TTI-2-18-83-332-1

TEXAS
TRANSPORTATION
INSTITUTE

STATE DEPARTMENT
OF HIGHWAYS AND
PUBLIC TRANSPORTATION

COOPERATIVE
RESEARCH

ANALYSIS OF TRUCK USE AND HIGHWAY COST ALLOCATION IN TEXAS

in cooperation with the
Department of Transportation
Federal Highway Administration

RESEARCH REPORT 332-1
STUDY 2-18-83-332
TRUCK USE AND HIGHWAY COST ALLOCATION

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA/TX-85/41+332-1		

4. Title and Subtitle	5. Report Date June 1985 6. Performing Organization Code
Analysis of Truck Use and Highway Cost Allocation in Texas	
7. Author(s) Alberto Garcia-Diaz, Dock Burke, and Arturo Villarreal-Cavazos	8. Performing Organization Report No. Research Report 332-1
9. Performing Organization Name and Address Texas Transportation Institute The Texas A&M University System College Station, TX 77843	10. Work Unit No. 11. Contract or Grant No. Study No. 2-18-83-332
12. Sponsoring Agency Name and Address Texas State Department of Highways and Public Transportation; Transportation Planning Division P.O. Box 5051 Austin, TX 78763	13. Type of Report and Period Covered Interim - September 1982 June 1985 14. Sponsoring Agency Code

15. Supplementary Notes
Research performed in cooperation with DOT and FHWA.
Research Study Title: Analysis of Truck Use and Highway Cost Allocation in Texas.

16. Abstract
The highway cost allocation problem is one of determining equitable charges for each of the vehicle classes sharing transportation facilities such as highways and bridges. Previous attempts at solving this problem can essentially be reduced to two major approaches: (a) proportional allocation methods, which determine costs in proportion to one or more measures of highway usage; and (b) incremental methods, which allocate costs on the basis of highway design differences necessary to accommodate gradually heavier vehicle classes.

This report develops two new highway cost allocation methodologies that actually extend the basic concepts of the incremental and proportional allocation procedures. The new methods are referred to as the "Modified Incremental Approach" and the "Generalized Method." Both methods fulfill the following conditions: (a) highway costs are completely financed by users (completeness condition); (b) vehicle classes reduce their cost responsibilities by sharing the facilities with other vehicle classes (rationality principle); and (c) vehicle classes are charged at least enough to cover their corresponding marginal costs (marginality principle). An example using Texas pavement data is utilized to illustrate the application of the proposed methods.

17. Key Words	18. Distribution Statement
Highway Cost Allocation, Truck Use, Highway Design Differences, Vehicle Classes.	No restrictions. This document is available to the public through the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	unclassified	185	

Form DOT F 1700.7 (8-69)

ANALYSIS OF TRUCK USE AND HIGHWAY COST ALLOCATION IN TEXAS

Alberto Garcia-Diaz
Dock Burke
Arturo Villarreal-Cavazos

Research Report 332-1

Research Study 2-18-83-332

Sponsored by

Texas State Department of Highways and Public Transportation

in cooperation with

The U. S. Department of Transportation
Federal Highway Administration

June 1985

Texas Transportation Institute
The Texas A&M University System
College Station, TX 77843

ABSTRACT

The highway cost allocation problem is one of determining equitable charges for each of the vehicle classes sharing transportation facilities such as highways and bridges. Previous attempts at solving this problem can essentially be reduced to two major approaches: (a) proportional allocation methods, which determine costs in proportion to one or more measures of highway usage; and (b) incremental methods, which allocate costs on the basis of highway design differences necessary to accommodate gradually heavier vehicle classes.

This report develops two new highway cost allocation methodologies that actually extend the basic concepts of the incremental and proportional allocation procedures. The new methods are referred to as the "Modified Incremental Approach" and the "Generalized Method". Both methods fulfill the following conditions: (a) highway costs are completely financed by users (completeness condition); (b) vehicle classes reduce their cost responsibilities by sharing the facilities with other vehicle classes (rationality principle); and (c) vehicle classes are charged at least enough to cover their corresponding marginal costs (marginality principle). An example using Texas pavement data is utilized to illustrate the application of the proposed methods.

IMPLEMENTATION STATEMENT

The new cost allocation methodologies developed in this project have been computerized and tested using limited rehabilitation data from the Texas pavement data base. Proposed changes to the RENU2 program have been implemented and are currently being validated.

DISCLAIMER

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

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

An important problem currently receiving a great deal of attention from state legislatures is the highway financing problem. In recent years, this problem has become more acute due to the fact that a significant portion of U. S. highway pavements is reaching unacceptable levels of user serviceability.

Two basic questions must be answered by a highway cost allocation procedure: (a) how much is needed to keep a highway (or other transportation facilities) operational during a specified planning horizon? And (b) what fraction of the total cost must be charged to each vehicle class in the traffic stream served by the facility? This paper summarizes the results of recent work aimed at providing adequate answers to both of these questions. In particular, two new cost allocation methods are developed: a modified incremental approach and an optimization method; the optimization method will be referred to as a generalized procedure, since in fact it is based on an extension of the concepts used in the incremental [4,10,11] and proportional allocation [6,7,10] methods.

Proportional allocation methods determine cost responsibilities on the basis of a measure that reflects the amount of use of a highway facility by each of the various vehicle classes. Common measures include gross vehicle weight, vehicle-miles of travel, and equivalent single axle loads (ESALs). It must be noted that these methods may yield results that conflict with the perception of fairness by individual vehicle classes; this indeed hinders the acceptability of the results by all the users of the facility and questions the overall applicability of the proportional methods.

Incremental allocation methods identify cost responsibilities on the basis of the cost differences associated with the sequential introduction of vehicle classes into the traffic stream. Inconsistent results are obtained when vehicle classes are introduced in different sequences, however. This inconsistency constitutes a serious flaw in any cost allocation method that seeks to be equitable.

The two procedures developed in this paper exhibit properties which make them superior to those previously used in the context of highway facility planning. In particular, they fulfill three fundamental requirements:

(a) **Completeness**: the provision of highway facilities must be entirely financed by the various vehicle classes that utilize them.

(b) **Rationality**: the common facility is the most economically attractive alternative for all vehicle classes to meet their transportation needs; that is, any other alternative to satisfy this need, such as using an exclusive facility, would be more expensive for any vehicle class.

(c) **Marginality**: the allocated costs associated with any vehicle class must be sufficient to at least cover its corresponding marginal costs.

The completeness requirement insures that only funds provided by highway users are considered to finance the common highway facility. The rationality requirement is a well established concept in the economics literature [16] which deals with a fundamental characteristic of economic behavior. Any procedure which violates this condition would be strongly objected. The marginality requirement is another widely accepted economical principle [19]. The violation of this principle implies the existence of cross-subsidization among the vehicle classes involved. The rationality and marginality requirements establish an essential element of equity in the cost allocation procedure.

In conclusion, having an equitable cost allocation methodology (which satisfies the rationality and marginality principles) to analyze the many aspects related to the highway financing problem enhances the acceptability of the results among the various vehicle classes which must cover the total cost of the facility. In Chapter 2 this important issue is briefly discussed for both the proportional and the incremental methodologies.

This report presents a summary of the relevant previous work related to highway cost allocation procedures in Chapter 2. A brief description of the RENU2 program is given in Chapter 3. Chapter 4 delineates the basic methodology developed for this study. Such methodology includes two cost allocation procedures -the Modified Incremental Method and the Generalized Method- and a procedure to study the effect of environmental factors on highway costs. Applications of the proposed methodology using Texas pavement data are given in Chapter 5.

2. LITERATURE SURVEY

Currently available solution procedures for the highway cost allocation problem are not economically justifiable. Indeed, perhaps a non-controversial solution methodology to that problem does not exist; nevertheless, cost must be allocated in some rational way. Traditionally, it has been an accepted practice to define cost responsibilities on the basis of some criterion of efficiency which represents the use of the facility by the various vehicle classes.

One of the most widely used methods in highway cost allocation is the so-called incremental approach, which was adopted in the earlier cost allocation studies conducted in the United States. This approach was adequate while new construction was the principal cause of highway cost. However, now that a larger portion of the budget must be assigned to the maintenance and rehabilitation of existing facilities, the incremental approach has been reviewed and questioned, and some important problems, which will be discussed later, have been found.

The incremental method has been used in a number of cost allocation studies such as the first Federal Highway Cost Allocation Study [4], and studies conducted in several states including Virginia, Washington, North Dakota, Montana, Kentucky, and Rhode Island [3,10,11,12,13,15].

According to the incremental method, the cost of a highway facility designed for the lightest vehicle class is initially calculated; then vehicle classes are sequentially included in increasing order of axle weight and corresponding highway design or rehabilitation costs are calculated for the resulting traffic streams and a specified design period. At each step of the procedure, the cost difference between one design and the next is allocated to the vehicle class incorporated in that step. Some minor variations of the basic incremental method have also been considered [6].

Although it meets the completeness, rationality, and marginality requirements aforementioned, there is one important difficulty with the incremental method: it is not consistent. The method produces different results if vehicle classes are introduced in different orderings. This is due to the presence of overlapping requirements among the various vehicle classes. Figure 2.1 illustrates this inconsistency. The shaded areas in Figures 2.1(a), 2.1(b), and 2.1(c) represent costs allocated to vehicle classes 1, 2, and 3 if they are sequentially introduced in that order. However, the shaded areas in Figures 2.1(d), 2.1(e), and 2.1(f) represent the same costs when vehicle class 3 is included first, followed by vehicle classes 1 and 2.

Another accepted approach to the problem under consideration is to allocate costs in proportion to a single numerical criterion which, in the context of transportation systems, represents a measure of use or damage caused by the vehicle classes using a common highway facility. This method is known as the proportional method or the consumption approach [6,7]. The appeal of this method lies on its simplicity and on the fact that, if the appropriate basis is selected, the fairness of its results is less open to dispute.

A major issue with the proportional allocation method, however is that it may yield cost allocations which conflict with the interests of the individual vehicle classes. This difficulty is due to the fact that the method ignores the strategic alternatives (coalitions) available to the vehicle classes in order to meet their transportation needs. Such strategic alternatives include sharing a common facility with all vehicle classes, sharing a facility with some of the other vehicle classes, and having an exclusive facility. In other words, under the proportional allocation me-

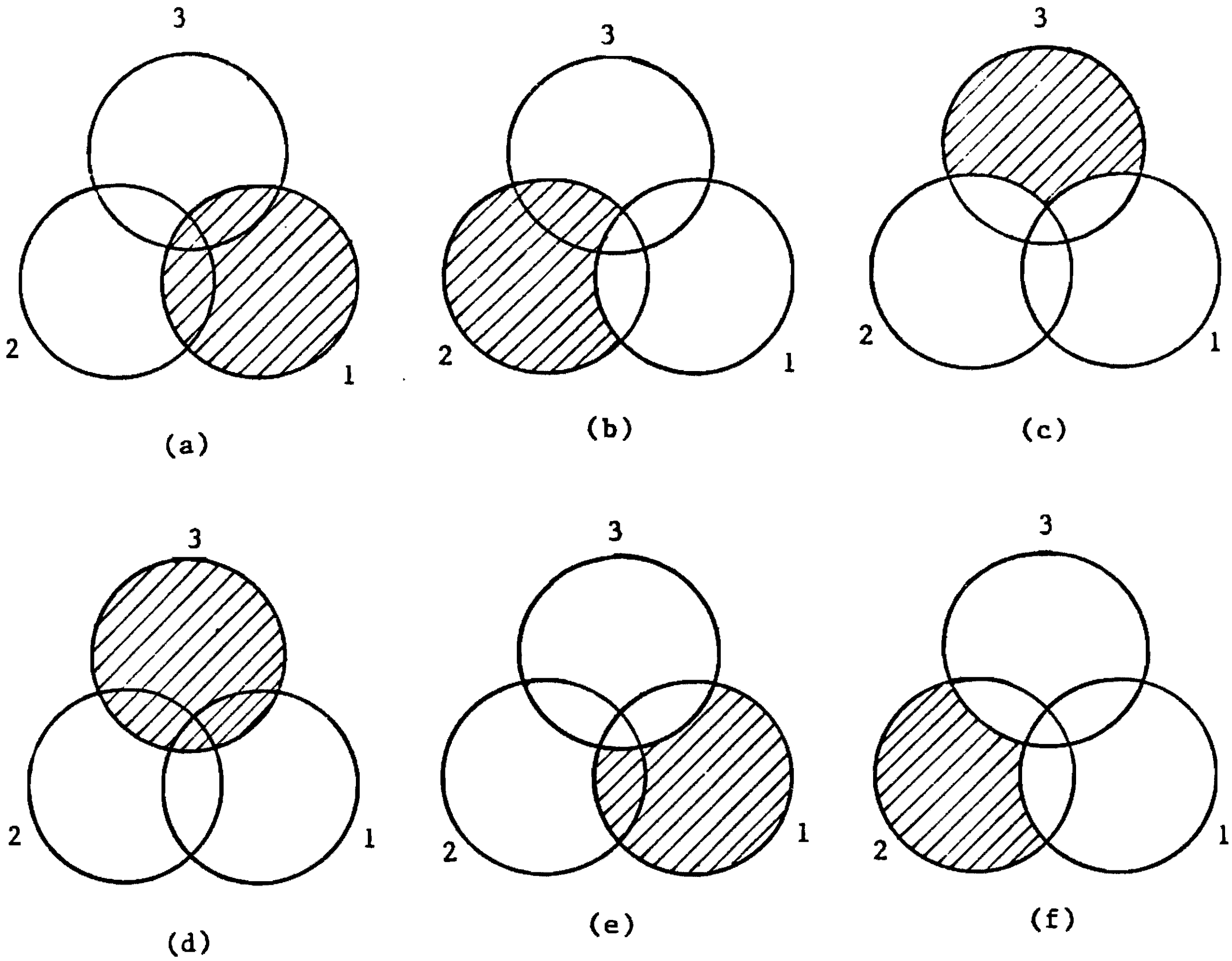


Figure 2.1 Basic Approach of the Incremental Method

thod, it is possible for a particular vehicle class to pay more by sharing a common facility than it would have to pay by having its own exclusive one.

In a pioneering and enlightening article, Young, Okada, and Hashimoto [18] analyze several cost allocation methods used in water resources management. Among the methods discussed, those that stem from the theory of cooperative games [16,17] are of particular interest. These methods provide means for approaching the cost allocation problem by taking into account all the possible strategic alternatives available to each vehicle class in the provision of highway facilities needed to meet a specified traffic demand. These various strategic possibilities actually establish constraints which define a set of feasible solutions that satisfy the completeness, rationality and marginality requirements. The cost allocations resulting from these methods are more likely to be accepted because they are formulated on the basis of fundamental economic principles.

3. PAVEMENT REHABILITATION: THE RENU2 PROGRAM

The RENU2 program [8] estimates the maintenance and rehabilitation costs associated with changes in legal load limits. In this program, the number of equivalent loads (ESAL) is calculated for both current and proposed legal limits from given traffic compositions and axle distributions. Based on the resulting ESALs, life cycles of typical pavements are analyzed and rehabilitation and maintenance costs are estimated. The RENU2 program essentially performs five functions:

(a) Pavement performance function.

(b) Pavement survival function.

(c) Pavement age adjustment function.

(d) Load shifting function.

(e) Cost estimating function.

The rest of this chapter will be dedicated to the discussion of each of these points.

3.1 Pavement Performance Function

The pavement performance function predicts the deterioration trend of a pavement in terms of the loss of PSI (present serviceability index) or the increase in area or severity of a distress (cracking, rutting, flushing, etc.) as the level of traffic loads increases. In this function, the life cycle of a pavement is identified for given traffic conditions. It is assumed that a terminal performance index (either a minimal PSI value or maximal distress area/severity values) is specified and that the life cycle of a pavement is completed when this critical value is reached.

A functional form that has been found to adequately represent the loss of PSI for Texas highways is:

$$g_t = e^{-(\rho/W)^{\beta}} \qquad (3.1)$$

where

W: No. of cumulative ESALs,

ρ: Scale parameter, and

β: Form parameter

The damage function g(W) can also be expressed as the ratio of the loss in serviceability after W 18-kip ESALs to a specified maximum design loss.

Let P_0 be the initial PSI (at W = 0), P_t be the PSI after W_t 18-kip ESALs, and P_f be a lower bound on the PSI. Then the relative loss after W_t ESALs can be expressed as:

$$g_t = (P_0-P_t)/(P_0-P_f) \qquad (3.2)$$

From Eq. (3.2) it is possible to express P_t as a function of g_t, as follows:

$$P_t = P_0-(P_0-P_f)g_t \qquad (3.3)$$

Eq. (3.3) can be further rewritten after using Eq. (1). The final result is given by:

$$P_t = P_0-(P_0-P_f)e^{-(\rho/W)^{\beta}} \qquad (3.4)$$

Figure 3.1(a) shows the form of the loss of PSI (P_t) as a function of the cummulative number of ESALs (W) according to Eq. (3.4). Note that P_f

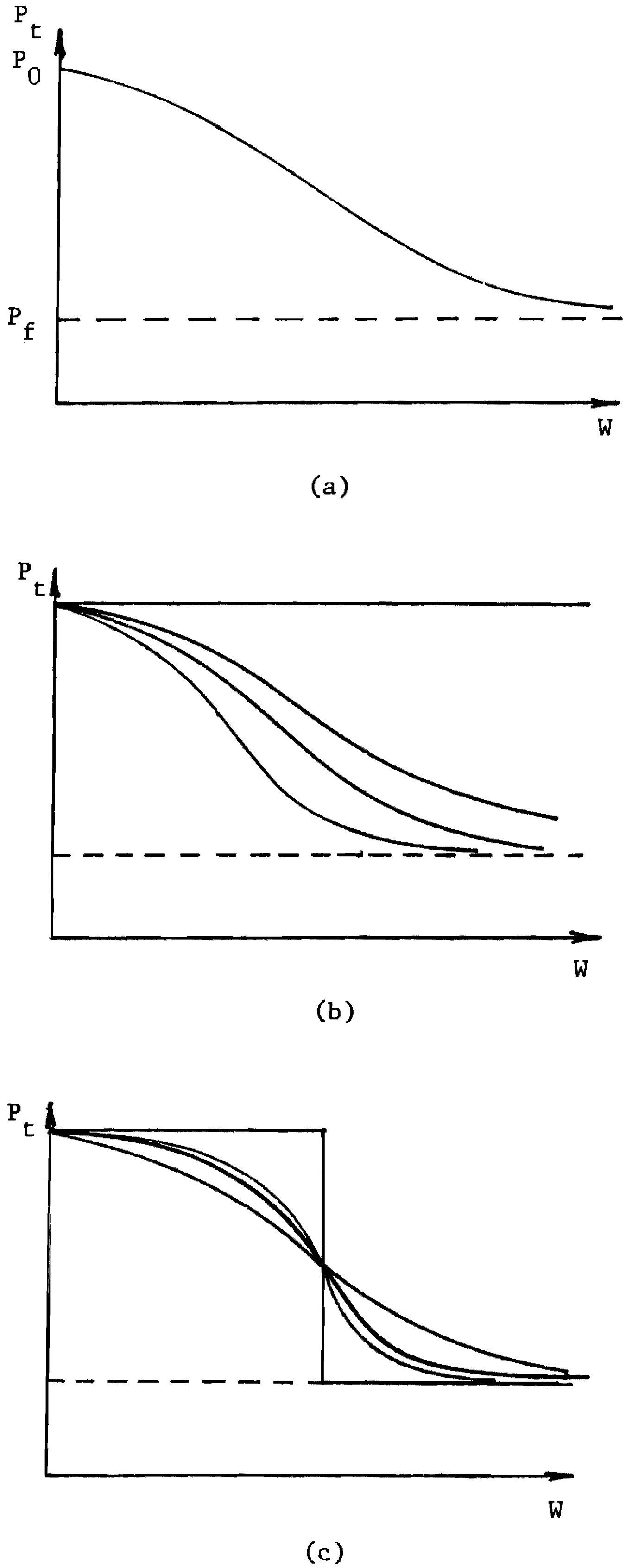


Figure 3.1 Performance Curves

represents an asymptotic minimum PSI value. Figures 3.1(b) and 3.1(c) show, respectively, the influence of parameters ρ and β on the form of the function.

The Texas Transportation Institute has estimated, through statistical procedures, values for P_f, ρ and β from measured pavement data [9]. Table 3.1 indicates mean, maximum, and minimum values of these parameters for hot mix, black base, and overlaid pavements.

Very frequently pavements may be seriously distressed an in need of major rehabilitation before the seviceability index drops to its terminal value. This is particularly true of pavements with severe alligator and transverse cracks. In cases where the asymptotic serviceability index, P_f, is higher than the terminal serviceability index, P_t, or when the remaining life calculated from the serviceability index equation is excessively long (say 30 to 40 years), the pavement will probably need major rehabilitation due to distress.

Pavement distress can appropriately be represented by estimating two separate components: density and severity. Density may be expressed either as the percent of the total pavement surface area that is covered by the distress, or total crack length per unit area, or crack spacing, or similar measures. Severity may be expressed as either an objective or subjective measure. Examples of objective measures are crack width, crack depth, and relative displacement at joint. Subjective measures may be assessed reliably by comparing the observed distress with photographs of different levels of severity which may be described as none, slight, moderate, or severe and may be given numerical ratings such as 0, 1, 2, and 3, respectively, or be assigned numbers that are proportional to these in a range between 0 and 1. The change of either area or severity of distress can be evaluated using the previously discussed equations.

Table 3.1 Serviceability Performance Curve Parameters by Pavement Type

Pavement Type	Black Base	Hot Mix Asphalt concrete	Overlays
Number of Test Sections	51	36	77
ρ (mean)	2.321	1.960	1.974
ρ (min)	0.005	0.100	0.013
ρ (max)	17.239	11.098	9.188
β (mean)	1.337	1.952	1.196
β (min)	0.300	0.095	0.095
β (max)	6.277	7.259	2.893
P_o (mean)	4.15	3.87	3.92
P_o (min)	2.79	2.86	2.07
P_o (max)	4.77	4.78	4.88
P_f (mean)	1.962	1.661	2.121
P_f (min)	0.000	0.000	0.004
P_f (max)	4.295	4.305	4.391

In order to study the behavior of the area covered by a given type of distress, and the corresponding level of severity, two indices will be introduced: (a) the distress area index, and (b) the distress severity index. Each of these indices represents a number between 1 and 0 which <u>decreases</u> as the level of traffic is increased. Note that the present serviceability index (PSI) has a similar behavior, with the exception that it decreases from P_0 to P_f.

Specifically, the distress area index decreases from a value A_0 ($A_0 \le 1$) to a value A_f ($0 \le A_f \le A_0$) as the traffic increases; similarly, the distress severity index decreases from a value of S_0 ($S_0 \le 1$) to a value S_f ($0 \le S_f \le S_0$) as the traffic level increases; that is, a recently rehabilitated pavement will have indices close to one, as opposed to pavements in need of rehabilitation which will have indices close to zero.

The distress area index, A, is expressed by a relationship similar to that of Eq. (3.4), namely,

$$A = A_0 - (A_0 - A_f) e^{-(\rho/W)^{\beta}} \qquad (3.5)$$

Similarly the distress severity index, S, is expressed as

$$S = S_0 - (S_0 - S_f) e^{-(\rho/W)^{\beta}} \qquad (3.6)$$

Using the A, S, and W data from the Texas Transportation Institute data base, the parameters and have been estimated for the most significant distress types affecting black base, hot mix, and overlaid pavements which are, respectively, alligator cracking area, alligator cracking severity, and transverse cracking severity [1]. Table 3.2 summarizes the results obtained [9]. .cp 5

Table 3.2 Primary Distress Type and Curve Fit Parameters by Pavement Type

Pavement Type	Black Base	Hot Mix Asphalt concrete	Overlays
Type of Distress	Alligator Cracking Severity	Alligator Cracking Area	Transverse Cracking Severity(*)
ρ (mean)	1.19	0.93	85.57
ρ (min)	0.14	0.07	24.13
ρ (max)	3.01	3.63	194.83
β (mean)	2.54	3.43	1.47
β (min)	0.89	0.50	0.50
β (max)	8.78	18.21	5.52

(*) The ρ and β terms for this case are determined in terms of the number of months the pavement has been in service.

3.2 Pavement Survival Function

The pavement survival function estimates the percent of miles of pavement that do not need to be rehabilitated when the pavement performance function indicates that the specified critical performance index is reached. From this information it is possible to subsequently estimate the predicted number of miles that will need to be rehabilitated after a given number of load applications.

A survivor curve is a functional relationship that predicts the percentage of mileage in a given pavement category that does not require immediate rehabilitation at a specified time. This specified time can be considered as the time at which the pavement has reached a given traffic load level, or the time since last rehabilitation. Evidently, to decide if a pavement requires or does not require some kind of rehabilitation, it is first necessary to define a measure of pavement performance. This measure of performance has been defined in terms of PSI or distress as shown in the previous sections. The fundamental idea behind the development of a survivor curve is the concept that since the performance relationship is deterministic, it would be meaningful to determine a second relationship that estimates the percent of pavement mileage that actually survives when the performance function reaches a critical value.

Survival times are data that measure the time to failure. These times are subject to random variations, and like any random variables, form a distribution; the two-parameter Weibull distribution [14] is assumed as the survival distribution for predicting the survival or failure rate of pavements. The Weibull distribution is one of the well-known survival distributions; its applicability to various failure situations, such as electron tube failure, the fatigue life of deep-groove ball bearings, etc., has been extensively investigated and recommended.

The Weibull distribution is characterized by two non-negative parameters λ and γ ; its probability density function, f(w), and the cumulative distribution function, F(w), are defined as follows:

$$f(w) = \lambda\gamma(\lambda w)^{\gamma-1} e^{-(\lambda w)^{\gamma}} \tag{3.7}$$

$$F(w) = 1-e^{-(\lambda w)^{\gamma}} \tag{3.8}$$

In the specific application of the Weibull distribution to the study of pavement survivability, w represents the traffic load at which the pavement reaches a critical performance level. The parameters λ and γ are referred to as a "scale parameter" and a "shape parameter", respectively.

The survival function, denoted by s(w), is defined as the probability that an individual mile of pavement of a given type survives a traffic load larger that w. From the definition of the cumulative distribution function F(w), it can be concluded that s(w) = 1-F(w). That is,

$$s(w) = e^{-(\lambda w)^{\gamma}} \tag{3.9}$$

Figure 3.2 illustrates the typical shape of the survival function. As explained here, s(w) is the survival rate of a given type of pavement structure under w traffic loads.

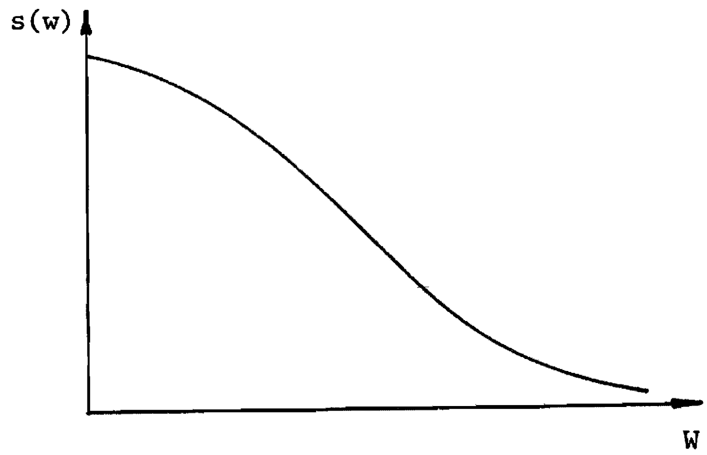


Figure 3.2 Typical Survival Curve

Survival curves have been obtained for Texas flexible pavements using different critical levels for PSI and the most relevant types of distress [9]. Table 3.3 shows λ and γ values for PSI survival curves corresponding to hot mix, black base and overlaid pavements. Table 3.4 gives λ and γ values associated with distress survival curves for the same types of pavements.

3.3 Pavement Age Adjustment Function

The pavement age adjustment function updates the age distribution of the pavement mileage when it is rehabilitated. Typical rehabilitation actions include regular thin overlays when the pavement fails because of several distress types or medium to thick overlays when it fails as a result of PSI loss.

Figure 3.3 summarizes the age adjustment procedure for a representative pavement section and a given year of a specified analysis period. The procedure can be described as follows:

Table 3.3 Design Parameters for PSI survivor curves

Black Base		
P_c	λ	γ
1.0	0.276	2.111
2.0	0.417	1.549
3.0	0.607	1.497
Hot Mix		
P_c	λ	γ
1.0	0.423	1.363
2.0	0.687	1.365
3.0	0.787	1.012
Overlays		
P_c	λ	γ
1.0	0.327	1.524
2.0	0.555	1.163
3.0	0.818	1.088

Table 3.4 Design Parameters for Distress Survivor Curves

	Black Base			
	$g_c = 0.25$		$g_c = 0.50$	
	λ	γ	λ	γ
Rutting				
Area	0.010	2.801	0.006	2.133
Severity	0.010	3.143	0.006	2.782
Alligator Cracking				
Area	0.006	3.065	0.003	2.129
Severity	0.007	4.380	0.004	2.681
Longitudinal Cracking				
Area	0.006	2.815	0.003	2.068
Severity	0.008	3.285	0.005	2.279
Transverse Cracking				
Area	0.006	2.760	0.003	1.878
Severity	0.008	3.382	0.004	2.443

Table 3.4 Design Parameters for Distress Survivor Curves (Cont'd)

	Hot Mix			
	g_c = 0.25		g_c = 0.50	
	λ	γ	λ	γ
Rutting				
Area	0.007	2.617	0.004	2.696
Severity	0.007	2.039	0.004	1.781
Alligator Cracking				
Area	0.006	3.304	0.004	3.343
Severity	0.007	3.227	0.005	2.610
Longitudinal Cracking				
Area	0.006	2.819	0.003	1.836
Severity	0.007	3.059	0.005	2.182
Transverse Cracking				
Area	0.006	2.111	0.004	1.696
Severity	0.008	2.551	0.005	2.129

Table 3.4. Design Parameters for Distress Survivor Curves (Cont'd)

	Overlays			
	$g_c = 0.25$		$g_c = 0.50$	
	λ	γ	λ	γ
Rutting				
Area	0.009	1.604	0.004	1.219
Severity	0.010	1.819	0.005	1.804
Alligator Cracking				
Area	0.007	2.575	0.003	2.080
Severity	0.009	2.280	0.005	2.056
Longitudinal Cracking				
Area	0.008	1.519	0.004	1.197
Severity	0.011	1.919	0.006	1.693
Transverse Cracking				
Area	0.008	1.792	0.004	1.397
Severity	0.011	1.916	0.006	1.797

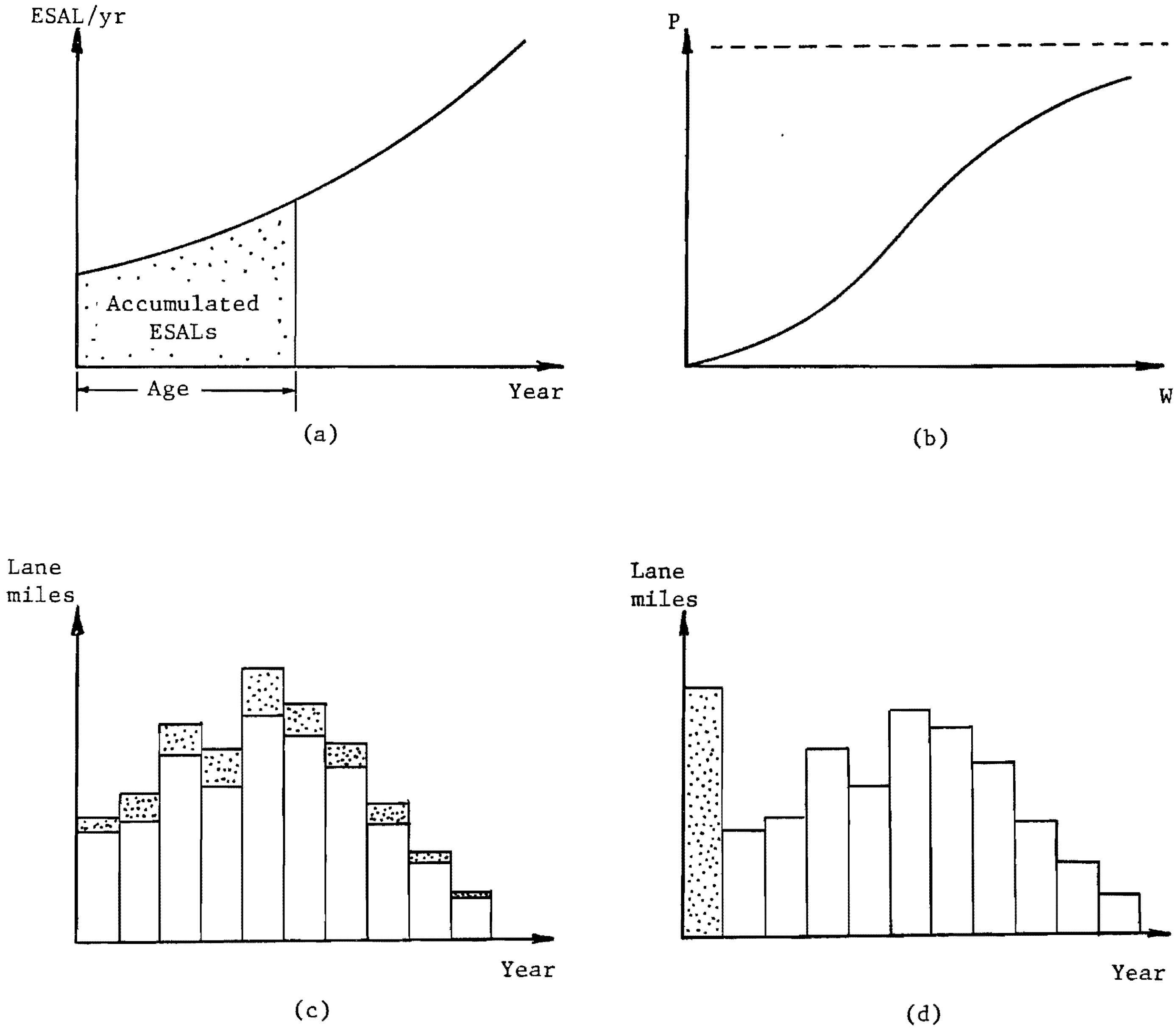


Figure 3.3 Age Distribution Adjustment

(a) Calculate the number of accumulated ESALs for each pavement age category as illustrated in Figure 3.3(a) from traffic data and forecasts.

(b) Obtain the expected fraction P of pavements to fail in the current year for each age category by using the number of accumulated ESALs calculated in the previous step and the appropriate survival curve as shown in Figure 3.3(b). The number of lane-miles that fail and hence are due for rehabilitation in the current year are indicated by the shaded portions of the rectangles representing the number of lane-miles in each pavement age category in Figure 3.3(c). The remaining portions of these rectangles represent the surviving pavement.

(c) The new age distribution is obtained by creating a new age category composed of the total number of lane-miles just rehabilitated and updating the ages of the lane-miles that survived, as indicated in Figure 3.3(d).

3.4 Load Shifting Function

The load shifting function is a procedure that modifies a given axle load distribution to reflect a change in the current legal load limits. This module can be used to establish the most likely truck traffic distribution that will occur on a highway system after changing the legal axle load limits. This function will not be used in this study.

3.5 Cost Estimating Function

The cost estimating function computes maintenance and rehabilitation costs for both the present and proposed legal load limits in order to determine the impact of the change in such limits. The program computes maintenance costs for both routine and preventive maintenance. All costs can be calculated for each year of a specified planning horizon and can be broken down according to several types of highway systems (i.e., Interstate, US, State, FM).

