CORE
Metadata, citation and similar papers at core.ac.uk
Provided by University of Kentucky

Research Report
KTC-93-7

ENHANCEMENTS TO PROCEDURE FOR ESTIMATING ESALS

by

John A. Deacon
Professor

Jerry G. Pigman
Research Engineer

Neil W. Tollner
Senior Scientific Programmer Analyst

and

David H. Cain
Engineering Technician II

Kentucky Transportation Center
College of Engineering
University of Kentucky
Lexington, Kentucky

in cooperation with

Kentucky Transportation Cabinet
Commonwealth of Kentucky

The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the University of Kentucky or the Kentucky Transportation Cabinet. This report does not constitute a standard, specification, or regulation. The inclusion of manufacturer names and trade names is for identification purposes and is not to be considered an endorsement.

February 1993

INTRODUCTION

Kentucky's current procedure for estimating equivalent single axle loads (ESALs) was updated in 1990 to incorporate traffic data categorized by functional class rather than by statewide averages. This change resulted from the influx of data generated by automatic equipment used to classify and weigh vehicles in motion. A much wider range of geographic conditions and road conditions was sampled, and the expectation was that the data used in the estimation procedure would be more accurate as a result.

The 1990 revisions were quite extensive, and, although model calibrations using 1989 and 1990 data have proven useful, the Division of Planning has sought refinements which would improve the calibrations and make them more useful for specific needs. In addition, the continuing analysis of weight and classification data for coal trucks has identified potential improvements that could produce more representative data for these types of vehicles.

In response to these needs, a research study was proposed for FY 1992 with funds totaling $25,000 (Appendix A). Funding limitations delayed the study until FY 1993 and reduced the amount to $15,000. Although there was some decrease in scope, the study has been successful in 1) enhancing the accuracy of the calibration process; 2) improving the appearance, clarity, and utility of the output; and 3) potentially reducing year-to-year variations in the estimation of key quantities. In addition, a possible revision has been outlined which would eventually offer other benefits including a more definitive and accurate method for reflecting effects of coal movement.

The purpose of this report is to explain and document the progress that has been made toward improving the ESAL-estimation process and exploiting the wealth of data being generated by the new vehicle classification and weighing program.

WIM-TO-STATIC CONVERSION

Kentucky's load equivalency factors, on which ESAL-computations depend, are based on static truck weight measurements. Because of the dynamics of vehicles in motion, it is necessary to convert weigh-in-motion (WIM) measurements to their static equivalents. Factors for such conversions were developed in earlier work (1), and computer code was prepared for preliminary processing of all Kentucky WIM data. Incorporated in the code was the capability for classifying trucks into two "commodity" categories, normal and heavy/coal.

During preliminary phases of the current model-calibration process, the algorithm for commodity classification was found to be erroneous. Apparently, changes made following the 1990 model calibration had not been properly debugged. As a result one of the first of the current revisions was to correct this portion of the WIM computer code.

The second revision was to add a routine for validity checking. In the original ESAL programs, validity checking of the weight data was a part of the LOADMTR program. Because programs, validity checking of the weight data was a part of the LOADMTR program. Because the WIM program precedes the LOADMTR program in the calibration sequence and because WIM program output is used directly for other purposes, it was desirable to move validity checking routines to the WIM program[1]. Additional validity checks were added in response to specific error types observed in a few of the WIM data records. The following summarizes validity checks which are being applied to each WIM record:

- The gross vehicle weight must be within 10 percent of the sum of the individual axle weights;
- The wheelbase must be within 10 percent of the sum of the individual axle spacings;
- The number of individual axle weights must be exactly one more than the number of axle spacings;
- Each individual axle weight must be greater than zero;
- Each axle spacing must be greater than 1.5 feet; and
- Gross weight and wheelbase limits, identical to those used in earlier years, must be within the following limits:

[1]For convenience several of the validity checks were eventually added to the UNITEAL program instead of the WIM program.

