INPUT X$
61 "     PRINT ""Please Wait ....."""
62 "     OPEN ""I"", #1, X$"
```

```
63 "         INPUT #1, EQ"
64 "         INPUT #1, ID"
65 "         INPUT #1, C$"
66 "         INPUT #1, F$"
67 "      INPUT #1, D$"
68 "      INPUT #1, a$"
69 "      INPUT #1, NC, NT, RX, RM, AL, PA, GR, ESAL1"
70       FOR NTT = 1 TO NT
71 "         INPUT #1, NX(NTT)"
72          FOR I = 1 TO NX(NTT)
73 "            INPUT #1, WL(NTT, I)"
74            NEXT I
75          NEXT
76       FOR I = 1 TO NC
77 "         IF EQ = 0 THEN : INPUT #1, NS(I), BA(I), D(I, 1), D(I, 2), D(I, 3), D(I,
4), D(I, 5)"
78 "         IF EQ = 1 THEN : INPUT #1, NS(I), BA(I), D(I, 1), D(I, 2), D(I, 3)"
79 "         IF EQ = 1 THEN : INPUT #1, D(I, 4), D(I, 5), D(I, 6), D(I, 7), PFW(I)"
80          NEXT
81       CLOSE
82       GOTO 2290
83       '
84       '
85 2290 REM To print input card images ----------------------
86       '
87       '
88 "2320 OPEN ""SCRN:"" FOR OUTPUT AS #1"
89       REM ITERATE FOR THE NUMBER OF TRUCKS CONSIDERED
90 2340 FOR ITE = 1 TO NT
91 "         IF ITE > 1 THEN : PRINT #1, : GOTO 2440"
92 "         CLS : PRINT #1, ""TEXAS TRANSPORTATION INSTITUTE"""
93 "         PRINT #1, ""LOAD RATING OF LIGHT PAVEMENT"""
94 "         PRINT #1,"
95 "         PRINT #1, ""JOB : ""; a$; ""          (INPUT FILE ""; X$; "")"""
96 "         PRINT #1,"
97 "         PRINT #1, ""DISTRICT:""; ID; ""     COUNTY:""; C$; ""     ROAD:""; F$"
98 "         PRINT #1, ""ALLOWABLE RUT(INS):""; RX; ""     RECORDED RUT(INS):""; RM"
99 "         PRINT #1,"
100   2440    NTT = ITE
101   "         PRINT #1, ""TRUCK NO. ""; NTT"
102   "         PRINT #1, ""AXLE NUMBER   SINGLE WHEEL/ESWL(LBS)"""
103         '
104         '
105         ' To loop of the number of axles NX and number of trucks NTT
106            FOR I = 1 TO NX(NTT)
107   "             PRINT #1, TAB(5); I; TAB(20); WL(NTT, I)"
108              NEXT I
109   "         PRINT #1,"
110         '
111         '
112         'For more than one truck considered
113            IF ITE > 1 THEN : GOTO 3010
114   "         PRINT #1, ""ANNUAL TRAFFIC GROWTH RATE:""; GR; ""  # OF YEARS:""; AL;
""  FIRST YEAR TRAFFIC:""; ESAL1"
115   "         PRINT #1, ""TOTAL NUMBER OF PASSES DURING ABOVE PERIOD:""; PA"
116   "         IF EQ = 0 THEN : PRINT #1, ""DATE:""; D$; ""     DYNAFLECT"""
117   "            IF EQ = 1 THEN : PRINT #1, ""DATE:""; D$; ""    FALLING WEIGHT
DEFLECTOMETER"""
```

```
118           REM CONTINUE
119   "        PRINT #1,"
120   "         PRINT #1, ""SECTION    BASE                                    DEFLEC-
TION"""
121   "                IF  EQ  =  0  THEN  :  PRINT  #1,  ""    NO.      THICKNESS
(MILS) """
122   "                IF  EQ  =  1  THEN  :  PRINT  #1,  ""    NO.      THICKNESS
(MILS)                    LOAD"""
123   "        IF EQ = 0 THEN : PRINT #1, ""          (INS)      W1       W2       W3
W4      W5      """
124   "        IF EQ = 1 THEN : PRINT #1, ""          (INS)      W1       W2       W3
W4      W5       W6       W7    (LBS)"""
125   "        IF EQ = 1 THEN : PRINT #1, ""(RADIAL DISTANCE)"";"
126   "         IF EQ = 1 THEN : PRINT #1, USING ""  ###.#    ###.#    ###.#    ###.#
###.#    ###.#    ###.#"""; X(1), X(2), X(3), X(4), X(5), X(6), X(7)"
127   "        PRINT #1,"
128          FOR I = 1 TO NC
129          '
130          'CALCULATION OF BASE THICKNESS
131          IF TBC = 1 THEN : BA(I) = NH
132          IF BA(I) <> 0 THEN : GOTO 2890
133          '
134   "        WW = (DE(I, 1) - DE(I, 7)) / DE(I, 7)"
135          BA(I) = (SSO(I) / 1958 * WW) ^ 1.85
136          BA(I) = INT((BA(I) * 10 + 5) / 10)
137          IF NH = 0 THEN : PPR = 0
138          IF PPR = 2 THEN : GOTO 3000
139       REM To print image for Dynaflect ————————————————
140   2890 '
141       REM Check if hardcopy is needed
142   "            IF EQ = 0 THEN : PRINT #1, NS(I); TAB(12); BA(I); TAB(23);"
143   "             IF EQ = 0 THEN : PRINT #1, USING ""#####.##""; D(I, 1), D(I, 2),
D(I, 3), D(I, 4), D(I, 5)"
144       '
145       REM To print image for Falling Weight Deflectometer
146       '
147   "            IF EQ = 1 THEN : PRINT #1, NS(I); TAB(12); BA(I); TAB(19);"
148   "            IF EQ = 1 THEN : PRINT #1, USING ""###.##  ###.##  ###.##  ###.##
###.##  ###.##  ###.## #######""; D(I, 1), D(I, 2), D(I, 3), D(I, 4), D(I, 5), D(I,
6), D(I, 7); PFW(I)"
149   "     '      IF EQ = 1 THEN : PRINT #1,"
150   3000        NEXT
151   "     PRINT #1,"
152   3010    CLOSE
153       '
154       '
155       REM PR is 1 indicates that hardcopy is required
156          IF PR = 1 THEN GOTO 3650
157       '
158       REM OO is 1 indicates that all corrections had been done
159   3080    'IF OO=1 THEN GOTO 3570
160       '
161       '
162       '————————————————————————————————
163       '
164   "         LOCATE 25, 1: PRINT SPC(132);"
165       '
166       '
```

```
167         '
168            IF CO = 0 AND OO = 0 THEN : GOTO 3570
169            IF CO = 0 AND OO = 1 THEN : GOTO 3530
170          '
171          '
172    3530 IF OP <> 0 THEN : CLOSE : GOTO 1440
173         '
174    3570 'Check point for printing ————————————————————
175         '
176            IF ITE <> 1 THEN : GOTO 3640
177    "                        IF   TBC   =   0   THEN   :   LOCATE   25,   1:
INPUT ""DO YOU WANT HARDCOPY? 0=NO 1=YES 2=RESULTS ONLY""; PR"
178    "          IF TBC = 1 THEN : LOCATE 25, 1: INPUT ""DO YOU WANT HARDCOPY? 0=NO
1=YES ""; PR"
179    "                                  LOCATE    25,    1:         PRINT    ""
"""
180            IF PR = 2 THEN : PR = 1: PPR = 2: GOTO 3670
181    "3640    IF PR = 1 THEN : CLOSE : OPEN ""LPT1:"" FOR RANDOM AS #1: GOTO 2340"
182    3650 REM Computation for Dynaflect readings   EQ is 0
183         REM Computation for the FWD              EQ is 1
184    3670 REM CONTINUE
185            IF ITE <> 1 THEN : GOTO 4790
186         IF DESIGN = 1 THEN : GOTO 4340
187            IF EQ = 0 THEN : PL = 1000
188            IF EQ = 1 THEN : PL = 10956.3
189            FOR I = 1 TO NC
190    "             IF EQ = 1 THEN : SO(I) = PFW(I) / D(I, 1)"
191    "             IF EQ = 1 THEN : DE(I, 1) = PL / SO(I)"
192    "     REM D(I,7) may not be taken at 94.5"" then need to change"
193            IF (EQ = 1) THEN : GOSUB 7290: 'GEOMBASIN
194    "         IF (EQ = 1) THEN : DE(I, 7) = D7(I) * PL / PFW(I)"
195            REM D7(I) IS FROM SUBROUTINE GEOMBASIN
196         '
197         '
198         '
199         REM —— To convert Dynaflect readings to DE ————————
200    "             IF EQ = 0 THEN : DE(I, 7) = 3.38075 * D(I, 5) ^ .639462"
201    "             IF EQ = 0 THEN : DE(I, 7) = 4.5688721# * D(I, 5) ^ .578444"
202    "             IF EQ = 0 THEN : DE(I, 1) = -7.24474 + (29.6906 * D(I, 1))"
203         REM To obtain stiffnesses
204    "               IF EQ = 0 THEN : SO(I) = 86.0122 * EXP(.00187211# * PL / D(I,
1))"
205                  IF TBC = 0 THEN : SSO(I) = SO(I)
206         '
207         '
208     'SI — is used only when low overall stiffness is encountered
209                 SI(I) = -109.663 + 1.31393 * SO(I)
210                 IF SO(I) <= 83.46183 THEN : SI(I) = SO(I)
211                 H(I) = BA(I)
212                 NEXT
213         '
214         '
215         '
216         REM To determine the type of subgrade ————————————
217    "     PRINT ""Please Wait ..... """
218         '
219         REM To convert readings to standard deflection DE ————————
220         FOR I = 1 TO NC
```