Rehabilitation activities considered in RENU2 consist of overlays with asphaltic concrete. When the pavement fails by reaching a critical value of PSI, the thickness of the overlay is calculated so that the pavement remains serviceable throughout the rest of the analysis period. If the pavement fails due to a critical value of distress index (area or serviceability), thin overlays are applied periodically. The time period between overlays is specified by the user in this case. The rehabilitation cost is a function of:

(a) The geometry of the road (lane width, shoulder width, etc.).

(b) The critical performance levels set for PSI and distress, which determine the timing of pavement rehabilitations.

(c) The overlay thicknesses.

Routine maintenance costs are estimated using the EAROMAR procedure [5] and cost information provided by the user. These equations were actually developed to predict maintenance work loads for multilane freeways in terms of: (a) patching, (b) crack sealing, and (c) base and surface repairs. The general form of the EAROMAR model can be formulated as follows:

$$C_t = (1100C_1+1000C_2+5C_3)/(1+e^{-(t-10)/1.16}) \qquad (3.10)$$

where

C_t = annual maintenance cost in year t per lane-mile,

C_1 = \$/sq yd of bituminous skin patching,

C_2 = \$/linear foot of crack sealing,

C_3 = \$/cu yd of bituminous base and surface repair.

Cost estimates by the Texas Highway Department for C_1, C_2, and C_3 are \$3.47, \$0.25 and \$450, respectively.

Preventive maintenance is implemented in terms of seal coats. These seal coats waterproof and improve the texture of the pavement. The user of the RENU2 program specifies the time between seal coats and the corresponding cost per lane mile. The preventive maintenance option is applicable to all flexible pavements and different values can be considered for each representative pavement section.

Cost estimates obtained from RENU2 are used to construct a cost function similar to that portrayed in Figure 3.4. This function represents costs incurred during a specified analysis period, and is dependent upon the number of cumulative ESALs acting on the pavement. The cost estimate for each vehicle class combination is obtained from Figure 3.4 by using the number of ESALs associated with that combination.

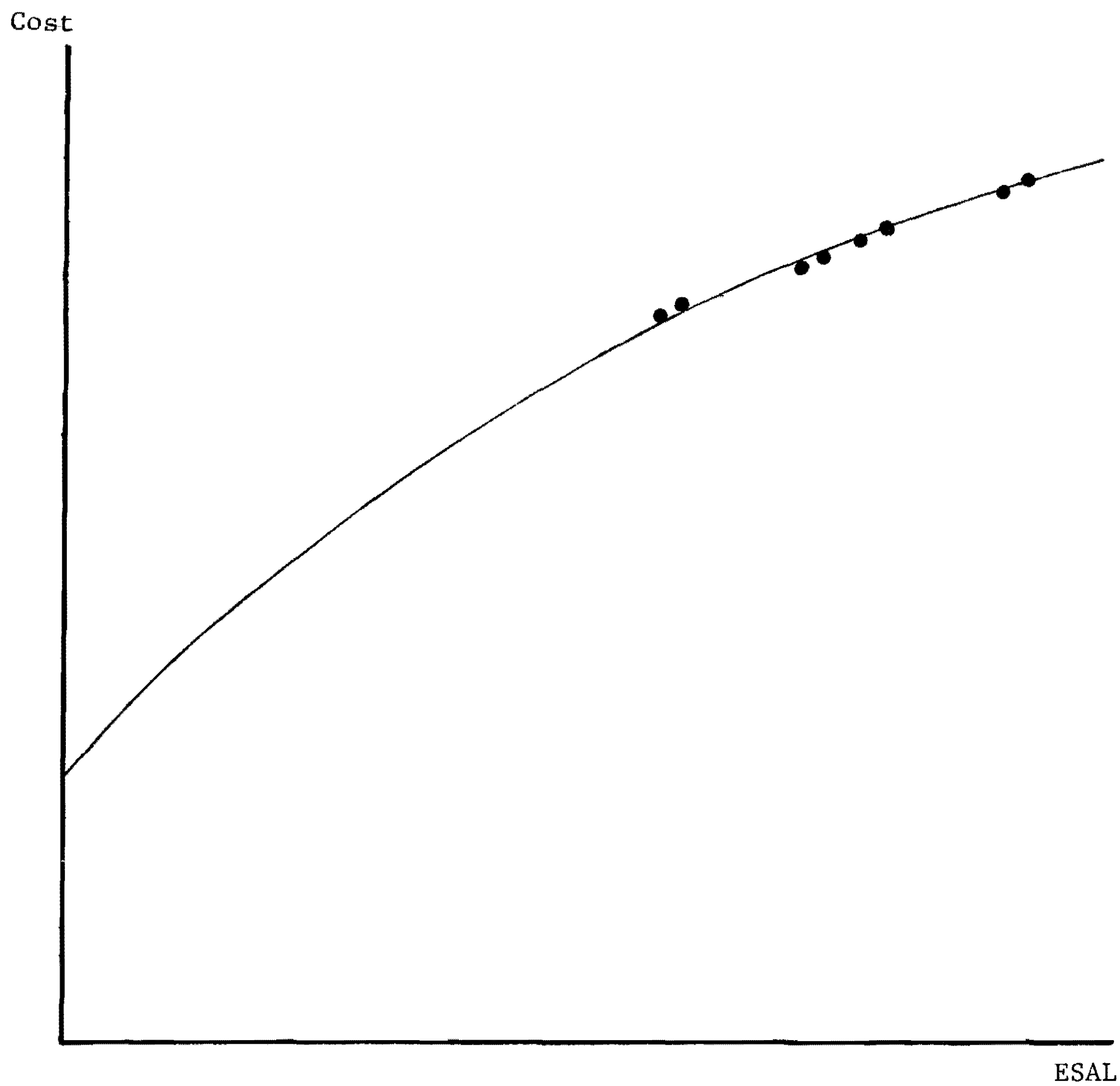


Figure 3.4 Rehabilitation Cost as a Function of ESALs

4. METHODOLOGY

4.1 The Modified Incremental Approach

A modified version of the incremental approach is proposed as a suitable methodology to allocate construction, reconstruction or rehabilitation costs. The proposed modification to the incremental approach attempts to overcome the lack of consistency mentioned in Chapter 2; however, an indirect result of this modification is that the computational complexity of the new procedure is increased.

In the Modified Incremental Approach cost estimates are prepared for every vehicle class, as well as for every combination of two or more vehicle classes. As an illustration, if a highway is designed to accomodate three types of vehicle classes 1, 2, and 3, the final cost allocation for each class is determined only after considering hypothetical designs for the following vehicle class combinations and computing the corresponding design costs: (a) class 1, (b) class 2, (c) class 3, (d) classes 1 and 2, (e) classes 1 and 3, (f) classes 2 and 3, and (g) classes 1, 2, and 3.

Using the cost estimates obtained for the above class combinations and a few fundamental operations, the total cost (corresponding to the combination including classes 1, 2, and 3) is partitioned into as many cost components as vehicle combinations; moreover, each cost component can be considered as the estimate of the cost effect of a vehicle class combination. In order to simplify the description of the method, the following notation is used:

C_1 = cost of a highway designed for vehicle class 1 alone,

C_2 = cost of a highway designed for vehicle class 2 alone,

C_3 = cost of a highway designed for vehicle class 3 alone,

C_{12} = cost of a highway designed for vehicle classes 1 and 2,

C_{13} = cost of a highway designed for vehicle classes 1 and 3,

C_{23} = cost of a highway designed for vehicle classes 2 and 3,

C_{123} = Total cost of a highway designed (for vehicle classes 1, 2, and 3);

The shaded areas in Figure 4.1 illustrate the notation described above. In this Figure, each individual vehicle class is represented by a circle. When two or more vehicle classes are simultaneously considered, the corresponding circles exhibit a certain degree of overlapping. This overlapping represents the portion of the total cost that is due to a combined effect of two or more vehicle classes.

As can be illustrated in Figure 4.2(a), the portion of the total cost that can be attributed to only individual classes 1, 2, and 3 is given by Equations (4.1), (4.2), and (4.3), respectively:

$$P_1 = C_{123} - C_{23} \tag{4.1}$$

$$P_2 = C_{123} - C_{13} \tag{4.2}$$

$$P_3 = C_{123} - C_{12} \tag{4.3}$$

Similarly, the portions of the total cost attributed to the interaction of any two vehicle classes, (1 and 2, 1 and 3, and 2 and 3) can be calculated using Equations (4.1), (4.2), and (4.3) and the initial cost estimates (C_1, C_2, C_3, and C_{123}), as follows:

$$P_{12} = C_{123} - C_3 - P_1 - P_2 \tag{4.4}$$

$$P_{13} = C_{123} - C_2 - P_1 - P_3 \tag{4.5}$$

$$P_{23} = C_{123} - C_1 - P_2 - P_3 \tag{4.6}$$

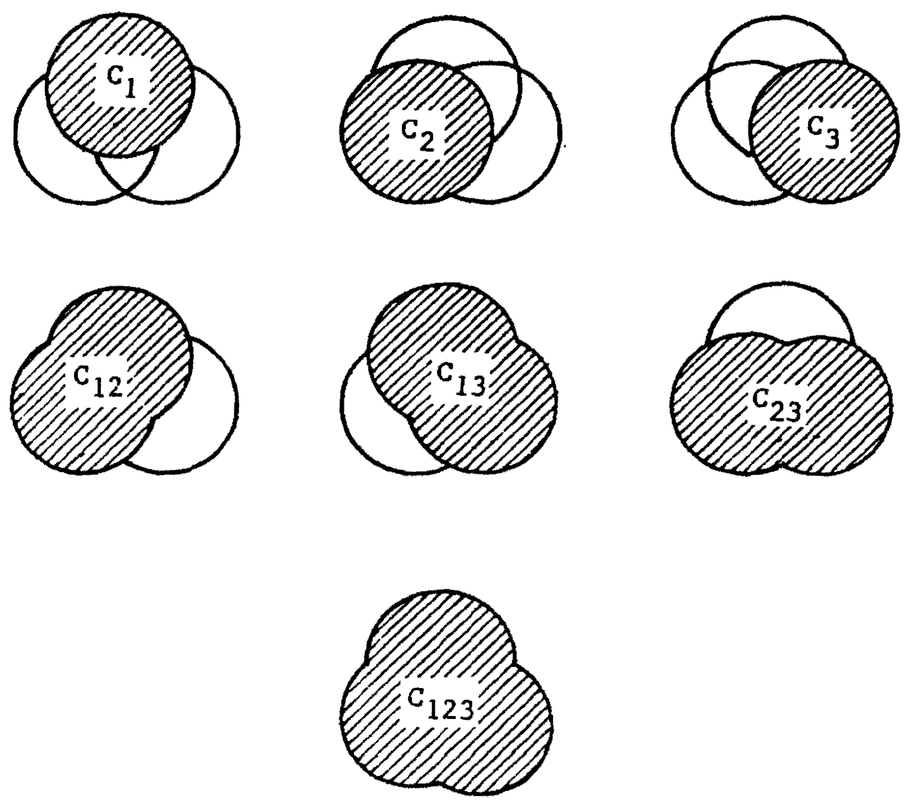


Figure 4.1 Input Cost Estimates

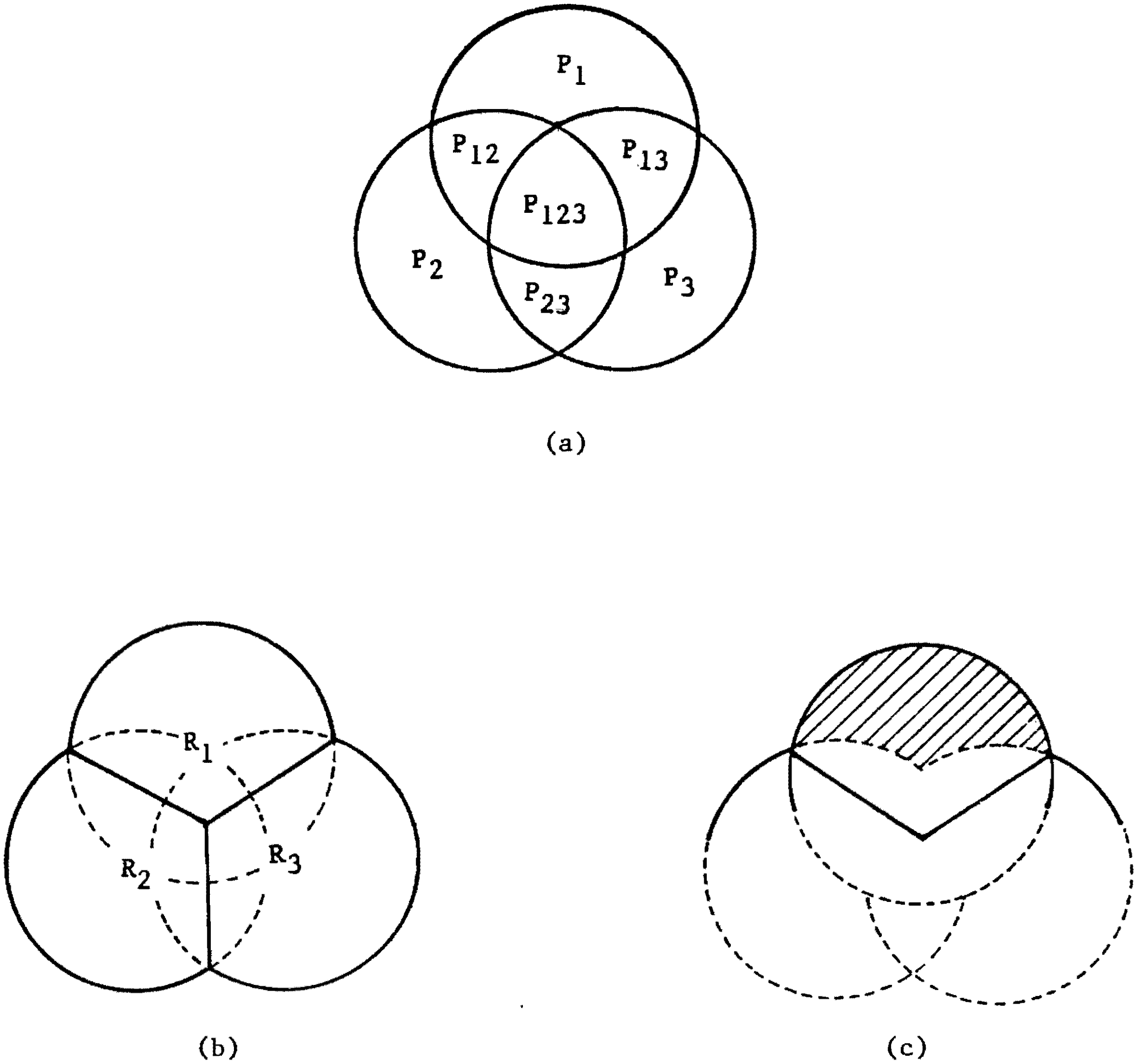


Figure 4.2 Cost Allocation Using the Modified Incremental Approach

Finally, the results from Equation (4.4), (4.5), and (4.6) are used to obtain P_{123}, the total portion of the cost attributed to the interaction of all vehicle classes, as shown below:

$$P_{123} = C_{123} - P_1 - P_2 - P_3 - P_{12} - P_{13} - P_{23} \qquad (4.7)$$

Figure 4.2(a) depicts the partitioning of the total cost C_{123} into the portions defined in Equations (4.1) through (4.7). As can be seen in this Figure, the allocated cost for vehicle class 1, for example, is equal to P_1 plus appropriate fractions of the portions P_{12}, P_{13}, and P_{123}. These fractions can be defined in terms of relative facility usage, as measured by vehicle miles of travel (VMT). If V_1, V_2, and V_3 represent the number of VMTs associated with classes 1, 2, and 3, respectively, the final allocated cost R_1, is given by Equation (4.8):

$$R_1 = P_1 + \frac{V_1}{V_1+V_2} P_{12} + \frac{V_1}{V_1+V_3} P_{13} + \frac{V_1}{V_1+V_2+V_3} P_{123} \qquad (4.8)$$

Similar results can be obtained for the cost allocations corresponding to classes 2 and 3:

$$R_2 = P_2 + \frac{V_2}{V_1+V_2} P_{12} + \frac{V_2}{V_2+V_3} P_{23} + \frac{V_2}{V_1+V_2+V_3} P_{123} \qquad (4.9)$$

$$R_3 = P_3 + \frac{V_3}{V_1+V_3} P_{13} + \frac{V_3}{V_2+V_3} P_{23} + \frac{V_3}{V_1+V_2+V_3} P_{123} \qquad (4.10)$$

Figure 4.2(b) represents the final cost allocations given in Equations (4.8), (4.9), and (4.10). In this Figure, it can be observed that the

Modified Incremental Method meets the completeness condition since the sum of the areas representing R_1, R_2, and R_3 is equal to the area representing the total cost C_{123} of Figure 4.1. The shaded area shown in Figure 4.2(c) represents the marginal cost of vehicle class 1. As can be seen by comparing this Figure with Figure 4.2(a) this marginal cost is exactly equal to P_1. Also, comparing Figures 4.2(b) and 4.2(c) it is clear that $P_1 \leq R_1$. Therefore, the cost allocated to vehicle class 1 is at least equal to its marginal cost. This shows that the marginality requirement is satisfied. Similarly, the fact that $R_1 \leq C_1$ indicates that the cost allocation corresponding to class 1 in a joint design is less that it would be in a design intended only for class 1. This means that the rationality requirement is satisfied.

The Modified Incremental Approach does not have the inconsistency limitation of the standard incremental method, since it considers all possible combinations of vehicle classes and does not require that vehicle classes be included in any sequence. The development presented in this Section can be generalized for any number of vehicle classes.

4.2 The Generalized Method

This procedure is based on concepts from the theory of cooperative games [16,17]. A linear programming model which includes a set of meaningful economic constraints is formulated and solved to determine the appropriate cost allocation among the vehicle classes that share a transportation facility. Although the procedure developed in this Section is valid for any number of vehicle classes, it will be illustrated with three vehicle classes 1, 2, and 3. The same notation given in Section 4.1 will be used in this illustration.

The Generalized Method expresses the completeness, rationality, and marginality principles in terms of a mathematical model. The completeness requirement, which establishes that the vehicle classes must entirely finance a highway facility, is stated below:

$$R_1 + R_2 + R_3 = C_{123} \tag{4.11}$$

The rationality principle, which imposes the condition that the common facility must be the best alternative for all individual vehicle classes 1, 2, and 3 and for all subgroups of vehicle classes 1 and 2, 1 and 3, and 2 and 3, is represented as follows:

$$R_1 \leq C_1 \tag{4.12}$$

$$R_2 \leq C_2 \tag{4.13}$$

$$R_3 \leq C_3 \tag{4.14}$$

$$R_1 + R_2 \leq C_{12} \tag{4.15}$$

$$R_1 + R_3 \leq C_{13} \tag{4.16}$$

$$R_2 + R_3 \leq C_{23} \tag{4.17}$$

The marginality principle establishes that the cost allocations for vehicle classes 1, 2, and 3, and the sum of allocations for subgroups 1 and 2, 1 and 3, and 2 and 3, must at least equal the corresponding marginal costs; this requirement is expressed by the following relationships:

$$R_1 \geq C_{123} - C_{23} \tag{4.18}$$

$$R_2 \geq C_{123} - C_{13} \tag{4.19}$$

$$R_3 \geq C_{123} - C_{12} \tag{4.20}$$

$$R_1 + R_2 \geq C_{123} - C_3 \tag{4.21}$$

$$R_1 + R_3 \geq C_{123} - C_2 \tag{4.22}$$

$$R_2 + R_3 \geq C_{123} - C_1 \tag{4.23}$$

As indicated by Young et al. [18], if Constraint (4.11) holds, then Constraints (4.12)-(4.17) are equivalent to Constraints (4.18)-(4.23). This means that Constraints (4.18)-(4.23) are redundant and need not be considered in the analysis.

Constraints (4.11)-(4.17) define the set of feasible solutions for the cost allocation problem. This set is called the "core" [17] of the problem and is represented in Figure 4.3(a). In this Figure, the core is the shaded segment on the plane representing Constraint (4.11). The boundaries or sides of the core are indicated by Constraints (4.12)-(4.17).

The core may contain several solutions of which only one must be selected. One way to accomplish this is to systematically reduce the set of feasible solutions until it contains exactly one solution. The core reduction procedure is illustrated in Figure 4.3(b). The core is reduced by "moving" its sides (constraints) in the directions of the corresponding arrows while keeping them parallel to the original positions. Mathematically, the size of the core is reduced if an amount t is substracted from each right-hand side of Constraints (4.12)-(4.17). Since only one point is desired, the amount t should be as large as possible without violating any of the Constraints. In conclusion, the core reduction procedure can be formulated in terms of the following linear programming model:

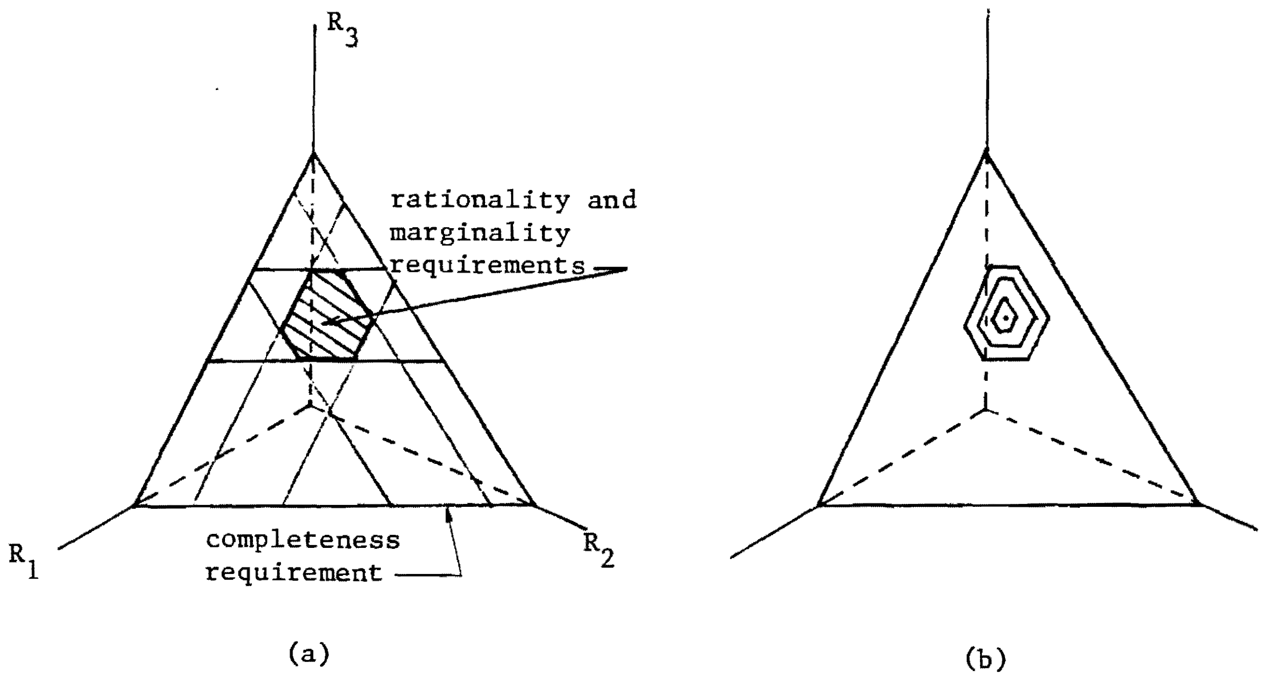


Figure 4.3 The Core in the Generalized Method

$$\text{maximize} \quad t$$

subject to

$$R_1 \leq C_1 - t \quad (4.24)$$

$$R_2 \leq C_2 - t \quad (4.25)$$

$$R_3 \leq C_3 - t \quad (4.26)$$

$$R_1 + R_2 \leq C_{12} - t \quad (4.27)$$

$$R_1 + R_3 \leq C_{13} - t \quad (4.28)$$

$$R_2 + R_3 \leq C_{23} - t \quad (4.29)$$

$$R_1 + R_2 + R_3 = C_{123} \quad (4.30)$$

$$R_1, R_2, R_3, t \geq 0 \quad (4.31)$$

4.3 **Environmental Factors**

An attractive feature of the generalized method is that it lends itself to a meaningful analysis of environmental costs. Environmental costs are those caused by factors other than traffic loads and, therefore, cannot be directly attributed to the individual vehicle classes.

The procedure described in this section can be easily extended to more that three vehicle classes. Only for convenience in the presentation it is assumed that only three classes are involved. The total number of vehicle combinations in this case is equal to 8. Each of these eight combinations can be represented in terms of a sequence of "+" and "-" signs, as indicated in Table 4.1. In this Table a negative sign indicates that a vehicle is not included in a combination, and a positive sign indicates that it is included. As an illustration, Combination 2 corresponds to a design for class 1 only with cost C_1, while Combination 4 corresponds to a design for classes 1 and 2, with cost C_{12}. In particular, Combination 8 corresponds to a design for vehicle classes 1, 2, and 3; this is the design whose cost C_{123} is to be allocated to the three vehicle classes. Combination 1 corresponds to a scenario with no vehicle classes. Since the cost C_0 associated with this scenario is not traffic-load related, it is assumed that it estimates the cost effect due to environmental factors.

It is always possible to express C_0 as a fraction of the total cost; that is,

$$C_0 = eC_{123} \qquad (4.33)$$

where e is an unknown number between 0 and 1. The methodology given in this Section can be used to find a maximal value for e for given C_1, $C_2,\ldots,C_{123}$.

The proposed method is based on the concept of effects associated with a two-level factorial experiment [2]. This concept is illustrated here using Table 4.1. As can be seen in this Table, 4 combinations include vehicle class 1 and 4 combinations do not include it. The average cost associated with the combination <u>not</u> including class 1 is given by

$$E_1^- = (C_0 + C_2 + C_3 + C_{23})/4 \qquad (4.34)$$

Similarly, the average cost associated with the vehicle combinations including class 1 is equal to

$$E_1^+ = (C_1 + C_{12} + C_{13} + C_{123})/4 \qquad (4.35)$$

The statistical effect of class 1 is defined as $E_1^+ - E_1^-$ since this difference measures the average increase in cost due to vehicle class 1. Letting E_1 be equal to $E_1^+ - E_1^-$, and using Equations (4.34) and (4.35), E_1 can be written as

Table 4.1 Vehicle Combinations

Combination No.	Vehicle Class 1	Vehicle Class 2	Vehicle Class 3	Cost
1	-	-	-	C_0
2	+	-	-	C_1
3	-	+	-	C_2
4	+	+	-	C_{12}
5	-	-	+	C_3
6	+	-	+	C_{13}
7	-	+	+	C_{23}
8	+	+	+	C_{123}

$$E_1 = (C_1 - C_2 + C_{12} - C_3 + C_{13} - C_{23} + C_{123})/4 - eC_{123}/4 \qquad (4.36)$$

Setting $A_1 = (C_1 - C_2 + C_{12} - C_3 + C_{13} - C_{23} + C_{123})/4$ and $B = C_{123}/4$, it is possible to rewrite Equation (4.36) as

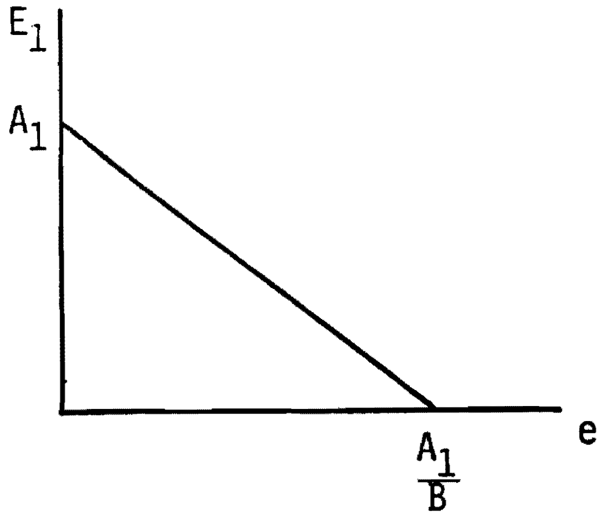
$$E_1 = A_1 - Be \qquad (4.37)$$

The relationship given in Equation (4.37) is linear and indicates that the effect due to vehicle class 1 decreases as the impact of the environmental factors is increased. This behavior is illustrated in Figure 4.4(a).

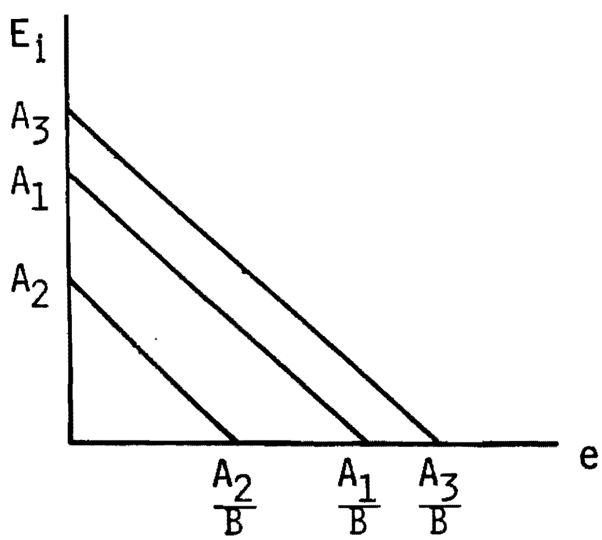
A similar procedure is followed to find the relationships for vehicle classes 2 and 3. Figure 4.4(b) shows three hypothetical linear relationships for the three vehicle classes under consideration. Since E_1, E_2, and E_3 must be positive, the range for e is between zero and the minimal A_i/B value. In the case of the illustration given in Figure 4.4(b) this value is A_2/B. In general, $0 \leq e \leq e'$ where

$$e' = \min \{A_1/B, A_2/B, A_3/B\} \qquad (4.38)$$

Summarizing, the cost effect due to the environmental factors can at most be a fraction e' of the total cost. Values of e exceeding e' are not valid since they would yield a negative value for the effect associated with at least one vehicle class.



(a)



(b)

Figure 4.4 Effects of the Vehicle Classes on Cost as Functions of e

5. APPLICATION OF THE METHODOLOGY

An application of the Modified Incremental Approach and the Generalized Method using a small sample from Texas pavement data is presented in this Section. Although realistic, these data are by no means comprehensive and are utilized only for illustrative purposes.

It is intended to allocate the estimated rehabilitation costs incurred in an analysis period of 18 years among four vehicle classes for a highway system consisting of two kinds of pavements. Table 5.1 describes the vehicle classes considered in this example, accumulated ESALs throughout the analysis period for each vehicle class, and percentages of VMTs corresponding to each vehicle class. Table 5.2 displays highway classification, pavement type, and pavement mileage for each of the two kinds of pavement.

A modification of the RENU program [8] was performed in order to obtain rehabilitation costs for the various vehicle combinations. Using these figures and ESAL data, rehabilitation costs were estimated for all vehicle combinations using the cost function discussed in Chapter 3. Table 5.3 gives the rehabilitation cost estimates associated with each vehicle class combination.

The results obtained from the Modified Incremental Approach and the Generalized Method are given below:

(a) **Modified Incremental Approach**

Vehicle Class 1	\$ 947,000
Vehicle Class 2	\$ 33,000
Vehicle Class 3	\$1,047,000
Vehicle Class 4	\$ 213,000

Table 5.1 Vehicle Class Data

Vehicle Class	Truck Type	ESALs (millions)	VMT (%)
1	2D	3.590	96.43
2	3A	0.647	1.18
3	3-S2	15.317	2.06
4	2-S1-S2	5.172	0.33

Table 5.2 Illustrative Pavement System.

Pavement	Highway Classification	Pavement Type	Mileage (lane-miles)
1	Interstate	Flexible Overlaid	57
2	U. S.	Hot Mix	135

Table 5.3 Rehabilitation Cost Estimates

Combination	Cost (millions)
1	1.06
2	0.76
1,2	1.11
3	1.87
1,3	2.04
2,3	1.90
1,2,3	2.06
4	1.18
1,4	1.46
2,4	1.24
1,2,4	1.51
3,4	2.105
1,3,4	2.22
2,3,4	2.13
1,2,3,4	2.24

(b) **Generalized Method**

Vehicle Class 1	$ 410,000
Vehicle Class 2	$ 320,000
Vehicle Class 3	$1,030,000
Vehicle Class 4	$ 480,000

It can be verified that both methods yield results which are consistent with the completeness, rationality and marginality principles. A considerable difference in the results can be observed between the two methods. This difference is explained by the influence of the measure of highway usage (VMT) on the allocation of common costs in the Modified Incremental Approach. In this example, a significant portion of the total cost C_{1234} is attributed to the interaction of all vehicle classes. A large percentage of this portion is allocated to vehicle class 1 due to the high percent of VMTs associated with it. On the other hand, the Generalized Method distributes the cost among the vehicle classes without considering VMTs; in case that the degree of pavement damage, and not highway utilization, is the dominant criterion in the decision making process, the generalized results are appropriate. The maximum percentage of the total cost that can be atributed to the environment e' is equal to 45%, as indicated by Equation (4.38).

6. SUMMARY

This report summarizes the work performed in relation to study 2-18-83-332 "Analysis of Truck Use and Highway Cost Allocation in Texas" during F. Y. 83-84. This Fiscal Year, emphasis was placed on the development of a sound conceptual methodology for the allocation of costs related to the provision and upkeeping of highway facilities.

In particular, two cost allocation methods were developed: the Modified Incremental Approach and the Generalized Method. These methods exhibit significant conceptual advantages over those previously used in the context of highway cost allocation. In particular, they fulfill the following conditions: (a) highway costs are completely financed by users; (b) vehicle classes reduce their cost responsibilities by sharing the facilities with other vehicle classes; and (c) vehicle are charged at least enough to cover their corresponding marginal costs.

In addition, a procedure was developed to assess a range of values for the percentage of the total rehabilitation and maintenance costs that can be attributed to the effect of the environment, that is, independent of traffic.

The proposed methodology was illustrated using two representative sections from the Texas pavement data base.

7. REFERENCES

1. Allison, J. T., A. Garcia-Diaz, and R. L. Lytton. A Model for Predicting Flexible Pavement Service Life and Its Impact on Rehabilitation Decisions. Paper presented at the 62nd Annual Meeting of the Transportation Research Board, Washington, D. C., January 1983.
2. Box, E.P., W.G. Hunter and J.S. Hunter. Statistics for Experimenters: an Introduction to Design Data Analysis and Model Building, Wiley, New York, 1978.
3. Brown, J.E. and W.C. Allinson. The 1956 Rhode Island Highway Finance Study. Highway Research Board Bulletin No. 175, pp. 44-52.
4. Bureau of Public Roads. Final Report of the Highway Cost Allocation Study. H. Doc. 564 and 77, 87th. Congress, 1961.
5. Butler, B. D. Jr. Economic Analysis of Roadway Occupancy for Freeway Pavement Maintenance and Rehabilitation. Vols. I, II, and III, Report Nos. FHWA-RD-76-14 and FHWA-RD-76-15, 1974.
6. FHWA. Final Report on the Federal Highway Cost Allocation Study. May 1982.
7. Freeman P.N.W. The Recovery of Costs from Road Users in South Africa, Part II: Cost Allocation Study. National Institute for Transport and Road Research, CSIR, South Africa, 1981.
8. Garcia-Diaz, A. Documentation of the Modified Computerized Procedure RENU2 to Estimate Pavement Network Rehabilitation and Maintenance Costs. Research Report 992-1, Texas Transportation Institute, March 1983.
9. Garcia-Diaz, A., M. Riggins, and S. J. Liu. Development of Performance Equations and Survivor Curves for Flexible Pavements. SDHPT Research Report No. 284-5, Texas Transportation Institute, February

1984.