Vehicle Type	Gross Weight (Kips)		Wheelbase (Feet)	
	Lower	Upper	Lower	Upper[2]
4	2.0	66	2.0	40
5	2.5	82	3.2	46
6	3.0	99	4.3	52
7	3.5	115	5.5	58
8	4.0	132	6.6	64
9	4.5	149	7.8	70
10	5.0	165	8.9	76
11	5.5	180	10.1	82
12	6.0	180	11.2	88
13	6.5	180	12.4	90

Weight records identified as erroneous by the above criteria are purged from the data set, and a summary of the extent of "erroneous" data is provided so that the integrity of the entire data set can be quickly ascertained.

Summary statistics resulting from the application of the validity checking process to 1989, 1990, and 1991 weight data are presented in Tables 1-3, respectively. In general, it appears that the databases are quite good and that the edit specifications are quite reasonable. At the same time, the number of vehicles with excessive wheelbases seems to be too large. A detailed examination of a sample of these records reveals that a relatively small relaxation in the upper wheelbase limit would greatly reduce the number of these rejections (Figure 1). As a result, each upper limit will be increased by 6 feet for future calibrations. In addition to the large number of rejections due to restrictive wheelbase limits, a very large number of trucks were classified as "921" vehicles. Detailed examination of this matter has resulted in modifications to the computer code which will significantly reduce the number of vehicles that are coded this way in the future.

Appendix B contains a Fortran source listing of the revised WIM program.

---

[2]As noted in the text, each upper wheelbase limit will be increased for future work by 6 feet. The 90-foot cap will be increased to 100 feet.

Table 1. Summary of Rejected Weight Records in 1989

Description	Functional Classification									
	1	2	6	7	8	11	12	14	16	17
WIM PROGRAM										
Axles without recorded weight	0	0		0		0		0		
Axle spacing less than 1.5 feet	0	0		0		1		3		
Mismatch between numbers of axle weights and spacings	0	1		0		2		0		
More than 10 percent difference between gross weight and sum of axle weights	0	0		0		0		0		
More than 10 percent difference between wheelbase and sum of axle spacings	10	5		0		2		1		
UNITEAL PROGRAM										
Excessive gross weight	0	0		0		0		0		
Negligible gross weight	0	0		0		0		0		
Excessive wheelbase	99	10		0		13		4		
Negligible wheelbase	0	0		0		0		0		
Vehicle type "921"	202	91		19		94		185		
TOTAL NUMBER OF REJECTIONS	311	107		19		112		193		
TOTAL NUMBER OF RECORDS	18,744	3,652		777		9,899		4,868		
PERCENT REJECTION	1.66	2.93		2.44		1.13		3.96		

Table 2. Summary of Rejected Weight Records in 1990

Description	Functional Classification									
	1	2	6	7	8	11	12	14	16	17
WIM PROGRAM										
Axles without recorded weight	0	0	0	0		0	0	0	0	
Axle spacing less than 1.5 feet	0	0	0	0		0	0	0	0	
Mismatch between numbers of axle weights and spacings	7	0	0	0		0	0	0	0	
More than 10 percent difference between gross weight and sum of axle weights	0	0	0	0		0	0	0	0	
More than 10 percent difference between wheelbase and sum of axle spacings	0	0	0	0		0	0	0	0	
UNITEAL PROGRAM										
Excessive gross weight	0	0	0	0		0	0	0	0	
Negligible gross weight	0	0	0	0		0	0	0	0	
Excessive wheelbase	174	20	6	0		4	6	7	0	
Negligible wheelbase	0	0	0	0		0	0	0	0	
Vehicle type "921"	644	106	53	79		38	54	67	85	
TOTAL NUMBER OF REJECTIONS	825	126	59	79		42	60	74	85	
TOTAL NUMBER OF RECORDS	40,912	7,525	1,591	2,198		4,421	3,189	6,856	3,577	
PERCENT REJECTION	2.02	1.67	3.71	3.59		0.95	1.88	1.08	2.38	