```
221         '
222         '
223         REM To determine subgrade K1S-value
224    "       X = BA(I): Y = DE(I, 7): IF X > 18 THEN : X = 18"
225    "        K1S(I) = (9555.651 + 370.3322 * X) * DE(I, 7) ^ (-1.21665 + .016349 *
X)"
226           IF K1S(I) < 5450 THEN : K1S(I) = 5450
227           IF K1S(I) > 62200 THEN : K1S(I) = 62200
228           LL1 = .4342945 * LOG(K1S(I))
229           A11 = 10 ^ (10.50698 - 1.97986 * LL1)
230           A12 = 361.506 - 2131.05 / (K1S(I) * 10 ^ -3) - 18.305 * K1S(I) * 10 ^ -
3 + .36743 * (K1S(I) * 10 ^ -3) ^ 2 - .00256 * (K1S(I) * 10 ^ -3) ^ 3
231           A13 = -13.6825 + 88.26011 / (K1S(I) * 10 ^ -3) + 1.165388 * K1S(I) * 10
^ -3 - .0236 * (K1S(I) * 10 ^ -3) ^ 2 + .00017 * (K1S(I) * 10 ^ -3) ^ 3
232            A1 = A11 + A12 * BA(I) + A13 * BA(I) * BA(I)
233           B11 = .032285 - .109 / (K1S(I) * 10 ^ -4)
234         B12 = .123403 - .03883 * LL1
235         B13 = -.00426 + .001256 * LL1
236         B1 = B11 + B12 * BA(I) + B13 * BA(I) * BA(I)
237    "       K1(I) = (DE(I, 1) / A1) ^ (1 / B1)"
238          IF K1(I) <= 3000 THEN : K1(I) = 3000
239           IF K1(I) > 60000! THEN : K1(I) = 60000!
240          NEXT I
241         'To determine material properties ————————————————
242         '
243         '
244    4340 'FOR I = 1 TO NC
245    4790 '
246         AV = 0
247         '
248         'To calculate elastic modulus ECAL ————————————————
249         FOR I = 1 TO NC
250         GOSUB 6600
251         NEXT I
252         '
253    5080 'To calculate the rut depth and number of passes
254         IF RUT1 = 1 THEN : INIOPT = 3: GOSUB 12400
255         IF RUT1 <> 1 THEN : INIOPT = 3: GOSUB 12000
256         '
257         FOR I = 1 TO NC
258           PS(I) = passes(I)
259           AM(I) = RFINAL(I)
260           IF GR < 0 THEN : GOTO 5097
261         ' REMAINING LIFE
262           NY(I) = (LOG(PS(I) * GR + ESAL1) - LOG(ESAL1)) / LOG(1 + GR)
263           TOTYR = TOTYR + NY(I)
264           IF NY(I) > NYMAX THEN : NYMAX = NY(I)
265    5097    AV = AV + PS(I)
266         '
267         NEXT I
268         IF GR < 0 THEN : GOTO 5100
269         '
270         FOR II = 1 TO NYMAX
271           FOR I = 1 TO NC
272             IF NY(I) >= II THEN : DYR(II) = DYR(II) + 1
273           NEXT I
274           DYR(II) = (DYR(II) / NC) * 100
275           IF DYR(II) = 100 THEN : S100 = II
```

```
276         NEXT II
277         'CALCULATIONS OF ALLOWBLE AXLE LOADS
278           AL1 = ((S100 / 10) ^ .2432) * 18
279           AL2 = ((S100 / 10) ^ .2467) * 33.5
280           AL3 = ((S100 / 10) ^ .2534) * 48.25
281           IF AL1 > 18 THEN : AL1 = 18
282           IF AL2 > 34 THEN : AL2 = 34
283           IF AL3 > 46 THEN : AL3 = 46
284         'FOR PLOTS
285         MINX = NS(1)
286         MAXX = NS(NC)
287    5100 'To print results ------------------------------
288    "     IF PR = 1 THEN : OPEN ""LPT1:"" FOR RANDOM AS #1: CLS : PRINT ""Printing
.....""
289    "     IF PR = 0 THEN : OPEN ""SCRN:"" FOR OUTPUT AS #1: CLS"
290         '
291    "        IF GR < 0 THEN : PRINT #1, "" SECTION          LAYER PROPERTIES
NO. OF      RUT"""
292         '
293    "        IF GR < 0 THEN : PRINT #1, ""     NO.             ELASTIC MODULUS
ALLOWABLE   DEPTH"""
294    "     IF (EQ = 0) AND (GR < 0) THEN : PRINT #1, ""             E1-BASE     E2-
SUBGR                  PASSES     (IN)"""
295    "      IF (EQ = 1) AND (GR < 0) THEN : PRINT #1, ""                BASE/SUBB
SUBGRADE                   PASSES     (IN)   """
296         '
297    "        IF GR >= 0 THEN : PRINT #1, "" SECTION          LAYER PROPERTIES
REMAINING       NO. OF      RUT"""
298         '
299    "        IF GR >= 0 THEN : PRINT #1, ""     NO.             ELASTIC MODULUS
LIFE         ALLOWABLE   DEPTH"""
300    "     IF (EQ = 0) AND (GR >= 0) THEN : PRINT #1, ""             E1-BASE     E2-
SUBGR      (YEARS)        PASSES     (IN)"""
301    "      IF (EQ = 1) AND (GR >= 0) THEN : PRINT #1, ""                BASE/SUBB
SUBGRADE      (YEARS)        PASSES     (IN)   """
302    "     'PRINT #1,"
303         '
304         '
305         REM CONTINUE
306         FOR I = 1 TO NC
307         '
308    "          IF (EQ = 1) AND (NH <> 0) THEN : PRINT #1, NS(I); TAB(8); E1(I);
TAB(19); E2(I); TAB(29); SS(I); TAB(39); BC(I); TAB(54); PS(I);"
309    "         IF (EQ = 1) AND (NH <> 0) THEN : PRINT #1, TAB(64); AM(I); TAB(74);
""DESIGN"""
310    "        IF (EQ = 1) AND (NH = 0) THEN : PRINT #1, TAB(3); NS(I);"
311    "        IF (EQ = 1) AND (NH = 0) THEN : PRINT #1, TAB(13); USING ""########"";
E1(I);"
312    "        IF (EQ = 1) AND (NH = 0) THEN : PRINT #1, TAB(25); USING ""######"";
E2(I);"
313    "          IF (EQ = 1) AND (NH = 0) AND (GR >= 0) THEN : PRINT #1, TAB(39);
USING ""###.#""; NY(I);"
314    "             IF (EQ = 1) AND (NH = 0) THEN : PRINT #1, TAB(51); USING
""##########""; PS(I);"
315    "        IF (EQ = 1) AND (NH = 0) THEN : PRINT #1, USING ""######.##""; AM(I)"
316         '
317            NEXT I
318    "        PRINT #1,"
```

```
319   "         PRINT #1,"
320           IF GR < 0 THEN : GOTO 5740
321   "         PRINT #1,"
322   "         PRINT #1, ""                        Allowable Axle Load Limits (kips)"""
323   "              PRINT #1, ""               SINGLE                              TANDEM
TRIPLE"""
324   "         PRINT #1,"
325   "         IF S100 >= 10 THEN : PRINT #1, ""         LEGAL                    LEGAL
LEGAL"""
326   "              IF S100 < 10 THEN : PRINT #1, USING ""                  ##
##                  ##""; AL1; AL2; AL3"
327   "         PRINT #1,"
328   "         PRINT #1,"
329   5740    AV = AV / NC
330   "          PRINT #1, ""AVERAGE NUMBER OF PASSES TO CAUSE SPECIFIED RUT (""; RX;
""): ""; AV"
331   "         PRINT #1,"
332   "         PRINT #1, ""NUMBER OF PASSES USED FOR RUT DEPTH CALCULATION : ""; PA"
333   "         PRINT #1,"
334   "          PRINT #1, ""RUT DEPTH USED FOR REMAINING LIFE AND NUMBER OF PASSES
CALCULATION :""; RX"
335         '
336            NEXT ITE
337         PR = 0
338         PRINT
339         CLOSE
340         '
341         IF TBC = 0 THEN : GOTO 5960
342         DESIGN = 0
343         NH = 0
344         '
345         GOTO 2320
346         '
347         REM CONTINUE to get output menu OUTMENU
348   5960 GOSUB 7580: 'To OUTMENU
349   5980 END
350         '
351         '
352   6510 REM SUBROUTINE to open an output file OUTFILE
353         '
354   6512 CLS : KEY OFF
355   "     LOCATE 5, 1: PRINT ""TO CREATE OUTPUT FILE"""
356   "     LOCATE 6, 1: INPUT ""ENTER OUTPUT FILE NAME AND <RET>:""; XX$"
357         CLS : KEY OFF
358         IF TEMRUT = 1 THEN : SLF = 1: GOTO 6536
359   "     LOCATE 7, 1: PRINT ""Select the format:"""
360   "     LOCATE 9, 1: PRINT ""(1)"""
361   "       IF GR < 0 THEN : LOCATE 10, 1: PRINT ""Sec.     Thick     E1(Base)
E2(Subgr)  Passes  Rut depth  """
362   "       IF GR >= 0 THEN : LOCATE 10, 1: PRINT ""Sec.     Thick     E1(Base)
E2(Subgr)  Passes  Rut depth  Remaining life"""
363   "     LOCATE 11, 1: PRINT ""(2)"""
364   "      IF (EQ = 1) AND (GR < 0) THEN : LOCATE 12, 1: PRINT ""Sect.  Thick  D1
D7   FWD load   E1(Base)  E2(Subgr)  Passes  """
365   "      IF (EQ = 0) AND (GR < 0) THEN : LOCATE 12, 1: PRINT ""Sect.  Thick  D1
D5   FWD load   E1(Base)  E2(Subgr)  Passes  """
366   "      IF (EQ = 1) AND (GR >= 0) THEN : LOCATE 12, 1: PRINT ""Sect.  Thick  D1
D7   FWD load   E1(Base)  E2(Subgr)  Passes  Remaining life"""
```