10. Hall, W.L. and R.D. Johnson. Comparison of Incremental and Ton-Mile Allocation Methods. Highway Research Board Bulletin No. 175, pp. 27-32.

11. Hennes, R.G. and W.W. Mylroie. Washington State Highway Cost Allocation Study. Highway Research Board Bulletin No. 175, pp. 1-27.

12. JLARC. Vehicle Cost Responsibility in Virginia. Commonwealth of Virginia, Richmond, November, 1981.

13. Koenker, E.W. Comparative Highway Cost Allocation: Incremental and Other Methods. Highway Research Board Bulletin No. 175, pp. 22-26.

14. Lee, E. T. Statistical Methods for Survival Data Analysis. Lifetime Learning Publications, Behmst, California, 1980.

15. Martin, J.W., C.R. Lockyer and E.C. Holshouser. Results of the Kentucky Highway Finance Study. Highway Research Board Bulletin No. 175, pp. 33-43.

16. Newman, J. von and O. Morgenstern. Theory of Games and Economic Behavior, Princeton University Press, Princeton, 1944.

17. Owen, G. Game Theory, Academic Press, New York, 1982.

18. Young, H.P., N. Okada and T. Hashimoto. Cost Allocation in Water Resources Development -- A Case Study of Sweden. IIASA Research Report RR-80-32, Laxenburg, Austria, 1980.

19. Zajac, E. Fairness and Efficiency: An Introduction to Public Utility Pricing, Ballinger, Cambridge, 1978.

APPENDIX 1

MODIFIED RENU2 FORTRAN CODE

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

               C
               C
               C                      R  E  N  U
               C
               C
               C     PROGRAM TO DETERMINE EFFECT OF LEGAL LOAD LIMITS ON LONG-RANGE
               C     PAVEMENT COSTS.
               C
               C     THIS VERSION CREATED AUG 7-1981
               C
               C
 ISN 0002            REAL XLAMB
 ISN 0003            COMMON /MECH/XKT,NRU,NLH,ND,NDEL,IACR,IYR,JYR,CONSTR(20)
 ISN 0004            COMMON /COSTS/ COSM(20,2), COSV(20,2), COSMS(20,2), COSVS(20,2),
                    1               CSMPW(2), CSVPW(2), CSMUA(2),CSVUA(2),COSC(20,2)
 ISN 0005            COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
 ISN 0006            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
 ISN 0007            COMMON /FUNDS/ APOF(20,2), RTINT, RTINF
 ISN 0008            COMMON /IO/ LI, LO, LD
 ISN 0009            COMMON /LABELS/ MATLAB(5,10)
 ISN 0010            COMMON /LMP/ XLM(30),YLM(30),POTLM(20,2),OUTP(20,2),
                    1        TOTALM, PPF, TPF, PFNO, NASL, NSLR,TOVLM(30,2),XLM2(30)
 ISN 0011            COMMON /MISC/ IPOT, IARMS, OLDMNT, AGF
 ISN 0012            COMMON /OUT/ PSIE(30,2), EALREM(30,2), COSTM(20,30,2),CSTOV(30,2)
                    1          ,PSIB(30)
 ISN 0013            COMMON /OVER/ TOV(30,2), SNOV(30,2), THOV(30,2)
 ISN 0014            COMMON /OVRLAY/ XHCIO,XHCIM,WLANE, WPSH, WGSH, PPVDSH, CAC, CGR
                    1     , CSCOAT
 ISN 0015            COMMON / POV/ SNOVP(20,2), THOVP(20,2), CSTOVP(20,2), PP(20,2)
                    1          , RLP(20,2)
 ISN 0016            COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
 ISN 0017            COMMON /STRCOE/ STRCD(8),CC(4),MC(11),NC,STRC(5),RFS(4),RFB(4)
 ISN 0018            COMMON /TEMPC/ CONTP(25),DISTCT
 ISN 0019            COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
 ISN 0020            COMMON /SUMARY/ SECTLE(2,10,8),SYSTLE(60,8),NSECT(8),DELC(10,8),
                    1                COSR(10,8),DELCPW(10,8),COSRPW(10,8),DELCUA(10,8),
                    2                COSRUA(10,8),RLRAT(10,8),TLM(10,8),DSLV(10,8),NSYS
 ISN 0021            COMMON /CMP/ COMP(30,34), PCOMP(30), AATP(30), PIH(30,8),QIH(30,8)
 ISN 0022            COMMON /SLVG/ ISLV, FLRP, VI(30), RI(30), VL(30), RL(30),
                    1               U(30), PL(30), MI(30), P(20), VP(20), RP(20),
                    2               PB,VPB,RPB, NS, NY, SV(6,2), SVB, FLRPTP(4)
 ISN 0023            COMMON /TIME/ ATP, OVLIF, NYAP, NYR,  YR(40)
 ISN 0024            COMMON /TITLE/ TITLE(20,3), SECTTL(20)
 ISN 0025            COMMON/HOR/A(10),B(10),C(10),DT(10),DF(10),S(10),T(10),TR(5),PI(5)
                    *,PT(5),AC(5),AA,SCT(5),XMNW18(10) ,XKTO
 ISN 0026            COMMON /EXTRA/ PTOVTK,TPE,PFO,XMNOTK,XMXOTK,NIS
 ISN 0027            COMMON /BURKE/ XLAMB, GAMMA, TFBAP
 ISN 0028            COMMON /COMBI/ ICOMB, NVC, COFVCT(6)
 ISN 0029            COMMON /COST/ COSTRH(20), COSTRM(20), COSTPM(20)
 ISN 0030            DIMENSION TITLES(5)
 ISN 0031            REWIND 2
```

```
ISN 0034            WRITE (6,600) TITLES,NVC
ISN 0035        600 FORMAT (1X,5A4,I5)
ISN 0036            NCOMB=2**NVC-1
ISN 0037            DO 1000 ICOMB=1,NCOMB
ISN 0038            DO 1 I=1,8
ISN 0039            DO 1 J=1,30
ISN 0040            PIH(J,I)=0.
ISN 0041            QIH(J,I)=0.
ISN 0042          1 CONTINUE
ISN 0043            CALL INIT(1)
ISN 0044            CALL COMGEN
ISN 0045            REWIND 10
ISN 0046        100 CALL INPUT (IGO)
ISN 0047            GO TO (110, 200, 300,300), IGO
ISN 0048        110 CALL INIT(2)
ISN 0049            CALL INPRNT
ISN 0050            CALL EALGET
ISN 0051            CALL COSCAL
ISN 0052            GO TO 100
ISN 0053        200 CONTINUE
ISN 0054            GO TO 100
ISN 0055        300 CONTINUE
ISN 0056       1000 CONTINUE
ISN 0057            ENDFILE 2
ISN 0058            STOP
ISN 0059            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      58, PROGRAM SIZE =      974, SUBPROGRAM NAME =  MAIN

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                    920K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

  ISN 0002          SUBROUTINE NPAGE
  ISN 0003          RETURN
  ISN 0004          END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       3, PROGRAM SIZE =      164, SUBPROGRAM NAME = NPAGE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

    ISN 0002          BLOCK DATA
    ISN 0003          COMMON /TEMPC/ CONTP(25),DISTCT
    ISN 0004          COMMON /MECH/XKT,NRU,NLH,ND,NDEL,IACR,IYR,JYR,CONSTR(20)
    ISN 0005          COMMON/HOR/A(10),B(10),C(10),DT(10),DF(10),S(10),T(10),TR(5),PI(5)
                     *,PT(5),AC(5),AA,SCT(5),XMNW18(10) ,XKTO
    ISN 0006          COMMON /EXTRA/ PTOVTK,TPE,PFO,XMNOTK,XMXOTK,NIS
    ISN 0007          COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
    ISN 0008          COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
    ISN 0009          COMMON /FUNDS/ APOF(20,2), RTINT, RTINF
    ISN 0010          COMMON /IO/ LI, LO, LD
    ISN 0011          COMMON /LABELS/ MATLAB(5,10)
    ISN 0012          COMMON /LMP/ XLM(30), YLM(30), POTLM(20,2), OUTP(20,2),
                     1      TOTALM, PPF, TPF, PFNO, NASL, NSLR,TOVLM(30,2),XLM2(30)
    ISN 0013          COMMON /MISC/ IPOT, IARMS, OLDMNT, AGF
    ISN 0014          COMMON /OVRLAY/ XHCIO,XHCIM,WLANE, WPSH, WGSH, PPVDSH, CAC, CGR
                     1     , CSCOAT
    ISN 0015          COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
    ISN 0016          COMMON /STEER/ EQFACT(15,5), PTST(4)
    ISN 0017          COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
    ISN 0018          COMMON /STRCOE/ STRCD(8),CC(4),MC(11),NC,STRC(5),RFS(4),RFB(4)
    ISN 0019          COMMON /TIME/ ATP, DVLIF, NYAP, NYR,  YR(40)
    ISN 0020          COMMON /SLVG/ ISLV, FLRP, VI(30), RI(30), VL(30), RL(30),
                     1              U(30), PL(30), MI(30), P(20), VP(20), RP(20),
                     2              PB,VPB,RPB, NS, NY, SV(6,2), SVB, FLRPTP(4)
    ISN 0021          COMMON /DCOM1/ TTOUTP(8),TTOC(8),TTT(8,20,3)
                C
                C ***************************************************
                C VARIBLES COMPARISON BETWEEN AASHO & TEXAS EQUATIONS
                C
                C       TEXAS        AASHO    DESCRIPTIONS
                C     ---------     -------  -----------------------------------
                C       C(1)         ALF      HARMONIC MEAN TEMPERATURE
                C       C(2)         TI       THORN THWAITE INDEX
                C       C(3)         FTC      ANNUAL AVERAGE FREEZE-THAW CYCLES
                C       C(4)         WFTC
                C       C(5)         PR       ANNUAL AVERAGE RAINFALL
                C       C(6)         TM       MEAN MONTHLY TEMPERATURE
                C       DF(1)        DMD      MAXIMUM DEFLEXION
                C       DF(2)        SCI      SURFACE CURVATURE INDEX
                C       DF(3)        VOL      VOLUME OF DINAFLEX BASIN
                C       DT(1)        AS       ASPHALT STIFFNESS
                C       S(1)         TTC      TEXAS TRIAXIAL CLASS
                C       S(2)         SLL      LIQUID LIMIT
                C       S(3)         SPI      PLASTICITY INDEX
                C       S(4)         SPP      PERCENT PASSING #200
                C       T(1)         T        AGE IN YEARS
                C       TR(1)        ADT      AVERAGE DAILY TRAFFIC
                C       TR(2)        18-KIP   18-KIP SINGLE AXLE LOADS
                C       TR(NPT)      W        18-KIP SINGLE AXLE LOADS
                C     ---------     -------  -----------------------------------
                C
                C *** REFER TO SUBROUTINES PSIT & RUTA
                C
```

```
                C ************************************************
                C
ISN 0022              DATA NAPOV, PAPOV, SIZE, AVRG /21, 5.0, 2.0, 100./
ISN 0023              DATA XHCIO/0.0/,XHCIM/0.0/
ISN 0024              DATA PICON, PTERM, PIOV, PTOV / 4*-1. /
ISN 0025              DATA IF, IR, IC /1, 2, 3 /
ISN 0026              DATA LI, LO, LD /10, 6, 1/
ISN 0027              DATA SS, R, AGG, XK, E /3., 1., 195.43, 150., 4.0E6/
ISN 0028              DATA NYAP, OVLIF, ATP, NYR / 20, 20., 20., 40 /
ISN 0029              DATA RTINT, RTINF /0., 0. /
                C     TABLE OF STEERING AXLE EQUIVALENCIES BY AXLE LOAD AND TERMINAL PSI
ISN 0030              DATA XMNW18/10*0.0/
ISN 0031              DATA SCT/.5,.5,.5,.5,.5/
ISN 0032              DATA A/13.,13.,10.,8.,10.,10.,10.,10.,10.,0./
ISN 0033              DATA AC/.5,.5,.5,.5,.5/
ISN 0034              DATA B/12.,12.,10.,7.,10.,10.,10.,10.,40.,0./
ISN 0035              DATA C/9.,-30.,125.,20.,16.,55.,0.,0.,0.,0./
ISN 0036              DATA DT/.5,0.,0.,0.,0.,0.,0.,0.,0.,0./
ISN 0037              DATA DF/1.5,1.,2.225,0.,0.,0.,0.,0.,0.,0./
ISN 0038              DATA T/15.,0.,0.,0.,0.,0.,0.,0.,0.,0./
ISN 0039              DATA TR/36000.,36000.,36000.,36000.,36000./
ISN 0040              DATA S/5.,50.,30.,40.,0.,0.,0.,0.,0.,0./
ISN 0041              DATA PI/4.7,4.73,4.41,4.81,4.6/
ISN 0042              DATA PT/2.5,2.5,2.5,2.5,2.5/
ISN 0043              DATA NAPOV, PAPOV, SIZE, AVRG /21, 5.0, 2.0, 100./
ISN 0044              DATA PICON, PTERM, PIOV, PTOV / 4*-1. /
ISN 0045              DATA IF, IR, IC /1, 2, 3 /
ISN 0046              DATA NYAP, OVLIF, ATP, NYR / 20, 20., 20., 40 /
ISN 0047              DATA PPF,TPF,PFNO /0., 0., 0. /
ISN 0048              DATA RTINT, RTINF /0., 0. /
ISN 0049              DATA PTST /1.5, 2.0, 2.5, 3.0/
ISN 0050              DATA EQFACT /2., 4., 6., 8., 10., 12., 14., 16., 18., 20., 22.,
                    1             24., 26., 28., 30.,
                    2             .0005, .008, .04, .13, .28, .52, .92, 1.42, 2.12,
                    3             2.95, 4.02, 5.29, 6.73, 8.31, 10.19,
                    4             .0009, .01, .05, .14, .31, .54, .86, 1.31, 1.94,
                    5             2.52, 3.35, 4.4, 5.49, 6.67, 8.05,
                    6             .002, .02, .06, .18, .36, .62, .93, 1.33, 1.9, 2.44,
                    7             3.15, 3.95, 4.82, 5.83, 6.8,
                    8             .004, .03, .09, .23, .41, .66, .94, 1.28, 1.74,
                    9             2.16, 2.7, 3.28, 3.89, 4.59, 5.23/
ISN 0051              DATA STRCD /.44, .34, .23, .14, .30, .18, .11, .14 /
ISN 0052              DATA RFS /.9, .7, .5, .5/
ISN 0053              DATA RFB /1., .9, .7, .5/
ISN 0054              DATA CC / 1.0, 0.85, 0.75, 0.75 /
ISN 0055              DATA NC /11/
ISN 0056              DATA MC /3HACP,3HATB,3HCTB,3HAGB,3HSAB,3HLTB,3HAGS,3HLTS,
                    1        3HJCP, 3HCRC, 3HACO /
ISN 0057              DATA MATLAB / 4HASPH, 4HALT , 4HSURF, 4HACE , 4H    ,
                    1              4HASPH, 4HALT , 4HBASE, 4H    , 4H    ,
                    2              4HCEME, 4HNT T, 4HREAT, 4HED B, 4HASE ,
                    3              4HAGGR, 4HEGAT, 4HE BA, 4HSE  , 4H    ,
                    4              4HSAND, 4H ASP, 4HHALT, 4H BAS, 4HE   ,
                    5              4HLIME, 4H TRE, 4HATED, 4H BAS, 4HE   ,
                    6              4HAGGR, 4HEGAT, 4HE SU, 4HBBAS, 4HE   ,
                    7              4HLIME, 4H TRE, 4HATED, 4H SUB, 4HBASE,
                    8              4HJCP , 4HSURF, 4HACE , 4H    , 4H    ,
                    9              4HCRC . 4HSURF, 4HACE , 4H    , 4H
```

```
                   A              /
  ISN 0058          DATA FLRPTP /1.2, 1.4, 1.6, 1.8 /
  ISN 0059          DATA CONTP / 21.,22.,22.,9.,16.,23.,26.,26.,28.,24.,28.,33.,33.,
                   1             31.,31.,36.,30.,26.,25.,32.,38.,31.,25.,24.,19./
  ISN 0060          END
```

```
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*    SOURCE STATEMENTS =      59, PROGRAM SIZE =        0, SUBPROGRAM NAME = TEMPC

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                              912K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE COMGEN
   ISN 0003            COMMON /COMBI/ ICOMB, NVC, COFVCT(6)
   ISN 0004            DO 100 IND=1,NVC
   ISN 0005               COFVCT (IND) = 0.
   ISN 0006        100 CONTINUE
   ISN 0007            IND = 1
   ISN 0008            INQUOT = ICOMB
   ISN 0009        500 IF (INQUOT .EQ. 0) GO TO 900
   ISN 0011               QUOT = FLOAT(INQUOT)/2.
   ISN 0012               INQUOT = INT(QUOT)
   ISN 0013               IRES = INT((QUOT-FLOAT(INQUOT))*2.)
   ISN 0014               COFVCT(IND) = FLOAT(IRES)
   ISN 0015               IND = IND+1
   ISN 0016               GO TO 500
   ISN 0017        900 RETURN
   ISN 0018            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      17, PROGRAM SIZE =      518, SUBPROGRAM NAME =COMGEN

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002           SUBROUTINE INPUT (IGO)
   ISN 0003           COMMON /TEMPC/ CONTP(25),DISTCT
   ISN 0004           COMMON /EXTRA/ PTOVTK,TPE,PFO,XMNOTK,XMXOTK,NIS
   ISN 0005           COMMON /MNTPAR/ UNTCST(4),USRMDL(31,3),WDTH,S,DISS,DCON,DIN,MFLG
   ISN 0006           COMMON /MECH/XKT,NRU,NLH,ND,NDEL,IACR,IYR,JYR,CONSTR(20)
   ISN 0007           COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
   ISN 0008           COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
   ISN 0009           COMMON /FUNDS/ APOF(20,2), RTINT, RTINF
   ISN 0010           COMMON /INTVLS/ STARTS(6)
   ISN 0011           COMMON /IO/ LI, LO, LD
   ISN 0012           COMMON /LABELS/ MATLAB(5,10)
   ISN 0013           COMMON /LDS/ PGVWL, PSAL, PTAL, PTRAL, FGVWL, FSAL, FTAL, FTRAL,
                     1             PSTAW(10), FSTAW(10)
   ISN 0014           COMMON /LMP/ XLM(30),YLM(30),POTLM(20,2),OUTP(20,2),TOTALM, PPF,
                     1    TPF, PFNO, NASL, NSLR,TOVLM(30,2),XLM2(30)
   ISN 0015           COMMON /MISC/ IPOT, IARMS, OLDMNT, AGF
   ISN 0016           COMMON /NEWSYS/ NEWSYS
   ISN 0017           COMMON /NMBR/ SA(30,11), TA(30,11), TR(50,11), VE(30,11),
                     1              VG(75,11), NLDI(6), EPI(10), ST(30,11)
   ISN 0018           COMMON /OUTSWH/ IOUT
   ISN 0019           COMMON /OVRLAY/ XHCIO,XHCIM,WLANE, WPSH, WGSH, PPVDSH, CAC, CGR
                     1     , CSCOAT
   ISN 0020           COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
   ISN 0021           COMMON /STRCOE/ STRCD(8),CC(4),MC(11),NC,STRC(5),RFS(4),RFB(4)
   ISN 0022           COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
   ISN 0023           COMMON /TIME/ ATP, OVLIF, NYAP, NYR,  YR(40)
   ISN 0024           COMMON /TITLE/ TITLE(20,3), SECTTL(20)
   ISN 0025           COMMON /TRTYP/ TTYP(2,10), PTTYP(10,20,2), PCTTR(20,2),PERCT(4),
                     1               NAXLES(10,4),NT(4), NTTY, NATT, NTT, NEWTRK
   ISN 0026           COMMON /SLVG/ ISLV, FLRP, VI(30), RI(30), VL(30), RL(30),
                     1              U(30), PL(30), MI(30), P(20), VP(20), RP(20),
                     2              PB,VPB,RPB, NS, NY, SV(6,2), SVB, FLRPTP(4)
   ISN 0027           COMMON /SWTCHS/ OVLIFE, PCTINT, PCTINF, TPFPC, PFNOPC, AGR, SPCJT,
                     1                XMLI, CACI, CGRI, ICAC, ACDENS, ICGR, GRDENS,
                     2                INTT, SAVMNT, IDST, NLD, MCODE(5), TFCDNS
   ISN 0028           DIMENSION KWORD(5), IVAL(2), VAL(5), KEY(22), STRCIN(5)
   ISN 0029           DATA ISTOP /4HSTOP/
   ISN 0030           DATA SATP /0./
   ISN 0031           DATA KEY /4HSTOP, 4HEXEC, 4HFLEX, 4HRIGI, 4HPERF, 4HAGE , 4HOVER,
                     1         4HMODE, 4HHIST, 4HNO M, 4HTRUC, 4HSYST, 4HOLD , 4HRUN ,
                     2         4HLOAD, 4HSING, 4HTAND, 4HTRID, 4HGVW , 4HEMPT, 4HSTEE,
                     3         4HOUTP/
   ISN 0032           DATA IACO /4HACO /
   ISN 0033           DATA NKEY /22/
   ISN 0034           IDST = 0
   ISN 0035           NEWTRK = 0
   ISN 0036           NEWSYS = 0
   ISN 0037           ATP = SATP
   ISN 0038           CALL NPAGE
                C
                C     READ AND ECHO PRINT A KEYWORD CARD
                C
   ISN 0039         2 READ (LI,3) KWORD, IVAL, VAL
```

```
ISN 0040          3 FORMAT(5A4,2I5,5F10.0)
ISN 0041            WRITE (LO,4) KWORD, IVAL, VAL
ISN 0042          4 FORMAT(1X,5A4,2I5,5(F10.2,2X))
              C
              C     TEST FOR NORMAL PROGRAM TERMINATION
              C
ISN 0043            IF (KWORD(1) .EQ. ISTOP) GO TO 9992
              C
              C     SEARCH THE KEY TABLE FOR THE KEYWORD READ IN
              C
ISN 0045            DO 10 I=1,NKEY
ISN 0046            IKEY = I
ISN 0047            IF (KWORD(1) .EQ. KEY(I)) GO TO 15
ISN 0049         10 CONTINUE
ISN 0050            GO TO 9996
ISN 0051         15 GO TO (9998, 9997, 100, 200, 300, 400, 500, 600, 700, 800, 900,
                  1        1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700, 1800, 1900,
                  2        2000) , IKEY
              C
              C     *** FLEXIBLE SECTION ***
              C
ISN 0052        100 IP = IF
ISN 0053            WLANE = VAL(1)
ISN 0054            WDTH = WLANE
ISN 0055            SS = VAL(2)
ISN 0056            R = VAL(3)
ISN 0057            PF=VAL(4)
ISN 0058            PFO=VAL(5)
              C
              C     READ A TITLE CARD FOR THIS SECTION
              C
ISN 0059        101 READ (LI,102) SECTTL
ISN 0060        102 FORMAT (20A4)
ISN 0061            WRITE (LO,103) SECTTL
ISN 0062        103 FORMAT (1X,20A4)
ISN 0063            IF(IP.EQ.IR) GO TO 105
              C
              C     READ AND ECHO PRINT THE MATERIALS CARD
ISN 0065            READ(LI,19) NDIST,NIS,NPT,NRU,NLH,NDEL,TPE,XMNOTK,XMXOTK,PTOVTK,
                  1      IACR,IYR,JYR
ISN 0066            DISTCT=FLOAT(NDIST)
ISN 0067            ND=1
ISN 0068            IF(NDIST.GT.1.AND.NDIST.LE.9) ND=2
ISN 0070            IF(NDIST.GE.22.AND.NDIST.LE.25) ND=2
ISN 0072      19    FORMAT(7I5,3F5.0,3I5)
ISN 0073            WRITE(LO,21) NDIST,NIS,NPT,NRU,NLH,NDEL,TPE,XMNOTK,XMXOTK,PTOVTK,
                  1      IACR,IYR,JYR
ISN 0074         21 FORMAT(1X,7I5,3F5.2,3I5)
ISN 0075            READ(LI,20)(CONSTR(I),I=1,20)
ISN 0076         20 FORMAT(15F5.0)
ISN 0077            WRITE(LO,22)(CONSTR(I),I=1,20)
ISN 0078         22 FORMAT(1X,15F8.1/1X,5F8.1)
ISN 0079      105   READ (LI,110) (MCODE(I), THICK(I), STRCIN(I), I=1,4)
ISN 0080            IF(IP.EQ.IR) GO TO 1010
ISN 0082            MCODE(1)=MC(1)
ISN 0083            MCODE(2)=MC(4)
ISN 0084            MCODE(3)=MC(8)
              C
```

```
               C
               C      THICK REPRESENTS THE LAYER THICKNESSES OF REPRESENTATIVE
               C      SECTIONS
               C
               C
  ISN 0085            IF(THICK(1).NE.0) GO TO 1010
  ISN 0087            IF(NPT.NE.3.OR.NRU.NE.1) GO TO 50
  ISN 0089            THICK(1)=.75
  ISN 0090            THICK(2)=6.0
  ISN 0091            GO TO 1010
  ISN 0092     50     IF(NPT.NE.3.OR.NRU.NE.2) GO TO 51
  ISN 0094            THICK(1)=0.75
  ISN 0095            THICK(2)=8.0
  ISN 0096            GO TO 1010
  ISN 0097       51   IF(NPT.NE.1.OR.NRU.NE.1.OR.NLH.NE.1) GO TO 52
  ISN 0099            THICK(1)=2.0
  ISN 0100            THICK(2)=8.0
  ISN 0101            GO TO 1010
  ISN 0102     52     IF(NPT.NE.1.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 53
  ISN 0104            THICK(1)=4.0
  ISN 0105            THICK(2)=12.0
  ISN 0106            GO TO 1010
  ISN 0107     53     IF(NPT.NE.1.OR.NRU.NE.2.OR.NLH.NE.1) GO TO 54
  ISN 0109            THICK(1)=2.0
  ISN 0110            THICK(2)=8.
  ISN 0111            THICK(3)=6.0
  ISN 0112            GO TO 1010
  ISN 0113     54     IF(NPT.NE.1.OR.NRU.NE.2.OR.NLH.NE.2) GO TO 55
  ISN 0115            THICK(1)=4.0
  ISN 0116            THICK(2)=10.0
  ISN 0117            THICK(3)=6.0
  ISN 0118            GO TO 1010
  ISN 0119     55     MCODE(2)=MC(2)
  ISN 0120            MCODE(3)=MC(4)
  ISN 0121            MCODE(4)=MC(8)
  ISN 0122            IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.1) GO TO 56
  ISN 0124            THICK(1)=2.0
  ISN 0125            THICK(2)= 2.0
  ISN 0126            THICK(3)=8.0
  ISN 0127            GO TO 1010
  ISN 0128     56     IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 57
  ISN 0130            THICK(1)=3.0
  ISN 0131            THICK(2)= 4.0
  ISN 0132            THICK(3)=12.0
  ISN 0133            GO TO 1010
  ISN 0134     57     IF(NPT.NE.4.OR.NRU.NE.2.OR.NLH.NE.1) GO TO 58
  ISN 0136            THICK(1)=2.0
  ISN 0137            THICK(2)= 2.0
  ISN 0138            THICK(3)=8.0
  ISN 0139            THICK(4)=6.0
  ISN 0140     58     IF(NPT.NE.4.OR.NRU.NE.2.OR.NLH.NE.2) GO TO 1010
  ISN 0142            THICK(1)=3.0
  ISN 0143            THICK(2)= 4.0
  ISN 0144            THICK(3)=10.0
  ISN 0145            THICK(4)=6.0
  ISN 0146      1010 CONTINUE
  ISN 0147       110 FORMAT(5(A3,2X,2F5.0,1X))
  ISN 0148            WRITE (LO,120) (MCODE(I), THICK(I), STRCIN(I), I=1,4)
```

```
ISN 0149        120 FORMAT(1X,5(A3,2X,F5.1,F5.3,1X))
             C
             C      DETERMINE THE NUMBER OF LAYERS IN THE PAVEMENT STRUCTURE
             C
ISN 0150            IPFLG = 0
ISN 0151            DO 140 I=1,4
ISN 0152            IF (THICK(I) .LE. 0.0) GO TO 160
ISN 0154            NLAY = I
ISN 0155            STRC(I) = STRCIN(I)
ISN 0156            DO 135 J=1,NC
ISN 0157            IF (MCODE(I) .NE. MC(J)) GO TO 135
ISN 0159            IF ((IP .EQ. IF) .AND. ((J .EQ. 9) .OR. (J .EQ. 10))) GO TO 9994
ISN 0161            IF ((IP .EQ. IR) .AND. (J .EQ. 1)) IPFLG = I
ISN 0163            MTYPE(I) = J
ISN 0164            GO TO 140
ISN 0165        135 CONTINUE
ISN 0166            GO TO 9993
ISN 0167        140 CONTINUE
ISN 0168        160 IF (IPFLG .EQ. 0) GO TO 165
ISN 0170            IF (MTYPE(2) .NE. 9 .AND. MTYPE(2) .NE. 10) GO TO 9989
ISN 0172            NIS=1
ISN 0173            IP = IC
ISN 0174        165 STRC(5) = STRC(1)
ISN 0175            MCODE(5) = IACO
ISN 0176            GO TO 2
             C
             C      *** RIGID SECTION ***
             C
ISN 0177        200 IP = IR
ISN 0178            WLANE = VAL(1)
ISN 0179            WDTH = WLANE
ISN 0180             XK=VAL(2)
ISN 0181            IF (VAL(3) .NE. 0.0) AGG = VAL(3)
ISN 0183            IF (VAL(4) .NE. 0.0) E = VAL(4)
ISN 0185            IF (VAL(5) .NE. 0.0) DISTCT = VAL(5)
ISN 0187            IF (VAL(4) .NE. 0.0) E = VAL(4)
ISN 0189            GO TO 101
             C
             C      *** PERFORMANCE SECTION ***
             C
ISN 0190        300 PICON = VAL(1)
ISN 0191            PTERM = VAL(2)
ISN 0192            PIOV = VAL(3)
ISN 0193            PTOV = PTERM
ISN 0194            OVLIFE = VAL(4)
ISN 0195            OVLIF = NYAP
ISN 0196            IF (VAL(4) .GT. 0.) OVLIF = VAL(4)
ISN 0198            READ (LI,310) ATP
ISN 0199        310 FORMAT(3F10.0)
ISN 0200            WRITE (LO,320) ATP
ISN 0201            IF(ATP.LT.1) ATP=13.
ISN 0203        320 FORMAT(1X,8F10.2)
ISN 0204            SATP = ATP
ISN 0205            GO TO 2
             C
             C      *** AGE DISTRIBUTION SECTION ***
             C
ISN 0206        400 NASL = IVAL(1)
```

```
ISN 0207          ISLV = IVAL(2)
ISN 0208          FLRP = VAL(1)
           C
           C      READ AND ECHO PRINT THE DISTRIBUTION OF LANE MILES BY AGE
           C
ISN 0209          READ (LI,410) (YLM(I),I=1,NASL)
ISN 0210      410 FORMAT(16F5.0,/,14F5.0)
ISN 0211          WRITE (LO,420) (YLM(I),I=1,NASL)
ISN 0212      420 FORMAT(1X,15F8.1/1X,15F8.1)
ISN 0213          IF (ISLV .EQ. 0) GO TO 404
ISN 0215          READ (LI,430) (VI(I),I=1,NASL)
ISN 0216          WRITE (LO,320) (VI(I),I=1,NASL)
ISN 0217      430 FORMAT(16F5.0)
ISN 0218          READ (LI,430) (RI(I),I=1,NASL)
ISN 0219          WRITE (LO,320) (RI(I),I=1,NASL)
ISN 0220     404  IF(NASL.LE.25) GO TO 421
ISN 0222          DO 422 I=26,NASL
ISN 0223      422 YLM(25)=YLM(25)+YLM(I)
ISN 0224          NASL=25
ISN 0225      421 CONTINUE
ISN 0226          GO TO 2
           C
           C      *** OVERLAY SECTION ***
           C
ISN 0227      500 ICAC = IVAL(1)
ISN 0228          ICGR = IVAL(2)
           C
           C      READ AND ECHO PRINT THE OVERLAY PARAMETERS
           C
ISN 0229          READ (LI,510) PPVDSH,WPSH, WGSH, CACI, CGRI, ACDENS, GRDENS,CSCOAT
ISN 0230      510 FORMAT(8F10.0)
ISN 0231          WRITE (LO,520)PPVDSH,WPSH, WGSH, CACI, CGRI, ACDENS, GRDENS,CSCOAT
ISN 0232      520 FORMAT (1X,8F10.2)
ISN 0233          GO TO 2
           C
           C      *** MODEL MAINTENANCE SECTION ***
           C
ISN 0234      600 IARMS = IVAL(1)
ISN 0235          MFLG = 1
           C
           C      READ AND ECHO PRINT THE UNIT COSTS FOR BOTH FLEXIBLE AND RIGID
           C      PAVEMENTS, AND THE JOINT SEALING PARAMETERS
           C
ISN 0236          READ (LI,610) (UNTCST(I),I=1,3)
ISN 0237      610 FORMAT(3F10.0)
ISN 0238          READ (LI,620) UNTCST(4),DISS,DCON,DIN
ISN 0239          WRITE (LO,630) (UNTCST(I),I=1,4), DISS, DCON, DIN
ISN 0240      620 FORMAT(4F10.0,2F5.0,I5)
ISN 0241      630 FORMAT(1X,3F10.2/1X,6F10.2,I5)
ISN 0242          GO TO 2
           C
           C      *** HISTORICAL MAINTENANCE SECTION ***
           C
ISN 0243      700 IARMS = IVAL(1)
ISN 0244          MFLG = 2
           C
           C      READ AND ECHO PRINT THE MAINTENANCE COSTS PER LANE MILE BY AGE FOR
           C      FLEXIBLE PAVEMENTS
```

```
                   C
  ISN 0245               READ (LI,710) (USRMDL(I,1),I=1,24)
  ISN 0246           710 FORMAT(8F10.0)
  ISN 0247               WRITE (LO,720) (USRMDL(I,1),I=1,24)
  ISN 0248           720 FORMAT(1X,8F10.0)
                   C
                   C     READ AND ECHO PRINT THE MAINTENANCE COSTS PER LANE MILE BY AGE FOR
                   C     RIGID PAVEMENTS
                   C
  ISN 0249               READ (LI,710) (USRMDL(I,2),I=1,24)
                   C     WRITE (LO,720) (USRMDL(I,2),I=1,24)
  ISN 0250               GO TO 2
                   C
                   C     *** NO MAINTENANCE SECTION ***
                   C
  ISN 0251           800 MFLG = 0
  ISN 0252               GO TO 2
                   C
                   C     *** TRUCK TYPES SECTION ***
                   C
  ISN 0253           900 NTTY = IVAL(1)
  ISN 0254               NATT = IVAL(2)
  ISN 0255               PERCT(1)=VAL(1)
  ISN 0256               PERCT(2)=VAL(2)
  ISN 0257               PERCT(3)=VAL(3)
  ISN 0258               PERCT(4)=VAL(4)
  ISN 0259               NEWTRK = NEWTRK + 1
  ISN 0260               IF ((NTTY+NATT) .GT. 10) GO TO 9995
  ISN 0262               NTT = NTTY
  ISN 0263               K = 0
  ISN 0264               INTT = NTT + NATT
                   C
                   C     READ AND ECHO PRINT THE TRUCK LABELS
                   C
  ISN 0265               READ (LI,910) ((TTYP(M,J),M=1,2),J=1,INTT)
  ISN 0266           910 FORMAT(8(2A4,2X))
  ISN 0267               WRITE (LO,920) ((TTYP(M,J),M=1,2),J=1,INTT)
  ISN 0268           920 FORMAT(1X,8(2A4,2X))
                   C
                   C     READ AND ECHO PRINT THE AXLE CONFIGURATIONS
                   C
  ISN 0269               READ (LI,921) ((NAXLES(M,J),J=1,4),M=1,INTT)
  ISN 0270           921 FORMAT(8(4I2,2X))
  ISN 0271               WRITE (LO,922) ((NAXLES(M,J),J=1,4),M=1,INTT)
  ISN 0272           922 FORMAT(1X,8(4I2,2X))
  ISN 0273               DO 929 J=1,4
  ISN 0274               NT(J) = 0
  ISN 0275               DO 928 M=1,NTT
  ISN 0276               NT(J) = NT(J) + NAXLES(M,J)
  ISN 0277           928 CONTINUE
  ISN 0278           929 CONTINUE
                   C
                   C     READ AND ECHO PRINT THE TRUCK PERCENTAGES
                   C
  ISN 0279           935 K = K+1
  ISN 0280               DO 950 N=1,NYAP
  ISN 0281               READ (LI,930) I, (PTTYP(J,I,K),J=1,10), PCTTR(I,K)
  ISN 0282           930 FORMAT(I3,1X,11F6.0)
```