Table 3. Summary of Rejected Weight Records in 1991

Description	Functional Classification									
	1	2	6	7	8	11	12	14	16	17
WIM PROGRAM										
Axles without recorded weight	35	0	0	58	0	0	0	0	0	0
Axle spacing less than 1.5 feet	0	0	0	0	0	0	0	0	0	0
Mismatch between numbers of axle weights and spacings	14	0	0	72	0	0	0	0	0	0
More than 10 percent difference between gross weight and sum of axle weights	0	0	0	0	0	0	0	0	0	0
More than 10 percent difference between wheelbase and sum of axle spacings	0	0	0	0	0	0	0	0	0	0
UNITEAL PROGRAM										
Excessive gross weight	0	3	0	3	0	0	0	2	0	0
Negligible gross weight	0	0	0	0	0	0	0	0	0	0
Excessive wheelbase	170	10	2	0	0	14	47	5	0	0
Negligible wheelbase	0	0	0	0	0	0	0	0	0	0
Vehicle type "921"	865	162	37	28	11	181	212	82	10	4
TOTAL NUMBER OF REJECTIONS	1,084	175	39	161	11	195	259	89	10	4
TOTAL NUMBER OF RECORDS	77,305	10,922	689	1,032	296	13,272	9,057	4,368	905	115
PERCENT REJECTION	1.40	1.60	5.67	15.60	3.72	1.47	2.86	2.04	1.10	3.48