```
367   "     IF (EQ = 0) AND (GR >= 0) THEN : LOCATE 12, 1: PRINT ""Sect.  Thick  D1
D5   FWD load   E1(Base)  E2(Subgr)  Passes  Remaining life"""
368   "     LOCATE 20, 1: INPUT ""ENTER SELECTION AND <RET> :""; SLF"
369         IF (SLF < 1) OR (SLF > 2) THEN : GOTO 6512
370   "6536 OPEN ""O"", #2, XX$: PRINT #2, ""    """
371   "       CLS : PRINT #2, "" TEXAS TRANSPORTATION INSTITUTE """
372   "       PRINT #2, ""LOAD RATING OF LIGHT PAVEMENT"""
373   "       PRINT #2,"
374   "       PRINT #2, ""JOB : ""; a$; ""          (INPUT FILE ""; X$; "")"""
375   "       PRINT #2,"
376   "       PRINT #2, ""DISTRICT:""; ID; ""     COUNTY:""; C$; ""     ROAD:""; F$"
377   "        PRINT #2, ""ALLOWABLE RUT(INS):""; RX; ""      RECORDED RUT(INS):"";
RM"
378   "       PRINT #2,"
379   "       PRINT #2, ""TRUCK NO. 1"""
380   "       PRINT #2, ""AXLE NUMBER   SINGLE WHEEL/ESWL(LBS)"""
381   "       PRINT #2, TAB(5); ""1""; TAB(20); ""9000"""
382   "       PRINT #2,"
383       '
384       '
385   "        PRINT #2, ""ANNUAL TRAFFIC GROWTH RATE:""; GR; ""  # OF YEARS:""; AL;
""  FIRST YEAR TRAFFIC:""; ESAL1"
386   "       PRINT #2, ""TOTAL NUMBER OF PASSES DURING ABOVE PERIOD:""; PA"
387   "       PRINT #2,"
388   "       IF EQ = 0 THEN : PRINT #2, ""DATE:""; D$; ""     DYNAFLECT"""
389   "          IF EQ = 1 THEN : PRINT #2, ""DATE:""; D$; ""    FALLING WEIGHT
DEFLECTOMETER"""
390         REM CONTINUE
391         IF GR < 0 THEN : GOTO 6562
392   "       IF TEMRUT = 1 THEN : PRINT #2, ""                  TEMP. CORRECTED
*              **              *"""
393   "          IF TEMRUT = 0 AND SLF = 1 THEN : PRINT #2, ""
*              **              *"""
394   "     IF TEMRUT = 0 AND SLF = 1 THEN : PRINT #2, "" SEC   BASE       E1-BASE
E2-SUBGR   NO. OF       RUT DEPTH    REMAINING LIFE"""
395   "     IF TEMRUT = 1 THEN : PRINT #2, "" SEC   BASE       E1-BASE    E2-SUBGR
NO. OF       RUT DEPTH    REMAINING LIFE"""
396   "     IF TEMRUT = 0 AND SLF = 1 THEN : PRINT #2, ""  #     (IN)        (PSI)
(PSI)     PASSES         (IN)         (YEARS)"""
397   "     IF TEMRUT = 1 THEN : PRINT #2, "" #     (IN)        (PSI)        (PSI)
PASSES         (IN)         (YEARS)"""
398   "      IF TEMRUT = 0 AND SLF = 1 THEN : FOR I = 1 TO NC: PRINT #2, USING
""##.###   ##.#     #########      #######   #########          #.##          ##.#"";
NS(I); BA(I); E1(I); E2(I); PS(I); AM(I); NY(I): NEXT"
399   "     IF TEMRUT = 1 THEN : FOR I = 1 TO NC: PRINT #2, USING ""##.###   ##.#
#########      #######   #########          #.##          ##.#""; NS(I); BA(I); E1(I);
E2(I); PS(I); AM(I); NY(I): NEXT"
400   "                IF   SLF   =   2   THEN   :   PRINT   #2,   ""
*              *"""
401   "     IF SLF = 2 AND EQ = 1 THEN : PRINT #2, "" SEC   BASE       D1         D7
FWD LOAD   E1-BASE     E2-SUBG        NO. OF   REMAINING LIFE"""
402   "     IF SLF = 2 AND EQ = 0 THEN : PRINT #2, "" SEC   BASE       D1         D5
FWD LOAD   E1-BASE     E2-SUBG        NO. OF   REMAINING LIFE"""
403   "     IF SLF = 2 THEN : PRINT #2, ""  #    (IN)     (MILS)    (MILS)    (LBS)
(PSI)       (PSI)         PASSES     (YEARS) """
404   "     IF SLF = 2 AND EQ = 1 THEN : FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#     ###.##      ###.##      ######      #########       ########       ###########
##.#""; NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I); NY(I): NEXT"
```

```
405   "      IF SLF = 2 AND EQ = 0 THEN : FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#      ###.##      ###.##      ######      ########       #######       ##########
##.#""; NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I); NY(I): NEXT"
406   "      PRINT #2,"
407   "      IF SLF = 1 THEN : PRINT #2, "" *  NUMBER OF PASSES AND REMAINING LIFE
FOR SPECIFIED RUT DEPTH OF ""; RX; ""in"""
408   "      IF SLF = 1 THEN : PRINT #2, "" ** RUT DEPTH FOR SPECIFIED NUMBER OF
PASSES OF ""; PA; "" IN ""; AL; ""YEARS"":"
409   "      IF SLF = 2 THEN : PRINT #2, "" *  NUMBER OF PASSES AND REMAINING LIFE
FOR SPECIFIED RUT DEPTH OF ""; RX; ""in"":"
410   "      PRINT #2,"
411   "      PRINT #2, ""      Remaining Life(yrs)          Cumulative % Sections"""
412         FOR II = 1 TO NYMAX
413   "        PRINT #2, USING ""            ##                                ###.#"";
II; DYR(II)"
414         NEXT II
415   "      PRINT #2,"
416   "      PRINT #2,"
417   "      PRINT #2, ""                    Allowable Axle Load Limits (kips)"""
418   "         PRINT #2, ""                SINGLE                          TANDEM
TRIPLE"""
419   "      PRINT #2,"
420   "      IF S100 >= 10 THEN : PRINT #2, ""          LEGAL                  LEGAL
LEGAL"""
421   "          IF  S100  <  10  THEN  :  PRINT  #2,  USING  ""                  ##
##                ##""; AL1; AL2; AL3"
422         GOTO 6563
423   6562 'SAVE DATA WITHOUT REMAINING LIFE - GR<0
424   "        IF TEMRUT = 1 THEN : PRINT #2, ""                  TEMP. CORRECTED
*              **              """
425   "          IF  TEMRUT  =  0  AND  SLF  =  1  THEN  :  PRINT  #2,  ""
*              **              """
426   "      IF TEMRUT = 0 AND SLF = 1 THEN : PRINT #2, "" SEC   BASE       E1-BASE
E2-SUBGR   NO. OF       RUT DEPTH    """
427   "      IF TEMRUT = 1 THEN : PRINT #2, "" SEC   BASE       E1-BASE    E2-SUBGR
NO. OF       RUT DEPTH    """
428   "      IF TEMRUT = 0 AND SLF = 1 THEN : PRINT #2, ""  #    (IN)        (PSI)
(PSI)     PASSES         (IN)         """
429   "      IF TEMRUT = 1 THEN : PRINT #2, "" #    (IN)        (PSI)        (PSI)
PASSES         (IN)         """
430   "       IF TEMRUT = 0 AND SLF = 1 THEN : FOR I = 1 TO NC: PRINT #2, USING
""##.###  ##.#    ########     #######   ########          #.##          ""; NS(I);
BA(I); E1(I); E2(I); PS(I); AM(I): NEXT"
431   "      IF TEMRUT = 1 THEN : FOR I = 1 TO NC: PRINT #2, USING ""##.###  ##.#
########     #######   ########          #.##          ""; NS(I); BA(I); E1(I); E2(I);
PS(I); AM(I): NEXT"
432   "                  IF   SLF   =   2   THEN   :   PRINT   #2,   ""
*              *"""
433   "      IF SLF = 2 AND EQ = 1 THEN : PRINT #2, "" SEC   BASE       D1        D7
FWD LOAD   E1-BASE     E2-SUBG        NO. OF   REMAINING LIFE"""
434   "      IF SLF = 2 AND EQ = 0 THEN : PRINT #2, "" SEC   BASE       D1        D5
FWD LOAD   E1-BASE     E2-SUBG        NO. OF   REMAINING LIFE"""
435   "      IF SLF = 2 THEN : PRINT #2, ""  #    (IN)     (MILS)    (MILS)    (LBS)
(PSI)       (PSI)         PASSES     (YEARS) """
436   "      IF SLF = 2 AND EQ = 1 THEN : FOR I = 1 TO NC: PRINT #2, USING ""##.###
##.#     ###.##     ###.##     ######     ########      #######      ##########      "";
NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I): NEXT"
437   "      IF SLF = 2 AND EQ = 0 THEN : FOR I = 1 TO NC: PRINT #2, USING ""##.###
```

```
##.#    ###.##    ###.##    ######    ########    #######    ##########    "";
NS(I); BA(I); D(I, 1); D(I, 7); PFW(I); E1(I); E2(I); PS(I): NEXT"
438   "     PRINT #2,"
439   "      IF SLF = 1 THEN : PRINT #2, "" *  NUMBER OF PASSES FOR SPECIFIED RUT
DEPTH OF ""; RX; ""in"""
440   "      IF SLF = 1 THEN : PRINT #2, "" ** RUT DEPTH FOR SPECIFIED NUMBER OF
PASSES OF ""; PA; "" IN ""; AL; ""YEARS"":"
441   "      IF SLF = 2 THEN : PRINT #2, "" *  NUMBER OF PASSES FOR SPECIFIED RUT
DEPTH OF ""; RX; ""in"":"
442        'TEMP. CORR. DATA
443   "6563  IF TEC = 1 THEN : INPUT ""    DO YOU WANT TO SAVE TEMPERATURE AND
MOISTURE CORRECTED DATA ? 1=YES, 0=NO ""; CH04"
444        IF (CH04 <> 1 AND CH04 <> 0) THEN GOTO 6563
445        IF (CH04 = 0) GOTO 6584
446   "     PRINT #2, ""  """
447   "     PRINT #2, ""          TEMPERATURE CORRECTION DATA """
448   "     PRINT #2,"
449   "     PRINT #2, ""    INPUT CONDITIONS:                WANTED CONDITIONS:"""
450   "      PRINT #2, USING ""    TEMPERATURE =    ###.# deg. F      TEMPERATURE =
###.# deg. F""; TTEMP; RTEMP"
451   "      PRINT #2, USING ""    SUCTION      = #####.# psi         SUCTION      =
#####.# psi""; HI; HL"
452   "     PRINT #2,"
453   "     PRINT #2, ""    INPUT     WDNTED CONDITION    MODULUS CHANGE    MODULUS
CHANGE"""
454   "     PRINT #2, ""   MODULUS          MODULUS         DUE TO TEMPER.    DUE TO
SUCTION"""
455   "       PRINT #2, ""      (psi)               (psi)                  (psi)
(psi)"""
456   "     PRINT #2,"
457        FOR I = 1 TO KK1
458   "        PRINT #2, USING ""    #######          #######            ######
######""; OE(I); NE(I); DET(I); DES(I)"
459        NEXT I: CLOSE
460   6584 RETURN
461   6585 REM SUBROUTINE TO SELECT TEMP. CORRECTION OR STRUCTURAL ANALYSIS
462   6586 CLS : KEY OFF
463   "     LOCATE 3, 10"
464   "     PRINT ""LOADRATE - STRUCTURAL ANALYSIS OF LIGHT PAVEMENTS"""
465   "     LOCATE 5, 10"
466   "     PRINT ""                    MAIN MENU"""
467   "     LOCATE 7, 10"
468   "     PRINT ""           SELECT THE OPERATION DESIRED:"""
469   "     LOCATE 9, 10"
470   "     PRINT ""(1)  Structural analysis of pavements"""
471   "     LOCATE 10, 10"
472   "       PRINT ""    ( moduli, # of passes, rut depth, remaining life and
allowable"""
473   "     LOCATE 11, 10"
474   "     PRINT ""     axle load calculations, and analysis of temperature and """
475   "     LOCATE 12, 10"
476   "     PRINT ""     moisture effects on base material )"""
477   "     LOCATE 13, 10"
478   "     PRINT ""(2)  Temperature correction of base modulus only"""
479   "     LOCATE 14, 10"
480   "     PRINT ""(3)  Rut depth calculation only"""
481   "     LOCATE 15, 10"
482   "     PRINT ""(4)  Exit program"""
```