```
ISN 0283          WRITE (LO,940) I, (PTTYP(J,I,K),J=1,10), PCTTR(I,K)
ISN 0284      940 FORMAT(1X,I3,1X,11F6.2)
ISN 0285      950 CONTINUE
ISN 0286          IF ((NATT .GT. 0) .AND. (K .EQ. 1)) GO TO 935
ISN 0288          IF (K .EQ. 2) GO TO 2
ISN 0290          DO 970 J=1,10
ISN 0291          DO 960 I=1,20
ISN 0292          PTTYP(J,I,2) = PTTYP(J,I,1)
ISN 0293      960 CONTINUE
ISN 0294      970 CONTINUE
ISN 0295          GO TO 2
            C
            C     *** TITLE CARD SECTION ***
            C
            C     READ AND ECHO PRINT THE THREE TITLE CARDS
            C
ISN 0296     1000 DO 1030 J=1,3
ISN 0297          READ (LI,102) (TITLE(I,J),I=1,20)
ISN 0298          WRITE (LO,103) (TITLE(I,J),I=1,20)
ISN 0299     1030 CONTINUE
ISN 0300          NEWSYS = 1
ISN 0301          GO TO 2
            C
            C     *** OLD SECTIONS ***
            C
ISN 0302     1100 SAVMNT = VAL(1)
ISN 0303          IPOT = IVAL(1)
ISN 0304          IFF = IVAL(2)
ISN 0305          IF (IPOT .EQ. 0.OR.IPOT.EQ.3) GO TO 2
ISN 0307          IF (IPOT .EQ. 1) GO TO 1150
ISN 0309          PFNOPC = VAL(3)
ISN 0310          PCTINF = VAL(4)
            C
            C     READ AND ECHO PRINT THE ANNUAL PROJECTED OVERLAY FUNDS FOR PRESENT
            C     REGULATIONS
            C
ISN 0311          READ (LI,1110) (APOF(I,1),I=1,NYAP)
ISN 0312     1110 FORMAT(8F10.0)
ISN 0313          WRITE (LO,1120) (APOF(I,1),I=1,NYAP)
ISN 0314     1120 FORMAT(1X,8F10.0)
ISN 0315          IF (IFF .EQ. 1) GO TO 1140
ISN 0317          DO 1130 I=1,NYAP
ISN 0318          APOF(I,2) = APOF(I,1)
ISN 0319     1130 CONTINUE
ISN 0320          GO TO 2
            C
            C     READ AND ECHO PRINT THE ANNUAL PROJECTED OVERLAY FUNDS FOR FUTURE
            C     REGULATIONS
            C
ISN 0321     1140 READ (LI,1110) (APOF(I,2),I=1,NYAP)
ISN 0322          WRITE (LO,1120) (APOF(I,2),I=1,NYAP)
ISN 0323          GO TO 2
ISN 0324     1150 TPFPC = VAL(2)
ISN 0325          PFNOPC = VAL(3)
ISN 0326          GO TO 2
            C
            C     *** RUN PARAMETERS ***
            C
```

63

```
ISN 0327      1200 IF (IVAL(1) .NE. 0) NYAP = MINO(IVAL(1),20)
ISN 0329           IEQTRP = IVAL(2)
ISN 0330           AGR = VAL(1)
ISN 0331           PCTINT = VAL(2)
ISN 0332           IF(VAL(3).NE.0.0)XHCIO=VAL(3)
ISN 0334           IF(VAL(4).NE.0.0)XHCIM=VAL(4)
ISN 0336           TFCDNS=VAL(5)
ISN 0337           GO TO 2
              C
              C    *** LOAD LIMITS SECTION ***
              C
              C    READ THE PRESENT AND FUTURE LOAD LIMITS
              C
ISN 0338      1300 IEWS = IVAL(1)
ISN 0339           IDST = 1
ISN 0340           NEWTRK = NEWTRK + 2
ISN 0341           READ (LI,1310) PGVWL, PSAL, PTAL, PTRAL
ISN 0342      1310 FORMAT(4F10.0)
ISN 0343           WRITE (LO,1315) PGVWL, PSAL, PTAL, PTRAL
ISN 0344      1315 FORMAT(1X,4F10.2)
ISN 0345           READ (LI,1310) FGVWL, FSAL, FTAL, FTRAL
ISN 0346           WRITE (LO,1315) FGVWL, FSAL, FTAL, FTRAL
              C
              C    READ THE PRESENT AND FUTURE STEERING AXLE WEIGHTS FOR EACH TRUCK TYPE
              C
ISN 0347           NTT = INTT
ISN 0348           READ (LI,1320) (PSTAW(I),I=1,NTT)
ISN 0349           READ (LI,1320) (FSTAW(I),I=1,NTT)
ISN 0350      1320 FORMAT(10F8.0)
ISN 0351           WRITE (LO,1325) (PSTAW(I),I=1,NTT)
ISN 0352           WRITE (LO,1325) (FSTAW(I),I=1,NTT)
ISN 0353      1325 FORMAT(1X,10F8.0)
              C
              C    READ THE NEW EMPTY WEIGHT (AS A PERCENTAGE OF THE CURRENT EMPTY WEIGHT)
              C    FOR EACH TRUCK TYPE
              C
ISN 0354           IF (IEWS .EQ. 0) GO TO 2
ISN 0356           READ (LI,1320) (EPI(I),I=1,NTT)
ISN 0357           WRITE (LO,1330) (EPI(I),I=1,NTT)
ISN 0358      1330 FORMAT(1X,10F8.2)
ISN 0359           GO TO 2
              C
              C    *** SINGLE AXLE SECTION ***
              C
ISN 0360      1400 NLDI(1) = IVAL(1)
ISN 0361           NLD = IVAL(1)
ISN 0362           NTT = INTT
ISN 0363           STARTS(1) = VAL(1)
ISN 0364           NEWTRK = NEWTRK + 2
              C
              C    READ THE LOAD INTERVALS AND, FOR EACH TRUCK TYPE, THE NUMBER OF
              C    SINGLE AXLES FOR EACH INTERVAL
              C
ISN 0365           DO 1420 L=1,NLD
ISN 0366           READ (LI,1410) ELDINT, (SA(L,J),J=1,NTT)
ISN 0367      1410 FORMAT(F10.0,10F7.0)
ISN 0368           WRITE (LO,1415) ELDINT, (SA(L,J),J=1,NTT)
ISN 0369      1415 FORMAT(1X,F10.0,10F7.0)
```

```
ISN 0370          SA(L,11) = ELDINT
ISN 0371     1420 CONTINUE
ISN 0372          DO 1422 K=1,NLD
ISN 0373          SA(K,2)=0.000001
ISN 0374          SA(K,3)=0.000001
ISN 0375     1422 CONTINUE
ISN 0376          GO TO 2
            C
            C     *** TANDEM AXLE SECTION ***
            C
ISN 0377     1500 NLDI(2) = IVAL(1)
ISN 0378          NLD = IVAL(1)
ISN 0379          NTT = INTT
ISN 0380          STARTS(2) = VAL(1)
ISN 0381          NEWTRK = NEWTRK + 2
            C
            C     READ THE LOAD INTERVALS AND NUMBER OF DOUBLES PER TRUCK TYPE PER INTERVAL
            C
ISN 0382          DO 1510 L=1,NLD
ISN 0383          READ (LI,1410) ELDINT, (TA(L,J),J=1,NTT)
ISN 0384          WRITE (LO,1415) ELDINT, (TA(L,J),J=1,NTT)
ISN 0385          TA(L,11) = ELDINT
ISN 0386     1510 CONTINUE
ISN 0387          GO TO 2
            C
            C     *** TRIPLE AXLE SECTION ***
            C
ISN 0388     1600 NLDI(3) = IVAL(1)
ISN 0389          NLD = IVAL(1)
ISN 0390          NTT = INTT
ISN 0391          STARTS(3) = VAL(1)
ISN 0392          NEWTRK = NEWTRK + 2
            C
            C     READ THE LOAD INTERVALS AND NUMBER OF TRIPLES PER TRUCK TYPE PER INTERVAL
            C
ISN 0393          DO 1610 L=1,NLD
ISN 0394          READ (LI,1410) ELDINT, (TR(L,J),J=1,NTT)
ISN 0395          WRITE (LO,1415) ELDINT, (TR(L,J),J=1,NTT)
ISN 0396          TR(L,11) = ELDINT
ISN 0397     1610 CONTINUE
ISN 0398          GO TO 2
            C
            C     *** GROSS VEHICLE WEIGHT SECTION ***
            C
ISN 0399     1700 NLDI(4) = IVAL(1)
ISN 0400          NLD = IVAL(1)
ISN 0401          NTT = INTT
ISN 0402          STARTS(4) = VAL(1)
ISN 0403          NEWTRK = NEWTRK + 2
            C
            C     READ THE LOAD INTERVALS AND THE NUMBER OF EACH TRUCK TYPE WHOSE GVW FALLS
            C     WITHIN EACH INTERVAL
            C
ISN 0404          DO 1710 L=1,NLD
ISN 0405          READ (LI,1410) ELDINT, (VG(L,J),J=1,NTT)
ISN 0406          WRITE (LO,1415) ELDINT, (VG(L,J),J=1,NTT)
ISN 0407          VG(L,11) = ELDINT
ISN 0408     1710 CONTINUE
```

```
   ISN 0409           GO TO 2
                C
                C     *** EMPTY VEHICLE WEIGHT SECTION ***
                C
   ISN 0410      1800 NLDI(5) = IVAL(1)
   ISN 0411           NLD = IVAL(1)
   ISN 0412           NTT = INTT
   ISN 0413           STARTS(5) = VAL(1)
   ISN 0414           NEWTRK = NEWTRK + 2
                C
                C     READ THE LOAD INTERVALS AND THE NUMBER OF EACH TRUCK TYPE WHOSE EVW FALLS
                C     WITHIN EACH INTERVAL
                C
   ISN 0415           DO 1810 L=1,NLD
   ISN 0416           READ (LI,1410) ELDINT, (VE(L,J),J=1,NTT)
   ISN 0417           WRITE (LO,1415) ELDINT, (VE(L,J),J=1,NTT)
   ISN 0418           VE(L,11) = ELDINT
   ISN 0419      1810 CONTINUE
   ISN 0420           GO TO 2
                C
                C     *** STEERING AXLES SECTION ***
                C
   ISN 0421      1900 NLDI(6) = IVAL(1)
   ISN 0422           NLD = IVAL(1)
   ISN 0423           NTT = INTT
   ISN 0424           STARTS(6) = VAL(1)
   ISN 0425           IDST = 6
   ISN 0426           NEWTRK = NEWTRK + 2
                C
                C     READ THE LOAD INTERVALS AND, FOR EACH TRUCK TYPE, THE NUMBER OF
                C     STEERING AXLES FOR EACH INTERVAL
                C
   ISN 0427           DO 1910 L=1,NLD
   ISN 0428           READ (LI,1410) ELDINT, (ST(L,J),J=1,NTT)
   ISN 0429           WRITE (LO,1415) ELDINT, (ST(L,J),J=1,NTT)
   ISN 0430           ST(L,11) = ELDINT
   ISN 0431      1910 CONTINUE
   ISN 0432           GO TO 2
                C
                C     *** OUTPUT KEYWORD SECTION ***
                C
   ISN 0433      2000 IOUT = IVAL(1)
   ISN 0434           GO TO 2
                C
                C     *** KEYWORD ERROR PROCESSING SECTION ***
                C
   ISN 0435      9989 WRITE (LO,9089) IPFLG
   ISN 0436      9089 FORMAT(/1X,19H*** ERROR IN LAYER ,I1,4H ***/
                     1       38H ACP NOT PERMITTED FOR RIGID PAVEMENT /
                     2       30H UNLESS ABOVE JCP OR CRC LAYER//
                     3       15H RUN TERMINATED)
   ISN 0437           GO TO 9999
   ISN 0438      9992 IGO = 3
   ISN 0439           GO TO 99999
   ISN 0440      9993 WRITE (LO,9093)
   ISN 0441      9093 FORMAT(/1X,37H*** UNRECOGNIZABLE MATERIALS CODE ***//
                     1       15H RUN TERMINATED)
   ISN 0442           GO TO 9999
```

```
  ISN 0443      9994 WRITE (LO,9094)
  ISN 0444      9094 FORMAT(/1X,51H*** ILLEGAL MATERIAL CODE FOR THIS TYPE OF PAVEMENT,
                   1       4H ***//15H RUN TERMINATED)
  ISN 0445           GO TO 9999
  ISN 0446      9995 WRITE (LO,9095)
  ISN 0447      9095 FORMAT(/1X,28H*** TOO MANY TRUCK TYPES ***//
                   1       15H RUN TERMINATED)
  ISN 0448           GO TO 9999
  ISN 0449      9996 WRITE (LO,9096)
  ISN 0450      9096 FORMAT(/1X,44H*** SPECIFIED KEYWORD NOT FOUND IN TABLE ***,
                   1       //15H RUN TERMINATED)
  ISN 0451           GO TO 9999
  ISN 0452      9997 IGO = 1
  ISN 0453           GO TO 99999
  ISN 0454      9998 WRITE (LO,9098)
  ISN 0455      9098 FORMAT(/1X,44H*** STOP DIRECTIVE FOUND OUT OF SEQUENCE ***,
                   1       //15H RUN TERMINATED)
  ISN 0456      9999 IGO = 4
  ISN 0457     99999 DO 3500 I=1,30
  ISN 0458           XLM(I) = YLM(I)
  ISN 0459      3500 CONTINUE
  ISN 0460           S = SPCJT
  ISN 0461           XML = 0.
  ISN 0462           IF (XMLI .NE. 0.) XML = XMLI
  ISN 0464           LP = MINO(4, MAXO(1,INT(7.1 - 2.*PTERM)))
  ISN 0465           IF (FLRP .LE. 0.) FLRP = FLRPTP(LP)
  ISN 0467           RETURN
  ISN 0468           END
```


*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

STATISTICS   SOURCE STATEMENTS =   467, PROGRAM SIZE =   10756, SUBPROGRAM NAME = INPUT

STATISTICS NO DIAGNOSTICS GENERATED

****** END OF COMPILATION ******   836K BYTES OF CORE NOT USED

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

      ISN 0002            SUBROUTINE INPRNT
      ISN 0003            COMMON /TEMPC/ CONTP(25),DISTCT
      ISN 0004            COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
      ISN 0005            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
      ISN 0006            COMMON /FUNDS/ APOF(20,2), RTINT, RTINF
      ISN 0007            COMMON /INTVLS/ STARTS(6)
      ISN 0008            COMMON /IO/ LI, LO, LD
      ISN 0009            COMMON /LABELS/ MATLAB(5,10)
      ISN 0010            COMMON /LDS/ PGVWL, PSAL, PTAL, PTRAL, FGVWL, FSAL, FTAL, FTRAL,
                         1             PSTAW(10), FSTAW(10)
      ISN 0011            COMMON /LMP/ XLM(30), YLM(30), POTLM(20,2), OUTP(20,2), TOTALM,
                         1      PPF, TPF, PFNO, NASL, NSLR,TOVLM(30,2),XLM2(30)
      ISN 0012            COMMON /MISC/ IPOT, IARMS, OLDMNT, AGF
      ISN 0013            COMMON /MNTPAR/ UNTCST(4),USRMDL(31,3),WDTH,S,DISS,DCON,DIN,MFLG
      ISN 0014            COMMON /EXTRA/ PTOVTK,TPE,PFO,XMNOTK,XMXOTK,NIS
      ISN 0015            COMMON /NEWSYS/ NEWSYS
      ISN 0016            COMMON /NMBR/ SA(30,11), TA(30,11), TR(50,11), VE(30,11),
                         1              VG(75,11), NLDI(6), EPI(10), ST(30,11)
      ISN 0017            COMMON /OUTSWH/ IOUT
      ISN 0018            COMMON /OVRLAY/ XHCIO,XHCIM,WLANE, WPSH, WGSH, PPVDSH, CAC, CGR
                         1    , CSCOAT
      ISN 0019            COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
      ISN 0020            COMMON /STRCOE/ STRCD(8), CC(4), MC(11), NC, STRC(5), RFS(4),
                         1              RFB(4)
      ISN 0021            COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
      ISN 0022            COMMON /TIME/ ATP, OVLIF, NYAP, NYR,  YR(40)
      ISN 0023            COMMON /TITLE/ TITLE(20,3), SECTTL(20)
      ISN 0024            COMMON /TRTYP/ TTYP(2,10), PTTYP(10,20,2), PCTTR(20,2),PERCT(4),
                         1              NAXLES(10,4), NT(4), NTTY, NATT, NTT, NEWTRK
      ISN 0025            COMMON /SLVG/ ISLV, FLRP, VI(30), RI(30), VL(30), RL(30),
                         1             U(30), PL(30), MI(30), P(20), VP(20), RP(20),
                         2             PB, VPB, RPB, NS, NY, SV(6,2), SVB, FLRPTP(4)
      ISN 0026            COMMON /SWTCHS/ OVLIFE, PCTINT, PCTINF, TPFPC, PFNOPC, AGR, SPCJT,
                         1              XMLI, CACI, CGRI, ICAC, ACDENS, ICGR, GRDENS,
                         2              INTT, SAVMNT, IDST, NLD, MCODE(5), TFCDNS
      ISN 0027            RTINT = PCTINT * 0.01
      ISN 0028            RTINF = PCTINF * 0.01
      ISN 0029            TPF = TPFPC*.01
      ISN 0030            PFNO = PFNOPC * 0.01
      ISN 0031            AGF = AGR * 0.01
      ISN 0032            CAC = CACI
      ISN 0033            CGR = CGRI
      ISN 0034            IF (ICAC .EQ. 1) GO TO 4000
      ISN 0036            IF (ICAC .EQ. 2) GO TO 4010
      ISN 0038            CAC = CACI * 36.
      ISN 0039            GO TO 4010
      ISN 0040       4000 CAC = CACI * (ACDENS * 27.) / 2000.
      ISN 0041       4010 IF (ICGR .EQ. 2) GO TO 99999
      ISN 0043            IF (ICGR .EQ. 1) GO TO 4020
      ISN 0045            CGR = CGRI * 36.
      ISN 0046            GO TO 99999
      ISN 0047       4020 CGR = CGRI * (GRDENS * 27.) / 2000.
      ISN 0048      99999 RETURN
```

```
LEVEL 2.3.0 (JUNE 78)        INPRNT         OS/360  FORTRAN H EXTENDED            DATE 84.262/15.03.11        PAGE    2

  ISN 0049           END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      48, PROGRAM SIZE =      676, SUBPROGRAM NAME =INPRNT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        920K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002              SUBROUTINE INIT (IGO)
   ISN 0003              COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
   ISN 0004              COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
   ISN 0005              COMMON /STRCOE/ STRCD(8),CC(4),MC(11),NC,STRC(5),RFS(4),RFB(4)
   ISN 0006              COMMON /TIME/ ATP, OVLIF, NYAP, NYR,  YR(40)
   ISN 0007              DATA ICON, F /2, 1. /
                   C     ICON IS THE INDEX ON CONDITION FACTOR USED TO RELATE AN  OLD PCC
                   C     PAVEMENT WITH AN AC OVERLAY TO AN EQUIVALENT SLAB THICKNESS.
                   C     F IS A FACTOR ALSO USED IN THE ABOVE RELATION.
   ISN 0008              GO TO (100, 200, 300), IGO
                   C     HERE FOR PROGRAM INITIALIZATION, FIRST EXECUTION.
   ISN 0009          100 DO 110 J=1,NYR
   ISN 0010              YR(J) = FLOAT(J)
   ISN 0011          110 CONTINUE
   ISN 0012              GO TO 900
                   C
                   C     HERE FOR SET UP CHORES AFTER READING INPUT DATA.
   ISN 0013          200 CONTINUE
                   C     WE HAVE ALL THE INPUT FOR A REPRESENTATIVE SECTION.  DETERMINE -SN-
                   C     OR -D- FOR COMPOSITE PAVTS, AS WELL AS SET UP STRUCTURAL COEF.
   ISN 0014              IF (IP .EQ. IR .OR. IP .EQ. IC) GO TO 230
   ISN 0016              SN = 0.
   ISN 0017              DO 215 L=1,NLAY
   ISN 0018              M = MTYPE(L)
                   C     REPLACE VALUE IN DATA STATEMENT WITH VALUE READ IN.
   ISN 0019              IF (STRC(L) .NE. 0.) STRCD(M) = STRC(L)
                   C     IF NO VALUE READ IN, SET VALUE FROM THE DATA STATEMENT.
   ISN 0021              IF (STRC(L) .EQ. 0.) STRC(L) = STRCD(M)
   ISN 0023          215 SN = SN + STRC(L)*THICK(L)
                   C     SET -A- VALUE FOR OVERLAY = -A- FOR AC IF NOT READ IN SEPARATELY.
   ISN 0024              IF (STRC(5) .EQ. 0.) STRC(5) = STRCD(1)
   ISN 0026              GO TO 250
   ISN 0027          230 XJ = 3.2
                   C     CONTINUITY FACTOR FOR PCC PAVEMENTS   3.2 FOR JCP, 2.2 FOR CRC.
                   C     TEST FOR COMPOSITE PAVEMENT (AC TOP LAYER READ UNDER -RIGID-.)
   ISN 0028              IF (MTYPE(1) .EQ. 1) GO TO 240
   ISN 0030              D = THICK(1)
   ISN 0031              IF (MTYPE(1) .EQ. 10) XJ = 2.2
   ISN 0033              GO TO 250
                   C     EQUIVALENT SLAB THICKNESS FOR INITIALLY COMPOSITE PAVT.
   ISN 0034          240 D = (THICK(1)/2.5 + CC(ICON)*THICK(2))/F
   ISN 0035              IP = IC
   ISN 0036              IF (MTYPE(2) .EQ. 10) XJ = 2.2
   ISN 0038          250 CONTINUE
   ISN 0039              GO TO 900
                   C
   ISN 0040          300 CONTINUE
                   C
   ISN 0041          900 CONTINUE
   ISN 0042              RETURN
   ISN 0043              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
```

```
 LEVEL 2.3.0 (JUNE 78)                          OS/360  FORTRAN H EXTENDED           DATE 84.262/15.03.14       PAGE    2

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      42, PROGRAM SIZE =      992, SUBPROGRAM NAME =  INIT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                               924K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED              DATE 84.262/15.03.15        PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

  ISN 0002            SUBROUTINE DISTR ( P, NHIST, NSLICE)
  ISN 0003            COMMON /SWTCHS/ OVLIFE, PCTINT, PCTINF, TPFPC, PFNOPC, AGR, SPCUT,
                     1                XMLI, CACI, CGRI, ICAC, ACDENS, ICGR, GRDENS,
                     2                INTT, SAVMNT, IDST, NLD, MCODE(5), TFCDNS
  ISN 0004            COMMON /BURKE/ XLAMB, GAMMA, TFBAP
  ISN 0005            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
  ISN 0006            DIMENSION P(25)
  ISN 0007            IF (IP .EQ. IF) GO TO 100
                C->FIX SURVIVAL CURVES FOR RIGID PAVEMENTS
  ISN 0009            P(1)=0
  ISN 0010            P(2)=0.
  ISN 0011            P(3)=0.
  ISN 0012            P(4)=0.0125
  ISN 0013            P(5)=.0125
  ISN 0014            P(6)=0.0295
  ISN 0015            P(7)=P(6)
  ISN 0016            P(8)=P(7)
  ISN 0017            P(9)=.03
  ISN 0018            P(10)=P(9)
  ISN 0019            P(11)=.085
  ISN 0020            P(12)=P(11)
  ISN 0021            P(13)=.0325
  ISN 0022            P(14)=P(13)
  ISN 0023            P(15)=.0595
  ISN 0024            P(16)=P(15)
  ISN 0025            P(17)=.0325
  ISN 0026            P(18)=.085
  ISN 0027            P(19)= P(18)
  ISN 0028            P(20)=.03
  ISN 0029            P(21)=.03
  ISN 0030            P(22)=.0295
  ISN 0031            P(23)=P(22)
  ISN 0032            P(24)=P(23)
  ISN 0033            P(25)=.025
  ISN 0034            GO TO 999
  ISN 0035        100 CONTINUE
                C->GET INITIAL TRAFFIC
  ISN 0036            AGF=AGR/100.
  ISN 0037            WO=TFBAP*(1+AGF)**(-NSLICE)
                C->GET P(I) FOR I=1 TO NHIST
  ISN 0038            ACUM=0
  ISN 0039            ACPLYR=0
  ISN 0040            DO 10 I=1,NHIST
                C-------->TRANSFORM YEARS INTO ACCUMULATED LOADS AT AGE I
  ISN 0041               ACUM=ACUM + WO*(1+AGF)**I
                C-------->GET CUMMULATIVE FRACTION OF PAVEMENTS THAT FAILED
                C          "ACUMIL" STANDS FOR ACCUMULATED EAL IN MILLIONS
  ISN 0042               ACUMIL = ACUM/1000000
  ISN 0043               POWER = -(XLAMB*ACUMIL)**GAMMA
  ISN 0044               IF (POWER .GT. -5.4E-79) POWER = -5.4E-79
  ISN 0046               ACPNOW=1-EXP(POWER)
                C-------->GET FRACTION OF PAVEMENTS THAT FAILED DURING YEAR I
  ISN 0047               P(I)=ACPNOW-ACPLYR
```

```
              C-------->UPDATE POINTER AND DO IT AGAIN
  ISN 0048              ACPLYR=ACPNOW
  ISN 0049         10   CONTINUE
  ISN 0050        999   RETURN
  ISN 0051              END
```

```
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       50, PROGRAM SIZE =      1066, SUBPROGRAM NAME = DISTR

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                                 924K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE EALGET
                 C     THIS ROUTINE CALCULATES THE RATIO OF EAL PER UNIT TIME UNDER THE
                 C     PROPOSED REGULATIONS TO THAT UNDER THE PRESENT REGULATIONS,
                 C     SUBJECT TO THE RESTRAINT OF EQUAL PAYLOAD PER UNIT TIME(IEQTRP=0),
                 C     OR TO THE RESTRAINT OF EQUAL NUMBER OF TRIPS (IEQTRP=1).
   ISN 0003            COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
   ISN 0004            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
   ISN 0005            COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
   ISN 0006            COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
   ISN 0007            COMMON /TIME/ ATP, OVLIF, NYAP, NYR,  YR(40)
   ISN 0008            COMMON /SWTCHS/ OVLIFE, PCTINT, PCTINF, TPFPC, PFNOPC, AGR, SPCUT,
                      1                XMLI, CACI, CGRI, ICAC, ACDENS, ICGR, GRDENS,
                      2                INTT, SAVMNT, IDST, NLD, MCODE(5), TFCDNS
   ISN 0009            COMMON /TRTYP/ TTYP(2,10), PTTYP(10,20,2), PCTTR(20,2),PERCT(4),
                      1               NAXLES(10,4),NT(4), NTTY, NATT, NTT, NEWTRK
   ISN 0010            COMMON /BURKE/ XLAMB, GAMMA, TFBAP
   ISN 0011            DIMENSION S1(10), S2(10), T1(10), T2(10)
   ISN 0012            IPVT = IP
   ISN 0013            IF (IP .EQ. IC) IPVT = IR
                 C     CALL -TRAFIC- ONLY IF NEW LIMITS OR WEIGHT DISTRIBUTIONS HAVE BEEN
                 C     READ FOR THIS PROBLEM
   ISN 0015            IF (NEWTRK .GT. 1) CALL TRAFIC
   ISN 0017            CALL EAL18 (SN, D, PTERM, IPVT)
                 C     EAL18 RETURNS 18K EAL PER AVERAGE TRUCK, EALPT, AND PAYLOAD PER
                 C     AVERAGE TRUCK, APPT, FOR EACH TRUCK TYPE.
                 C     FOR EACH YEAR OBTAIN THE (NORMALIZED) TOTAL PAYLOAD AND TOTAL 18K
                 C     EAL
   ISN 0018            DO 10 J=1,NYAP
   ISN 0019            CALL MULT (PTTYP(1,J,1), APPT(1,1), NTTY, S1)
   ISN 0020            CALL MULT (PTTYP(1,J,1), EALPT(1,1), NTTY, T1)
   ISN 0021            CALL SUM (S1, NTTY, SUM1)
   ISN 0022            CALL SUM (T1, NTTY, TUM1)
   ISN 0023            IF (J .EQ. 1) TFBAP=TUM1*TFCDNS
   ISN 0025         10 CONTINUE
   ISN 0026            RETURN
   ISN 0027            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       26, PROGRAM SIZE =       702, SUBPROGRAM NAME =EALGET

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        924K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                          OS/360  FORTRAN H EXTENDED           DATE 84.262/15.03.17       PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002              SUBROUTINE OVCOST (THOV, OVCST)
                   C     OBTAINS COST/(LANE MILE) FOR GIVEN OVERLAY THICKNESS
   ISN 0003              COMMON /OVRLAY/ XHCIO,XHCIM,WLANE, WPSH, WGSH, PPVDSH, CAC, CGR
                        1     , CSCOAT
   ISN 0004              DATA C1/16.2962963/
                   C     COSTS ARE INPUT TO THIS ROUTINE IN DOLLARS/CU YD.
                   C     C1 IS THE NUMBER OF CUBIC YDS IN A LAYER 1 MILE BY 1 FOOT BY 1 IN.
                   C
   ISN 0005              F = PPVDSH/100.
   ISN 0006              TH = THOV
                   C     FIND THE VOLUME/(LANE MILE) OF ROAD OVERLAY, OF PAVED SHOULDER
                   C     OVERLAY, AND OF GRANULAR SHOULDER OVERLAY
   ISN 0007              VPO = WLANE*TH*C1
   ISN 0008              VPSO = WPSH*TH*C1
   ISN 0009              VGSO = WGSH*TH*C1
                   C     PAVEMENT OVERLAY COST
   ISN 0010              PVTOC = VPO*CAC
                   C     UNPAVED SHOULDER OVERLAY COST
   ISN 0011              UPSHOC = CGR*(1.-F)*VGSO
                   C     PAVED SHOULDER COST
   ISN 0012              PSHOC = CAC*F*VPSO
                   C     TOTAL OVERLAY COST
   ISN 0013              OVCST  = PVTOC + UPSHOC + PSHOC
   ISN 0014              RETURN
   ISN 0015              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       14, PROGRAM SIZE =       386, SUBPROGRAM NAME =OVCOST

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          924K BYTES OF CORE NOT USED
```

```
  LEVEL 2.3.0 (JUNE 78)                            OS/360  FORTRAN H EXTENDED            DATE 84.262/15.03.18        PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002             SUBROUTINE ACCTFC (TFC1, AGF, NYR, TFCA)
                  C     CUMULATIVE TRAFFIC BY YEAR FROM BASE YEAR (18 KIP EAL).
                  C     INPUT
                  C     TFC1   - 18KIP EAL IN BASE YEAR (YEAR 1)
                  C     AGF    - ANNUAL GROWTH FACTOR (PERCENT/100.)
                  C     NYR    - NUMBER OF YEARS FOR WHICH ACCUMULATED TRAFFIC DESIRED.
                  C     OUTPUT
                  C     TFCA   - ARRAY OF CUMULATIVE 18 KIP EAL THROUGH END OF INDEX YEAR.
   ISN 0003             DIMENSION TFCA (NYR)
   ISN 0004             TFCA(1) = TFC1
   ISN 0005             T = TFC1
   ISN 0006             DO 10 I=2,NYR
   ISN 0007             T = T*(1. + AGF)
   ISN 0008             TFCA(I) = TFCA(I-1) + T
   ISN 0009          10 CONTINUE
   ISN 0010             RETURN
   ISN 0011             END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       10, PROGRAM SIZE =       398, SUBPROGRAM NAME =ACCTFC

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        928K BYTES OF CORE NOT USED
```

```
 LEVEL 2.3.0 (JUNE 78)                         OS/360  FORTRAN H EXTENDED            DATE 84.262/15.03.19       PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE OUTPUT (LOCSW)
   ISN 0003            RETURN
   ISN 0004            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       3, PROGRAM SIZE =      176, SUBPROGRAM NAME =OUTPUT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          928K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED              DATE 84.262/15.03.21        PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

      ISN 0002              SUBROUTINE OVTHKF (XNOV, THOV,YR)
      ISN 0003              REAL*8 THICK1(5),DMDRU,DMDRE
      ISN 0004              COMMON /MECH/XKT,NRU,NLH,ND,NDEL,IACR,IYR,JYR,CONSTR(20)
      ISN 0005              COMMON/HOR/A(10),B(10),C(10),DT(10),DF(10),S(10),T(10),TR(5),PI(5)
                           *,PT(5),AC(5),AA,SCT(5),XMNW18(10) ,XKTO
      ISN 0006              COMMON /EXTRA/ PTOVTK,TPE,PFO,XMNOTK,XMXOTK,NIS
      ISN 0007              COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
      ISN 0008              COMMON /PSI/PF,PICON, PTERM, PIOV, PTOV
      ISN 0009              COMMON /STRCOE/ STRCD(8),CC(4),MC(11),NC,STRC(5),RFS(4),RFB(4)
      ISN 0010              DIMENSION BETA(5,2,2),CO(5,2,2)
      ISN 0011              BETA(1,1,1)=-1.5287
      ISN 0012              BETA(1,1,2)=-1.5387
      ISN 0013              BETA(3,1,1)=-1.4370
      ISN 0014              BETA(3,1,2)=-1.4370
      ISN 0015              BETA(4,1,1)=-1.5605
      ISN 0016              BETA(4,1,2)=-1.5776
      ISN 0017              BETA(1,2,1)=-1.53
      ISN 0018              BETA(1,2,2)=-1.562
      ISN 0019              BETA(3,2,1)=-1.4649
      ISN 0020              BETA(3,2,2)=-1.4649
      ISN 0021              BETA(4,2,1)=-1.5700
      ISN 0022              BETA(4,2,2)=-1.6085
      ISN 0023              CO(3,1,1)=600.
      ISN 0024              CO(3,1,2)=600.
      ISN 0025              CO(1,1,1)=10000.0
      ISN 0026              CO(4,1,1)=10000.0
      ISN 0027              CO(1,1,2)=50000.0
      ISN 0028              CO(4,1,2)=50000.0
      ISN 0029              CO(3,2,1)=1000.
      ISN 0030              CO(3,2,2)=1000.0
      ISN 0031              CO(1,2,1)=10000.0
      ISN 0032              CO(4,2,1)=10000.0
      ISN 0033              CO(1,2,2)=100000.0
      ISN 0034              CO(4,2,2)=100000.0
      ISN 0035              NLAY1=NLAY+1
      ISN 0036              DO 10  K=2, NLAY1
      ISN 0037          10  THICK1(K)=THICK(K-1)
      ISN 0038              THICK1(1)=XMNOTK
      ISN 0039              IF (PF.GT.PTERM.OR.TPE.EQ.0) GO TO 100
      ISN 0041              TNPT=NPT
      ISN 0042              NPT=4
      ISN 0043              CALL PSIT (P,PFO,W,XKTO)
      ISN 0044              NPT=TNPT
      ISN 0045              IF (PFO.GE.PTOV) GOTO 3
      ISN 0047              GO TO 8
      ISN 0048         100  IF (PFO.GE.PTOV) GOTO 3
      ISN 0050              IF(PFO.EQ.PTERM) PTERM=PTERM+0.05
      ISN 0052              XKTO=-.8*XNOV*ALOG((PIOV-PTERM)/(PIOV-PFO))
      ISN 0053         8    DMDRE=(100.0+XKTO/CO(NPT,NRU,NLH))**(BETA(NPT,NRU,NLH))
      ISN 0054              N=(XMXOTK-XMNOTK)*4+0.9999
      ISN 0055              DO 1 I=1,N
      ISN 0056              CALL RUSIAN  (THICK1,DMDRU,NLAY1,NPT,NRU,NLH)
      ISN 0057              IF(DMDRU.LE.DMDRE)GO TO 2
```

```
ISN 0059        1 THICK1(1)=THICK1(1)+.25
ISN 0060        2 THOV=THICK1(1)
ISN 0061          GO TO 4
ISN 0062        3 THOV=XMNOTK
ISN 0063        4 RETURN
ISN 0064          END
```

```
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      63, PROGRAM SIZE =      1704, SUBPROGRAM NAME =OVTHKF

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  920K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 84.262/15.03.21      PAGE   1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002              FUNCTION RWT18L(D,PI,PT)
                  C      AASHO-RIGID PREDICTION OF 18 KIP EAL TO TERMINAL PSI
   ISN 0003              GT = ALOG10((PI-PT)/(PI-1.5))
   ISN 0004              GTERM = GT/(1.+1.624E7/(D+1.)**8.46)
   ISN 0005              RWT18L= 7.35*ALOG10(D+1.)-0.06+GTERM
   ISN 0006              RETURN
   ISN 0007              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       6, PROGRAM SIZE =       480, SUBPROGRAM NAME =RWT18L

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                       928K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 84.262/15.03.22       PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            FUNCTION RNAASH(DA)
                C      MODIFY AASHO-RIGID PREDICTION FOR NON-AASHO CONDITIONS
   ISN 0003            COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
   ISN 0004            Z = E/XK
   ISN 0005            CT = 223.3
   ISN 0006            IKK = AGG
   ISN 0007            IF( IKK .EQ. 0 ) CT=204.16
   ISN 0009            D75 = DA**.75
   ISN 0010            RNAASH = ALOG10((CT/215.63)*(D75-1.132)/
                      1 (D75-18.42/Z**0.25))
   ISN 0011            RETURN
   ISN 0012            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       11, PROGRAM SIZE =       478, SUBPROGRAM NAME =RNAASH

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002           FUNCTION GPSIR (XN, PI, D)
               C      AASHO-RIGID PREDICTION OF PSI AFTER GIVEN 18 KIP EAL
   ISN 0003           DATA MAX, TEST /10, .001 /
   ISN 0004           EXP10(X) = EXP(2.302585*X)
   ISN 0005           PTN = 3.
   ISN 0006           ITER = 0
   ISN 0007           RN = RNAASH(D)
   ISN 0008           XNL = ALOG10(XN)
   ISN 0009           DT1 = 7.35*ALOG10(D+1.) - 0.06
   ISN 0010           DT2 = 1. + 1.624E7/(D+1.)**8.46
   ISN 0011        10 ITER = ITER + 1
   ISN 0012           IF (ITER .GT. MAX) GO TO 30
   ISN 0014           PT = PTN
   ISN 0015           GT = (XNL - DT1 - (4.22 - 0.32*PT)*RN)*DT2
   ISN 0016           PTN = PI - (PI - 1.5)*EXP10(GT)
   ISN 0017           IF (ABS(PTN - PT) .LT. TEST) GO TO 20
   ISN 0019           GO TO 10
   ISN 0020        20 GPSIR = PTN
   ISN 0021           RETURN
   ISN 0022        30 GPSIR = PTN
   ISN 0023           WRITE (6,1) MAX, PTN, PT, XN
   ISN 0024         1 FORMAT (1X, 37HFUNCTION GPSIR DID NOT CONVERGE AFTER, I5,
                   1         11H ITERATIONS / 1X,33HLAST AND PREVIOUS PSI VALUES WERE,
                   2         2F10.6 / 1X,  3HFOR , F10.0,26H 18KIP EAL TO DATE. ABORT.)
   ISN 0025           STOP
   ISN 0026           END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      25, PROGRAM SIZE =      944, SUBPROGRAM NAME = GPSIR

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                  924K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002              SUBROUTINE GETD (W18, PI, PT, DB, DF)
                 C       AASHO-RIGID SLAB THICKNESS FOR GIVEN LIFE (18 KIP EAL) AND INITIAL
                 C       AND TERMINAL PSI
   ISN 0003              DATA MAX, TEST /10, .001 /
   ISN 0004              EXP10(X) = EXP(2.302585*X)
   ISN 0005              ITER = 0
   ISN 0006              DN = DB
   ISN 0007           10 ITER = ITER + 1
   ISN 0008              IF (ITER .GT. MAX) GO TO 99
   ISN 0010              D = DN
   ISN 0011              W = RWT18L(D,PI,PT) + (4.22-.32*PT)*RNAASH(D)
   ISN 0012              DTERM = 7.35*ALOG10(D + 1.)
   ISN 0013              D1NLOG = (W18 - (W - DTERM))/7.35
   ISN 0014              DN = EXP10(D1NLOG) - 1.
   ISN 0015              IF (ABS(D-DN) .LT. TEST) GO TO 20
   ISN 0017              GO TO 10
   ISN 0018           20 DF = DN
   ISN 0019              RETURN
   ISN 0020           99 DF = D
   ISN 0021              WRITE (6,1) D, DN, W18, PI,PT,DB
   ISN 0022              RETURN
   ISN 0023            1 FORMAT (1X, 27HTOO MANY ITERATIONS IN GETD /
                        1        1X, 20HLAST TWO VALUES WERE , 2F8.4 /
                        2        1X, 36HINPUT  LOG N18, PI, PT, STARTING D =  /
                        3        1X, 4F10.4 /)
   ISN 0024              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      23, PROGRAM SIZE =      870, SUBPROGRAM NAME =  GETD

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          924K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE OVTHKR (D, EXD, TH)
                 C     OBTAIN THICKNESS OF AC OVERLAY TO BRING EQUIVALENT SLAB
                 C     THICKNESS, D, OF COMBINATION UP TO NEW DESIGN VALUE.
                 C     (EXISTING D DISCOUNTED FOR USE)
   ISN 0003            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
   ISN 0004            COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
   ISN 0005            COMMON /STRCOE/ STRCD(8),CC(4),MC(11),NC,STRC(5),RFS(4),RFB(4)
   ISN 0006            DATA F/1./
   ISN 0007            INDX = 7.5 - 2.*PTERM
   ISN 0008            INDX = MINO(4,MAXO(1,INDX))
   ISN 0009            C = CC(INDX)
   ISN 0010            TH = 2.5*(F*D - C*EXD)
   ISN 0011            RETURN
   ISN 0012            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       11, PROGRAM SIZE =       424, SUBPROGRAM NAME =OVTHKR

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                         924K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002          SUBROUTINE TRAFIC
               C
               C     THIS ROUTINE COMPUTES THE FOLLOWING
               C        1. THE ADJUSTED AVERAGE EMPTY WEIGHT OF VEHICLES WEIGHED EMPTY
               C        2. ADJUSTED GROSS WEIGHT AND TOTAL PAYLOAD CARRIED - PRESENT
               C           AND PROPOSED REGULATIONS
               C        3. DISTRIBUTION OF AXLE WEIGHTS - PRESENT AND PROPOSED REGS.
               C        4. AXLE WEIGHT DISTRIBUTIONS BY VEHICLE CLASSIFICATION -
               C           PROPOSED REGULATIONS
               C
               C     THE INPUTS ARE
               C       1. NAXLES(10,4) - THE NUMBER OF SINGLE, TANDEM, TRIPLE AND
               C                         STEERING AXLES FOR EACH TRUCK TYPE
               C       2. NTTY - NUMBER OF TRUCK TYPES TO BE CONSIDERED (EXISTING)
               C       3. NATT - NUMBER OF ADDED TRUCK TYPES (FUTURE DESIGN)
               C       4. NEWTRK - SHIFTING INDICATOR
               C            0 - SHIFTING PROCEDURE TO BE DONE
               C            1 - SHIFTING PROCEDURE NOT TO BE DONE (ALREADY DONE)
               C       5. SA(30,11) - NUMBER OF SINGLE AXLES WEIGHED BY INTERVAL AND
               C                      TRUCK TYPE
               C       6. TA(30,11) - NUMBER OF TANDEM AXLES WEIGHED BY INTERVAL AND
               C                      TRUCK TYPE
               C       7. TR(50,11) - NUMBER OF TRIPLE AXLES WEIGHED BY INTERVAL AND
               C                      TRUCK TYPE
               C       8. ST(30,11) - NUMBER OF STEERING AXLES WEIGHED BY INTERVAL AND
               C                      TRUCK TYPE
               C       9. VE(30,11) - NUMBER OF VEHICLES WEIGHED EMPTY BY INTERVAL AND
               C                      TRUCK TYPE
               C      10. VG(75,11) - NUMBER OF VEHICLES WEIGHED GROSS BY INTERVAL AND
               C                      TRUCK TYPE
               C      11. NLDI(6) - NUMBER OF INTERVALS INPUT FOR EACH OF THE ABOVE SIX
               C                    ARRAYS, WHERE,
               C                        1 = SA   2 = TA   3 = TR   4 = VG   5 = VE   6 = ST
               C      12. EMPTY(10) - PERCENT INCREASE IN AVERAGE EMPTY WEIGHT FOR EACH
               C                      TRUCK TYPE
               C      13. PGVWL - PRESENT GROSS VEHICLE WEIGHT LIMIT
               C      14. PSAL -- PRESENT SINGLE AXLE WEIGHT LIMIT
               C      15. PTAL -- PRESENT TANDEM AXLE WEIGHT LIMIT
               C      16. PTRAL - PRESENT TRIPLE AXLE WEIGHT LIMIT
               C      17. PSTAW(10) - PRESENT STEERING AXLE WEIGHT LIMIT BY TRUCK TYPE
               C      18-22.
               C          FGVWL, FSAL, FTAL, FTRAL, FSTAW(10) - SAME AS 13 THROUGH 17
               C                EXCEPT THAT THESE ARE VALUES UNDER PROPOSED REGULATIONS
               C      23. SIZE - STANDARD INTERVAL SIZE (2-KIPS)
               C      24. AVRG - AVERAGE VARIABLE (AVRG = 100. GIVES AVERAGE VALUES
               C                                  PER 100 TRUCKS)
               C      25. NAPOV - NUMBER OF SELECTED CUMULATIVE PERCENTAGES FOR THE
               C                  DISTRIBUTION OF AXLE WEIGHTS - PROPOSED REGS. SECTION
               C      26. PAPOV - PERCENTAGE INCREMENT CORRESPONDING TO NAPOV ABOVE
               C
   ISN 0003          COMMON /TRFFIC/ ELVWI(75), APVWE(75), APVWG(75), SAAPV(75),
                   1                 TAAPV(75), TRAPV(75), STAPV(75), NGVW
                                                ), MTYPE(4), NLAY, IP, IF, IR, IC
```

```
ISN 0005            COMMON /TRTYP/ TTYP(2,10), PTTYP(10,20,2), PCTTR(20,2), PERCT(4),
                  1               NAXLES(10,4),NT(4), NTTY, NATT, NTT, NEWTRK
ISN 0006            COMMON /NMBR/ SA(30,11), TA(30,11), TR(50,11), VE(30,11),
                  1              VG(75,11), NLDI(6), EMPTY(10), ST(30,11)
ISN 0007            COMMON /LDS/ PGVWL, PSAL, PTAL, PTRAL, FGVWL, FSAL, FTAL, FTRAL,
                  1             PSTAW(10), FSTAW(10)
ISN 0008            COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
ISN 0009            COMMON /TRINDX/ ITT
ISN 0010            COMMON /IO/ LI, LO, LD
ISN 0011            COMMON /OUTPTS/ TD4(10,6,2)
ISN 0012            COMMON EVWI(75), EVWMP(75), ELVWMP(75), GLVWNI(75), VWE(75),
                  2           PVWE(75), TWFAV(75), TPFAV(75), TVWE(75),
                  3           APPV(75), PPV(75), FACT(75), SAI(75), TAI(75), TRI(75),
                  4           SAA(75), TAA(75), TRA(75), SLA(75), TLA(75),
                  5           TRLA(75), APSA(75), APTA(75), APTR(75), APOV(75),
                  6           GWA(75), GWAF(75), SLAR(75), TLAR(75), TRLAR(75),
                  7           SANOV(75), TANOV(75), TRNOV(75), PSA(75), PTA(75),
                  8           PTR(75), SLAT(75), TLAT(75), TRLAT(75), STA(75),
                  9           PST(75), STLA(75), STLAR(75), STLAT(75), APST(75),
                  A           STI(75), STNOV(75), NLDISV(6)
ISN 0013            IF (NEWTRK .EQ. 1) GO TO 9999
ISN 0015            DO 6 K=1,1
ISN 0016            DO 4 J=1,6
ISN 0017            DO 2 I=1,10
ISN 0018            TD4(I,J,K) = 0.0
ISN 0019          2 CONTINUE
ISN 0020          4 CONTINUE
ISN 0021          6 CONTINUE
ISN 0022            DO 7 I=1,6
ISN 0023            NLDISV(I) = NLDI(I)
ISN 0024          7 CONTINUE
ISN 0025            DO 160 IT=1,NTT
ISN 0026            PERC =PERCT(IT)
ISN 0027            ITT = IT
ISN 0028            VTN = 0.
ISN 0029            NSA = 0
ISN 0030            NTA = 0
ISN 0031            NTR = 0
ISN 0032            NNA = 0
ISN 0033            NNT = 0
ISN 0034            NNR = 0
ISN 0035            APV = 0.
ISN 0036            PAPV = 0.
ISN 0037            DO 8 I=1,75
ISN 0038            PSA(I) = 0.
ISN 0039            PTA(I) = 0.
ISN 0040            PTR(I) = 0.
ISN 0041            PST(I) = 0.
ISN 0042            SAI(I) = 0.
ISN 0043            TAI(I) = 0.
ISN 0044            TRI(I) = 0.
ISN 0045            STI(I) = 0.
ISN 0046            SANOV(I) = 0.
ISN 0047            TANOV(I) = 0.
ISN 0048            TRNOV(I) = 0.
ISN 0049            STNOV(I) = 0.
ISN 0050            ELVWI(I) = 0.
ISN 0051            APVWE(I) = 0.
```

```
ISN 0052            APVWG(I) = 0.
ISN 0053            SAAPV(I) = 0.
ISN 0054            TAAPV(I) = 0.
ISN 0055            TRAPV(I) = 0.
ISN 0056            STAPV(I) = 0.
ISN 0057            FACT(I) = 0.
ISN 0058            GLVWNI(I) = 0.
ISN 0059            APSA(I) = 0.
ISN 0060            APTA(I) = 0.
ISN 0061            APTR(I) = 0.
ISN 0062            APST(I) = 0.
ISN 0063          8 CONTINUE
ISN 0064            DO 9 I=1,6
ISN 0065            NLDI(I) = NLDISV(I)
ISN 0066          9 CONTINUE
              C
ISN 0067         35 CONTINUE
ISN 0068         50 CONTINUE
ISN 0069            IF (NAXLES(IT,1) .EQ. 0) GO TO 64
              C
              C     SINGLE AXLES
              C
ISN 0071            NLDS = NLDI(1)
ISN 0072            CALL COUNT (SA(1,IT), NLDS)
ISN 0073            CALL INTVL (SA, SAI, NLDS, NSA, 1, 30, SAA, IT)
ISN 0074            CALL PCTAGE (SAA, NSA, PSA)
ISN 0075            CALL ACMLTE (PSA, NSA, APSA)
ISN 0076            NNA = NSA
ISN 0077         64 IF (NAXLES(IT,2) .EQ. 0) GO TO 66
              C
              C     TANDEM AXLES
              C
ISN 0079            NLDS = NLDI(2)
ISN 0080            CALL COUNT (TA(1,IT), NLDS)
ISN 0081            CALL INTVL (TA, TAI, NLDS, NTA, 2, 30, TAA, IT)
ISN 0082            CALL PCTAGE (TAA, NTA, PTA)
ISN 0083            CALL ACMLTE (PTA, NTA, APTA)
ISN 0084            NNT = NTA
ISN 0085         66 IF (NAXLES(IT,3) .EQ. 0) GO TO 68
              C
              C     TRIPLE AXLES
              C
ISN 0087            NLDS = NLDI(3)
ISN 0088            CALL COUNT (TR(1,IT), NLDS)
ISN 0089            CALL INTVL (TR, TRI, NLDS, NTR, 3, 50, TRA, IT)
ISN 0090            CALL PCTAGE (TRA, NTR, PTR)
ISN 0091            CALL ACMLTE (PTR, NTR, APTR)
ISN 0092            NNR = NTR
ISN 0093         68 IF ((NAXLES(IT,4) .EQ. 0) .OR. (IP .NE. IF)) GO TO 69
              C
              C     STEERING AXLES
              C
ISN 0095            NLDS = NLDI(6)
ISN 0096            CALL COUNT (ST(1,IT), NLDS)
ISN 0097            CALL INTVL (ST, STI, NLDS, NST, 6, 30, STA, IT)
ISN 0098            CALL PCTAGE (STA, NST, PST)
ISN 0099            CALL ACMLTE (PST, NST, APST)
ISN 0100            NNS = NST
```

```
ISN 0101        69 IF (IT .GT. NTTY) GO TO 146
ISN 0103           NGVW = NJ
              C
              C    *** DISTRIBUTION OF SINGLE/TANDEM/TRIDEM AXLE WEIGHTS - PROPOSED LIMITS **
              C
              C    SET UP THE TABLE OF SELECTED CUMULATIVE PERCENTAGES DEFINING THE
              C    GROSS WEIGHT AND AXLE WEIGHT CURVES
              C
ISN 0104           P = 0.0
ISN 0105           DO 70 I=1,NAPOV
ISN 0106           APOV(I) = P
ISN 0107           P = P + PAPOV
ISN 0108        70 CONTINUE
              C
ISN 0109           GO TO 150
ISN 0110       146 DO 147 I=1,NSA
ISN 0111           SAAPV(I) = APSA(I)
ISN 0112           SANOV(I) = PSA(I)
ISN 0113           PSA(I) = 0.
ISN 0114       147 CONTINUE
ISN 0115           NNA = NSA
ISN 0116           DO 148 I=1,NTA
ISN 0117           TAAPV(I) = APTA(I)
ISN 0118           TANOV(I) = PTA(I)
ISN 0119           PTA(I) = 0.
ISN 0120       148 CONTINUE
ISN 0121           NNT = NTA
ISN 0122           DO 149 I=1,NTR
ISN 0123           TRAPV(I) = APTR(I)
ISN 0124           TRNOV(I) = PTR(I)
ISN 0125           PTR(I) = 0.
ISN 0126       149 CONTINUE
ISN 0127           NNR = NTR
ISN 0128           DO 151 I=1,NST
ISN 0129           STAPV(I) = APST(I)
ISN 0130           STNOV(I) = PST(I)
ISN 0131           PST(I) = 0.
ISN 0132       151 CONTINUE
ISN 0133           NNS = NST
ISN 0134           DO 152 I=1,NJ
ISN 0135           APVWG(I) = APVWE(I)
ISN 0136       152 CONTINUE
ISN 0137           NGVW = MAXO(NSA,NTA,NTR,NST,NJ)
              C
              C    WRITE TO DISK FOR RECALL IN EQUIVALENT LOAD APPLICATIONS ROUTINE
              C
ISN 0138       150 CALL OUTPUT (3)
ISN 0139           WRITE (LD) NSA, NTA, NTR, NST, NNA, NNT, NNR, NNS,
                 1            (PSA(I),I=1,NNA), (PTA(I),I=1,NNT), (PTR(I),I=1,NNR),
                 2            (PST(I),I=1,NNS), (SANOV(I),I=1,NSA),
                 3            (TANOV(I),I=1,NTA), (TRNOV(I),I=1,NTR),
                 4            (STNOV(I),I=1,NST), (SAI(I),I=1,NSA), (TAI(I),I=1,NTA),
                 5            (TRI(I),I=1,NTR), (STI(I),I=1,NST), VTN, APV, PAPV
ISN 0140       160 CONTINUE
ISN 0141      9999 RETURN
ISN 0142           END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
```

```
*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      141, PROGRAM SIZE =      3510, SUBPROGRAM NAME =TRAFIC

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          900K BYTES OF CORE NOT USED
```