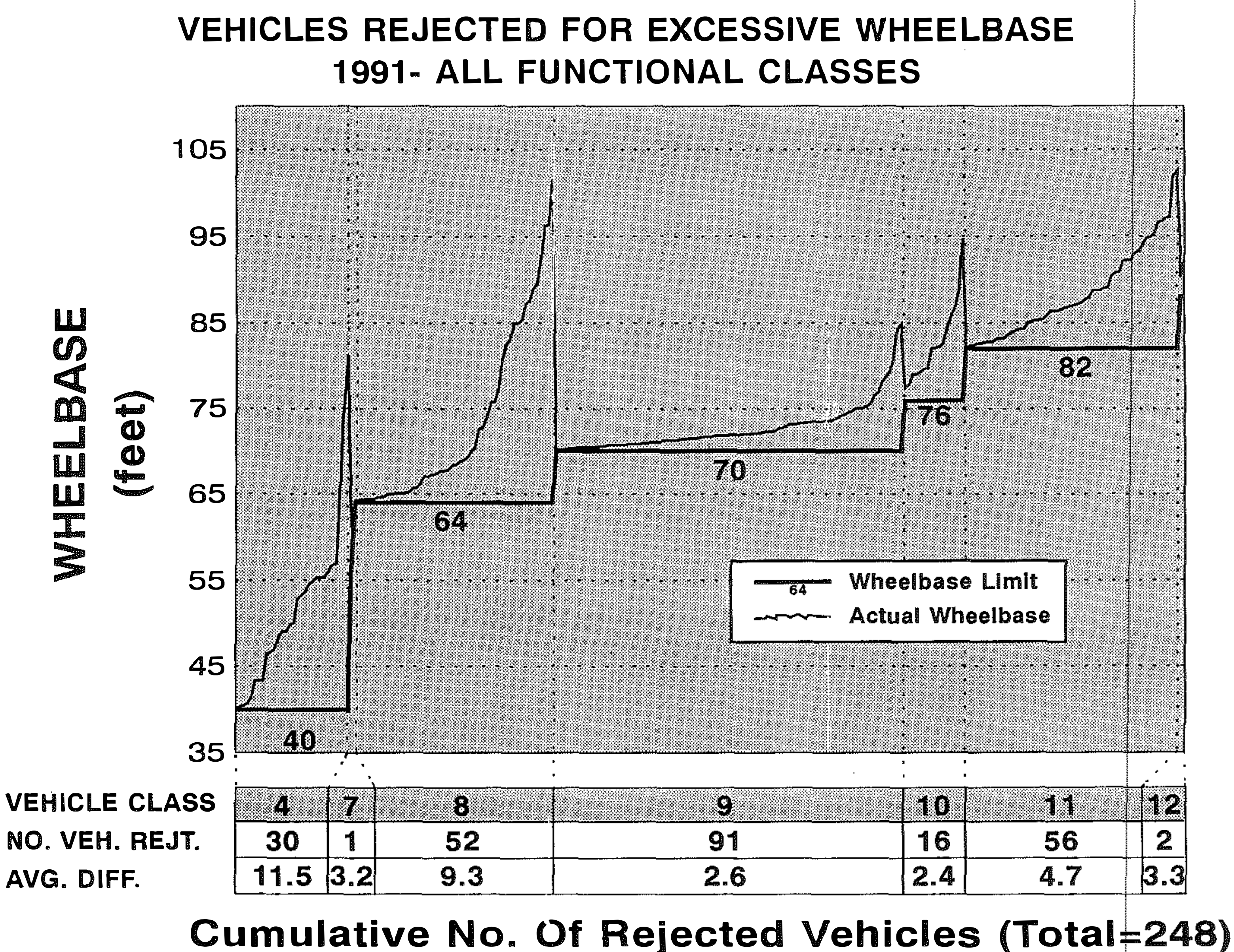

Figure 1. Frequency Distribution of Vehicles Exceeding Upper Wheelbase Limits

MODIFICATION OF EALCALC

The main program for model calibration is that named EALCALC. EALCALC merges weight data (previously processed by LOADMTR) with classification data (previously processed by CLASSUM) to produce the primary ESAL-model parameters (AADT, percent trucks, axles per truck, ESALs per truck axle, etc.).

During the 1990 model calibration last year, what should have been identical runs of EALCALC failed to produce duplicate output. After considerable searching, the error could not be definitively traced to its source. EALCALC is a rather large, complex program which has been continuously modified since its inception in response to new data sources and analysis requirements. As an expedient, it was decided to compute the unit ESALs, that is, the ESALs per truck axle, externally and to input them directly into EALCALC rather than to continue to try to locate and correct the internal error. As the code to compute the unit ESALs had been developed for other purposes, this correction required fairly small additional effort during the 1990 calibration.

The unit ESALs incorporated into the 1990 calibration were classified by commodity type and by functional classification but not by vehicle type. This departure from past calibrations yielded constant ESALs per axle for all classification sites within a given functional classification. Because this was neither necessary nor desirable, a change was made as part of the current study to reflect real differences in ESALs per axle as a function not only of commodity type and functional class but also vehicle type. The anticipated effects of this change include an increase in accuracy and the restoration of past conventions.

COMPUTATION OF UNIT ESALS

The ESALs for a given axle are very sensitive to the axle load. For example, the well known "fourth power law" suggests that the number of ESALs attributed to a given axle is approximately proportional to the fourth power of its weight. As a result, a handful of very heavily laden vehicles can have order-of-magnitude effects on the ESAL-loading of the traffic stream. To illustrate, 437 2-axle trucks were weighed at one 1991 site. The ESALs per axle averaged an enormous 8.09. By removing the single most damaging truck, however, the average dropped to 0.42 ESALs per axle. Recorded loads on this one truck included 9,400 pounds on the steering axle and 74,500 pounds on the remaining single axle: these resulted in an ESAL estimate totaling 6,700 for this one vehicle.

Although the new validity specifications for weight data attenuate such effects, additional care was required to assure that computations of damage factors for individual axles are not a result of extrapolations much beyond the range of historical data. The EALCALC program had limited the damage factors for very heavy loads--thus avoiding extreme estimates that may result from erroneous or spurious data. This practice was continued by modifying the UNITEAL program to incorporate the following specification:

Type of Axle	For Loads In Excess Of	Use Following Damage Factor
Steering	22,500 lb.	5.39
Other Single	37,500 lb.	78.6
Tandem	75,000 lb.	63.0
Tridem	112,500 lb.	59.0
Quad	150,000 lb.	58.3

These limits are the same that have been used in past years.