```
483   "     LOCATE 20, 10"
484   "     INPUT ""ENTER SELECTION AND <RET> : ""; SLECT1"
485         IF (SLECT1 < 1) OR (SLECT1 > 4) THEN : GOTO 6586
486         RETURN
487   6600 'SUBROUTINE for elastic modulus calculations ECAL
488         '
489         'Initializing various values - all changes will be made here
490         KO1 = .6: REM SIGR/SIGZ ratio for base course
491         KO2 = .82: REM SIGR/SIGZ ratio for subgrade
492         'EMOD will be calculated at test load
493         AREA = 3.1415927# * 5.9 * 5.9
494         'Looping
495         FOR I = 1 TO NC
496             IF EQ = 0 THEN : PFW(I) = 9000: REM DYNAFLECT - CALCULATE FOR WHEEL
LOAD
497            SL(I) = PFW(I) / AREA
498            ZZ1 = 1 + BA(I) / 2: REM Elastic modulus for base course depth
499            ZZ2 = 7! + BA(I): REM Elastic Modulus for subgrade depth
500            ALS1 = 2.416966 * SL(I) ^ .477234
501            ALS2 = .6007171 * K1S(I) ^ .058992
502            ALH = 1.338562 * (1 + BA(I)) ^ -.145887
503            RHB = .59614133# * K1(I) ^ 5.593462E-02
504            RHH = 3.0385722# * (1 + BA(I)) ^ .419624
505            RHS = 272.63462# * K1S(I) ^ -.58427
506             NSG = -1.01866 + 1.239625 / (K1S(I) * 10 ^ -3) + .000013 * (K1S(I) *
10 ^ -3) ^ 3 + .061115 * (K1S(I) * 10 ^ -3) - .00165 * (K1S(I) * 10 ^ -3) ^ 2
507            ALPHA = ALS1 * ALS2 * ALH
508            RHO = (1 + BA(I)) * LOG(RHH * RHS * RHB / ALPHA) / LOG(2.7182818#)
509            ER1 = ALPHA * EXP(-((-RHO / ZZ1) ^ 1)) + 5.9
510            ER2 = ALPHA * EXP(-((-RHO / ZZ2) ^ 1)) + 5.9
511            SIGZ1 = PFW(I) / (3.1415927# * ER1 * ER1)
512            SIGZ2 = PFW(I) / (3.1415927# * ER2 * ER2)
513            SIGR1 = KO1 * SIGZ1
514            SIGR2 = KO2 * SIGZ2
515            E2(I) = K1S(I) * (ABS(SIGZ2 - SIGR2)) ^ NSG
516            E1(I) = K1(I) * (SIGZ1 + 2 * SIGR1) ^ .33
517            NEXT
518         RETURN
519         '
520         '
521         REM SUBROUTINE  MAINMENU ————————————————————
522         '
523   6970 'KEY OFF: CLS
524         '
525         '
526         REM SUBROUTINE Input for geometric regression GEOMINPUT ————————
527         '    for non-standard FWD sensor spacing
528         GR = 1: CLS : REM To initialize and activate GEOMBASIN
529         'Standard FWD sensor spacing for Texas SDHPT is as follows
530   "     LOCATE 5, 10: PRINT ""FWD Sensor          #1  #2  #3  #4  #5  #6  #7"""
531   "     LOCATE 6, 10: PRINT ""Dist.(ins) from load""; 0; 12; 24; 36; 48; 60; 72"
532         '
533   "     LOCATE 7, 10: PRINT ""Type in new spacing ....."""
534   "     LOCATE 8, 10: INPUT ""Sensor #1 at (ins) ""; X(1)"
535   "     LOCATE 9, 10: INPUT ""Sensor #2 at (ins) ""; X(2)"
536   "     LOCATE 10, 10: INPUT ""Sensor #3 at (ins) ""; X(3)"
537   "     LOCATE 11, 10: INPUT ""Sensor #4 at (ins) ""; X(4)"
538   "     LOCATE 12, 10: INPUT ""Sensor #5 at (ins) ""; X(5)"
```

```
539   "     LOCATE 13, 10: INPUT ""Sensor #6 at (ins) ""; X(6)"
540   "     LOCATE 14, 10: INPUT ""Sensor #7 at (ins) ""; X(7)"
541         IF X(1) = 0 THEN : X(1) = .01
542         RETURN
543         '
544         '
545   7290 REM SUBROUTINE geometric regression GEOMBASIN
546            J = 0: K = 0: L = 0: M = 0: R2 = 0
547            N = 4
548   "         IF D(I, 7) = 0 THEN : N = 3"
549            IF X(7) = 0 THEN : N = 3
550            FOR II = 4 TO (3 + N)
551   "             X = X(II): Y = D(I, II)"
552                X = LOG(X): Y = LOG(Y)
553                J = J + X: K = K + Y: L = L + X * X: M = M + Y * Y
554                R2 = R2 + X * Y
555                NEXT II
556            B = (N * R2 - K * J) / (N * L - J * J)
557            a = (K - B * J) / N
558            J = B * (R2 - J * K / N)
559            M = M - K * K / N
560            K = M - J
561            R2 = J / M
562   "         'print ""D7@94.49 =""; exp(a)*94.49^b"
563            D7(I) = EXP(a) * 94.49 ^ B
564            RETURN
565         '
566   7580 REM SUBROUTINE  OUTMENU
567         CLS : KEY OFF
568   "     LOCATE 2, 10: PRINT ""        OUTPUT MENU """
569         '
570   "     LOCATE 4, 10: PRINT ""Select Optional Output Format"""
571         '
572   "     LOCATE 6, 10: PRINT "" (1) Plot Base Course Elastic Modulus"""
573   "     LOCATE 7, 10: PRINT "" (2) Plot Subgrade Elastic Modulus"""
574   "     LOCATE 8, 10: PRINT "" (3) Plot Number of Vehicle Passes"""
575   "     LOCATE 9, 10: PRINT "" (4) Plot Remaining Life"""
576   "     LOCATE 10, 10: PRINT "" (5) Temperature Correction of Base Modulus"""
577   "       LOCATE 11, 10: PRINT ""       (5.1) Rut depth, Passes and Remaining
Life"""
578   "     LOCATE 12, 10: PRINT ""            for corrected moduli"""
579   "     LOCATE 13, 10: PRINT "" (6) Save the Output as a file"""
580   "     LOCATE 14, 10: PRINT "" (7) Exit to Main Menu"""
581         '
582   "     LOCATE 20, 10: INPUT ""ENTER SELECTION AND <RET> : ""; SLO"
583         IF SLO = 7 OR SLO = 0 THEN : CLS : SLO = 0: TEMRUT = 0: TEC = 0: GOTO 730
584         IF SLO = 6 THEN : GOSUB 6510: GOTO 7580: 'To OUTFILE and back
585         IF SLO = 5 THEN : TEC = 1: GOSUB 10010
586         IF SLO = 5.1 THEN : FOR I = 1 TO 100: NY(I) = 0: NEXT
587         IF SLO = 5.1 THEN : FOR I = 1 TO 100: DYR(I) = 0: NEXT
588         IF SLO = 5.1 THEN : FOR I = 1 TO 100: PS(I) = 0: NEXT
589       '
590         IF SLO = 5.1 THEN : TEMRUT = 1: GOSUB 5080
591         GOTO 7800
592         END
593         '
594         '
595   7800 REM PROGRAM SBRPLOT
```

```
596         PRCHK = 0
597         IF PT = 0 THEN : PT = 1: REM 1=Epson/Panasonic    2=Mannesmann Tally
598             REM  *******  THIS  PART  OF  THE  PROGRAM  SHOULD  REMAIN  UNTOUCHED
*************
599         CLS
600         PRINT
601         REM ************* plot at low resolution graphics
602         PRINT
603         RD = 0
604         PRINT
605    "     'INPUT"" TITLE : "";T$"
606    "     IF SLO = 1 THEN : t$ = ""ELASTIC MODULUS OF BASE COURSE"""
607    "     IF SLO = 2 THEN : t$ = ""ELASTIC MODULUS OF SUBGRADE"""
608    "     IF SLO = 3 THEN : t$ = ""NUMBER OF PASSES FOR RUT DEPTH"""
609    "     IF SLO = 4 THEN : t$ = ""REMAINING LIFE"""
610    "     'PRINT ""plotting : "";T$"
611    "     'INPUT ""Starting Section Number: ""; MINX"
612    "     'INPUT ""Ending Section Number: ""; MAXX"
613         PRINT
614         MINY = 0
615         '
616    "     IF SLO = 1 THEN : INPUT ""Minimum E1  [Default=0]: ""; MINY"
617    "     IF SLO = 1 THEN : INPUT ""Maximum E1 [Default=100000]: ""; MAXY"
618         IF SLO = 1 AND MAXY = 0 THEN : MAXY = 100000!
619    "     IF SLO = 2 THEN : INPUT ""Minimum E2 [Default= 0   ]: ""; MINY"
620    "     IF SLO = 2 THEN : MAXY = 25000: INPUT ""Maximum E2 [Default=25000]: "";
MAXY"
621         IF SLO = 2 AND MAXY = 0 THEN : MAXY = 25000
622    "     IF SLO = 3 THEN : INPUT ""Minimum PASSES [Default=   0 ]: ""; MINY"
623    "     IF SLO = 3 THEN : INPUT ""Maximum PASSES [Default=0.5E6]: ""; MAXY"
624         IF SLO = 3 AND MAXY = 0 THEN : MAXY = 500000!
625    "       IF SLO = 4 THEN : INPUT ""Minimum REMAINING LIFE [Default=  0]: "";
MINY"
626    "       IF SLO = 4 THEN : INPUT ""Maximum REMAINING LIFE [Default= 24]: "";
MAXY"
627         IF SLO = 4 AND MAXY = 0 THEN : MAXY = 24
628         '
629         '
630         PRINT
631         SX = 2: 'Number of alphabet for NC
632         IF SX = 0 THEN : SX = 1
633         PRINT
634    "     'INPUT ""hardcopy?   1=yes  0=no  "";PR"
635         PR = 1
636         XSCALE = 60 / (MAXX - MINX)
637         YSCALE = 20 / (MAXY - MINY)
638         YORG = 22 + MINY * YSCALE
639         XORG = 10 - MINX * XSCALE
640         KEY OFF
641         CLS
642          '
643          '
644          '
645    8300 REM these are ASCII for the IBM
646          IF PRCHK = 0 THEN : PR1 = 196: PR2 = 179: PR3 = 218: PR4 = 191: PR5 =
192: PR6 = 217
647         FOR COL = 11 TO 69
648    "          LOCATE 2, COL: PRINT CHR$(PR1)"
```