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

```
  ISN 0002            SUBROUTINE EAL18 (STRNUM, SLBTHK, TPSI, IPVT)
                C
                C     THIS ROUTINE CALCULATES THE EQUIVALENT 18-KIP AXLE LOAD
                C     APPLICATIONS FOR EACH VEHICLE USING INFORMATION WRITTEN ON DISK BY
                C     SUBROUTINE TRAFIC
                C
                C     THE INPUTS ARE
                C       1. STRNUM - STRUCTURAL NUMBER FOR A FLEXIBLE PAVEMENT
                C       2. SLBTHK - SLAB THICKNESS FOR A RIGID PAVEMENT
                C       3. TPSI --- TERMINAL PSI
                C       4. IPVT --- PAVEMENT TYPE SWITCH
                C       5. APPT(10,2) - AVERAGE PAYLOAD PER VEHICLE, PRESENT + PROPOSED
                C       6. COFVCT(6) - A VECTOR WITH ZERO-ONE ELEMENTS THAT DEFINE THE
                C                      PRESENT TRAFFIC COMBINATION BEING CONSIDERED
                C
                C     THE OUTPUT IS
                C       EALPT(10,2) - 18-KIP EAL PER TRUCK - PRESENT AND PROPOSED REGS.
                C
  ISN 0003            DIMENSION PSA(75), PTA(75), PTR(75), SANOV(75), TANOV(75),
                    1           TRNOV(75), EFSA(75), EFTA(75), EFTR(75), SAN18(75),
                    2           TAN18(75), TRN18(75), SPN18(75), DPN18(75), TPN18(75),
                    3           SAI(75), TAI(75), TRI(75), SAM(75), TAM(75), TRM(75),
                    4           PST(75), STNOV(75), EFST(75), STN18(75), STPN18(75),
                    5           STI(75), STM(75)
  ISN 0004            COMMON /EALPAY/ EALPT(10,2), APPT(10,2), EALFCT(20), IEQTRP
  ISN 0005            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
  ISN 0006            COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
  ISN 0007            COMMON /TRTYP/ TTYP(2,10), PTTYP(10,20,2), PCTTR(20,2),PERCT(4),
                    1                NAXLES(10,4),NT(4), NTTY, NATT, NTT, NEWTRK
  ISN 0008            COMMON /IO/ LI, LO, LD
  ISN 0009            COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
  ISN 0010            COMMON /COMBI/ ICOMB, NVC, COFVCT(6)
  ISN 0011            DATA PSI1, PK1, PSI2, PK2 /4.2, 2.7, 4.5, 3.0/
  ISN 0012            REWIND 1
  ISN 0013            NTT = NTTY + NATT
  ISN 0014            DO 1000 IT=1,NTT
                C
                C     READ FROM DISK THE INFORMATION STORED BY SUBROUTINE TRAFIC
                C
  ISN 0015            READ (LD) NSA, NTA, NTR, NST, NNA, NNT, NNR, NNS,
                    1           (PSA(I),I=1,NNA), (PTA(I),I=1,NNT), (PTR(I),I=1,NNR),
                    2           (PST(I),I=1,NNS), (SANOV(I),I=1,NSA),
                    3           (TANOV(I),I=1,NTA), (TRNOV(I),I=1,NTR),
                    4           (STNOV(I),I=1,NST), (SAI(I),I=1,NSA), (TAI(I),I=1,NTA),
                    5           (TRI(I),I=1,NTR), (STI(I),I=1,NST), VTN, APV, PAPV
  ISN 0016            APPT(IT,1) = APV
                C
                C     COMPUTE THE 18-KIP EAL FOR EACH AXLE TYPE
                C
  ISN 0017            TSN18 = 0.
  ISN 0018            IF (NAXLES(IT,1) .EQ. 0) GO TO 50
                C
```