Appendix C contains a Fortran source listing of the revised UNITEAL program.

REVISED OUTPUT

Hard-copy output from the EALCALC program and, especially, the SMOOTH program is used extensively in site-specific forecasts of ESAL accumulations. As a result of improvements sought by the Division of Planning, significant revisions were made to the hard-copy output, and an experimental dBASE display was developed.

Driving changes to the hard-copy output was the desire to make the output not only more attractive but also more comprehensible. This was accomplished largely by eliminating unnecessary data, by reducing the number of years of displayed data to ten, and by careful attention to spacing and wording. In addition, historic data identifying the numbers of both classification and weigh stations[3] were added to assist the analyst in evaluating the reliability of the measurements. The most recent 3-year averages that had been included in the EALCALC output were more carefully distinguished from the historical data, and the most recent single-year data were added. To assist the analyst in identifying abnormal growth estimates, entries in the SMOOTH output were tagged with ? ... ? brackets when incremental percentages were negative or when they exceeded 5 percent. The new formats are illustrated in Figures 2 and 3.

In response to a Division of Planning request to display the historic data trends graphically, an experimental dBASE routine was developed which provides computer access to the EALCALC and SMOOTH output and creates the desired graphical display. The 1991 output has been provided in this format (in addition to the traditional hard-copy output), and, as

[3]The annual number of weigh stations was not conveniently accessible by computer. To avoid reprocessing pre-1990 data, the number of weigh stations has been shown only for 1990 and 1991. As data for future years is processed, this number will automatically be added to the historical file.

SUMMARY OF AVERAGE VALUES FOR
FUNCTIONAL CLASS 01 -- RURAL INTERSTATE

YEAR	3 YR AVG	91	90	89	88	87	86	85	84	83	82
NUMBER OF WEIGH STA.		7	4								
UNCLASSIFIED ROADS (ALL AVC LOCATIONS AND MANUAL LOCATIONS WITH LESS THAN 3% OF TRUCKS CLASSIFIED AS HEAVY/COAL)											
NO OF CLASSIFICATION STA.	46	13	23	10	9	14	10	14	9	8	14
AADT	22103	23609	21017	22643	20785	17661	16636	20837	19585	19132	18507
PERCENT TRUCKS	27.551	25.801	27.057	30.960	27.468	28.858	30.465	29.759	26.786	26.513	25.158
AXLES PER TRUCK	4.456	4.384	4.476	4.501	4.467	4.488	4.504	4.489	4.418	4.423	4.348
EAL'S PER TRUCK AXLE	0.177	0.189	0.186	0.142	0.159	0.166	0.167	0.157	0.154	0.136	0.135
CLASSIFIED ROADS (MANUAL LOCATION WITH 3% OR MORE OF TRUCKS CLASSIFIED AS HEAVY/COAL)											
NO OF CLASSIFICATION STA.	1	0	0	1	1	1	3	2	1	1	2
AADT	12213	0	0	12213	18903	26386	20497	22128	23310	22627	21573
PERCENT TRUCKS	28.808	0.000	0.000	28.808	24.018	23.962	24.408	22.534	23.085	24.955	24.270
PERCENT OF TRUCKS CLASSIFIED AS HEAVY/COAL	6.443	0.000	0.000	6.443	3.583	4.209	11.007	4.467	3.220	3.722	5.228
AXLES PER TRUCK NORMAL	4.432	0.000	0.000	4.432	4.311	4.404	4.247	4.249	4.313	4.376	4.365
AXLES PER TRUCK HEAVY/COAL	5.008	0.000	0.000	5.008	5.288	5.301	5.076	5.133	5.246	5.254	5.074
EAL'S PER TRUCK AXLE NORMAL	0.138	0.000	0.000	0.138	0.157	0.169	0.162	0.153	0.155	0.136	0.137
EAL'S PER TRUCK AXLE HEAVY/COAL	0.258	0.000	0.000	0.258	0.559	0.257	0.271	0.240	0.222	0.221	0.242