```
649   "          LOCATE 22, COL: PRINT CHR$(PR1)"
650             NEXT
651          FOR ROW = 3 TO 21
652   "          LOCATE ROW, 10: PRINT CHR$(PR2)"
653   "          LOCATE ROW, 70: PRINT CHR$(PR2)"
654             NEXT
655   "       LOCATE 2, 10: PRINT CHR$(PR3)"
656   "       LOCATE 2, 70: PRINT CHR$(PR4)"
657   "       LOCATE 22, 10: PRINT CHR$(PR5)"
658   "       LOCATE 22, 70: PRINT CHR$(PR6)"
659          IF PRCHK = 0 THEN : PR7 = 193: PR8 = 195
660          IF PRCHK = 0 THEN : PR9 = 180: PR10 = 194
661           '
662           '
663           '
664          FOR I = 1 TO 9
665   "          LOCATE 22, (10 + 6 * I): PRINT CHR$(PR7)"
666   "          LOCATE (2 + 2 * I), 10: PRINT CHR$(PR8)"
667   "          LOCATE (2 + 2 * I), 70: PRINT CHR$(PR9)"
668   "          LOCATE 2, (10 + 6 * I): PRINT CHR$(PR10)"
669             NEXT
670          IF PRCHK = 1 AND PT = 1 THEN : GOSUB 9350
671          IF PRCHK = 1 AND PT = 2 THEN : GOSUB 9210
672   "       LOCATE 1, 10: PRINT t$"
673   "       LOCATE 1, 55: PRINT X$"
674          XINT = (MAXX - MINX) / 10
675          YINT = (MAXY - MINY) / 10
676   "       LOCATE 23, 8: PRINT (MINX)"
677   "       LOCATE 23, 20: PRINT (MINX + 2 * XINT)"
678   "       LOCATE 23, 32: PRINT (MINX + 4 * XINT)"
679   "       LOCATE 23, 44: PRINT (MINX + 6 * XINT)"
680   "       LOCATE 23, 56: PRINT (MINX + 8 * XINT)"
681   "       LOCATE 23, 69: PRINT (MINX + 10 * XINT)"
682   "       'lOCATE 24,30 :PRINT ""Section Numbers"""
683   "       LOCATE 2, 1: PRINT MAXY"
684   "       LOCATE 6, 1: PRINT INT((MAXY - YINT * 2) * 100) / 100"
685   "       LOCATE 10, 1: PRINT INT((MAXY - YINT * 4) * 100) / 100"
686   "       LOCATE 14, 1: PRINT INT((MAXY - YINT * 6) * 100) / 100"
687   "       LOCATE 18, 1: PRINT INT((MAXY - YINT * 8) * 100) / 100"
688   "       LOCATE 22, 1: PRINT MINY"
689   "       IF SLO = 1 THEN : LOCATE 11, 2: PRINT "" E1"""
690   "       IF SLO = 1 THEN : LOCATE 12, 2: PRINT ""(psi)"""
691   "       IF SLO = 2 THEN : LOCATE 11, 2: PRINT "" E2"""
692   "       IF SLO = 2 THEN : LOCATE 12, 2: PRINT ""(psi)"""
693   "       IF SLO = 3 THEN : LOCATE 11, 2: PRINT ""No.of"""
694   "       IF SLO = 3 THEN : LOCATE 12, 2: PRINT ""Passes"""
695   "       IF SLO = 4 THEN : LOCATE 11, 2: PRINT ""Remain."""
696   "       IF SLO = 4 THEN : LOCATE 12, 2: PRINT "" Life"""
697   "       IF SLO = 4 THEN : LOCATE 13, 2: PRINT ""(years)"""
698          REM CONTINUE to exit loop
699          'get points
700   "       IF PRCHK = 1 THEN : LOCATE 25, 1: PRINT SPC(70);"
701   "       IF PRCHK = 1 THEN : LOCATE 25, 1: PRINT ""RE-PLOTTING ... "";"
702          FOR I = 1 TO NC
703             X = NS(I)
704             IF SLO = 1 THEN : Y = E1(I)
705             IF SLO = 1 AND Y > MAXY THEN : Y = MAXY
706             IF SLO = 2 THEN : Y = E2(I)
```

```
707           IF SLO = 2 AND Y > MAXY THEN : Y = MAXY
708           IF SLO = 3 THEN : Y = PS(I)
709           IF SLO = 3 AND Y > MAXY THEN : Y = MAXY
710           IF SLO = 4 THEN : Y = NY(I)
711           IF SLO = 4 AND Y > MAXY THEN : Y = MAXY
712           'display point and coord
713           'display point and coord
714        '
715           SCY = YORG - INT(Y * YSCALE): SCX = XORG + INT(X * XSCALE)
716           IF SCX < 10 THEN GOTO 9050
717           IF SCX > 70 THEN GOTO 9050
718           IF SCY < 2 THEN GOTO 9050
719           IF SCY > 22 THEN GOTO 9050
720   "        LOCATE SCY, SCX: PRINT CHR$(42): GOTO 9050"
721   9050    NEXT
722   "     LOCATE 25, 1: PRINT SPC(75);"
723   "     IF PRCHK = 1 THEN : LOCATE 25, 1: INPUT ""PrtSc?  <RET> if NO""; NN$"
724         IF PRCHK = 1 THEN : GOTO 9150
725   "     LOCATE 25, 1: PRINT SPC(75);"
726   "     LOCATE 25, 1: PRINT ""PRESS ENTER TO CONTINUE "";"
727         INPUT ; NN$
728         IF PRCHK = 1 THEN : GOTO 9150
729         IF PR = 1 AND PT = 1 THEN : GOSUB 9350: GOTO 8300
730         IF PR = 1 AND PT = 2 THEN : GOSUB 9210: GOTO 8300
731         IF PR = 1 AND PT = 3 THEN : GOTO 9150: 'No adjustment required
732   9150 REM CONTINUE
733         CLS
734         GOTO 7580: 'To return to OUTMENU
735         '
736         '
737         '
738   9210 REM SUBROUTINE MT180 to replot the graph for the mannesmann tally
739         PR1 = 157: REM FROM IBM 196
740         PR2 = 156: REM FROM IBM 179
741         PR3 = 134: REM FROM IBM 218
742         PR4 = 149: REM FROM IBM 191
743         PR5 = 153: REM FROM IBM 192
744         PR6 = 154: REM FROM IBM 217
745         PR7 = 158: REM FROM IBM 193
746         PR8 = 150: REM FROM IBM 195
747         PR9 = 151: REM FROM IBM 180
748         PR10 = 152: REM FROM IBM 194
749         PRCHK = 1
750         RETURN
751         '
752   9350 REM SUBROUTINE EPSON to replot the graph for the epson/panasonic
753         PR1 = 45: REM FROM IBM 196
754         PR2 = 124: REM FROM IBM 179
755         PR3 = 46: REM FROM IBM 218
756         PR4 = 46: REM FROM IBM 191
757         PR5 = 46: REM FROM IBM 192
758         PR6 = 46: REM FROM IBM 217
759         PR7 = 43: REM FROM IBM 193
760         PR8 = 43: REM FROM IBM 195
761         PR9 = 43: REM FROM IBM 180
762         PR10 = 43: REM FROM IBM 194
763         PRCHK = 1
764         RETURN
```

```
765         '
766         '
767         '
768    "        OPEN ""O"", #1, ""B:FM421.RES"": FOR II = 1 TO 26: PRINT #1, NS(II),
BA(II), D(II, 1), D(II, 7), K1(II), K1S(II), E1(II), E2(II), SS(II), BC(II), PS(II),
AM(II): NEXT: CLOSE"
769    "        OPEN ""O"", #1, ""B:FM421.SAN"": FOR II = 1 TO 26: PRINT #1, PS(II),
AM(II): NEXT: CLOSE"
770    10010 CLS
771          REM SUBROUTINE for temperature correction TEMP
772          'THIS SUBROUTINE PREDICTS THE MODULUS OF GRANULAR MATERIALS
773          'AT DIFFERENT TEMPERATURES AND MOISTURES BASED ON A
774          'MICROMECHANICAL APPROACH
775    10130 'CLS
776          '
777          '=======INPUT========
778          PRINT : PRINT : PRINT
779    "       PRINT TAB(25); ""TYPE OF BASE MATERIAL:"": PRINT"
780    "       PRINT TAB(25); ""  1 = CRUSHED LIMESTONE """
781    "       PRINT TAB(25); ""       LIME ORE GRAVEL"""
782    "       PRINT TAB(25); ""       IRON ORE GRAVEL"""
783    "       PRINT TAB(25); ""  2 = RIVER GRAVEL"""
784    "       PRINT TAB(25); ""       CALICHE"""
785    "       PRINT TAB(25); ""       CALICHE GRAVEL"""
786    "       PRINT TAB(25); ""  3 = SANDSHELL"""
787    "       PRINT TAB(25); : INPUT ""SELECT YOUR OPTION ""; MTYPE"
788          IF (MTYPE <= 0 OR MTYPE > 3) GOTO 10130
789          IF (MTYPE = 1) THEN ALP = 5 * 10 ^ (-6): E = 6.4 * 10 ^ 6: U = .17: K1 =
14000: K2 = .4
790          IF (MTYPE = 2) THEN ALP = 6.5 * 10 ^ (-6): E = 7.8 * 10 ^ 6: U = .2: K1
= 24000: K2 = .37
791          IF (MTYPE = 3) THEN ALP = 5 * 10 ^ (-6): E = 8.534001 * 10 ^ 6: U = .17:
K1 = 7210: K2 = .45
792    10260 CLS : PRINT : PRINT
793    "       PRINT TAB(20); ""PROPERTIES OF THE MATERIAL SELECTED:"""
794          PRINT
795    "       PRINT TAB(25); ""1. LINEAR THERMAL EXPANSION        = ""; ALP"
796    "       PRINT TAB(25); ""2. ELASTIC MODULUS OF SOIL GRAINS  = ""; E"
797    "       PRINT TAB(25); ""3. POISSON'S RATIO                 = ""; U"
798          IF SLO = 5 THEN : GOTO 10340
799    "       PRINT TAB(25); ""4. K1                              = ""; K1"
800    "       PRINT TAB(25); ""5. K2                              = ""; K2"
801    10340 PRINT : PRINT
802    "10350 PRINT TAB(20); : INPUT ""DO YOU WANT TO CHANGE? 0=NO  1=YES ""; CHOICE"
803          IF (CHOICE <> 1 AND CHOICE <> 0) THEN GOTO 10350
804          IF (CHOICE = 0) GOTO 10460
805    "       PRINT : PRINT TAB(20); : INPUT ""WHICH LINE ""; WL"
806    "       PRINT TAB(20); : INPUT ""INPUT NEW VALUE ""; NV"
807          IF (WL = 1) THEN ALP = NV
808          IF (WL = 2) THEN E = NV
809          IF (WL = 3) THEN U = NV
810          IF (WL = 4) THEN K1 = NV
811          IF (WL = 5) THEN K2 = NV
812          GOTO 10260
813    10460 CLS : PRINT : PRINT : PRINT TAB(25);
814    "       INPUT ""UNIT WEIGHT OF MATERIAL (pcf) :   "", UW"
815          CLS
816          PRINT : PRINT
```