90

```
                 C
   ISN 0020            CALL MIDPNT (SAI, NSA, SAM)
   ISN 0021            IF (IPVT .EQ. 2) GO TO 10
   ISN 0023            GT = ALOG10((PSI1 - TPSI) / PK1)
   ISN 0024            CALL FLEXEQ (SAM, NSA, 1.0, STRNUM, GT, EFSA)
   ISN 0025            GO TO 20
   ISN 0026         10 GT = ALOG10((PSI2 - TPSI) / PK2)
   ISN 0027            CALL RIGEQ (SAM, NSA, 1.0, SLBTHK, GT, EFSA)
   ISN 0028         20 CALL MULT (EFSA, PSA, NNA, SAN18)
   ISN 0029            CALL SUM (SAN18, NNA, TSN18)
   ISN 0030         50 CONTINUE
   ISN 0031            TDN18 = 0.
   ISN 0032            TYN18 = 0.
   ISN 0033            IF (NAXLES(IT,2) .EQ. 0) GO TO 100
                 C
                 C     TANDEM AXLES
                 C
   ISN 0035            CALL MIDPNT (TAI, NTA, TAM)
   ISN 0036            IF (IPVT .EQ. 2) GO TO 12
   ISN 0038            GT = ALOG10((PSI1 - TPSI) / PK1)
   ISN 0039            CALL FLEXEQ (TAM, NTA, 2.0, STRNUM, GT, EFTA)
   ISN 0040            GO TO 22
   ISN 0041         12 GT = ALOG10((PSI2 - TPSI) / PK2)
   ISN 0042            CALL RIGEQ (TAM, NTA, 2.0, SLBTHK, GT, EFTA)
   ISN 0043         22 CALL MULT (EFTA, PTA, NNT, TAN18)
   ISN 0044            CALL SUM (TAN18, NNT, TDN18)
   ISN 0045        100 CONTINUE
   ISN 0046            TTN18 = 0.
   ISN 0047            TZN18 = 0.
   ISN 0048            IF (NAXLES(IT,3) .EQ. 0) GO TO 150
                 C
                 C     TRIPLE AXLES
                 C
   ISN 0050            CALL MIDPNT (TRI, NTR, TRM)
   ISN 0051            IF (IPVT .EQ. 2) GO TO 14
   ISN 0053            GT = ALOG10((PSI1 - TPSI) / PK1)
   ISN 0054            CALL FLEXEQ (TRM, NTR, 3.0, STRNUM, GT, EFTR)
   ISN 0055            GO TO 24
   ISN 0056         14 GT = ALOG10((PSI2 - TPSI) / PK2)
   ISN 0057            CALL RIGEQ (TRM, NTR, 3.0, SLBTHK, GT, EFTR)
   ISN 0058         24 CALL MULT (EFTR, PTR, NNR, TRN18)
   ISN 0059            CALL SUM (TRN18, NNR, TTN18)
   ISN 0060        150 CONTINUE
   ISN 0061            TSTN18 = 0.
   ISN 0062            TWN18 = 0.
   ISN 0063            IF ((NAXLES(IT,4) .EQ. 0) .OR. (IP .NE. IF)) GO TO 200
                 C
                 C     STEERING AXLES
                 C
   ISN 0065            CALL MIDPNT (STI, NST, STM)
   ISN 0066            IA = -1.5 + 2. * TPSI
   ISN 0067            IF(IP. EQ. IF) IA = -1*PF + 2* TPSI
   ISN 0069            IA = MAX0(1, MIN0(4,IA))
   ISN 0070            CALL STEREQ (IA, EFST, NST, STM)
   ISN 0071            CALL MULT (EFST, PST, NNS, STN18)
   ISN 0072            CALL SUM (STN18, NNS, TSTN18)
   ISN 0073        200 EALPT(IT,1) = (TSN18*FLOAT(NAXLES(IT,1)) + TDN18 *
                     1                FLOAT(NAXLES(IT,2)) + TTN18*FLOAT(NAXLES(IT,3)) +
```

```
                     2              TSTN18*FLOAT(NAXLES(IT,4))) * 0.01
  ISN 0074             EALPT(IT,1) = EALPT(IT,1)*COFVCT(IT)
  ISN 0075        1000 CONTINUE
  ISN 0076             RETURN
  ISN 0077             END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       76, PROGRAM SIZE =      9882, SUBPROGRAM NAME = EAL18

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          908K BYTES OF CORE NOT USED
```

```
 LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 84.262/15.03.30       PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE RIGEQ (XL, NL, ST, D, GT, EQ)
   ISN 0003            DIMENSION XL(1), EQ(1)
   ISN 0004            D1 = D + 1.0
   ISN 0005            D1P = D1 ** 8.46
   ISN 0006            C = 3.28 * ALOG10(ST)
   ISN 0007            GTB18 = GT / (1.0 + 1.620E+7 / D1P)
   ISN 0008            STP = ST ** 3.52
   ISN 0009            CON = 5.908 + C - GTB18
   ISN 0010            DO 10 L=1,NL
   ISN 0011            B2 = 3.63 * (XL(L) + ST) ** 5.20
   ISN 0012            BX = 1.0 + B2 / (D1P * STP)
   ISN 0013            E = CON - 4.62 * ALOG10(XL(L) + ST) + GT / BX
   ISN 0014         10 EQ(L) = 10.0 ** (-E)
   ISN 0015            RETURN
   ISN 0016            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =        15, PROGRAM SIZE =        836, SUBPROGRAM NAME = RIGEQ

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                         928K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                        OS/360  FORTRAN H EXTENDED             DATE 84.262/15.03.32         PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE FLEXEQ (XL, NL, ST, SN, GT, EQ)
   ISN 0003            DIMENSION XL(1), EQ(1)
   ISN 0004            SNP = (SN + 1.0) ** 5.19
   ISN 0005            GTB18 = GT / (0.40 + 1094.0 / SNP)
   ISN 0006            B1 = SNP * ST ** 3.23
   ISN 0007            CON = 6.125 + 4.33 * ALOG10(ST) - GTB18
   ISN 0008            DO 20 L=1,NL
   ISN 0009            B2 = 4.79 * ALOG10(XL(L) + ST)
   ISN 0010            BX = 0.40 + 0.081 * (XL(L) + ST) ** 3.23 / B1
   ISN 0011            E = CON - B2 + GT / BX
   ISN 0012         20 EQ(L) = 10.0 ** (-E)
   ISN 0013            RETURN
   ISN 0014            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       13, PROGRAM SIZE =      810, SUBPROGRAM NAME =FLEXEQ

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002             SUBROUTINE STEREQ (IEQ, SEQ, NEQ, EQM)
                  C
                  C     THIS ROUTINE COMPUTES STEERING AXLE EQUIVALENCY FACTORS
                  C
                  C     THE INPUTS ARE
                  C       1. EQM - ARRAY OF INTERVAL MIDPOINTS
                  C       2. NEQ - NUMBER OF MIDPOINTS IN EQM
                  C       3. IEQ - INDICATES WHICH COLUMN OF THE EQUIVALENCY FACTOR TABLE
                  C                (BY PSI) IS TO BE USED
                  C
                  C     THE OUTPUT IS
                  C       SEQ - ARRAY OF STEERING AXLE EQUIVALENCIES
                  C
   ISN 0003             DIMENSION SEQ(1), EQM(1)
   ISN 0004             COMMON /STEER/ EQFACT(15,5), PTST(4)
                  C
                  C     EQFACT(J,1) CONTAINS THE LOAD VALUES (J).
                  C     EQFACT(J,K) CONTAINS THE EQUIVALENCY FOR LOAD J, TERM PSI PTST(K-1)
                  C
   ISN 0005             DO 30 I=1,NEQ
   ISN 0006             IF (EQM(I) .LT. EQFACT(1,1)) GO TO 25
   ISN 0008             DO 10 J=2,15
   ISN 0009             IF (EQFACT(J,1) .GE. EQM(I)) GO TO 20
   ISN 0011          10 CONTINUE
   ISN 0012             SEQ(I) = EQFACT(15,IEQ)
   ISN 0013          20 K = J-1
   ISN 0014             SEQ(I) = EQFACT(K,IEQ) + (EQM(I) - EQFACT(K,1)) *
                       1         ((EQFACT(J,IEQ)-EQFACT(K,IEQ)) / (EQFACT(J,1)-EQFACT(K,1)
                       2                                                                ))
   ISN 0015             GO TO 30
   ISN 0016          25 SEQ(I) = EQFACT(1,IEQ) * EQM(I) / EQFACT(1,1)
   ISN 0017          30 CONTINUE
   ISN 0018             RETURN
   ISN 0019             END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      18, PROGRAM SIZE =      852, SUBPROGRAM NAME =STEREQ

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE INTVL (A1, A2, N, N1, IS, NN, A3, NM)
                 C
                 C     THIS ROUTINE CONVERTS THE END-OF-INTERVAL KIP TABLES TO EVENLY
                 C     DISTRIBUTED INTERVALS BASED ON THE VARIABLE *SIZE*.
                 C
                 C     THE INPUTS ARE
                 C       1. A1 - ARRAY OF END-OF-INTERVAL KIP VALUES
                 C       2. N -- NUMBER OF VALUES IN A1
                 C       3. IS - ARRAY IDENTIFIER WHERE,
                 C                IS=1 - SINGLE AXLE ARRAY
                 C                IS=2 - TANDEM AXLE ARRAY
                 C                IS=3 - TRIPLE AXLE ARRAY
                 C                IS=4 - GROSS WEIGHT ARRAY
                 C                IS=5 - EMPTY WEIGHT ARRAY
                 C                IS=6 - STEERING AXLE ARRAY
                 C       4. NN - MAXIMUM ALLOWABLE ROW LENGTH OF A1
                 C       5. NM - INDICATES WHICH TRUCK TYPE IS CURRENTLY BEING CONSIDERED
                 C
                 C     THE OUTPUTS ARE
                 C       1. N1 - THE NEW LENGTH OF THE END-OF-INTERVAL KIP TABLE
                 C       2. A2 - THE NEW END-OF-INTERVAL KIP TABLE
                 C       3. A3 - THE NUMBER OF TRUCKS (OR AXLES) WEIGHED IN EACH INTERVAL
                 C
   ISN 0003            COMMON /INTVLS/ STARTS(6)
   ISN 0004            COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
   ISN 0005            DIMENSION A1(NN,11), A2(75), A3(75), ACC(75)
   ISN 0006            XMLOAD = A1(N,11)
   ISN 0007            A2(1) = SIZE
                 C
                 C     SET *S* TO  THE LARGEST EVEN NUMBER GREATER THAN OR EQUAL TO THE
                 C     FIRST END-OF-INTERVAL KIP VALUE
                 C
   ISN 0008            S = 0.
   ISN 0009            K = 0
   ISN 0010          5 IF (S .GE. STARTS(IS)) GO TO 7
   ISN 0012            S = S + SIZE
   ISN 0013            K = K+1
   ISN 0014            GO TO 5
                 C
                 C     SET UP THE EVENLY DISTRIBUTED END-OF-INTERVAL KIP TABLE AND ZERO
                 C     ALL INTERVALS AT BEGINNING OF TABLE IN WHICH NO TRUCKS/AXLES WERE
                 C     WEIGHED
                 C
   ISN 0015          7 I = 1
   ISN 0016            J = 1
   ISN 0017         10 IF (A2(I) .GE. XMLOAD) GO TO 20
   ISN 0019            I = I+1
   ISN 0020            A2(I) = A2(J) + SIZE
   ISN 0021            J = J+1
   ISN 0022            GO TO 10
   ISN 0023         20 N1 = I
   ISN 0024            DO 30 I=1,K
   ISN 0025            A3(I) = 0.
```

```
  ISN 0026       30 CONTINUE
  ISN 0027          I = K+1
  ISN 0028          CALL ACMLTE (A1(1,NM), N, ACC)
  ISN 0029          CALL ITRP (A1(1,11), ACC, A2, I, N1, N, A3, 1)
  ISN 0030          RETURN
  ISN 0031          END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      30, PROGRAM SIZE =     1236, SUBPROGRAM NAME = INTVL

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002             SUBROUTINE ITRP (V1, V2, V3, LIS, NV, NL, V4, IV)
                  C
                  C     THIS ROUTINE PERFORMS LINEAR INTERPOLATION
                  C
                  C     THE INPUTS ARE
                  C       1. V1 -- ARRAY OF X1 VALUES
                  C       2. V2 -- ARRAY OF F2(X) VALUES
                  C       3. V3 -- ARRAY OF X-VALUES
                  C       4. LIS - FIRST NON-ZERO VALUE IN V3
                  C       5. NV -- LAST VALUE IN V3
                  C       6. NL -- LAST VALUE IN V1
                  C       7. IV -- INTERPOLATION INDICATOR WHERE,
                  C                  IV=1 - VALUES ARE CUMULATIVE
                  C                     0 - VALUES ARE NOT CUMULATIVE
                  C
                  C     THE OUTPUT IS
                  C       V4 -- ARRAY OF INTERPOLATED RESULTS
                  C
   ISN 0003             DIMENSION V1(75), V2(75), V3(75), V4(75)
   ISN 0004             IF (LIS .EQ. 1) V4(1) = 0.0
   ISN 0006             J = 1
   ISN 0007             DO 50 I=LIS,NV
   ISN 0008             DO 10 K=J,NL
                  C
                  C     FIND THE SMALLEST X1 GREATER THAN OR EQUAL TO X
                  C
   ISN 0009             IF (V1(K) .GE. V3(I)) GO TO 20
   ISN 0011          10 CONTINUE
   ISN 0012             K = NL+1
   ISN 0013             V2SV = V2(K)
   ISN 0014             V1SV = V1(K)
   ISN 0015             V2(K) = V2(NL)
   ISN 0016             V1(K) = V3(I)
   ISN 0017             L = NL
   ISN 0018             GO TO 25
                  C
                  C     SET X1 AND F1 VALUES APPROPRIATELY, THEN INTERPOLATE
                  C
   ISN 0019          20 J = K
   ISN 0020             L = K-1
   ISN 0021             IF (L .EQ. 0) GO TO 30
   ISN 0023          25 F1 = V2(L)
   ISN 0024             X1 = V1(L)
   ISN 0025             GO TO 40
   ISN 0026          30 X1 = 0.0
   ISN 0027             F1 = V4(1)
   ISN 0028          40 V4(I) = F1 + (V3(I)-X1) * ((V2(K)-F1) / (V1(K)-X1))
   ISN 0029             IF (K .LE. NL) GO TO 50
   ISN 0031             V2(K) = V2SV
   ISN 0032             V1(K) = V1SV
   ISN 0033          50 CONTINUE
                  C
```

```
                  C      INTERVAL
                  C
   ISN 0034              IF (IV .EQ. 0) GO TO 999
   ISN 0036              J = NV
   ISN 0037              DO 60 I=2,NV
   ISN 0038              V4(J) = V4(J) - V4(J-1)
   ISN 0039              J = J-1
   ISN 0040           60 CONTINUE
   ISN 0041          999 RETURN
   ISN 0042              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      41, PROGRAM SIZE =     1132, SUBPROGRAM NAME =  ITRP

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                    928K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                 OS/360  FORTRAN H EXTENDED          DATE 84.262/15.03.44        PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002             SUBROUTINE PCTAGE (P1, NP, P2)
                  C
                  C     THIS ROUTINE SUMS THE *NP* VALUES IN ARRAY P1 AND DETERMINES, FOR
                  C     EACH VALUE IN P1, ITS PERCENTAGE OF THE TOTAL
                  C
   ISN 0003             DIMENSION P1(75), P2(75)
   ISN 0004             TOT = 0.0
   ISN 0005             DO 10 I=1,NP
   ISN 0006             TOT = TOT + P1(I)
   ISN 0007          10 CONTINUE
   ISN 0008             DO 20 I=1,NP
   ISN 0009             P2(I) = P1(I) / TOT * 100.0
   ISN 0010          20 CONTINUE
   ISN 0011             RETURN
   ISN 0012             END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       11, PROGRAM SIZE =       440, SUBPROGRAM NAME =PCTAGE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                      928K BYTES OF CORE NOT USED
```

```
 LEVEL 2.3.0 (JUNE 78)                              OS/360  FORTRAN H EXTENDED            DATE 84.262/15.03.45        PAGE   1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002               SUBROUTINE COUNT (CA, ICA)
                C
                C         THIS ROUTINE DETERMINES WHICH OF THE *ICA* VALUES IN ARRAY CA IS
                C         THE LAST NON-ZERO VALUE
                C
   ISN 0003               DIMENSION CA(75)
   ISN 0004               DO 10 I=1,ICA
   ISN 0005               IF (CA(I) .GT. 0.0) J = I
   ISN 0007            10 CONTINUE
   ISN 0008               ICA = J
   ISN 0009               RETURN
   ISN 0010               END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       9, PROGRAM SIZE =      326, SUBPROGRAM NAME = COUNT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                              928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE ACMLTE (AIN, NA, AOUT)
                 C
                 C     THIS ROUTINE CONVERTS ARRAY AIN TO A CUMULATIVE ARRAY
                 C
   ISN 0003            DIMENSION AIN(75), AOUT(75)
   ISN 0004            AOUT(1) = AIN(1)
   ISN 0005            NB = NA-1
   ISN 0006            DO 10 I=1,NB
   ISN 0007            J = I+1
   ISN 0008            AOUT(J) = AOUT(I) + AIN(J)
   ISN 0009         10 CONTINUE
   ISN 0010            RETURN
   ISN 0011            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       10, PROGRAM SIZE =       410, SUBPROGRAM NAME =ACMLTE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                      928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

  ISN 0002            SUBROUTINE MIDPNT (X1, NM, X2)
                C
                C     THIS ROUTINE DETERMINES THE MIDPOINT OF EACH INTERVAL IN ARRAY X1,
                C     WHERE EACH VALUE IN X1 IS AN END-OF-INTERVAL KIP VALUE
                C
  ISN 0003            COMMON /CNSTS/ NAPOV, PAPOV, SIZE, AVRG
  ISN 0004            DIMENSION X1(75), X2(75)
  ISN 0005            I = 0
  ISN 0006            J = 1
  ISN 0007            ELI = X1(NM)
  ISN 0008            X2(1) = X1(1) - (SIZE/2.)
  ISN 0009         10 I = I+1
  ISN 0010            J = J+1
  ISN 0011            X2(J) = X2(I) + SIZE
  ISN 0012            IF (X1(J) .LT. ELI) GO TO 10
  ISN 0014            RETURN
  ISN 0015            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       14, PROGRAM SIZE =      448, SUBPROGRAM NAME =MIDPNT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                 928K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                    OS/360  FORTRAN H EXTENDED          DATE 84.262/15.03.50        PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

  ISN 0002              SUBROUTINE MULT (YA, YB, NU, YC)
                  C
                  C     THIS ROUTINE MULTIPLIES TWO VECTORS SUCH THAT YC(I) = YA(I)*YB(I)
                  C
  ISN 0003              DIMENSION YA(75), YB(75), YC(75)
  ISN 0004              DO 10 I=1,NU
  ISN 0005              YC(I) = YA(I) * YB(I)
  ISN 0006           10 CONTINUE
  ISN 0007              RETURN
  ISN 0008              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =        7, PROGRAM SIZE =       396, SUBPROGRAM NAME =  MULT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002              SUBROUTINE AVRGE (AV, NV, AN, AVG)
                   C
                   C     THIS ROUTINE COMPUTES THE AVERAGE OF THE VALUES IN ARRAY AV
                   C     OVER *AN*
                   C
   ISN 0003              DIMENSION AV(75)
   ISN 0004              AVG = 0.0
   ISN 0005              DO 10 I=1,NV
   ISN 0006              AVG = AV(I) + AVG
   ISN 0007           10 CONTINUE
   ISN 0008              AVG = AVG / AN
   ISN 0009              RETURN
   ISN 0010              END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       9, PROGRAM SIZE =      350, SUBPROGRAM NAME = AVRGE

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                      928K BYTES OF CORE NOT USED
```

```
LEVEL 2.3.0 (JUNE 78)                 OS/360  FORTRAN H EXTENDED          DATE 84.262/15.03.52        PAGE    1

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002             SUBROUTINE SUM (S1, NS, S2)
                  C
                  C     THIS ROUTINE COMPUTES THE SUM OF THE VALUES IN ARRAY S1
                  C
   ISN 0003             DIMENSION S1(75)
   ISN 0004             S2 = 0.0
   ISN 0005             DO 10 I=1,NS
   ISN 0006             S2 = S2 + S1(I)
   ISN 0007          10 CONTINUE
   ISN 0008             RETURN
   ISN 0009             END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       8, PROGRAM SIZE =      322, SUBPROGRAM NAME =   SUM

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                      928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002           SUBROUTINE ZERO (A,N)
   ISN 0003           DIMENSION A(N)
   ISN 0004           DO 10 I=1,N
   ISN 0005        10 A(I) = 0.
   ISN 0006           RETURN
   ISN 0007           END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =        6, PROGRAM SIZE =       280, SUBPROGRAM NAME =  ZERO

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                       928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE PSIT(P,PF,W,XKT)
   ISN 0003            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
   ISN 0004            DOUBLE PRECISION PWR
   ISN 0005            COMMON/HOR/A(10),B(10),C(10),DT(10),DF(10),S(10),T(10),TR(5),PI(5)
                      *,PT(5),AC(5),AA,SCT(5),XMNW18(10) ,XKTO
   ISN 0006            COMMON /EXTRA/ PTOVTK,TPE,PFO,XMNOTK,XMXOTK,NIS
   ISN 0007            W = TR(1)*0.1
   ISN 0008            P=PI(NPT)
   ISN 0009            GO TO (10,20,30,40,50),NPT
   ISN 0010         10 CONTINUE
   ISN 0011            XKT=89.15+.00367*T(1)**.99*DT(1)**(-2.83)*S(1)**2.1*DF(2)**.85
   ISN 0012            PF=3.663+1236.1*S(2)**(-.086)*C(6)**.3*C(3)**.12*C(4)**(-.21)
                      **C(2)**(-.22)*DT(1)**.25*C(1)**(-3.13)*C(5)**.31
   ISN 0013            GO TO 60
   ISN 0014         20 CONTINUE
   ISN 0015            XKT=92.83+.27*10.0**(-12.0)*S(2)**1.64*DF(1)**(-.46)*C(1)**7.97
                      **DT(1)**(-1.45)*C(5)**(-3.38)*TR(NPT)**(-.25)*T(1)**1.09
   ISN 0016            PF=3.667+117.44*S(2)**(-.08)*DF(2)**(-.034)*C(1)**(-1.67)*C(4)**
                      *(-.085)*DT(1)**.49*T(1)**(-.059)*C(5)**.25
   ISN 0017            GO TO 60
   ISN 0018         30 CONTINUE
   ISN 0019            XKT=91.51+.6837*DF(1)**.23*C(2)**.38*C(4)**(-.18)*TR(NPT)**(-.15)*
                      *T(1)**1.45
   ISN 0020            PF=2.367+15.598*S(3)**(-.018)*C(1)**(-.55)*C(3)**(-.24)*S(1)**
                      *(-.17)*T(1)**(-.085)*TR(NPT)**.03
   ISN 0021            GO TO 60
   ISN 0022         40 CONTINUE
   ISN 0023            XKT=81.84+5.052*DF(2)**(-.32)*DF(1)**1.4*C(2)**.89*T(1)**.25*
                      *S(1)**(-1.74)
   ISN 0024            PF=3.719+3.327*C(1)**(-.38)*C(4)**.033*S(3)**(-.09)*C(3)**
                      *(-.061)*S(4)**.071*S(2)**.16*S(1)**(-.017)*T(1)**(-.075)
   ISN 0025            GO TO 60
   ISN 0026         50 CONTINUE
   ISN 0027            XKT=-.0737+231.63*T(1)**1.26*S(4)**.3*TR(NPT)**(-.47)
   ISN 0028            PF=.00804+7.6131*DF(2)**(-.15)*T(1)**.021*C(5)**(-1.37)
   ISN 0029        60  CONTINUE
   ISN 0030            PFO=PF
   ISN 0031            RETURN
   ISN 0032            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       31, PROGRAM SIZE =     3838, SUBPROGRAM NAME =  PSIT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                      912K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002             SUBROUTINE RUSIAN (DPTEMP,DMDRU,NLAY,NPT,NRU,NLH)
                  C
                  C    E REPRESENTS THE ELASTICITY MODULI FOR THE REPRESENTATIVE
                  C     SECTIONS
                  C
                  C
   ISN 0003             IMPLICIT REAL*8 (A-H,O-Z)
   ISN 0004             REAL*8 MTERM, NTERM
   ISN 0005             DIMENSION X(5), EX(5) ,DP(5), R(6) ,E(6) ,WHAT(5) ,DPTEMP(5)
   ISN 0006             IF(NPT.NE.3.OR.NRU.NE.1) GO TO 112
   ISN 0008             E(1)=65000.0
   ISN 0009             E(2)=20000.0
   ISN 0010             E(3)=12000.0
   ISN 0011             E(4)=5000.0
   ISN 0012       112   IF(NPT.NE.3.OR.NRU.NE.2) GO TO 120
   ISN 0014             E(1)=65000.0
   ISN 0015             E(2)=20000.0
   ISN 0016             E(3)=12800.0
   ISN 0017             E(4)=5100.0
   ISN 0018       120   IF(NPT.NE.1.OR.NRU.NE.1.OR.NLH.NE.1)GO TO 30
   ISN 0020             E(1)=300000.0
   ISN 0021             E(2)=80000.0
   ISN 0022             E(3)=15000.0
   ISN 0023             E(4)=6000.0
   ISN 0024       30    IF(NPT.NE.1.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 40
   ISN 0026             E(1)=305000.
   ISN 0027             E(2)=100000.0
   ISN 0028             E(3)=16500.0
   ISN 0029             E(4)=6000.0
   ISN 0030       40    IF(NPT.NE.1.OR.NRU.NE.2.OR.NLH.NE.1) GO TO 50
   ISN 0032             E(1)=300000.
   ISN 0033             E(2)= 85000.0
   ISN 0034             E(3)=22000.0
   ISN 0035             E(4)=16400.0
   ISN 0036             E(5)=6000.
   ISN 0037       50    IF(NPT.NE.1.OR.NRU.NE.2.OR.NLH.NE.2) GO TO 60
   ISN 0039             E(1)=325000.
   ISN 0040             E(2)= 95000.0
   ISN 0041             E(3)=35000.0
   ISN 0042             E(4)=18500.0
   ISN 0043             E(5)=6000.
   ISN 0044       60    IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.1) GO TO 70
   ISN 0046             E(1)=325000.
   ISN 0047             E(2)=130000.0
   ISN 0048             E(3)=90000.0
   ISN 0049             E(4)=16800.0
   ISN 0050             E(5)=6000.0
   ISN 0051       70    IF(NPT.NE.4.OR.NRU.NE.1.OR.NLH.NE.2) GO TO 80
   ISN 0053             E(1)=325000.
   ISN 0054             E(2)=130000.0
   ISN 0055             E(3)=90000.0
```

```
 ISN 0058      80    IF(NPT.NE.4.OR.NRU.NE.2.OR.NLH.NE.1) GO TO 90
 ISN 0060            E(1)=325000.
 ISN 0061            E(2)=130000.0
 ISN 0062            E(3)=90000.0
 ISN 0063            E(4)=38000.0
 ISN 0064            E(5)=19000.0
 ISN 0065            E(6)=6000.
 ISN 0066      90    IF(NPT.NE.4.OR.NRU.NE.2.OR.NLH.NE.2) GO TO 100
 ISN 0068            E(1)=325000.
 ISN 0069            E(2)=150000.0
 ISN 0070            E(3)=115000.0
 ISN 0071            E(4)=42000.0
 ISN 0072            E(5)=22000.0
 ISN 0073            E(6)=6000.
 ISN 0074       100  CONTINUE
 ISN 0075            DO 915 K=1,5
 ISN 0076             RTEMP =10.**2.+(12*(K-1))**2.
 ISN 0077      915    R(K)=DSQRT(RTEMP)
               C
 ISN 0078            DO 916 K=1,NLAY
 ISN 0079      916   DP(K)=DPTEMP(K)
 ISN 0080            IPVMT = 3
 ISN 0081            NW = 5
 ISN 0082            LEQ = 0
 ISN 0083            NOL = NLAY
               C
 ISN 0084            DO 5 K = 1, NLAY
 ISN 0085          5 X(K) = E(K)/1000000.
               C
 ISN 0086            IF( DP(  NOL-1) .LE. 10.0 ) LEQ = 1
 ISN 0088            NC = NOL
 ISN 0089            IF( LEQ .EQ. 1 ) NC = NOL - 1
 ISN 0091            IF( LEQ .EQ. 1 ) X(NOL-1) = X(NOL)
               C
 ISN 0093         53 BTERM = 10.0 ** (-0.05071) * DP(1) ** 0.10148
 ISN 0094            NTERM = 10.0 ** (-0.50233) * DP(1) ** 0.087879
 ISN 0095            CTERM = 10.0 ** (-0.060039) * DP(1) ** 0.0095198
 ISN 0096            MTERM = 0.704 - 0.026 * DP(1)
 ISN 0097            HTERM = 10.0 ** 1.8631 * DP(1) ** (-0.0038499)
               C
 ISN 0098            TMB = 2.0 * MTERM * BTERM
               C
 ISN 0099            NL = NLAY
 ISN 0100            EI = X(NC)
 ISN 0101            N1 = NL - 1
 ISN 0102            SUM = 0.0
 ISN 0103            DO 10 I = 1, N1
 ISN 0104         10 SUM = SUM + DP(I)
 ISN 0105            HS = HTERM - SUM
 ISN 0106            DP(   NL) = HS
               C
 ISN 0107            NT = NC - 1
               C
 ISN 0108            DO 11 I = 1, NT
 ISN 0109         11 EX(I) = X(I)
 ISN 0110            EX(NL) = X(NC)
 ISN 0111            EXT = EX(NL) * 1000000.0
 ISN 0112            IF( LEQ .EQ. 0 ) GO TO 14
```

```
                  C
  ISN 0114              GO TO ( 12, 12, 13, 13, 12, 12 ), IPVMT
  ISN 0115              GO TO 14
                  C
  ISN 0116           12 EX(NL - 1) = EX(NL) * (1.0 + 7.18 * DLOG10( DP(    NC)) - 1.56 *
                       = ( DLOG10( EXT     )* DLOG10( DP(    NC)) ))
  ISN 0117              GO TO 14
                  C
  ISN 0118           13 EX(NL - 1) = EX(NL) * (1.0 + 10.52 * DLOG10( DP(    NC)) - 2.10 *
                       = ( DLOG10( EXT     ) * DLOG10( DP(    NC))  ))
                  C
  ISN 0119           14 CONTINUE
                  C
  ISN 0120              HPR = 0.0
  ISN 0121              DO 15 I = 1, NL
  ISN 0122              XNUM = EX(I)/EI
  ISN 0123              HPR = HPR + (   XNUM      ** NTERM) * DP(    I)
  ISN 0124           15 CONTINUE
                  C
                  C
  ISN 0125              PHIALF = TMB * ((TMB + 1.0)/(TMB - 1.0)) ** 0.5
  ISN 0126              ALPHA = PHIALF/HPR
  ISN 0127              TTM = 2.0 * MTERM
                  C
  ISN 0128              DO 20 I = 1, NW
  ISN 0129              ARG = ALPHA * R(I)
                  C
  ISN 0130              WHAT(    I) = 0.47746 * (CTERM/( EI *1000000.0)) * (1000.0/HPR) *
                       = (TTM + 1.0) * BESJO( ARG )
                  C
  ISN 0131           20 CONTINUE
  ISN 0132              DMDRU=WHAT(1)
  ISN 0133              RETURN
  ISN 0134              END
*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      133, PROGRAM SIZE =      4166, SUBPROGRAM NAME =RUSIAN

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                        908K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002          FUNCTION BESJO ( X )
   ISN 0003          IMPLICIT REAL*8 (A-H, O-Z)
               C     A FUNCTION TO CALCULATE BESSEL FUNCTION JO(X) USING POLYNOMIAL    BES   20
               C     APPROXIMATION - REFERENCE HANDBOOK OF MATH. FUNCTIONS, BUREAU OF  BES   30
               C     STANDARDS, PAGES 369-370                                          BES   40
   ISN 0004          ASSIGN 2 TO JOJ1                                                  BES   60
   ISN 0005        1 CONTINUE                                                          BES   70
   ISN 0006          X3 = X/3.0                                                        BES   80
   ISN 0007          IF( X.GT. 3.0) X3 = 3.0/ X                                        BES   90
   ISN 0009          X32= X3*X3                                                        BES  100
   ISN 0010          X33=X32*X3                                                        BES  110
   ISN 0011          X34=X32*X32                                                       BES  120
   ISN 0012          X35=X32*X33                                                       BES  130
   ISN 0013          X36=X33*X33                                                       BES  140
   ISN 0014          GO TO JOJ1,(2,10)                                                 BES  150
   ISN 0015        2 IF(DABS( X ) .LE. 3.3 ) GO TO 3
   ISN 0017          X1 =    X    - 0.7853982  -0.04166397*X3 - 0.3954E-04 * X32 +     BES  200
                    +   0.262573E-02*X33  -  0.54125E-03* X34  -  0.29333E-03 * X35  + BES  210
                    +   0.13558E-03 * X36                                              BES  220
   ISN 0018          BESJO=((.7978846 -.77E-6   * X3  -  0.552740E-02 * X32  -         BES  170
                    -   0.9512E-04 * X33  +  0.137237E-02 * X34  - 0.72805E-03 * X35  +BES  180
                    +   0.14476E-03 * X36 )  /DSQRT(X)  )  * DCOS(X1  )                BES  190
   ISN 0019          RETURN                                                            BES  230
   ISN 0020        3 BES JO= 1.0  -  2.2499997 * X32  +  1.2656208 * X34  -            BES  240
                    -   0.3163866 * X36  +  0.0444479*(X34*X34)-0.0039444 *(X35*X35) + BES  250
                    *   0.000210* (X36*X36)                                            BES  260
   ISN 0021          RETURN                                                            BES  270
   ISN 0022          ENTRY BES J1(X)                                                   BES  280
               C     BESSEL FUNCTION J1 WHERE X IS BETWEEN -3 AND + INFINITY.          BES  290
   ISN 0023          ASSIGN 10 TO JOJ1                                                 BES  320
   ISN 0024          GO TO 1                                                           BES  360
   ISN 0025       10 IF ( DABS (X) .LT. 3.0   )  GO  TO  30
   ISN 0027          X1 =           X - 2.3561945  + 0.1249961  * X3 + 0.565E-4 * X32 -
                    ¢ 0.637879E-02 * X33 + 0.74348E-03 * X34 + 0.79824E-03 * X35
                    ' - 0.29166E-03 * X36
   ISN 0028          BESJ1 = DCOS( X1  )  *
                    =                          ( 0.79788456 + 0.156E-05 * X3
                    1   + 0.01659667 *  X32  + 0.17105E-03 * X33  - 0.249511E-02 * X34 BES  340
                    2   + 0.113653E-02 * X35  - 0.20033E-03 * X36 )  / DSQRT(X)        BES  350
   ISN 0029          RETURN                                                            BES  400
   ISN 0030       30 BES J1 =  X *  ( 0.5  - 0.5624999  * X32  + 0.2109357  * X34  -   BES  410
                    1   0.03954289 * X36  + 0.443319E-02 * (X34 * X34)  -  0.31761E-03 BES  420
                    2   * (X35*X35)  +  0.1109E-04* (X36*X36)  )                       BES  430
   ISN 0031          RETURN                                                            BES  440
   ISN 0032          END                                                               BES  450

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       31, PROGRAM SIZE =      1934, SUBPROGRAM NAME = BESJO

*STATISTICS*  NO  DIAGNOSTICS GENERATED
```

****** END OF COMPILATION ******   924K BYTES OF CORE NOT USED

REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

```
ISN 0002          SUBROUTINE SURVIV (IACR)
ISN 0003          COMMON /EXPVT/ NPT, THICK (4), MTYPE(4), NLAY, IP, IF, IR, IC
ISN 0004          COMMON /PSI/ PF, PICON, PTERM, PIOV, PTOV
ISN 0005          COMMON /BURKE/ XLAMB, GAMMA, TFBAP
ISN 0006          DIMENSION XLAM (5,3), GAM(5,3)
            C->THIS SUBROUTINE SETS SURVIVAL CURVE PARAMETER VALUES (XLAMB AND GAMMA)
            C  FOR FLEXIBLE PAVEMENTS
ISN 0007          IF (IP .NE. IF) GO TO 999
            C->FIND OUT TYPE OF FAILURE (PSI OR DISTRESS)
ISN 0009          IF (PF .GE. PTERM) GO TO 200
            C->SET XLAMB AND GAMMA FOR PSI SURVIVAL CURVES BY QUALITY STD. AND PAVT TYPE
ISN 0011          XLAM (1,1)=0.276
ISN 0012          GAM (1,1)=2.111
ISN 0013          XLAM (1,2)=0.417
ISN 0014          GAM (1,2)=1.549
ISN 0015          XLAM (1,3)=0.607
ISN 0016          GAM (1,3)=1.497
ISN 0017          XLAM (3,1)=0.423
ISN 0018          GAM (3,1)=1.363
ISN 0019          XLAM (3,2)=0.687
ISN 0020          GAM (3,2)=1.365
ISN 0021          XLAM (3,3)=0.787
ISN 0022          GAM (3,3)=1.012
ISN 0023          XLAM (4,1)=0.327
ISN 0024          GAM (4,1)=1.524
ISN 0025          XLAM (4,2)=0.555
ISN 0026          GAM (4,2)=1.163
ISN 0027          XLAM (4,3)=0.818
ISN 0028          GAM (4,3)=1.088
ISN 0029          GO TO 300
ISN 0030      200 CONTINUE
            C->SET XLAMB AND GAMMA FOR DISTRESS SURV. CURVES BY QUAL. STD. AND PAVT. TYPE
ISN 0031          XLAM (1,1)=0.004
ISN 0032          GAM (1,1)=2.681
ISN 0033          XLAM (1,2)=0.007
ISN 0034          GAM (1,2)=4.380
ISN 0035          XLAM (1,3)=0.009
ISN 0036          GAM (1,3)=5.000
ISN 0037          XLAM (3,1)=0.003
ISN 0038          GAM (3,1)=2.129
ISN 0039          XLAM (3,2)=0.006
ISN 0040          GAM (3,2)=3.065
ISN 0041          XLAM (3,3)=0.008
ISN 0042          GAM (3,3)=4.023
ISN 0043          XLAM (4,1)=0.004
ISN 0044          GAM (4,1)=2.443
ISN 0045          XLAM (4,2)=0.008
ISN 0046          GAM (4,2)=3.382
ISN 0047          XLAM (4,3)=0.011
ISN 0048          GAM (4,3)=4.119
ISN 0049      300 XLAMB=XLAM (NPT,IACR)
ISN 0050          GAMMA=GAM(NPT,IACR)
ISN 0051      999 RETURN
```

```
LEVEL 2.3.0 (JUNE 78)        SURVIV          OS/360  FORTRAN H EXTENDED            DATE 84.262/15.04.02       PAGE    2

  ISN 0052            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =      51, PROGRAM SIZE =      918, SUBPROGRAM NAME =SURVIV

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                   928K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

    ISN 0002            SUBROUTINE COSCAL
    ISN 0003            COMMON /PSI/ PF, PICON, PTERM, PIOV, PTOV
    ISN 0004            COMMON /TITLE/ TITLE(20,3), SECTTL(20)
    ISN 0005            COMMON /MECH/ XKT, NRU, NLH, ND, NDEL,IACR, IYR, JYR, CONSTR(20)
    ISN 0006            COMMON /LMP/ XLM(30),YLM(30),POTLM(20,2),OUTP(20,2),
                       1        TOTALM, PPF, TPF, PFNO, NASL, NSLR,TOVLM(30,2),XLM2(30)
    ISN 0007            COMMON /OVRLAY/ XHCIO,XHCIM,WLANE,WPSH,WGSH,PPVDSH,CAC,CGR,CSCOAT
    ISN 0008            COMMON /TIME/ ATP, OVLIF, NYAP,NYR,YR(40)
    ISN 0009            COMMON /MNTPAR/ UNTCST(4),USRMDL(31,3),WDTH,S,DISS,DCON,DIN,MFLG
    ISN 0010            COMMON /SWTCHS/ OVLIFE, PCTINT, PCTINF, TPFPC, PFNOPC, AGR, SPCJT,
                       1                 XMLI, CACI, CGRI, ICAC, ACDENS, ICGR, GRDENS,
                       2                 INTT, SAVMNT, IDST, NLD, MCODE(5), TFCDNS
    ISN 0011            COMMON /BURKE/ XLAMB, GAMMA, TFBAP
    ISN 0012            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
    ISN 0013            COMMON /COST/ COSTRH(20), COSTRM(20), COSTPM(20)
    ISN 0014            COMMON /IO/ LI, LO, LD
    ISN 0015            COMMON /COMBI/ ICOMB, NVC, COFVCT(6)
    ISN 0016            DIMENSION PR(50),PR1(30,20),PR2(19,20),ZLM(30),YMILES(20),
                       1          ZMILES(20), REHPLM(20)
    ISN 0017            EARMAR(T)=(1100*UNTCST(1)+1000*UNTCST(2)+5*UNTCST(3))/
                       1           (1+EXP(-(T-10)/1.16))
    ISN 0018            WRITE (LO,610) (IN,IN=1,6)
    ISN 0019        610 FORMAT (1H1,20X,32HVEHICLE COMBINATION (1=IN,0=OUT),/
                       1         ,30X,6I2)
    ISN 0020            WRITE (LO,611) (COFVCT(IN),IN=1,NVC)
    ISN 0021        611 FORMAT (30X,6F2.0)
    ISN 0022            CALL SURVIV (IACR)
    ISN 0023            DO 3 K=1,NASL
    ISN 0024               CALL DISTR (PR,K+NYAP-1,K)
    ISN 0025               DO 4 IAHE=1,NYAP
    ISN 0026                  PR1(K,IAHE)=PR(K+IAHE-1)
    ISN 0027          4    CONTINUE
    ISN 0028          3 CONTINUE
    ISN 0029            LL=NYAP-1
    ISN 0030            DO 5 L=1,LL
    ISN 0031               CALL DISTR (PR,NYAP-L,-L)
    ISN 0032               IAHEF=L+1
    ISN 0033               DO 6 IAHE=IAHEF,NYAP
    ISN 0034                  PR2(L,IAHE)=PR(IAHE-L)
    ISN 0035          6    CONTINUE
    ISN 0036          5 CONTINUE
    ISN 0037            CALL RHBLT(REHPLM)
    ISN 0038            XMAIN1=0.
    ISN 0039            FAILML=0.
    ISN 0040            DO 10 K=1,NASL
    ISN 0041               FAILML=FAILML+PR1(K,1)*YLM(K)
    ISN 0042               ZLM(K)=YLM(K)*(1-PR1(K,1))
    ISN 0043               XMAIN1=XMAIN1+EARMAR(K)*XLM(K)
    ISN 0044         10 CONTINUE
    ISN 0045            COSTRH(1)=(FAILML*REHPLM(1))*(1.+XHCIO)
    ISN 0046            YMILES(1)=CONSTR(1)+FAILML
    ISN 0047            ZMILES(1)=YMILES(1)
    ISN 0048            COSTRM(1)=XMAIN1*(1.+XHCIM)
```

```
ISN 0049            DO 20 I=2,NYAP
ISN 0050               XMAIN1=0.
ISN 0051               FAILML=0.
ISN 0052               DO 30 K=1,NASL
ISN 0053                  FAILML=FAILML+PR1(K,I)*YLM(K)
ISN 0054                  ZLM(K)=ZLM(K)-YLM(K)*PR1(K,I)
ISN 0055                  XMAIN1=XMAIN1+ZLM(K)*EARMAR(K+I-1)
ISN 0056         30    CONTINUE
ISN 0057               XMAIN2=0.
ISN 0058               IF (PF.GE.PTERM.AND.NDEL.GT.0.AND.I.GT.NDEL) GO TO 70
ISN 0060               LL=I-1
ISN 0061               DO 40 L=1,LL
ISN 0062                  FAILML=FAILML+YMILES(L)*PR2(L,I)
ISN 0063                  ZMILES(L)=ZMILES(L)-YMILES(L)*PR2(L,I)
ISN 0064                  XMAIN2=XMAIN2+ZMILES(L)*EARMAR(I-L)
ISN 0065         40    CONTINUE
ISN 0066               COSTRH(I)=(FAILML*REHPLM(I))*(1.+XHCIO)**I
ISN 0067               YMILES(I)=CONSTR(I)+FAILML
ISN 0068               ZMILES(I)=YMILES(I)
ISN 0069               COSTRM(I)=(XMAIN1+XMAIN2)*(1.+XHCIM)**I
ISN 0070               GO TO 20
ISN 0071         70    DELTA=YMILES(I-NDEL)-CONSTR(I-NDEL)
ISN 0072               YMILES(I-NDEL)=CONSTR(I-NDEL)
ISN 0073               LL=I-1
ISN 0074               DO 50 L=1,LL
ISN 0075                  FAILML=FAILML+YMILES(L)*PR2(L,I)
ISN 0076                  ZMILES(L)=ZMILES(L)-YMILES(L)*PR2(L,I)
ISN 0077                  XMAIN2=XMAIN2+ZMILES(L)*EARMAR(I-L)
ISN 0078         50    CONTINUE
ISN 0079               PROB=0.
ISN 0080               ML=NDEL-1
ISN 0081               DO 60 M=1,ML
ISN 0082                  PROB=PROB+PR2(I-NDEL,I-NDEL+M)
ISN 0083         60    CONTINUE
ISN 0084               YMILES(I)=CONSTR(I)+FAILML+DELTA*(1-PROB)
ISN 0085               ZMILES(I)=YMILES(I)
ISN 0086               COSTRH(I)=((FAILML+DELTA*(1-PROB))*REHPLM(I))*(1.+XHCIO)**I
ISN 0087               COSTRM(I)=(XMAIN1+XMAIN2)*(1.+XHCIM)**I
ISN 0088         20 CONTINUE
ISN 0089            DO 100 I=1,20
ISN 0090               COSTPM(I)=0.
ISN 0091        100 CONTINUE
ISN 0092            IF (IP.NE.IF.OR.JYR.EQ.0) GO TO 99
ISN 0094            TINTML=0.
ISN 0095            DO 110 K=1,NASL
ISN 0096               TINTML=TINTML+YLM(K)
ISN 0097        110 CONTINUE
ISN 0098            TCNSTR=0.
ISN 0099            DO 120 I=1,NYAP
ISN 0100               TCNSTR=TCNSTR+CONSTR(I)
ISN 0101        120 CONTINUE
ISN 0102            DO 130 I=1,NYAP
ISN 0103               COSTPM(I)=CSCOAT*(TINTML+(TCNSTR/2))/FLOAT(JYR)
ISN 0104               COSTPM(I)=COSTPM(I)*(1.+XHCIM)**I
ISN 0105        130 CONTINUE
ISN 0106            WRITE (LO,613) (SECTTL(J),J=1,20)
ISN 0107        613 FORMAT (//,20X,20A4,///)
ISN 0108            WRITE (LO,600)
```

```
  ISN 0109       600 FORMAT (10X,5HYEAR ,10X,10HROUT MAINT,10X,10H  REHAB   ,10X,
                   1         10HPREV MAINT,/,25X,10H   COST   ,10X,10H   COST   ,10X,
                   2         10H   COST   ,/)
  ISN 0110           WRITE (LO,601) (I,COSTRM(I),COSTRH(I),COSTPM(I),I=1,NYAP)
  ISN 0111       601 FORMAT ((10X,I5,3(10X,E10.3)))
  ISN 0112           WRITE (LO,620)
  ISN 0113       620 FORMAT (1H1)
  ISN 0114        99 RETURN
  ISN 0115           END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =     114, PROGRAM SIZE =     8812, SUBPROGRAM NAME =COSCAL

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                          900K BYTES OF CORE NOT USED
```

```
REQUESTED OPTIONS: NODUMP

OPTIONS IN EFFECT: NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)
                   SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

   ISN 0002            SUBROUTINE RHBLT(REHPLM)
   ISN 0003            COMMON /MISC/ IPOT, IARMS, OLDMNT, AGF
   ISN 0004            COMMON /BURKE/ XLAMB, GAMMA, TFBAP
   ISN 0005            COMMON /TIME/ ATP, OVLIF, NYAP, NYR,  YR(40)
   ISN 0006            COMMON /EXPVT/ NPT, THICK(4), MTYPE(4), NLAY, IP, IF, IR, IC
   ISN 0007            COMMON /PSI/ PF,PICON, PTERM, PIOV, PTOV
   ISN 0008            COMMON /IO/ LI, LO, LD
   ISN 0009            COMMON /STRUC/ SN,SS,R,D,AGG,XJ,XK,E
   ISN 0010            DIMENSION REHPLM(20),CUMEAL(50)
   ISN 0011            CALL ACCTFC(TFBAP,AGF,50,CUMEAL)
   ISN 0012            DO 10 I=1,NYAP
   ISN 0013               NOVLF=IFIX(OVLIF+.5)
   ISN 0014               IF (I+NOVLF.GT.50) GO TO 40
   ISN 0016               XNOV=CUMEAL(I+NOVLF)-CUMEAL(I)
   ISN 0017               IF (IP.NE.IF) GO TO 20
   ISN 0019               CALL OVTHKF(XNOV,THOV,YR)
   ISN 0020               GO TO 30
   ISN 0021         20    CALL GETD(ALOG10(XNOV),PIOV,PTOV,D,DOV)
   ISN 0022               DEX=D
   ISN 0023               CALL OVTHKR(DOV,D,THOV)
   ISN 0024         30    CALL OVCOST(THOV,REHPLM(I))
   ISN 0025         10 CONTINUE
   ISN 0026            GO TO 99
   ISN 0027         40 WRITE (LO,601)
   ISN 0028        601 FORMAT (10X,32HTRAFFIC FORECAST PERIOD EXCEEDED)
   ISN 0029         99 RETURN
   ISN 0030            END

*OPTIONS IN EFFECT*NAME(MAIN) NOOPTIMIZE LINECOUNT(60) SIZE(MAX) AUTODBL(NONE)

*OPTIONS IN EFFECT*SOURCE EBCDIC NOLIST NODECK OBJECT NOMAP NOFORMAT GOSTMT NOXREF ALC NOANSF NOTERM IBM FLAG(I)

*STATISTICS*     SOURCE STATEMENTS =       29, PROGRAM SIZE =      970, SUBPROGRAM NAME = RHBLT

*STATISTICS*  NO  DIAGNOSTICS GENERATED

****** END OF COMPILATION ******                                   928K BYTES OF CORE NOT USED

*STATISTICS*  NO  DIAGNOSTICS THIS STEP
```