Figure 2. Illustrative Hard-Copy Output from EALCALC

FUNCTIONAL CLASS 01 -- RURAL INTERSTATE
AVERAGE VALUES (SMOOTHED)

YEAR	ANNUAL CHANGE (%)	91	90	89	88	87	86	85	84	83	82
NO. OF WEIGH STA.		7	4								
UNCLASSIFIED ROADS (ALL AVC LOCATIONS AND MANUAL LOCATIONS WITH LESS THAN 3% OF TRUCKS CLASSIFIED AS HEAVY/COAL)											
NO. OF CLASSIFICATION STA.		13	23	10	9	14	10	14	9	8	14
AADT	2.288	22304	21794	21284	20773	20263	19753	19243	18732	18222	17712
PERCENT TRUCKS	0.114	28.088	28.056	28.024	27.992	27.960	27.928	27.896	27.864	27.832	27.800
AXLES PER TRUCK	0.074	4.466	4.463	4.460	4.457	4.453	4.450	4.447	4.443	4.440	4.437
EAL'S PER TRUCK AXLE	2.635	0.180	0.176	0.171	0.166	0.161	0.157	0.152	0.147	0.142	0.138
CLASSIFIED ROADS (MANUAL LOCATION WITH 3% OR MORE OF TRUCKS CLASSIFIED AS HEAVY/COAL)											
NO. OF CLASSIFICATION STA.		0	0	1	1	1	3	2	1	1	2
AADT	? -7.133 ?	15099	16177	17254	18331	19408	20485	21562	22640	23717	24794
PERCENT TRUCKS	1.718	27.020	26.556	26.092	25.628	25.164	24.700	24.236	23.772	23.308	22.844
PERCENT OF TRUCKS CLASSIFIED AS HEAVY/COAL	2.957	6.262	6.077	5.892	5.707	5.521	5.336	5.151	4.966	4.781	4.596
AXLES PER TRUCK NORMAL	0.186	4.381	4.373	4.364	4.356	4.348	4.340	4.332	4.324	4.315	4.307
AXLES PER TRUCK HEAVY/COAL	? -0.103 ?	5.145	5.150	5.155	5.161	5.166	5.171	5.176	5.182	5.187	5.192
EAL'S PER TRUCK AXLE NORMAL	0.817	0.158	0.157	0.156	0.154	0.153	0.152	0.151	0.149	0.148	0.147
EAL'S PER TRUCK AXLE HEAVY/COAL	? 5.717 ?	0.414	0.390	0.366	0.343	0.319	0.295	0.272	0.248	0.224	0.201

Figure 3. Illustrative Hard-Copy Output from SMOOTH

experience develops, an evaluation of its utility and, as appropriate, an identification of possible future enhancements can be made.

REVISED SMOOTHING ROUTINE

One of the criticisms of the ESAL modeling has been that incremental percentages, used in forecasting future ESAL accumulations, display excessive year-to-year variation. Such variation is thought to be largely due to sampling error (insufficient sample size) and, particularly in the last two years, to the transition to the new automated data collection program. Nevertheless, the variation can be significantly reduced by altering the nature of the regression fit that produces the incremental percentages. Past procedures have employed a weighted, least-squares process utilizing 15 years of data. Weights have been proportional to the number of stations contributing to the estimate and have emphasized recent data by assigning a weight of 15 to current-year data, 14 to the prior year data, etc.

The weighted regression routine in the SMOOTH program was modified to produce more stable estimates of incremental percentages. Only the most recent 10 years of data were used with weights ranging linearly from 2 for the present year to 1 for data collected 10 years earlier. No weighting was given to the number of stations. The net effect of this change is expected to be more stable estimates of the incremental percentages. Unfortunately these revised estimates will not respond as quickly to real changes in the character of the traffic stream as did past estimates. As experience with the new data collection program develops, it would seem prudent to reexamine the nature of the weighted regression.