```
817         IF SLECT1 = 2 THEN : GOTO 10478
818         CHO1 = 1
819         GOTO 10500
820   10478 KK1 = 1
821         PRINT : PRINT TAB(25);
822   "       INPUT ""MODULUS OF MATERIAL (psi)          : "", EI"
823   10500 PRINT : PRINT
824   "10510 PRINT TAB(20); ""CONDITIONS AT WHICH THE MODULUS IS OBTAINED: """
825         PRINT
826   "       PRINT TAB(25); : INPUT ""TEMPERATURE, Ti (40 deg.F < Ti < 110 deg.F)
:   "", TTEMP"
827         IF (TTEMP < 40 OR TTEMP > 110) GOTO 10510
828   "10540 PRINT TAB(25); : INPUT ""SUCTION, Hi (-145 psi < Hi < 0 psi)          :
"", HI"
829         IF (HI < -145 OR HI > 0) GOTO 10540
830         PRINT : PRINT
831   "10560 PRINT TAB(20); ""INPUT WANTED CONDITIONS: """
832         PRINT
833   "       PRINT TAB(25); : INPUT ""TEMPERATURE, To (40 deg.F < To < 110 deg.F)
:   "", RTEMP"
834         IF (RTEMP < 40 OR RTEMP > 110) GOTO 10560
835   "10590 PRINT TAB(25); : INPUT ""SUCTION, Ho (-145 psi < Ho < 0 psi)          :
"", HL"
836         IF (HL < -145 OR HL > 0) GOTO 10590
837         IF CHO1 = 1 THEN KK1 = NC
838         FOR I = 1 TO KK1
839         IF CHO1 = 1 THEN EI = E1(I)
840         IF CHO1 = 1 THEN K1 = K1(I)
841         IF CHO1 = 1 THEN K2 = .33
842         '
843         '=======CALCULATION========
844         VOID = 1 / (UW / 2.67 / 62.4) - 1
845         N = VOID / (1 + VOID)          'POROSITY
846         X = (.4764 - N) / (.4764 - .2595)
847         IF (X < 0) THEN X = 0
848         IF (X > 1) THEN X = 1
849         PI = 3.1415927#
850         PRO1 = 3 / 4 * (1 - U ^ 2) / E
851         '==K2 IS THE POWER IN THE EQUATION E=K1*TETHA^K2==
852         'K2 = 1 / 3
853         THETA = 10 ^ (LOG(EI / K1) / LOG(10) / K2)
854         DTEMP = RTEMP - TTEMP'RTEMP IS THE REFERENCE TEMP.
855         DT = ABS(DTEMP)
856         IF (DTEMP = 0) THEN NE = EI: GOTO 10800
857         DELV = DT * ALP * 3
858         PSC = X / 2 ^ (1 / 2) / PRO1 * (1 / 3 * DELV) ^ (3 / 2)
859         PFCC = (1 - X) / 4 / PRO1 * (1 / 3 * DELV) ^ (3 / 2)
860         PT = (PSC + PFCC) * DTEMP / DT
861         DET(I) = K1 * K2 * THETA ^ (K2 - 1) * PT
862   10800 PS = -(HL - HI) * .13
863         DES(I) = K1 * K2 * THETA ^ (K2 - 1) * PS
864         TOTDE(I) = DET(I) + DES(I)
865         NE(I) = EI + TOTDE(I)
866         OE(I) = EI
867         NEXT I
868         '
869         '=======OUTPUT========
870         CLS
```

```
871         PRINT : PRINT : PRINT TAB(10);
872   "      PRINT ""INPUT CONDITIONS:                    WANTED CONDITIONS:"""
873         PRINT
874   "       PRINT USING ""  TEMPERATURE =     ###.#    deg. F        TEMPERATURE =
###.#   deg. F""; TTEMP; RTEMP"
875   "       PRINT USING ""  SUCTION      =  #####.##  psi            SUCTION      =
#####.##  psi   ""; HI; HL"
876         PRINT
877   "       PRINT ""      INPUT     WANTED CONDITION     MODULUS CHANGE    MODULUS
CHANGE"""
878   "       PRINT ""     MODULUS        MODULUS           DUE TO TEMPER.    DUE TO
SUCTION"""
879   "      PRINT ""    (psi)          (psi)              (psi)            (psi)"""
880         PRINT
881         FOR I = 1 TO KK1
882   "          PRINT USING ""     ######.##          ######.##           ######.##
########.##""; OE(I); NE(I); DET(I); DES(I)"
883         NEXT I
884         PRINT : PRINT
885   "      INPUT ""         PRESS RETURN TO CONTINUE........."; PRESS"
886         CLS
887         IF SLECT1 <> 2 THEN : GOTO 10988
888   "      PRINT : PRINT : PRINT ""          1. ANOTHER TEMPERATURE CORRECTION """
889   "      PRINT : PRINT ""          2. EXIT TO MAIN MANUAL """
890   "      PRINT : PRINT : INPUT ""       Select the Operation Desired: ""; CHO2"
891         IF CHO2 = 1 THEN : GOTO 10010
892         GOTO 730
893   "10988 PRINT : PRINT : PRINT ""          1. EXIT TO MAIN MENU """
894   "      PRINT : PRINT ""          2. EXIT TO OUTPUT MENU """
895   "      PRINT : PRINT : INPUT ""       Select the Operation Desired: ""; CHO3"
896         IF CHO3 = 1 THEN : GOTO 730
897         GOTO 7580
898         RETURN
899         '
900   12000 '*************************************************
901         ' RUT DEPTH PREDICTION SUBROUTINE FOR LOW-VOLUME ROADS
902         '                        BY
903         '                  K. A. S. YAPA
904         '*************************************************
905   "      ' Pavement Systems, Texas Transportation Institute,"
906   "      ' TTI Building, Texas A & M University,"
907   "      ' College Station, Texas 77843."
908         ' (409)-845-9910.
909         ' 24th MARCH 1988.
910         '
911         'This subroutine predicts the rut depth of a low-volume road by using a
data
912         'base of rut depths calcualted by the Mechano-lattice program. A
913         'multi-dimensional polynomial interpolation routine is used to
914         'interpolate among the input parameters. Required inputs are the
915         'resilient modulus of both the base course and the subgrade
916   "      'layers, the material classification of the subgrade layer"
917         'and the thickness of the base layer.
918   "      'Optionally, laboratory data from a permanent deformation test for"
919   "      'each material layer can be input, in place of the material"
920         'classification.
921         '
922   "      DIM rut(2, 3, 3, 3, 3)"
```

```
923         '
924         ' LOAD THE DATA BASE INTO AN ARRAY
925         '
926         FOR I = 1 TO 2
927         FOR J = 1 TO 3
928         FOR K = 1 TO 3
929         FOR L = 1 TO 3
930   "      READ rut(I, J, K, L, 1), rut(I, J, K, L, 2), rut(I, J, K, L, 3)"
931         NEXT L: NEXT K: NEXT J: NEXT I
932         '
933         ' READ THE ORIGINAL PARAMETER LEVELS OF THE DATABASE
934         '
935         FOR I = 1 TO 3
936   "      READ XM(I), XL(I), XK(I), XJ(I), XI(I)"
937         NEXT I
938         '
939         ' GOTO SUBROUTINE INPUT1 ————————————————
940         RUT1 = 1
941         '
942   12400 GOSUB 14240
943         ' LOOP TO CALCULATE RUT DEPTHS FOR EACH FWD SECTION
944         '
945         IF INIOPT = 2 THEN NC = 1
946         '
947         FOR INC = 1 TO NC
948         IF TEMRUT = 1 THEN : E1(INC) = NE(INC)
949         REM GOTO SUBROUTINE INPUT2 ————————————————
950         GOSUB 14850
951         '
952         'GOTO INPUT1 IF ANY CORRECTIONS ARE NEEDED
953         IF CORR = 1 THEN GOSUB 14240
954         CLS
955   "      PRINT "" Calculating....."""
956         '
957         ' SELECT PARAMETERS FOR INTERPOLATION
958         '
959         FOR I = 1 TO 2
960         FOR J = 1 TO 3
961         FOR K = 1 TO 3
962         FOR L = 1 TO 3
963         FOR M = 1 TO 3
964         '
965         ' M - BASE THICKNESS
966         '
967   "      Y1(M) = rut(I, J, K, L, M)"
968         X1(M) = XM(M): NEXT M
969         '
970         ' CALL THE INTERPOLATION ROUTINE
971         '
972         NUM = 3: X = XMM: GOSUB 13700
973         YLTEMP(L) = Y: XMFLAG = XFLAG
974         NEXT L
975         '
976         ' L - SUBGRADE RUTTING POTENTIAL
977         '
978         FOR LL = 1 TO 3: X1(LL) = XL(LL): Y1(LL) = YLTEMP(LL): NEXT LL
979         NUM = 3: X = XLL: GOSUB 13700
980         YKTEMP(K) = Y: XLFLAG = XFLAG
```