APPENDIX 2

BASIC INPUT FOR THE MODIFIED RENU2 PROGRAM

SAMPLE SECTION DATA

Item	Section 1	Section 2
No. of vehicle classes	4	4
Analysis Period	18 yrs.	18 yrs.
Annual Growth Rate in 18-kip ESAL per year	3.35 %	3.35 %
Interest Rate	4.0 %	4.0 %
Average No. of Vehicles per Year	320,000	320,000
Type of Pavement	Overlaid	Hot Mix
Pavement Classification	Interstate Flexible	U. S. Flexible
Truck Types	2D 3A 3-S2 2-S1-2	2D 3A 3-S2 2-S1-2
Load Limits:		
GVW	80,000 lb.	80,000 lb.
Single Axle	20,000 lb.	20,000 lb.
Tandem Axle	34,000 lb.	34,000 lb.
Critical PSI	1.50	1.50
PSI after Overlay	4.80	4.70
Overlay Design Life	20 yrs.	20 yrs.
Avg. percent of Paved Shoulders	95 %	10 %
Avg. Paved Shoulder Width per lane	4.75 ft.	0.80 ft.
Avg. Granular Shoulder Width per lane	0.25 ft.	7.20 ft.

```
VEHICLE CLASSES          4
RUN PARAMETERS          18     0      3.35        4.00        0.0         0.0    320000.00
SYSTEM TITLE             0     0      0.0         0.0         0.0         0.0         0.0
INTERSTATE FLEX PAVEMENTS DISTRICT 1
 TEXAS TRANSPORTATION INSTITUTE
 SAMPLE RUN FOR FLEXIBLE PAVEMENTS
FLEXIBLE                 0     0     12.00        0.0         0.0         2.10        2.10
INTFLX B  INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF
   17    1     4     1     2    10     0 1.75 6.00 4.50    1    10     6
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0     0.0     0.0     0.0     0.0
ACP    3.00.0   ATB     4.00.0   AGB    12.00.0   LTS     0.00.0
AGE DISTRIBUTION        30     0      0.0         0.0         0.0         0.0         0.0
     0.0     0.0     0.0     0.0     0.0     0.0     0.0    36.0     0.0     0.0     0.0     0.0     0.0     3.0    16.0
     0.0     2.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
TRUCK TYPE               4     0      0.0         0.0         0.0         0.0         0.0
2D          3A          3-S2        2-S1-2
 1 0 0 0   0 1 0 0   0 2 0 0   4 0 0 0
  1   3.77  0.58 18.22  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.11
  2   3.83  0.58 18.29  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.25
  3   3.94  0.58 18.38  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.44
  4   3.98  0.57 18.46  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.56
  5   4.06  0.57 18.55  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.73
  6   4.12  0.57 18.53  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.77
  7   4.14  0.57 18.61  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.89
  8   4.23  0.56 18.59  0.55  0.0   0.0   0.0   0.0   0.0   0.0  23.82
  9   4.27  0.56 18.67  0.55  0.0   0.0   0.0   0.0   0.0   0.0  24.05
 10   4.33  0.56 18.65  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.10
 11   4.37  0.56 18.64  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.15
 12   4.43  0.55 18.63  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.19
 13   4.47  0.55 18.61  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.21
 14   4.50  0.55 18.59  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.22
 15   4.54  0.55 18.57  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.24
 16   4.60  0.54 18.55  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.27
 17   4.65  0.54 18.54  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.30
 18   4.69  0.54 18.52  0.56  0.0   0.0   0.0   0.0   0.0   0.0  24.34
LOAD LIMITS              0     0      0.0         0.0         0.0         0.0         0.0
     80.00      20.00      34.00      56.00
    120.00      22.40      36.00      56.00
     13.       13.       12.        8.
     16.       16.       16.       16.
SINGLE AXLES            11     0      3.00        0.0         0.0         0.0         0.0
         3.        1.       0.        0.        0.
         7.      566.      26.       71.       47.
         8.       70.      20.      329.       18.
        12.       92.      33.     1459.      126.
        16.       59.       8.       24.      101.
        18.       30.       0.        2.       40.
        19.        9.       0.        0.       12.
        20.       31.       0.        0.       13.
        22.       10.       0.        0.       13.
        24.        0.       0.        0.        3.
        26.        0.       0.        0.        2.
TANDEM AXLES            15     0      6.00        0.0         0.0         0.0         0.0
         6.        0.       0.        0.        0.
        12.        0.      38.      568.        0.
        18.        0.      18.      552.        0.
        24.        0.      15.      518.        0.
        30.        0.       9       815.        0.
        32.        0.       2.      428.        0
```

	33.	0.	1.	88.	0.				
	34.	0.	0.	244.	0.				
	36.	0.	0.	246.	0.				
	38.	0.	3.	133.	0.				
	40.	0.	1.	70.	0.				
	42.	0.	1.	26.	0.				
	44.	0.	0.	15.	0.				
	46.	0.	0.	14.	0.				
	50.	0.	0.	3.	0.				
GVW			28	0	10.00	0.0	0.0	0.0	0.0
	10.	130.	0.	0.	0.				
	14.	100.	5.	0.	0.				
	20.	86.	34.	0.	0.				
	22.	33.	4.	0.	0.				
	24.	26.	5.	5.	0.				
	26.	33.	5.	24.	0.				
	28.	19.	3.	54.	1.				
	30.	6.	3.	92.	4.				
	32.	1.	9.	82.	1.				
	34.	0.	3.	74.	2.				
	36.	0.	5.	54.	3.				
	38.	0.	1.	41.	0.				
	40.	0.	2.	45.	3.				
	45.	0.	2.	91.	1.				
	50.	0.	5.	96.	4.				
	55.	0.	0.	112.	7.				
	60.	0.	1.	126.	6.				
	65.	0.	0.	154.	7.				
	70.	0.	0.	221.	8.				
	72.	0.	0.	112.	2.				
	75.	0.	0.	159.	9.				
	80.	0.	0.	171.	8.				
	85.	0.	0.	99.	7.				
	90.	0.	0.	31.	0.				
	95.	0.	0.	9.	1.				
	100.	0.	0.	7.	0.				
	105.	0.	0.	3.	0.				
	110.	0.	0.	0.	0.				
EMPTY			13	0	4.00	0.0	0.0	0.0	0.0
	6.	14.	0.	0.	0.				
	8.	78.	0.	0.	0.				
	10.	143.	4.	0.	0.				
	12.	107.	10.	0.	0.				
	14.	75.	26.	0.	0.				
	16.	50.	47.	2.	0.				
	18.	9.	35.	4.	0.				
	20.	7.	14.	19.	0.				
	25.	4.	23.	290.	3.				
	30.	0.	6.	262.	10.				
	35.	0.	0.	120.	4.				
	40.	0.	0.	24.	0.				
	45.	0.	0.	4.	2.				
PERFORMANCE			0	0	4.80	1:50	4.80	20.00	0.0
	0.0								
OVERLAY			2	3	0.0	0:0	0.0	0.0	0.0
	95.00	4.75		0.25	66.00	0.50	0.0	0.0	3200.00
MODEL MAINT			1	0	0.0	0.0	0.0	0.0	0.0
	3.47	0.25		414.00					
	1000.00	3.86		8.75	12.90				
OLD SECTIONS			3	0	1800.00	10.00	0.0	0.0	0.0
						0.0	0.0	0.0	0.0

```
EXECUTE                   0    0     0.0          0.0          0.0          0.0          0.0
FLEXIBLE                  0    0    12.00         0.0          0.0          1.70         2.10
USFLX  A   US RURAL HOT MIX HIGH TRAFFIC
   17    3     1     1     2    10     0 1.25 6.00 3.30     1    10     6
     0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0
     0.0    0.0    0.0    0.0    0.0
ACP    4.00.0   AGB    12.00.0   LTS    0.00.0          0.00.0
AGE DISTRIBUTION         30    0     0.0          0.0          0.0          0.0          0.0
     0.0    8.0    8.0    2.0    1.0    9.0    2.0   15.0    2.0   12.0    6.0    3.0    2.0    6.0   30.0
     0.0   12.0    0.0    8.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    4.0    4.0    1.0
SINGLE AXLES             11    0     3.00         0.0          0.0          0.0          0.0
          3.     4.     0.     0.     0.
          7.   354.    10.    26.    12.
          8     53.     7.   117.     5.
         12.    75.    38.   856.    36.
         16.    44.    14.    14.    24.
         16.    23.     1.     1.     7.
         19.     3.     0.     1.     3.
         20.     2.     0.     1.     1.
         22.     4.     0.     0.     2.
         24.     2.     0.     0.     0.
         26      0.     0.     0.     0.
TANDEM AXLES             16    0     6.00         0.0          0.0          0.0          0.0
          6.     0.     0.     0.     0.
         12.     0.    12.   237.     0.
         18.     0.    16.   442.     0.
         24.     0.     7.   238.     0.
         30.     0.    20.   347.     0.
         32.     0.     5.   187.     0.
         33.     0.     1.    62.     0.
         34.     0.     0.   125.     0.
         36.     0.     1.   154.     0.
         38.     0.     4.    98.     0.
         40.     0.     2.    51.     0.
         42.     0.     0.    29.     0.
         44      0.     1.    22      0.
         46      0.     0.    11.     0.
         50      0.     0.     6.     0.
         55.     0.     1.     1.     0.
GVW                      29    0    10.00         0.0          0.0          0.0          0.0
         10.    97.     0.     0.     0.
         14.    53.     3.     0.     0.
         20.    80.     9.     0.     0.
         22.     8.     3.     0.     0.
         24.    15.     4.     0.     0.
         26.    29.     4.     0.     0.
         28.     5.     2.     6.     0.
         30      1.     5.    13.     0.
         32.     3.     3.    44.     2.
         34.     1.     6.    59.     0.
         36.     0.     3.    53.     1.
         38.     0.     7.    61.     0.
         40.     0.     4.    44.     1.
         45.     0.     8.    59.     0.
         50.     0.     4.    46.     0.
```

```
        55.      0.      3.     46.      3.
        60.      0.      1.     49       3.
        65.      0.      1.     58.      1.
        70.      0.      0.     88.      3.
        72.      0.      0.     55.      0.
        75.      0.      0.     79.      2.
        80.      0.      0.    117.      1.
        85.      0.      0.     25.      1.
        90.      0.      0.     28.      0.
        95.      0.      0.      8.      0.
       100.      0.      0.      5.      0.
       105.      0.      0.      1.      0.
       110.      0.      0.      1.      0.
       120.      0.      0.      1.      0.
OLD SECTIONS             3     0     750.00         10.00         0.0          0.0          0.0
PERFORMANCE              0     0       4.70          1.50         4.70        20.00         0.0
       0.0
OVERLAY                  2     3       0.0           0.0          0.0          0.0          0.0
      10.00       0.80      7.20     66.00        0.50       0.0        0.0     3200.00
EXECUTE                  0     0       0.0           0.0          0.0          0.0          0.0
```

APPENDIX 3

COST OUTPUT FROM THE MODIFIED RENU2 PROGRAM

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.0.0.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.193E+01	0.304E+05
2	0.194E+06	0.235E+01	0.304E+05
3	0.239E+06	0.272E+01	0.304E+05
4	0.284E+06	0.337E+01	0.304E+05
5	0.316E+06	0.389E+01	0.304E+05
6	0.334E+06	0.456E+01	0.304E+05
7	0.343E+06	0.528E+01	0.304E+05
8	0.347E+06	0.615E+01	0.304E+05
9	0.349E+06	0.707E+01	0.304E+05
10	0.349E+06	0.811E+01	0.304E+05
11	0.350E+06	0.112E+02	0.304E+05
12	0.350E+06	0.128E+02	0.304E+05
13	0.350E+06	0.147E+02	0.304E+05
14	0.350E+06	0.169E+02	0.304E+05
15	0.350E+06	0.191E+02	0.304E+05
16	0.350E+06	0.218E+02	0.304E+05
17	0.350E+06	0.245E+02	0.304E+05
18	0.350E+06	0.278E+02	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.0.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.926E-02	0.304E+05
2	0.194E+06	0.389E-01	0.304E+05
3	0.239E+06	0.100E+00	0.304E+05
4	0.284E+06	0.389E-01	0.304E+05
5	0.316E+06	0.106E+00	0.304E+05
6	0.334E+06	0.109E+00	0.304E+05
7	0.343E+06	0.141E+00	0.304E+05
8	0.347E+06	0.109E+00	0.304E+05
9	0.349E+06	0.145E+00	0.304E+05
10	0.349E+06	0.211E+00	0.304E+05
11	0.350E+06	0.193E+00	0.304E+05
12	0.350E+06	0.254E+00	0.304E+05
13	0.350E+06	0.350E+00	0.304E+05
14	0.350E+06	0.395E+00	0.304E+05
15	0.350E+06	0.456E+00	0.304E+05
16	0.350E+06	0.465E+00	0.304E+05
17	0.350E+06	0.602E+00	0.304E+05
18	0.350E+06	0.574E+00	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.1.0.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.305E+01	0.304E+05
2	0.194E+06	0.370E+01	0.304E+05
3	0.239E+06	0.448E+01	0.304E+05
4	0.284E+06	0.533E+01	0.304E+05
5	0.316E+06	0.622E+01	0.304E+05
6	0.334E+06	0.732E+01	0.304E+05
7	0.343E+06	0.846E+01	0.304E+05
8	0.347E+06	0.980E+01	0.304E+05
9	0.349E+06	0.113E+02	0.304E+05
10	0.349E+06	0.129E+02	0.304E+05
11	0.350E+06	0.179E+02	0.304E+05
12	0.350E+06	0.206E+02	0.304E+05
13	0.350E+06	0.236E+02	0.304E+05
14	0.350E+06	0.270E+02	0.304E+05
15	0.350E+06	0.306E+02	0.304E+05
16	0.350E+06	0.348E+02	0.304E+05
17	0.350E+06	0.393E+02	0.304E+05
18	0.350E+06	0.444E+02	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
0.0.1.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.937E+04	0.304E+05
2	0.191E+06	0.113E+05	0.304E+05
3	0.234E+06	0.134E+05	0.304E+05
4	0.276E+06	0.158E+05	0.304E+05
5	0.304E+06	0.185E+05	0.304E+05
6	0.317E+06	0.214E+05	0.304E+05
7	0.321E+06	0.246E+05	0.304E+05
8	0.319E+06	0.280E+05	0.304E+05
9	0.315E+06	0.318E+05	0.304E+05
10	0.309E+06	0.358E+05	0.304E+05
11	0.303E+06	0.490E+05	0.304E+05
12	0.296E+06	0.552E+05	0.304E+05
13	0.289E+06	0.618E+05	0.304E+05
14	0.282E+06	0.688E+05	0.304E+05
15	0.274E+06	0.760E+05	0.304E+05
16	0.266E+06	0.834E+05	0.304E+05
17	0.257E+06	0.910E+05	0.304E+05
18	0.249E+06	0.987E+05	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.0.1.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.101E+05	0.304E+05
2	0.191E+06	0.121E+05	0.304E+05
3	0.233E+06	0.144E+05	0.304E+05
4	0.275E+06	0.170E+05	0.304E+05
5	0.303E+06	0.198E+05	0.304E+05
6	0.316E+06	0.229E+05	0.304E+05
7	0.319E+06	0.263E+05	0.304E+05
8	0.317E+06	0.300E+05	0.304E+05
9	0.313E+06	0.339E+05	0.304E+05
10	0.307E+06	0.382E+05	0.304E+05
11	0.300E+06	0.523E+05	0.304E+05
12	0.293E+06	0.588E+05	0.304E+05
13	0.285E+06	0.657E+05	0.304E+05
14	0.277E+06	0.729E+05	0.304E+05
15	0.269E+06	0.804E+05	0.304E+05
16	0.261E+06	0.881E+05	0.304E+05
17	0.252E+06	0.959E+05	0.304E+05
18	0.243E+06	0.104E+06	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
0.1.1.0.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.951E+04	0.304E+05
2	0.191E+06	0.115E+05	0.304E+05
3	0.234E+06	0.136E+05	0.304E+05
4	0.276E+06	0.160E+05	0.304E+05
5	0.304E+06	0.187E+05	0.304E+05
6	0.317E+06	0.217E+05	0.304E+05
7	0.321E+06	0.249E+05	0.304E+05
8	0.319E+06	0.284E+05	0.304E+05
9	0.315E+06	0.322E+05	0.304E+05
10	0.309E+06	0.363E+05	0.304E+05
11	0.302E+06	0.497E+05	0.304E+05
12	0.296E+06	0.560E+05	0.304E+05
13	0.288E+06	0.626E+05	0.304E+05
14	0.281E+06	0.696E+05	0.304E+05
15	0.273E+06	0.769E+05	0.304E+05
16	0.265E+06	0.844E+05	0.304E+05
17	0.256E+06	0.921E+05	0.304E+05
18	0.248E+06	0.998E+05	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.1.1.O.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.102E+05	0.304E+05
2	0.191E+06	0.123E+05	0.304E+05
3	0.233E+06	0.146E+05	0.304E+05
4	0.275E+06	0.172E+05	0.304E+05
5	0.303E+06	0.201E+05	0.304E+05
6	0.316E+06	0.232E+05	0.304E+05
7	0.319E+06	0.266E+05	0.304E+05
8	0.317E+06	0.304E+05	0.304E+05
9	0.312E+06	0.344E+05	0.304E+05
10	0.306E+06	0.387E+05	0.304E+05
11	0.299E+06	0.530E+05	0.304E+05
12	0.292E+06	0.595E+05	0.304E+05
13	0.284E+06	0.665E+05	0.304E+05
14	0.276E+06	0.738E+05	0.304E+05
15	0.268E+06	0.813E+05	0.304E+05
16	0.260E+06	0.890E+05	0.304E+05
17	0.251E+06	0.969E+05	0.304E+05
18	0.242E+06	0.105E+06	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.0.0.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.625E+01	0.304E+05
2	0.194E+06	0.757E+01	0.304E+05
3	0.239E+06	0.903E+01	0.304E+05
4	0.284E+06	0.107E+02	0.304E+05
5	0.316E+06	0.127E+02	0.304E+05
6	0.334E+06	0.148E+02	0.304E+05
7	0.343E+06	0.172E+02	0.304E+05
8	0.347E+06	0.199E+02	0.304E+05
9	0.349E+06	0.229E+02	0.304E+05
10	0.349E+06	0.263E+02	0.304E+05
11	0.350E+06	0.363E+02	0.304E+05
12	0.350E+06	0.417E+02	0.304E+05
13	0.350E+06	0.478E+02	0.304E+05
14	0.350E+06	0.545E+02	0.304E+05
15	0.350E+06	0.622E+02	0.304E+05
16	0.350E+06	0.706E+02	0.304E+05
17	0.350E+06	0.798E+02	0.304E+05
18	0.350E+06	0.902E+02	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.0.0.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.202E+02	0.304E+05
2	0.194E+06	0.244E+02	0.304E+05
3	0.239E+06	0.293E+02	0.304E+05
4	0.284E+06	0.347E+02	0.304E+05
5	0.316E+06	0.410E+02	0.304E+05
6	0.334E+06	0.479E+02	0.304E+05
7	0.343E+06	0.558E+02	0.304E+05
8	0.347E+06	0.645E+02	0.304E+05
9	0.348E+06	0.742E+02	0.304E+05
10	0.349E+06	0.851E+02	0.304E+05
11	0.349E+06	0.117E+03	0.304E+05
12	0.350E+06	0.135E+03	0.304E+05
13	0.350E+06	0.155E+03	0.304E+05
14	0.350E+06	0.177E+03	0.304E+05
15	0.350E+06	0.201E+03	0.304E+05
16	0.350E+06	0.228E+03	0.304E+05
17	0.350E+06	0.259E+03	0.304E+05
18	0.350E+06	0.292E+03	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.0.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.842E+01	0.304E+05
2	0.194E+06	0.102E+02	0.304E+05
3	0.239E+06	0.122E+02	0.304E+05
4	0.284E+06	0.145E+02	0.304E+05
5	0.316E+06	0.171E+02	0.304E+05
6	0.334E+06	0.200E+02	0.304E+05
7	0.343E+06	0.233E+02	0.304E+05
8	0.347E+06	0.268E+02	0.304E+05
9	0.349E+06	0.310E+02	0.304E+05
10	0.349E+06	0.355E+02	0.304E+05
11	0.350E+06	0.489E+02	0.304E+05
12	0.350E+06	0.563E+02	0.304E+05
13	0.350E+06	0.646E+02	0.304E+05
14	0.350E+06	0.737E+02	0.304E+05
15	0.350E+06	0.839E+02	0.304E+05
16	0.350E+06	0.952E+02	0.304E+05
17	0.350E+06	0.108E+03	0.304E+05
18	0.350E+06	0.122E+03	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.1.0.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.244E+02	0.304E+05
2	0.194E+06	0.295E+02	0.304E+05
3	0.239E+06	0.355E+02	0.304E+05
4	0.284E+06	0.420E+02	0.304E+05
5	0.316E+06	0.495E+02	0.304E+05
6	0.334E+06	0.579E+02	0.304E+05
7	0.343E+06	0.674E+02	0.304E+05
8	0.347E+06	0.779E+02	0.304E+05
9	0.348E+06	0.898E+02	0.304E+05
10	0.349E+06	0.103E+03	0.304E+05
11	0.349E+06	0.142E+03	0.304E+05
12	0.350E+06	0.163E+03	0.304E+05
13	0.350E+06	0.187E+03	0.304E+05
14	0.350E+06	0.214E+03	0.304E+05
15	0.350E+06	0.243E+03	0.304E+05
16	0.350E+06	0.276E+03	0.304E+05
17	0.350E+06	0.313E+03	0.304E+05
18	0.349E+06	0.353E+03	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
O.O.1.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.105E+05	0.304E+05
2	0.191E+06	0.126E+05	0.304E+05
3	0.233E+06	0.150E+05	0.304E+05
4	0.275E+06	0.177E+05	0.304E+05
5	0.302E+06	0.206E+05	0.304E+05
6	0.315E+06	0.238E+05	0.304E+05
7	0.318E+06	0.274E+05	0.304E+05
8	0.316E+06	0.312E+05	0.304E+05
9	0.311E+06	0.353E+05	0.304E+05
10	0.305E+06	0.397E+05	0.304E+05
11	0.298E+06	0.543E+05	0.304E+05
12	0.290E+06	0.610E+05	0.304E+05
13	0.283E+06	0.681E+05	0.304E+05
14	0.275E+06	0.755E+05	0.304E+05
15	0.266E+06	0.831E+05	0.304E+05
16	0.258E+06	0.909E+05	0.304E+05
17	0.249E+06	0.988E+05	0.304E+05
18	0.240E+06	0.107E+06	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.O.1.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.112E+05	0.304E+05
2	0.190E+06	0.135E+05	0.304E+05
3	0.233E+06	0.161E+05	0.304E+05
4	0.274E+06	0.189E+05	0.304E+05
5	0.301E+06	0.220E+05	0.304E+05
6	0.314E+06	0.254E+05	0.304E+05
7	0.317E+06	0.292E+05	0.304E+05
8	0.314E+06	0.332E+05	0.304E+05
9	0.309E+06	0.375E+05	0.304E+05
10	0.302E+06	0.421E+05	0.304E+05
11	0.295E+06	0.576E+05	0.304E+05
12	0.287E+06	0.647E+05	0.304E+05
13	0.279E+06	0.720E+05	0.304E+05
14	0.270E+06	0.797E+05	0.304E+05
15	0.261E+06	0.875E+05	0.304E+05
16	0.252E+06	0.955E+05	0.304E+05
17	0.243E+06	0.104E+06	0.304E+05
18	0.235E+06	0.112E+06	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
O.1.1.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.107E+05	0.304E+05
2	0.191E+06	0.128E+05	0.304E+05
3	0.233E+06	0.152E+05	0.304E+05
4	0.274E+06	0.179E+05	0.304E+05
5	0.302E+06	0.209E+05	0.304E+05
6	0.315E+06	0.242E+05	0.304E+05
7	0.318E+06	0.277E+05	0.304E+05
8	0.316E+06	0.316E+05	0.304E+05
9	0.311E+06	0.358E+05	0.304E+05
10	0.304E+06	0.402E+05	0.304E+05
11	0.297E+06	0.550E+05	0.304E+05
12	0.290E+06	0.618E+05	0.304E+05
13	0.282E+06	0.689E+05	0.304E+05
14	0.274E+06	0.764E+05	0.304E+05
15	0.265E+06	0.840E+05	0.304E+05
16	0.256E+06	0.919E+05	0.304E+05
17	0.248E+06	0.998E+05	0.304E+05
18	0.239E+06	0.108E+06	0.304E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.1.1.1.