To demonstrate the stability of the incremental percentages, a comparison was made between increments developed by a 1990 recalibration and by the new 1991 calibration. Complete results are summarized in the following table. Figures 4-11 provide example graphical comparisons of traffic parameter trends for unclassified roads on the rural interstate and principal arterial systems. Figure 12 indicates that more than half of the differences between the 111 1990 and 1991 estimates are less than 1 percent in magnitude. Thus, while there is generally good agreement between 1990 and 1991 estimates, the revisions that have been made do not eliminate the necessity for informed professional judgement when applying the ESAL model for traffic forecasts and evaluations. Although it's not clear that further change in the smoothing routine is desirable at this time, certainly additional comparisons must be made when 1992 data are processed.

Functional Class	Highway Type	ESAL Parameter	Growth Percentage	
			1990	1991
1	Normal	AADT	2.5	2.3
		Percent Trucks	1.1	0.1
		Axles per Truck	0.3	0.1
		ESALs per Truck Axle	2.0	2.6
	Coal	AADT	-2.8	-7.1
		Percent Trucks	1.8	1.7
		Percent Heavy/Coal Trucks	-0.1	3.0
		Axles per Normal Truck	0.3	0.2
		Axles per Heavy/Coal Truck	0.3	-0.1
		ESALs per Normal Truck Axle	0.8	0.8
		ESALs per Heavy/Coal Truck Axle	5.2	5.7
2	Normal	AADT	2.8	4.5
		Percent Trucks	2.4	1.8
		Axles per Truck	1.4	1.3
		ESALs per Truck Axle	-5.5	-1.5
	Coal	AADT	0.9	0.2
		Percent Trucks	-1.5	-1.7
		Percent Heavy/Coal Trucks	-4.3	-4.0
		Axles per Normal Truck	0.0	-0.3
		Axles per Heavy/Coal Truck	2.4	2.4
		ESALs per Normal Truck Axle	-2.8	-0.5
		ESALs per Heavy/Coal Truck Axle	-13.3	-19.2
6	Normal	AADT	0.0	1.0
		Percent Trucks	-3.6	-2.6
		Axles per Truck	0.3	0.4
		ESALs per Truck Axle	4.8	2.9
	Coal	AADT	3.4	6.6
		Percent Trucks	-4.7	-4.7
		Percent Heavy/Coal Trucks	-1.7	2.0
		Axles per Normal Truck	-0.1	-0.3
		Axles per Heavy/Coal Truck	0.5	0.9
		ESALs per Normal Truck Axle	6.0	4.7
		ESALs per Heavy/Coal Truck Axle	5.7	7.8

Functional Class	Highway Type	ESAL Parameter	Growth Percentage	
			1990	1991
7	Normal	AADT	-0.4	0.1
		Percent Trucks	-0.6	-0.8
		Axles per Truck	0.7	0.4
		ESALs per Truck Axle	5.6	7.5
	Coal	AADT	1.5	1.4
		Percent Trucks	-0.2	0.4
		Percent Heavy/Coal Trucks	-2.0	-1.3
		Axles per Normal Truck	0.8	0.5
		Axles per Heavy/Coal Truck	0.9	0.9
		ESALs per Normal Truck Axle	6.7	8.7
		ESALs per Heavy/Coal Truck Axle	8.3	8.1
8	Normal	AADT	-3.9	0.0
		Percent Trucks	-2.8	-1.2
		Axles per Truck	0.9	0.7
		ESALs per Truck Axle	-2.9	-7.2
	Coal	AADT	0.1	0.0
		Percent Trucks	0.9	0.4
		Percent Heavy/Coal Trucks	2.8	1.3
		Axles per Normal Truck	2.6	1.7
		Axles per Heavy/Coal Truck	2.2	1.6
		ESALs per Normal Truck Axle	-3.2	-7.0
		ESALs per Heavy/Coal Truck Axle	3.7	-1.7
9	Normal	AADT	-5.2	-0.9
		Percent Trucks	-31.0	-4.2
		Axles per Truck	2.8	1.6
		ESALs per Truck Axle	3.9	4.0
	Coal	AADT	---	---
		Percent Trucks	---	---
		Percent Heavy/Coal Trucks	---	---
		Axles per Normal Truck	---	---
		Axles per Heavy/Coal Truck	---	---
		ESALs per Normal Truck Axle	---	---
		ESALs per Heavy/Coal Truck Axle	---	---