```
981         NEXT K
982         '
983         ' K - SUBGRADE RESILIENT MODULUS
984         '
985         FOR KK = 1 TO 3: X1(KK) = XK(KK): Y1(KK) = YKTEMP(KK): NEXT KK
986         NUM = 3: X = XKK: GOSUB 13700
987         YJTEMP(J) = Y: XKFLAG = XFLAG
988         NEXT J
989         '
990         ' J - BASE RESILIENT MODULUS
991         '
992         FOR JJ = 1 TO 3: X1(JJ) = XJ(JJ): Y1(JJ) = YJTEMP(JJ): NEXT JJ
993         NUM = 3: X = XJJ: GOSUB 13700
994         YITEMP(I) = Y: XJFLAG = XFLAG
995         NEXT I
996         '
997         ' I - BASE RUTTING POTENTIAL
998         '
999         FOR II = 1 TO 2: X1(II) = XI(II): Y1(II) = YITEMP(II): NEXT II
1000        NUM = 2: X = XII: GOSUB 13700
1001        RUTCAL = Y * CYL / 300000!: XIFLAG = XFLAG
1002        '
1003        ' GOTO SUBROUTINE OUTPUT ————————————————
1004        '
1005        GOSUB 15680
1006        NEXT INC
1007        IF INIOPT <> 2 THEN : GOTO 13685
1008        PRINT : PRINT : PRINT
1009  "      PRINT ""            1. ANOTHER RUT DEPTH CALCULATION """
1010  "      PRINT : PRINT ""            2. EXIT TO MAIN MANUAL """
1011  "        PRINT : PRINT : PRINT : INPUT ""              Select the Operation
Desired: ""; CHO5"
1012        IF CHO5 = 1 THEN : GOTO 12400
1013        GOTO 730
1014        '
1015        ' DATA BASE OF RUT DEPTHS
1016        '
1017  "      DATA       0.2609 ,          0.2804 ,          0.3191"
1018  "      DATA       0.2216 ,          0.2152 ,          0.2223"
1019  "      DATA       0.1831 ,          0.1502 ,          0.1253"
1020  "      DATA       0.2705 ,          0.3030 ,          0.3431"
1021  "      DATA       0.2451 ,          0.2491 ,          0.2556"
1022  "      DATA       0.2168 ,          0.1951 ,          0.1680"
1023  "      DATA       0.3674 ,          0.4519 ,          0.5286"
1024  "      DATA       0.3503 ,          0.4172 ,          0.4580"
1025  "      DATA       0.3331 ,          0.3831 ,          0.3882"
1026  "      DATA       0.2702 ,          0.2912 ,          0.3019"
1027  "      DATA       0.2249 ,          0.2185 ,          0.2002"
1028  "      DATA       0.1797 ,          0.1458 ,          0.0991"
1029  "      DATA       0.2804 ,          0.3099 ,          0.3549"
1030  "      DATA       0.2473 ,          0.2490 ,          0.2600"
1031  "      DATA       0.2137 ,          0.1882 ,          0.1652"
1032  "      DATA       0.3900 ,          0.4610 ,          0.5322"
1033  "      DATA       0.3688 ,          0.4231 ,          0.4561"
1034  "      DATA       0.3475 ,          0.3842 ,          0.3798"
1035  "      DATA       0.2946 ,          0.3130 ,          0.3110"
1036  "      DATA       0.2344 ,          0.2263 ,          0.2031"
1037  "      DATA       0.1750 ,          0.1403 ,          0.0837"
```

```
1038  "       DATA        0.3013 ,          0.3356 ,          0.3755"
1039  "       DATA        0.2548 ,          0.2600 ,          0.2687"
1040  "       DATA        0.2087 ,          0.1842 ,          0.1618"
1041  "       DATA        0.3977 ,          0.4840 ,          0.5424"
1042  "       DATA        0.3750 ,          0.4343 ,          0.4541"
1043  "       DATA        0.3498 ,          0.3834 ,          0.3670"
1044  "       DATA        0.1613 ,          0.2376 ,          0.3056"
1045  "       DATA        0.1258 ,          0.1727 ,          0.2086"
1046  "       DATA        0.0934 ,          0.1082 ,          0.1119"
1047  "       DATA        0.1901 ,          0.2528 ,          0.3247"
1048  "       DATA        0.1588 ,          0.1993 ,          0.2374"
1049  "       DATA        0.1280 ,          0.1457 ,          0.1500"
1050  "       DATA        0.2607 ,          0.3681 ,          0.4915"
1051  "       DATA        0.2453 ,          0.3355 ,          0.4217"
1052  "       DATA        0.2300 ,          0.3038 ,          0.3523"
1053  "       DATA        0.1951 ,          0.2524 ,          0.2767"
1054  "       DATA        0.1491 ,          0.1794 ,          0.1747"
1055  "       DATA        0.1039 ,          0.1072 ,          0.0732"
1056  "       DATA        0.1957 ,          0.2643 ,          0.3400"
1057  "       DATA        0.1628 ,          0.2037 ,          0.2453"
1058  "       DATA        0.1296 ,          0.1433 ,          0.1505"
1059  "       DATA        0.2920 ,          0.4006 ,          0.5025"
1060  "       DATA        0.2713 ,          0.3648 ,          0.4262"
1061  "       DATA        0.2508 ,          0.3230 ,          0.3502"
1062  "       DATA        0.2254 ,          0.2785 ,          0.2834"
1063  "       DATA        0.1652 ,          0.1918 ,          0.1736"
1064  "       DATA        0.1056 ,          0.1059 ,          0.0643"
1065  "       DATA        0.2256 ,          0.2974 ,          0.3642"
1066  "       DATA        0.1790 ,          0.2216 ,          0.2574"
1067  "       DATA        0.1328 ,          0.1461 ,          0.1506"
1068  "       DATA        0.3322 ,          0.4287 ,          0.5197"
1069  "       DATA        0.3034 ,          0.3786 ,          0.4327"
1070  "       DATA        0.2749 ,          0.3287 ,          0.3456"
1071        '
1072        'INPUT PARAMETERS USED IN CREATING THE DATA BASE
1073        '
1074  "       DATA 18, 0.0100, 25000, 100000, 0.0075"
1075  "       DATA 12, 0.0060, 15000,  70000, 0.0025"
1076  "       DATA  6, 0.0020,  5000,  40000, 0.0000"
1077        END
1078  13685 RETURN
1079        '*****************************************************************
1080  13700 'SUBROUTINE FOR POLYNOMIAL INTERPOLATION
1081        '*****************************************************************
1082        ' X - VALUE OF THE PARAMETER
1083        ' X1(I) - PARAMETER VALUES USED IN THE DATA BASE
1084        ' Y1(I) - RUT DEPTHS FROM DATA BASE CORRESPONDING TO PARAMETER
1085        ' Y - INTERPOLATED VALUE
1086        ' NUM - NUMBER OF LEVELS OF THE PARAMETER
1087        '
1088        NS = 1
1089        DIF = ABS(X - X1(1))
1090        FOR a = 1 TO NUM
1091        DIFT = ABS(X - X1(a))
1092        '
1093        ' SELECT THE BEST STARTING POINT
1094        '
1095        IF DIFT < DIF THEN : NS = a: DIF = DIFT
```

```
1096        C(a) = Y1(a): DD(a) = Y1(a)
1097        NEXT a
1098        XFLAG = 0
1099        '
1100        'ENFORCE LIMITS ON EXTRAPOLATION (MAXIMUM = 1.5 * DIFFERENCE BETWEEN TWO
1101        'CONSECUTIVE PARAMETER LEVELS)
1102        '
1103        IF DIF > 1.5 * (ABS(X1(1) - X1(2))) THEN GOTO 13940 ELSE GOTO 13980
1104  13940 DIF = 1.5 * (ABS(X1(1) - X1(2)))
1105        IF NS = 1 THEN : X = X1(1) + DIF
1106        IF NS = NUM THEN : X = X1(NUM) - DIF
1107        XFLAG = X
1108  13980 Y = Y1(NS)
1109        NS = NS - 1
1110        BEND = 1
1111        FOR B = 1 TO BEND
1112        AEND = NUM - B
1113        FOR a = 1 TO AEND
1114        HO = X1(a) - X
1115        HP = X1(a + B) - X
1116        W = C(a + 1) - DD(a)
1117        DEN = HO - HP
1118        DEN = W / DEN
1119        '
1120        ' D - CORRECTION FROM THE LOWER LEVEL
1121        ' C - CORRECTION FROM THE UPPER LEVEL
1122        '
1123        DD(a) = HP * DEN
1124        C(a) = HO * DEN
1125        NEXT a
1126        '
1127        ' PICK THE SHORTEST PATH TO MOVE
1128        '
1129             IF (2 * NS) < AEND THEN : DY = C(NS + 1): GOTO 14210:
ELSE GOTO 14200
1130  14200 DY = DD(NS): NS = NS - 1
1131  14210 Y = Y + DY
1132        NEXT B
1133        RETURN
1134  14240 '*************************************************************************
1135        ' SUBROUTINE INPUT1
1136        '*************************************************************************
1137        '
1138        CLS
1139        'USE IF ONLY THE RUT LEVELS ARE NEEDED
1140        '
1141        IF INIOPT <> 2 GOTO 14400
1142  "      CLS : INPUT ""JOB DESCRIPTION :""; AA$"
1143  "      INPUT ""Resilient Modulus - Base Course (psi)""; EBA"
1144  "      INPUT ""Resilient Modulus - Subgrade (psi)""; ESG"
1145  "      INPUT ""Thickness of Base Layer (in)""; TBA"
1146  "      INPUT ""# of Equivalent Standard Wheel (9000 lbs) Passes""; EQPASS"
1147  "      INPUT ""Allowable Rut Depth (in.)""; RALLOW"
1148  "      INPUT ""Existing Rut Depth (in.)""; REXIST"
1149  14380 CLS
1150        '
1151  14400 CLS
1152  "       LOCATE 4, 10: PRINT "" INPUT DATA OPTIONS FOR RUT DEPTH CALCULATION:
```