INTFLX B INTERSTATE FLEX RURAL OVRLAY HIGH TRAFF

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.160E+06	0.114E+05	0.304E+05
2	0.190E+06	0.137E+05	0.304E+05
3	0.233E+06	0.163E+05	0.304E+05
4	0.274E+06	0.192E+05	0.304E+05
5	0.301E+06	0.223E+05	0.304E+05
6	0.314E+06	0.258E+05	0.304E+05
7	0.316E+06	0.296E+05	0.304E+05
8	0.314E+06	0.336E+05	0.304E+05
9	0.308E+06	0.380E+05	0.304E+05
10	0.301E+06	0.427E+05	0.304E+05
11	0.294E+06	0.583E+05	0.304E+05
12	0.286E+06	0.654E+05	0.304E+05
13	0.278E+06	0.729E+05	0.304E+05
14	0.269E+06	0.806E+05	0.304E+05
15	0.260E+06	0.885E+05	0.304E+05
16	0.251E+06	0.965E+05	0.304E+05
17	0.242E+06	0.105E+06	0.304E+05
18	0.233E+06	0.113E+06	0.304E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.0.0.0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.108E+01	0.720E+05
2	0.549E+06	0.134E+01	0.720E+05
3	0.593E+06	0.160E+01	0.720E+05
4	0.635E+06	0.193E+01	0.720E+05
5	0.671E+06	0.226E+01	0.720E+05
6	0.703E+06	0.263E+01	0.720E+05
7	0.732E+06	0.316E+01	0.720E+05
8	0.759E+06	0.365E+01	0.720E+05
9	0.785E+06	0.425E+01	0.720E+05
10	0.804E+06	0.487E+01	0.720E+05
11	0.817E+06	0.678E+01	0.720E+05
12	0.823E+06	0.784E+01	0.720E+05
13	0.826E+06	0.907E+01	0.720E+05
14	0.827E+06	0.104E+02	0.720E+05
15	0.828E+06	0.120E+02	0.720E+05
16	0.828E+06	0.137E+02	0.720E+05
17	0.828E+06	0.157E+02	0.720E+05
18	0.828E+06	0.179E+02	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.0.0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.888E-02	0.720E+05
2	0.549E+06	0.333E-02	0.720E+05
3	0.593E+06	0.388E-01	0.720E+05
4	0.635E+06	0.155E-01	0.720E+05
5	0.671E+06	0.621E-01	0.720E+05
6	0.703E+06	0.277E-01	0.720E+05
7	0.732E+06	0.344E-01	0.720E+05
8	0.759E+06	0.732E-01	0.720E+05
9	0.785E+06	0.566E-01	0.720E+05
10	0.804E+06	0.910E-01	0.720E+05
11	0.817E+06	0.832E-01	0.720E+05
12	0.823E+06	0.943E-01	0.720E+05
13	0.826E+06	0.162E+00	0.720E+05
14	0.827E+06	0.136E+00	0.720E+05
15	0.828E+06	0.203E+00	0.720E+05
16	0.828E+06	0.243E+00	0.720E+05
17	0.828E+06	0.195E+00	0.720E+05
18	0.828E+06	0.321E+00	0.720E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.1.0.0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.184E+01	0.720E+05
2	0.549E+06	0.222E+01	0.720E+05
3	0.593E+06	0.270E+01	0.720E+05
4	0.635E+06	0.318E+01	0.720E+05
5	0.671E+06	0.382E+01	0.720E+05
6	0.703E+06	0.443E+01	0.720E+05
7	0.732E+06	0.521E+01	0.720E+05
8	0.759E+06	0.611E+01	0.720E+05
9	0.785E+06	0.713E+01	0.720E+05
10	0.804E+06	0.819E+01	0.720E+05
11	0.817E+06	0.113E+02	0.720E+05
12	0.823E+06	0.131E+02	0.720E+05
13	0.826E+06	0.152E+02	0.720E+05
14	0.827E+06	0.175E+02	0.720E+05
15	0.828E+06	0.201E+02	0.720E+05
16	0.828E+06	0.229E+02	0.720E+05
17	0.828E+06	0.263E+02	0.720E+05
18	0.828E+06	0.299E+02	0.720E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
O.O.1.O.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.123E+05	0.720E+05
2	0.541E+06	0.147E+05	0.720E+05
3	0.580E+06	0.175E+05	0.720E+05
4	0.615E+06	0.207E+05	0.720E+05
5	0.644E+06	0.243E+05	0.720E+05
6	0.666E+06	0.283E+05	0.720E+05
7	0.684E+06	0.328E+05	0.720E+05
8	0.699E+06	0.377E+05	0.720E+05
9	0.711E+06	0.431E+05	0.720E+05
10	0.716E+06	0.490E+05	0.720E+05
11	0.712E+06	0.672E+05	0.720E+05
12	0.702E+06	0.762E+05	0.720E+05
13	0.688E+06	0.858E+05	0.720E+05
14	0.671E+06	0.961E+05	0.720E+05
15	0.652E+06	0.107E+06	0.720E+05
16	0.631E+06	0.118E+06	0.720E+05
17	0.610E+06	0.130E+06	0.720E+05
18	0.588E+06	0.141E+06	0.720E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.O.1.O.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.133E+05	0.720E+05
2	0.541E+06	0.159E+05	0.720E+05
3	0.579E+06	0.189E+05	0.720E+05
4	0.613E+06	0.224E+05	0.720E+05
5	0.642E+06	0.262E+05	0.720E+05
6	0.663E+06	0.305E+05	0.720E+05
7	0.680E+06	0.353E+05	0.720E+05
8	0.695E+06	0.405E+05	0.720E+05
9	0.706E+06	0.463E+05	0.720E+05
10	0.709E+06	0.525E+05	0.720E+05
11	0.705E+06	0.720E+05	0.720E+05
12	0.694E+06	0.815E+05	0.720E+05
13	0.678E+06	0.916E+05	0.720E+05
14	0.660E+06	0.102E+06	0.720E+05
15	0.640E+06	0.114E+06	0.720E+05
16	0.619E+06	0.125E+06	0.720E+05
17	0.597E+06	0.137E+06	0.720E+05
18	0.574E+06	0.149E+06	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.1.0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.125E+05	0.720E+05
2	0.541E+06	0.150E+05	0.720E+05
3	0.580E+06	0.178E+05	0.720E+05
4	0.615E+06	0.211E+05	0.720E+05
5	0.643E+06	0.247E+05	0.720E+05
6	0.666E+06	0.288E+05	0.720E+05
7	0.683E+06	0.333E+05	0.720E+05
8	0.698E+06	0.383E+05	0.720E+05
9	0.710E+06	0.438E+05	0.720E+05
10	0.715E+06	0.497E+05	0.720E+05
11	0.711E+06	0.682E+05	0.720E+05
12	0.700E+06	0.773E+05	0.720E+05
13	0.686E+06	0.870E+05	0.720E+05
14	0.668E+06	0.974E+05	0.720E+05
15	0.649E+06	0.108E+06	0.720E+05
16	0.629E+06	0.120E+06	0.720E+05
17	0.607E+06	0.131E+06	0.720E+05
18	0.585E+06	0.143E+06	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.1.1.0.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.135E+05	0.720E+05
2	0.541E+06	0.162E+05	0.720E+05
3	0.579E+06	0.192E+05	0.720E+05
4	0.613E+06	0.227E+05	0.720E+05
5	0.641E+06	0.266E+05	0.720E+05
6	0.663E+06	0.310E+05	0.720E+05
7	0.680E+06	0.358E+05	0.720E+05
8	0.694E+06	0.411E+05	0.720E+05
9	0.704E+06	0.470E+05	0.720E+05
10	0.708E+06	0.533E+05	0.720E+05
11	0.703E+06	0.730E+05	0.720E+05
12	0.692E+06	0.826E+05	0.720E+05
13	0.676E+06	0.928E+05	0.720E+05
14	0.658E+06	0.104E+06	0.720E+05
15	0.637E+06	0.115E+06	0.720E+05
16	0.616E+06	0.127E+06	0.720E+05
17	0.594E+06	0.139E+06	0.720E+05
18	0.571E+06	0.150E+06	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.0.0.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.406E+01	0.720E+05
2	0.549E+06	0.486E+01	0.720E+05
3	0.593E+06	0.582E+01	0.720E+05
4	0.635E+06	0.697E+01	0.720E+05
5	0.671E+06	0.823E+01	0.720E+05
6	0.703E+06	0.966E+01	0.720E+05
7	0.732E+06	0.114E+02	0.720E+05
8	0.759E+06	0.133E+02	0.720E+05
9	0.785E+06	0.154E+02	0.720E+05
10	0.804E+06	0.179E+02	0.720E+05
11	0.817E+06	0.246E+02	0.720E+05
12	0.823E+06	0.285E+02	0.720E+05
13	0.826E+06	0.330E+02	0.720E+05
14	0.827E+06	0.380E+02	0.720E+05
15	0.828E+06	0.436E+02	0.720E+05
16	0.828E+06	0.499E+02	0.720E+05
17	0.828E+06	0.571E+02	0.720E+05
18	0.828E+06	0.650E+02	0.720E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1.O.O.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.146E+02	0.720E+05
2	0.549E+06	0.177E+02	0.720E+05
3	0.593E+06	0.212E+02	0.720E+05
4	0.635E+06	0.252E+02	0.720E+05
5	0.671E+06	0.299E+02	0.720E+05
6	0.703E+06	0.352E+02	0.720E+05
7	0.732E+06	0.413E+02	0.720E+05
8	0.759E+06	0.482E+02	0.720E+05
9	0.784E+06	0.560E+02	0.720E+05
10	0.804E+06	0.648E+02	0.720E+05
11	0.817E+06	0.895E+02	0.720E+05
12	0.823E+06	0.104E+03	0.720E+05
13	0.826E+06	0.120E+03	0.720E+05
14	0.827E+06	0.138E+03	0.720E+05
15	0.828E+06	0.158E+03	0.720E+05
16	0.828E+06	0.181E+03	0.720E+05
17	0.828E+06	0.207E+03	0.720E+05
18	0.828E+06	0.236E+03	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.0.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.559E+01	0.720E+05
2	0.549E+06	0.674E+01	0.720E+05
3	0.593E+06	0.810E+01	0.720E+05
4	0.635E+06	0.966E+01	0.720E+05
5	0.671E+06	0.115E+02	0.720E+05
6	0.703E+06	0.135E+02	0.720E+05
7	0.732E+06	0.158E+02	0.720E+05
8	0.759E+06	0.184E+02	0.720E+05
9	0.785E+06	0.214E+02	0.720E+05
10	0.804E+06	0.248E+02	0.720E+05
11	0.817E+06	0.342E+02	0.720E+05
12	0.823E+06	0.397E+02	0.720E+05
13	0.826E+06	0.458E+02	0.720E+05
14	0.827E+06	0.527E+02	0.720E+05
15	0.828E+06	0.607E+02	0.720E+05
16	0.828E+06	0.694E+02	0.720E+05
17	0.828E+06	0.793E+02	0.720E+05
18	0.828E+06	0.904E+02	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.1.0.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.180E+02	0.720E+05
2	0.549E+06	0.217E+02	0.720E+05
3	0.593E+06	0.261E+02	0.720E+05
4	0.635E+06	0.311E+02	0.720E+05
5	0.671E+06	0.368E+02	0.720E+05
6	0.703E+06	0.434E+02	0.720E+05
7	0.732E+06	0.508E+02	0.720E+05
8	0.759E+06	0.593E+02	0.720E+05
9	0.784E+06	0.689E+02	0.720E+05
10	0.804E+06	0.798E+02	0.720E+05
11	0.817E+06	0.110E+03	0.720E+05
12	0.823E+06	0.128E+03	0.720E+05
13	0.826E+06	0.147E+03	0.720E+05
14	0.827E+06	0.170E+03	0.720E+05
15	0.828E+06	0.195E+03	0.720E+05
16	0.828E+06	0.223E+03	0.720E+05
17	0.828E+06	0.255E+03	0.720E+05
18	0.828E+06	0.291E+03	0.720E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
0.0.1.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.139E+05	0.720E+05
2	0.540E+06	0.167E+05	0.720E+05
3	0.578E+06	0.198E+05	0.720E+05
4	0.612E+06	0.234E+05	0.720E+05
5	0.640E+06	0.274E+05	0.720E+05
6	0.661E+06	0.319E+05	0.720E+05
7	0.678E+06	0.369E+05	0.720E+05
8	0.692E+06	0.423E+05	0.720E+05
9	0.702E+06	0.483E+05	0.720E+05
10	0.705E+06	0.547E+05	0.720E+05
11	0.700E+06	0.750E+05	0.720E+05
12	0.688E+06	0.848E+05	0.720E+05
13	0.672E+06	0.952E+05	0.720E+05
14	0.653E+06	0.106E+06	0.720E+05
15	0.633E+06	0.118E+06	0.720E+05
16	0.611E+06	0.129E+06	0.720E+05
17	0.588E+06	0.141E+06	0.720E+05
18	0.566E+06	0.153E+06	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
1.0.1.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.150E+05	0.720E+05
2	0.540E+06	0.179E+05	0.720E+05
3	0.577E+06	0.213E+05	0.720E+05
4	0.611E+06	0.251E+05	0.720E+05
5	0.638E+06	0.294E+05	0.720E+05
6	0.658E+06	0.342E+05	0.720E+05
7	0.674E+06	0.395E+05	0.720E+05
8	0.687E+06	0.453E+05	0.720E+05
9	0.696E+06	0.516E+05	0.720E+05
10	0.698E+06	0.584E+05	0.720E+05
11	0.692E+06	0.800E+05	0.720E+05
12	0.679E+06	0.903E+05	0.720E+05
13	0.662E+06	0.101E+06	0.720E+05
14	0.642E+06	0.113E+06	0.720E+05
15	0.620E+06	0.124E+06	0.720E+05
16	0.598E+06	0.136E+06	0.720E+05
17	0.575E+06	0.149E+06	0.720E+05
18	0.552E+06	0.161E+06	0.720E+05

VEHICLE COMBINATION (1=IN,0=OUT)
1 2 3 4 5 6
0.1.1.1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.141E+05	0.720E+05
2	0.540E+06	0.169E+05	0.720E+05
3	0.578E+06	0.201E+05	0.720E+05
4	0.612E+06	0.238E+05	0.720E+05
5	0.640E+06	0.278E+05	0.720E+05
6	0.661E+06	0.324E+05	0.720E+05
7	0.677E+06	0.374E+05	0.720E+05
8	0.691E+06	0.429E+05	0.720E+05
9	0.701E+06	0.490E+05	0.720E+05
10	0.704E+06	0.555E+05	0.720E+05
11	0.698E+06	0.761E+05	0.720E+05
12	0.686E+06	0.859E+05	0.720E+05
13	0.670E+06	0.965E+05	0.720E+05
14	0.651E+06	0.108E+06	0.720E+05
15	0.630E+06	0.119E+06	0.720E+05
16	0.608E+06	0.131E+06	0.720E+05
17	0.585E+06	0.143E+06	0.720E+05
18	0.563E+06	0.155E+06	0.720E+05

VEHICLE COMBINATION (1=IN,O=OUT)
1 2 3 4 5 6
1. 1. 1. 1.

USFLX A US RURAL HOT MIX HIGH TRAFFIC

YEAR	ROUT MAINT COST	REHAB COST	PREV MAINT COST
1	0.505E+06	0.152E+05	0.720E+05
2	0.540E+06	0.182E+05	0.720E+05
3	0.577E+06	0.216E+05	0.720E+05
4	0.610E+06	0.255E+05	0.720E+05
5	0.637E+06	0.299E+05	0.720E+05
6	0.658E+06	0.347E+05	0.720E+05
7	0.673E+06	0.401E+05	0.720E+05
8	0.686E+06	0.459E+05	0.720E+05
9	0.695E+06	0.523E+05	0.720E+05
10	0.697E+06	0.592E+05	0.720E+05
11	0.690E+06	0.811E+05	0.720E+05
12	0.677E+06	0.914E+05	0.720E+05
13	0.659E+06	0.102E+06	0.720E+05
14	0.639E+06	0.114E+06	0.720E+05
15	0.618E+06	0.126E+06	0.720E+05
16	0.595E+06	0.138E+06	0.720E+05
17	0.572E+06	0.150E+06	0.720E+05
18	0.549E+06	0.162E+06	0.720E+05

APPENDIX 4

MODIFIED INCREMENTAL APPROACH BASIC CODE

MODIFIED INCREMENTAL APPROACH BASIC CODE

FILE NAME: MIA

THE MODIFIED INCREMENTAL APPROACH IS EXECUTED BY CALLING THIS ROUTINE, WHICH ASKS FOR INPUT DATA AND CALCULATES COST PARTITIONS (P) AS DISCUSSED IN CHAPTER 4.

```
1 COMMON NVC,NDIM,P()
10 '
20 '      INPUT ROUTINE
30 '
35  PRINT  CHR$(26):PRINT:PRINT:PRINT
40 INPUT "NUMBER OF VEHICLE CLASSES";NVC
41 LPRINT "NUMBER OF VEHICLE CLASSES: ";NVC
50 NDIM = 2©NVC - 1
60  DIM  COMBI(NDIM),BIN(NVC),P(NDIM)
61 DIM SUMP(NDIM)
70  PRINT:  PRINT:PRINT:LPRINT:LPRINT:LPRINT
80 FOR INDEX=0 TO NDIM
85   PRINT
90   IEXP = 1
95       IF INDEX=0 THEN PRINT"COST OF DESIGN FOR NO
                        CLASSES";:LPRINT "COST OF DESIGN FOR
                        NO CLASSES";:GOTO 220
100 PRINT "COST OF DESIGN FOR CLASSES ";
101      LPRINT "COST OF DESIGN FOR CLASSES ";
110 INQUOT = INDEX
120 QUOT = INQUOT/2
130 INQUOT = INT(QUOT)
140 RES = (QUOT-INQUOT)*2
150          IND = RES*IEXP
160 IF IND <> 0 THEN PRINT IND;:LPRINT IND;
170 WHILE INQUOT>0
180          IF IND<>0 THEN PRINT ",";:LPRINT ",";
190          IEXP = IEXP+1
200          GOTO 120
210 WEND
215  PRINT  TAB(60);:LPRINT  TAB(60);
220 INPUT ":  ",COMBI(INDEX)
221 LPRINT ":  ";COMBI(INDEX)
230 NEXT INDEX
240 GOSUB 1000
300 '
309 '   CREATE COST PARTITIONS
310 '   CONVERT INDEXES TO BINARY
315 PRINT CHR$(26)
320 '
```

```
330 FOR IOTA=1 TO NDIM
340 FOR J=1 TO NVC
350  BIN(J)=0
360 NEXT J
370 INQUOT=IOTA
380 J=1
390 QUOT=INQUOT/2
400  INQUOT=INT(QUOT)
410  RES=(QUOT-INQUOT)*2
420  BIN(J)=RES
430 IF INQUOT=0 THEN GOTO 493
440 J=J+1 : GOTO 390
493 '
600 '
610 '   IDENTIFY POSITIONS FOR ONES
620 '
622 FOR K=1 TO NVC:PST(K)=0:NEXT K
630 IDX=0
640 FOR J=1 TO NVC
650 IF BIN(J)=0 THEN GOTO 680
660 IDX=IDX+1
670  PST(IDX)=J
680 NEXT J
700 NCOMB=2^IDX-2
705 DIM CB(NCOMB)
710 FOR J=1 TO NCOMB
720  CB(J)=0
730 INQUOT=J
740 PINX=0
750 INX=1
760 QUOT=INQUOT/2
770  INQUOT=INT(QUOT)
780  RES=(QUOT-INQUOT)*2
790 PINX=PINX+RES*(2^(PST(INX)-1))
800 INX=INX+1
810 IF INQUOT<>0 THEN GOTO 760
820 CB(J)=P(PINX)
830 NEXT J
900 '
901 '   SUM OF COMBINATIONS
902 '
923 SUMP(IOTA)=0
925 FOR J=1 TO NCOMB
930   SUMP(IOTA)=SUMP(IOTA)+CB(J)
940 NEXT J
950 P(IOTA)=P(IOTA)-SUMP(IOTA)
952 PRINT"COST DUE TO THE ACTION OF CLASSES";
953 FOR W=1 TO IDX:PRINT PST(W);:NEXT W:PRINT "IS    ";
960 PRINT P(IOTA):PRINT
962 ERASE CB
970 NEXT IOTA
980 CHAIN "GENER",,ALL
```

```
1000 FOR INDEX=0 TO NDIM
1010      P(INDEX)=COMBI(NDIM)-COMBI(NDIM-INDEX)
1020      NEXT INDEX
1030 RETURN
```

FILE NAME: GENER.BAS

THIS PROGRAM ACCEPTS VEHICLE-MILES-OF-TRAVEL DATA FOR EACH VEHICLE CLASS, CALCULATES ALLOCATED COSTS, AND REPORTS RESULTS FROM THE MODIFIED INCREMENTAL APPROACH.

```
10 DIM PMAT(NVC,NDIM),VREC(NDIM),VMT(NVC),VMTMAT(NVC,NVC)
20 '
30 '   *** INPUT VALUES OF VMT ***
40 '
45 LPRINT:LPRINT:LPRINT
50 FOR I=1 TO NVC
60    PRINT "VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS";I;
70    INPUT ": ",VMT(I)
71    LPRINT "VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS";
            I;": ";VMT(I)
80 NEXT I
90 '
100 '   *** GENERATE J'th COLUMN OF PMAT AND J'th ELEMENT OF
        VREC ***
110 '
120 FOR J=1 TO NDIM
130    POSITION=1
140    SUMVMT=0
150 '
160 '  CONVERT J TO BINARY
170 '
180    INQUOT=J
190    QUOT=INQUOT/2
200    INQUOT=INT(QUOT)
210    RES=(QUOT-INQUOT)*2
220 '      STORE P(J) IN APPROPRATE POSITION IN COLUMN J
230    PMAT(POSITION,J)=RES*P(J)
240 '      ACCUMULATE DENOMINATOR OF J'th VREC ELEMENT
250    SUMVMT=SUMVMT+RES*(VMT(POSITION))
260    POSITION=POSITION+1
270    IF INQUOT>0 THEN GOTO 190
280 '      GENERATE J'th ELEMENT OF VREC
290    IF SUMVMT=0 THEN VREC(J)=0 ELSE VREC(J)=1/SUMVMT
300 NEXT J
310 '
320 '   *** GENERATE VMTMAT ***
330 '
340 FOR I=1 TO NVC
350    FOR J=1 TO NVC
360       IF I=J THEN VMTMAT(I,J)=VMT(I) ELSE VMTMAT(I,J)=0
370 NEXT J:NEXT I
470 '
480 '  *** MULTIPLY [VMTMAT] [PMAT] [VREC] TO GET [ALLOC]
481 '
490 FLAG=0
500 ADIM=NVC
```

```
510 ABDIM=NVC
520 BDIM=NDIM
530  DIM A(ADIM,ABDIM),B(ABDIM,BDIM)
540 FOR I=1 TO ADIM
550     FOR J=1 TO ABDIM
560        A(I,J)=VMTMAT(I,J)
570     NEXT J
580     FOR K=1 TO BDIM
590         B(I,K)=PMAT(I,K)
600     NEXT K
610 NEXT I
620 GOSUB 2000
630 DIM ALLOC(NVC,NDIM)
640 FOR I=1 TO NVC
650     FOR J=1 TO NDIM
660        ALLOC(I,J)=C(I,J)
670 NEXT J:NEXT I
680 ERASE A,B,C
690 ADIM=NVC
700 ABDIM=NDIM
710 BDIM=1
720  DIM A(ADIM,ABDIM),B(ABDIM,BDIM)
730 FOR J=1 TO ABDIM
740     FOR I=1 TO ADIM
750        A(I,J)=ALLOC (I,J)
760     NEXT I
770     B(J,BDIM)=VREC(J)
780 NEXT J
785 FLAG=1
790 GOSUB 2000
800 ERASE ALLOC
810  PRINT  CHR$(26):PRINT:PRINT:PRINT
811  LPRINT:LPRINT:LPRINT
820 FOR I=1 TO NVC
830     ALLOC (I)=C(I,1)+VMT(I)*VREC(NDIM)*COMBI(0)
840     PRINT:LPRINT
850     PRINT "COST ALLOCATED TO VEHICLE CLASS";I;": $";
            ALLOC(I)
851     LPRINT "COST ALLOCATED TO VEHICLE CLASS";I;":  ";
             ALLOC(I)
860 NEXT I
870 ZARRAP$=INKEY$
890 IF ZARRAP$="" THEN GOTO 870
900 END
2000 CHAIN "MATPROD",,ALL
2010 IF FLAG THEN 800 ELSE 630
2020 RETURN
```

FILE NAME: MATPROD.BAS

THIS ROUTINE PERFORMS MATRIX PRODUCTS.

```
10 DIM C(ADIM,BDIM)
20 '
30 '        PERFORM MATRIX MULTIPLICATION
40 '
50 FOR I=1 TO ADIM
60  FOR J=1 TO BDIM
70          C(I,J)=0
80          FOR K=1 TO ABDIM
90                    C(I,J)=C(I,J)+A(I,K)*B(K,J)
100         NEXT K
110 NEXT J
120 NEXT I
130  CHAIN "GENER",2010,ALL
```

APPENDIX 5

SAMPLE RUN FOR THE MODIFIED INCREMENTAL APPROACH

NUMBER OF VEHICLE CLASSES: 4

COST OF DESIGN FOR NO CLASSES: 0	
COST OF DESIGN FOR CLASSES 1	: 1.06
COST OF DESIGN FOR CLASSES 2	: .76
COST OF DESIGN FOR CLASSES 1 , 2	: 1.11
COST OF DESIGN FOR CLASSES 3	: 1.87
COST OF DESIGN FOR CLASSES 1 , 3	: 2.04
COST OF DESIGN FOR CLASSES 2 , 3	: 1.9
COST OF DESIGN FOR CLASSES 1 , 2 , 3	: 2.06
COST OF DESIGN FOR CLASSES 4	: 1.18
COST OF DESIGN FOR CLASSES 1 , 4	: 1.46
COST OF DESIGN FOR CLASSES 2 , 4	: 1.24
COST OF DESIGN FOR CLASSES 1 , 2 , 4	: 1.51
COST OF DESIGN FOR CLASSES 3 , 4	: 2.105
COST OF DESIGN FOR CLASSES 1 , 3 , 4	: 2.22
COST OF DESIGN FOR CLASSES 2 , 3 , 4	: 2.13
COST OF DESIGN FOR CLASSES 1 , 2 , 3 , 4	: 2.24

VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 1 : 96.43
VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 2 : 1.18
VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 3 : 2.06
VEHICLE MILES OF TRAVEL FOR VEHICLE CLASS 4 : .33

COST ALLOCATED TO VEHICLE CLASS 1 : 1.03065

COST ALLOCATED TO VEHICLE CLASS 2 : .0393068

COST ALLOCATED TO VEHICLE CLASS 3 : .95709

COST ALLOCATED TO VEHICLE CLASS 4 : .212958

APPENDIX 6

GENERALIZED METHOD BASIC CODE

GENERALIZED METHOD BASIC CODE

FILE NAME: GM2.BAS

THIS ACCEPTS INPUT DATA AND CONSTRUCTS THE INITIAL MATRIX FOR THE LINEAR PROGRAMMING PROCEDURE INVOLVED IN THE GENERALIZED METHOD.

```
10  PRINT CHR$(26)
20 INPUT "Number of vehicle classes: ",NVC
21 LPRINT "Number of vehicle classes: ";NVC
30 PRINT CHR$(26)
40 NDIM=2©NVC-1
50 DIM INTABLE(NDIM+2,NVC+2)
60 FOR I=1 TO NDIM+2
70    FOR J=1 TO NVC+2
80        INTABLE(I,J)=0
90 NEXT J:NEXT I
95 LPRINT:LPRINT:LPRINT
100 FOR I=1 TO NDIM
110  JCOLUMN=1
120  INQUOT=I
130  PRINT"COST WHEN SYSTEM IS USED BY VEHICLE CLASSES";
131  LPRINT "COST WHEN SYSTEM IS USED BY VEHICLE CLASSES";
140  QUOT=INQUOT/2
150  INQUOT=INT(QUOT)
160  RES=(QUOT-INQUOT)*2
170  INTABLE(I,JCOLUMN)=RES
180  LABL=RES*JCOLUMN
190  IF LABL><0 THEN PRINT LABL;:LPRINT LABL;
200  WHILE INQUOT><0
210     IF LABL<>0 THEN PRINT ",";:LPRINT ",";
220     JCOLUMN=JCOLUMN+1
230     GOTO 140
240  WEND
250  PRINT TAB(60);:LPRINT TAB(60);
260  INPUT ": ", INTABLE(I,NVC+2)
261  LPRINT ": "; INTABLE(I,NVC+2)
270 NEXT I
280 FOR I=1 TO NDIM-1
290    INTABLE(I,NVC+1)=1
300 NEXT I
310 INTABLE(NDIM,NVC+1)=0
320 FOR J=1 TO NVC+2
330    INTABLE(NDIM+1,J)=-INTABLE(NDIM,J)
340 NEXT J
350 INTABLE(NDIM+2,NVC+1)=1
360 PRINT CHR$(26)
```

```
370 FOR I=1 TO NDIM+2
380    FOR J=1 TO NVC+2
390       IF J><NVC+2 THEN PRINT INTABLE(I,J);"   ";
                         ELSE PRINT INTABLE(I,J)
400 NEXT J:NEXT I
410 DIM TABLE(NDIM+2,NVC+2),IROW(NDIM+1),JCOL(NVC+1),
      R$(NDIM+1),C$(NVC+1)
420 NR=NDIM+1:NC=NVC+1:NI=NR+1:NJ=NC+1
430 CHAIN "SLP3",,ALL
440 END
```

FILE NAME: SLP.BAS

THIS FILE SOLVES THE LINEAR PROGRAMMING PROBLEM ASSOCIATED WITH THE GENERALIZED METHOD AND PERFORMS A TEST FOR MULTIPLE SOLUTIONS. THE LINEAR PROGRAMMING PROBLEM IS SOLVED THROUGH THE SO-CALLED SYMMETRIC METHOD.

```
30 X$="X":Y$="Y":TOL=.0001
40 GOSUB 1200
50 FOR I=1 TO NR:R$(I)=Y$:IROW(I)=I:NEXT I
60 FOR J=1 TO NC:C$(J)=X$:JCOL(J)=J:NEXT J
70 GOTO 90
80 GOTO 390
90 PIV=-1E+38
100 LL=1
110 MM=5
120 IF MM>NC THEN MM=NC
130 REM    PRINT TABLE HEADINGS.  COLUMN INDICATORS
140 PRINT:PRINT "                    -----------------    STEP
                 ";KNT;"  -----------------"
150 PRINT
160 PRINT TAB(9);
170 FOR J=LL TO MM
180     PRINT USING"    !";C$(J);
190     PRINT USING"##      ";JCOL(J);
200 NEXT J
210 IF MM=NC THEN PRINT "   R.H.S." ELSE PRINT
220 PRINT
230 MMM=MM+1
240 FOR I=1 TO NR
250     PRINT USING"  !";R$(I);
260     PRINT USING"##   ";IROW(I);
270     FOR J=LL TO MMM
280         PRINT USING" ##.###©©©© ";TABLE(I,J);
290     NEXT J:PRINT
300 NEXT I:PRINT
310 PRINT "  OBJ  ";
320 FOR J=LL TO MMM
330     PRINT USING" ##.###©©©© ";TABLE(NI,J);
340 NEXT J:PRINT:PRINT
350 LL=LL+5
360 IF LL>NC GOTO 390
370 MM=MM+5
380 GOTO 120
390 KNT=KNT+1
400 MM=0
410 REM    CHECK ROW INDICATORS
420 FOR I=1 TO NR
430 IF TABLE(I,NJ)>=0 GOTO 600
440 MM=1
450 CH=1E+38
460 FOR J=NEP TO NC
```

```
470 IF TABLE(I,J)>=0 GOTO 530
480 IF TABLE(NI,J)>0 GOTO 530
490 VALUE=TABLE(NI,J)/TABLE(I,J)
500 IF VALUE>=CH GOTO 530
510 CH=VALUE
520 JC=J
530 NEXT J
540 IF CH=1E+38 GOTO 600
550 VALUE=-TABLE(NI,JC)*TABLE(I,NJ)/TABLE(I,JC)
560 IF VALUE<=PIV GOTO 600
570 PIV=VALUE
580 II=I
590 JJ=JC
600 NEXT I
610 REM    CHECK COLUMN INDICATORS
620 FOR J=NEP TO NC
630 IF TABLE(NI,J)<=0 GOTO 800
640 MM=1
650 CH=1E+38
660 FOR I=1 TO NR
670        IF TABLE(I,J)<=0 GOTO 730
680        IF TABLE(I,NJ)<0 GOTO 730
690         VALUE=TABLE(I,NJ)/TABLE(I,J)
700        IF VALUE>=CH GOTO 730
710        CH=VALUE
720        IR=I
730     NEXT I
740     IF CH=1E+38 GOTO 800
750     VALUE=TABLE(NI,J)*TABLE(IR,NJ)/TABLE(IR,J)
760     IF VALUE<PIV GOTO 800
770     PIV=VALUE
780     II=IR
790     JJ=J
800 NEXT J
810 REM     OPTIMAL SOLUTION
820 IF MM=0 GOTO 1050
830 REM     INFEASIBLE SOLUTION
840 IF PIV=-1E+38 GOTO 1070
850 REM PERFORM INVERSION WITH INDICATED PIVOT ELEMENT
860 FOR I=1 TO NI
870     IF I=II GOTO 920
880     FOR J=1 TO NJ
890        IF J=JJ GOTO 910
900        TABLE(I,J)=TABLE(I,J)-
                      TABLE(II,J)*TABLE(I,JJ)/TABLE(II,JJ)
910     NEXT J
920 NEXT I
930 FOR J=1 TO NJ
940 IF J=JJ GOTO 960
950     TABLE(II,J)=TABLE(II,J)/TABLE(II,JJ)
960 NEXT J
970 FOR I=1 TO NI
```

```
980     IF I=II GOTO 1000
990      TABLE(I,JJ)=-TABLE(I,JJ)/TABLE(II,JJ)
1000 NEXT I
1010 TABLE(II,JJ)=1/TABLE(II,JJ)
1020 SWAP  IROW(II),JCOL(JJ)
1030 SWAP  R$(II),C$(JJ)
1040 GOTO 90
1050 PRINT "                         *****  OPTIMAL SOLUTION
             *****":PRINT
1060 GOTO 1080
1070 PRINT "                       *****  INFEASIBLE SOLUTION
             *****":PRINT:END
1080 FOR J=1 TO NC
1090      IF TABLE (NI,J)<>0 THEN GOTO 1130
1100       PRINT "                 ***** MULTIPLE SOLUTIONS
                 *****"
1110      INPUT "Press <RETURN> to continue",DUM$
1120      CHAIN "EFECT",,ALL
1130 NEXT J
1140 INPUT "Press <RETURN> to continue",DUM$
1150 CHAIN "RESULTS",,ALL
1200 FOR I=1 TO NI
1210   FOR J=1 TO NJ
1220       TABLE(I,J)=INTABLE(I,J)
1230   NEXT J
1240 NEXT I
1250 RETURN
```

FILE NAME: EFECT.BAS

THIS ROUTINE CALCULATES RELATIVE STATISTICAL COST EFFECTS FOR EACH OF THE VEHICLE CLASSES.

```
10 DIM EFFC(NVC),A1(NDIM+1,NVC),B(NDIM+1)
20 SUMEFC=0
30 INPUT "FRACTION OF THE TOTAL COST ATTRIBUTABLE TO THE
          ENVIRONMENT :", PENV
40 FOR JCL=1 TO NVC
50    EFFC(JCL)=0
60       FOR IRW=1 TO NDIM
70          IF INTABLE(IRW,JCL)=1 THEN EFFC(JCL)=EFFC(JCL)+
                                        INTABLE(IRW,NVC+2)
                                   ELSEEFFC(JCL)=EFFC(JCL)-
                                        INTABLE(IRW,NVC+2)
75          A1(IRW,JCL)=INTABLE(IRW,JCL)
80       NEXT IRW
85       A1(NDIM+1,JCL)=-A1(NDIM,JCL)
90       EFFC(JCL)=(EFFC(JCL)-PENV*INTABLE (NDIM,NVC+2))/2©(NVC-1)
100      SUMEFC=SUMEFC+EFFC(JCL)
110 NEXT JCL
120 FOR JCL=1 TO NVC
130    EFFC(JCL)=EFFC(JCL)/SUMEFC
140    PRINT EFFC(JCL),EFFC(JCL)*SUMEFC
150 NEXT JCL
160 FOR I=1 TO NDIM+1
170    B(I)=INTABLE(I,NVC+2)
180 NEXT I
190 FOR I=1 TO NR
200    IF IROW(I)=NVC+1 THEN IF R$(I)="X" THEN T=TABLE(I,NJ)
                                               : GOTO 250
210 NEXT I
220 T=0 : PRINT "      ***** NO BASIC SOLUTION FOR T *****"
250 CHAIN "NEWMAT",,ALL
260 END
```

FILE NAME: NEWMAT.BAS

AN INITIAL LINEAR PROGRAMMING MATRIX FOR THE SECOND PHASE OF THE GENERALIZED METHOD IS GENERATED IN THIS ROUTINE. THIS ROUTINE IS EXECUTED ONLY WHEN THE FIRST PHASE YIELDS MULTIPLE SOLUTIONS.

```
5   ERASE INTABLE,TABLE,R$,C$,IROW,JCOL
10 DIM INTABLE (2©NVC+2*NVC+1,3*NVC+1)
20 FOR IRW=1 TO 2©NVC
30     FOR JCL=1 TO NVC
40         INTABLE (IRW,JCL)=A1(IRW,JCL)
45     NEXT JCL
50     IF IRW<2©NVC-1 THEN INTABLE (IRW,3*NVC+1)=B(IRW)-T
                      ELSE INTABLE (IRW,3*NVC+1)=B(IRW)
60 NEXT IRW
100 FOR JCL=1 TO NVC
110     IRW=2*JCL-1
120     IX=2©NVC+IRW
130    JX=NVC+IRW
140     INTABLE(IX,JCL)=1/B(2©NVC-1)
150    INTABLE(IX+1,JCL)=-1/B(2©NVC-1)
160     INTABLE(IX,JX)=1
170    INTABLE(IX+1,JX)=-1
180     INTABLE(IX,JX+1)=-1
190    INTABLE(IX+1,JX+1)=1
200     INTABLE(IX,3*NVC+1)=EFFC(JCL)
210    INTABLE(IX+1,3*NVC+1)=-EFFC(JCL)
220 NEXT JCL
230 FOR J=NVC+1 TO 3*NVC
240     INTABLE(2©NVC+2*NVC+1,J)=-1
250 NEXT J
260 FOR I=1 TO 2©NVC+2*NVC+1
270     FOR J=1 TO 3*NVC+1
280         IF J<>3*NVC+1 THEN PRINT INTABLE(I,J);"   ";
                          ELSE PRINT INTABLE(I,J)
290 NEXT J,I
300 DIM TABLE(2©NVC+2*NVC+1,3*NVC+1),IROW(2©NVC+2*NVC+1),
        JCOL(3*NVC+1)
310 DIM R$(2©NVC+2*NVC+1),C$(3*NVC+1)
320 NR=2©NVC+2*NVC : NC=3*NVC
330 NI=NR+1 : NJ=NC+1
340 CHAIN "SLP2",,ALL
```

FILE NAME: RESULTS.BAS

FINAL RESULTS OF THE GENERALIZED METHOD ARE OUTPUT USING THIS PROGRAM.

```
10 DIM ALLOC(NVC)
20 REM  THIS PROGRAM PRINTS THE RESULTS OF THE
30 REM  GENERALIZED COST ALLOCATION METHOD
40 '
50 '   LOOK FOR BASIC VARIABLES
60 '
70 FOR I=1 TO NR
80     IF R$(I)<>"X" THEN GOTO 110
90        IF IROW(I)>NVC THEN GOTO 110
100           ALLOC(IROW(I))=TABLE(I,NJ)
110 NEXT I
120 '
130 '   PRINT COST ALLOCATIONS
140 '
145 LPRINT:LPRINT:LPRINT
150 PRINT CHR$(26):PRINT:PRINT:PRINT
160 FOR J=1 TO NVC
170    PRINT:LPRINT
180    PRINT "COST ALLOCATED TO VEHICLE CLASS";J;":";
            ALLOC(J)
181    LPRINT "COST ALLOCATED TO VEHICLE CLASS";J;":";
             ALLOC(J)
190 NEXT J
200 END
```

APPENDIX 7

SAMPLE RUN FOR THE GENERALIZED METHOD

Number of vehicle classes: 4

COST FOR VEHICLE CLASSES 1	: 1.06
COST FOR VEHICLE CLASSES 2	: .76
COST FOR VEHICLE CLASSES 1 , 2	: 1.11
COST FOR VEHICLE CLASSES 3	: 1.87
COST FOR VEHICLE CLASSES 1 , 3	: 2.04
COST FOR VEHICLE CLASSES 2 , 3	: 1.9
COST FOR VEHICLE CLASSES 1 , 2 , 3	: 2.06
COST FOR VEHICLE CLASSES 4	: 1.18
COST FOR VEHICLE CLASSES 1 , 4	: 1.46
COST FOR VEHICLE CLASSES 2 , 4	: 1.24
COST FOR VEHICLE CLASSES 1 , 2 , 4	: 1.51
COST FOR VEHICLE CLASSES 3 , 4	: 2.105
COST FOR VEHICLE CLASSES 1 , 3 , 4	: 2.22
COST FOR VEHICLE CLASSES 2 , 3 , 4	: 2.13
COST FOR VEHICLE CLASSES 1 , 2 , 3 , 4	: 2.24

COST ALLOCATED TO VEHICLE CLASS 1 : .41

COST ALLOCATED TO VEHICLE CLASS 2 : .32

COST ALLOCATED TO VEHICLE CLASS 3 : 1.03

COST ALLOCATED TO VEHICLE CLASS 4 : .48