```
“””
1153  “      LOCATE 6, 10: PRINT “” 1) Require subgrade material”””
1154  “      LOCATE 7, 10: PRINT “”    classification to determine approximate “””
1155  “      LOCATE 8, 10: PRINT “”    rutting potentials.”””
1156  “      LOCATE 10, 10: PRINT “” 2) Require laboratory data on residual deforma-
“””
1157  “      LOCATE 11, 10: PRINT “”    tion behavior of base and subgrade.”””
1158  “      LOCATE 15, 10: INPUT “”OPTION: 1=SOIL CLASS, 2=LAB DATA “”; OPP”
1159        ‘
1160        IF OPP = 2 THEN GOTO 14590
1161        IF OPP <> 1 THEN GOTO 14380
1162        ‘
1163        REM ALL BASE COURSE MATERIALS ARE CLASSIFIED AS ONE GROUP.
1164        ‘
1165  “      CLS : LOCATE 4, 10: PRINT “”Subgrade Material Type :”””
1166  “      LOCATE 7, 10: PRINT “”1) Heavy Clay - (CH)”””
1167  “      LOCATE 9, 10: PRINT “”2) Light/Silty Clay, Clayey Silt - (CL-ML)”””
1168  “      LOCATE 11, 10: PRINT “”3) Clayey/Silty/Uniform Sand - (SC-SM)”””
1169  “      LOCATE 15, 10: INPUT “”ENTER SELECTION & <RET> “”; MSG”
1170        GOTO 14750
1171  “14590 CLS : LOCATE 4, 10: PRINT “”Laboratory Data Input:”””
1172  “       LOCATE 6, 10: PRINT “”Log a - Intercept of the Straight Line Fit on a
“””
1173  “       LOCATE 7, 10: PRINT “”          Log-Log Plot of Accumulated Residual
Strain”””
1174  “      LOCATE 8, 10: PRINT “”        vs. Number of Load Repetitions”””
1175  “      LOCATE 10, 10: PRINT “”b     - Slope of the Straight Line Fit”””
1176        PRINT : PRINT
1177  “      INPUT “”Log a - Base Course Material””; LGABA”
1178  “      INPUT “”b     - Base Course Material””; BBA”
1179  “      INPUT “”Log a - Subgrade Material””; LGASG”
1180  “      INPUT “”b     - Subgrade Material””; BSG”
1181        IF INIOPT = 2 GOTO 14718
1182        CLS
1183        IF OPP <> 1 GOTO 14704
1184  “      LOCATE 16, 10: IF MSG = 1 THEN PRINT “”Subgrade      - CH - Clay”””
1185  “      LOCATE 16, 10: IF MSG = 2 THEN PRINT “”Subgrade      - CL-ML “””
1186  “      LOCATE 16, 10: IF MSG = 3 THEN PRINT “”Subgrade      - SC-SM”””
1187        GOTO 14714
1188        ‘
1189  “14704 LOCATE 16, 10: PRINT “”Base Course Material - “””
1190  “       LOCATE 17, 10: PRINT “”             Log a = “”; LGABA; “”,     b = “”;
BBA”
1191  “      LOCATE 19, 10: PRINT “”Subgrade Material    - “””
1192  “       LOCATE 20, 10: PRINT “”             Log a = “”; LGASG; “”,     b = “”;
BSG”
1193        ‘
1194  “14714 LOCATE 22, 10: INPUT “”DO YOU WANT ANY CORRECTIONS - 0=NO, 1=YES””;
CORR”
1195        IF CORR = 1 THEN GOTO 14380
1196        IF CORR <> 0 GOTO 14714
1197  14718 ABA = 10 ^ LGABA: ASG = 10 ^ LGASG
1198        IF ((ABA * 50000! ^ BBA) < .015) AND ((ASG * 50000! ^ BSG) < .016) GOTO
14750
1199  “      CLS : BEEP: PRINT “”Input Data are Incompatible. Check & Re-enter!””:”
1200        GOTO 14400
1201        ‘
1202  14750 ‘ SKIP IF ONLY RUT LEVELS ARE CALCULATED
```

```
1203        IF INIOPT = 2 GOTO 14840
1204        '
1205        ' OBTAIN THE TRUCK PASSES AND ALLOWABLE AND MEASURED RUT LEVELS FROM
1206        ' THE MAIN PROGRAM
1207        '
1208        EQPASS = PA
1209        RALLOW = RX
1210        REXIST = RM
1211  14840 RETURN
1212  14850 '*****************************************************
1213        'SUBROUTINE INPUT2 ————————————————
1214        '*****************************************************
1215        'SKIP IF ONLY RUT LEVELS ARE CALCULATED
1216        IF INIOPT = 2 GOTO 14940
1217        '
1218         EBA = E1(INC)
1219         ESG = E2(INC)
1220         TBA = BA(INC)
1221  14940 IF OPP <> 1 GOTO 15280
1222        '
1223        REM APPROXIMATE METHOD TO DETERMINE RUTTING BEHAVIOR
1224        '
1225        ' e(p) = a* N^b
1226  "      ' a - FIRST CYCLE STRAIN      (ABA,ACH,ACL,ASM)"
1227  "      ' b - RESIDUAL DEFORMATION RATE  (BBA,BCH,BCL,BSM)"
1228        ' N - # OF WHEEL PASSES  (CYL)
1229        ' a = aa * Mr^ab
1230  "      ' aa - COEFFICIENT   (AABA,AACH,AACL,AASM)"
1231  "      ' ab - EXPONENT   (ABBA,ABCH,ABCL,ABSM)"
1232  "      ' Mr - RESILIENT MODULUS (ksi)    (EBA,ESG)"
1233        '
1234        AABA = .0174: AACH = .0933: AACL = .001: AASM = .095
1235        ABBA = -.57: ABCH = -2.64: ABCL = -.73: ABSM = -1.95
1236        BBA = .125: BCH = .236: BCL = .162: BSM = .142
1237        '
1238  "      'CALCULATE ""a"" AND ""b"" (""a"" IS KEPT WITHIN PRACTICAL LIMITS)"
1239        '
1240        ABA = AABA * (EBA / 1000) ^ ABBA
1241        IF ABA > .0035 THEN ABA = .0035
1242        IF MSG <> 1 GOTO 15190
1243        BSG = BCH
1244        ASG = AACH * (ESG / 1000) ^ ABCH
1245        IF ASG > .004 THEN ASG = .004
1246  15190 IF MSG <> 2 GOTO 15230
1247        BSG = BCL
1248        ASG = AACL * (ESG / 1000) ^ ABCL
1249        IF ASG > .001 THEN ASG = .001
1250  15230 IF MSG <> 3 GOTO 15270
1251        BSG = BSM
1252        ASG = AASM * (ESG / 1000) ^ ABSM
1253        IF ASG > .005 THEN ASG = .005
1254  15270 '
1255  15280 REM USE IF ONLY RUT LEVELS ARE NEEDED
1256        '
1257        IF INIOPT <> 2 GOTO 15550
1258        CLS : REM DISPLAY INPUT DATA
1259  "      LOCATE 2, 10: PRINT ""JOB : ""; AA$"
1260  "      LOCATE 4, 10: PRINT ""Resilient Modulus (psi) - Base     = ""; EBA"
```

```
1261  "       LOCATE 6, 10: PRINT ""                          - Subgrade = ""; ESG"
1262  "       LOCATE 8, 10: PRINT ""Thickness of Base Layer            = ""; TBA; ""
in."""
1263  "         LOCATE 10, 10: PRINT ""# of Equivalent Standard Wheel Passes ="";
EQPASS"
1264  "       LOCATE 12, 10: PRINT ""Allowable Rut Depth =""; RALLOW; "" in."""
1265  "       LOCATE 14, 10: PRINT ""Measured Rut Depth =""; REXIST; "" in."""
1266        IF OPP <> 1 GOTO 15460
1267  "       LOCATE 16, 10: IF MSG = 1 THEN PRINT ""Subgrade      - CH - Clay"""
1268  "       LOCATE 16, 10: IF MSG = 2 THEN PRINT ""Subgrade      - CL-ML """
1269  "       LOCATE 16, 10: IF MSG = 3 THEN PRINT ""Subgrade      - SC-SM"""
1270        GOTO 15510
1271        '
1272        '
1273  "15460 LOCATE 16, 10: PRINT ""Base Course Material - """
1274  "       LOCATE 17, 10: PRINT ""             Log a = ""; LGABA; "",      b = "";
BBA"
1275  "       LOCATE 19, 10: PRINT ""Subgrade Material    - """
1276  "       LOCATE 20, 10: PRINT ""             Log a = ""; LGASG; "",      b = "";
BSG"
1277        '
1278  "15510 LOCATE 22, 10: INPUT ""DO YOU WANT ANY CORRECTIONS - 0=NO, 1=YES"";
CORR"
1279        IF CORR = 1 THEN RETURN
1280        IF CORR <> 0 GOTO 15510
1281        '
1282  15550 'ASSIGN VALUES TO INTERPOLATION PARAMETERS
1283        '
1284        CYL = 300000!
1285        XII = ABA * CYL ^ BBA
1286        XLL = ASG * CYL ^ BSG
1287        IF XII > .015 THEN XII = .015
1288        IF XLL > .016 THEN XLL = .016
1289        IF XLL < .001 AND TBA < 10 THEN XLL = .001
1290        '
1291        '
1292        XMM = TBA: XJJ = EBA: XKK = ESG
1293        IF TBA < 10 AND ESG > 20000! AND EBA > 100000 THEN XJJ = 100000
1294        IF TBA < 10 AND EBA < 60000 AND ESG > 30000 THEN XKK = 30000: IF TBA < 6
THEN XMM = 6: IF EBA < 30000 THEN XJJ = 30000
1295        RETURN
1296        '*********************************************************************
1297  15680 'SUBROUTINE OUTPUT
1298        '*********************************************************************
1299        '
1300        IF RUTCAL < 0 THEN RUTCAL = 0!
1301        RFINAL(INC) = RUTCAL * EQPASS / CYL
1302        REXTRA = RALLOW - REXIST
1303        IF RUTCAL = 0 THEN passes(INC) = 0: GOTO 15760
1304        passes(INC) = CYL * REXTRA / RUTCAL
1305  15760 IF INIOPT <> 2 GOTO 15860
1306        CLS
1307        IF EQPASS = 0 GOTO 15810
1308  "       LOCATE 6, 10: PRINT ""EQUIVALENT STANDARD WHEEL PASSES =""; EQPASS"
1309  "         LOCATE 8, 10: PRINT ""RUT DEPTH CAUSED          = ""; RFINAL(1); ""
in."""
1310  "15810 LOCATE 10, 10: PRINT ""ALLOWABLE RUT DEPTH     = ""; RALLOW; "" in."""
1311  "       LOCATE 12, 10: PRINT ""EXISTING RUT DEPTH      = ""; REXIST; "" in."""
```

```
1312  "       LOCATE 14, 10: PRINT ""ALLOWABLE WHEEL PASSES  = ""; passes(1)"
1313  "         IF passes(1) = 0 THEN : LOCATE 16, 5: PRINT ""INPUT DATA MAY BE
INCOMPATIBLE. CHECK & RERUN!"""
1314         PRINT : PRINT : PRINT
1315  "15860 IF INIOPT = 2 THEN : INPUT ""      PRESS RETURN TO CONTINUE ......."";
PRESS"
1316         CLS
1317         RETURN
```