Functional Class	Highway Type	ESAL Parameter	Growth Percentage	
			1990	1991
11	Normal	AADT	1.6	0.9
		Percent Trucks	1.4	2.0
		Axles per Truck	0.5	0.4
		ESALs per Truck Axle	-0.3	0.5
	Coal	AADT	---	---
		Percent Trucks	---	---
		Percent Heavy/Coal Trucks	---	---
		Axles per Normal Truck	---	---
		Axles per Heavy/Coal Truck	---	---
		ESALs per Normal Truck Axle	---	---
		ESALs per Heavy/Coal Truck Axle	---	---
12	Normal	AADT	6.6	10.5
		Percent Trucks	5.3	-0.6
		Axles per Truck	2.3	1.1
		ESALs per Truck Axle	1.0	2.1
	Coal	AADT	3.2	2.7
		Percent Trucks	-5.2	-5.5
		Percent Heavy/Coal Trucks	-9.7	-9.4
		Axles per Normal Truck	0.6	0.7
		Axles per Heavy/Coal Truck	0.2	0.3
		ESALs per Normal Truck Axle	0.5	1.1
		ESALs per Heavy/Coal Truck Axle	-2.4	-1.2
14	Normal	AADT	7.1	5.8
		Percent Trucks	-1.9	1.0
		Axles per Truck	0.6	0.6
		ESALs per Truck Axle	4.6	5.2
	Coal	AADT	3.1	1.6
		Percent Trucks	1.5	1.5
		Percent Heavy/Coal Trucks	-4.5	-6.0
		Axles per Normal Truck	2.1	0.7
		Axles per Heavy/Coal Truck	1.6	1.3
		ESALs per Normal Truck Axle	2.2	3.6
		ESALs per Heavy/Coal Truck Axle	6.0	2.3

Functional Class	Highway Type	ESAL Parameter	Growth Percentage	
			1990	1991
16	Normal	AADT	4.2	4.4
		Percent Trucks	-1.4	-0.6
		Axles per Truck	0.3	0.0
		ESALs per Truck Axle	2.8	7.3
	Coal	AADT	-1.6	-1.5
		Percent Trucks	2.3	-1.3
		Percent Heavy/Coal Trucks	-14.6	-24.4
		Axles per Normal Truck	1.3	-1.2
		Axles per Heavy/Coal Truck	3.7	3.6
		ESALs per Normal Truck Axle	6.0	9.8
		ESALs per Heavy/Coal Truck Axle	-3.6	-4.4
17	Normal	AADT	5.9	6.9
		Percent Trucks	2.8	3.9
		Axles per Truck	0.1	0.5
		ESALs per Truck Axle	4.8	6.0
	Coal	AADT	3.8	-6.4
		Percent Trucks	2.5	12.0
		Percent Heavy/Coal Trucks	6.1	-21.2
		Axles per Normal Truck	3.9	3.3
		Axles per Heavy/Coal Truck	1.8	-3.9
		ESALs per Normal Truck Axle	3.4	5.1
		ESALs per Heavy/Coal Truck Axle	9.0	9.5
19	Normal	AADT	11.6	13.6
		Percent Trucks	8.6	5.1
		Axles per Truck	2.2	3.3
		ESALs per Truck Axle	2.8	-1.3
	Coal	AADT	---	---
		Percent Trucks	---	---
		Percent Heavy/Coal Trucks	---	---
		Axles per Normal Truck	---	---
		Axles per Heavy/Coal Truck	---	---
		ESALs per Normal Truck Axle	---	---
		ESALs per Heavy/Coal Truck Axle	---	---

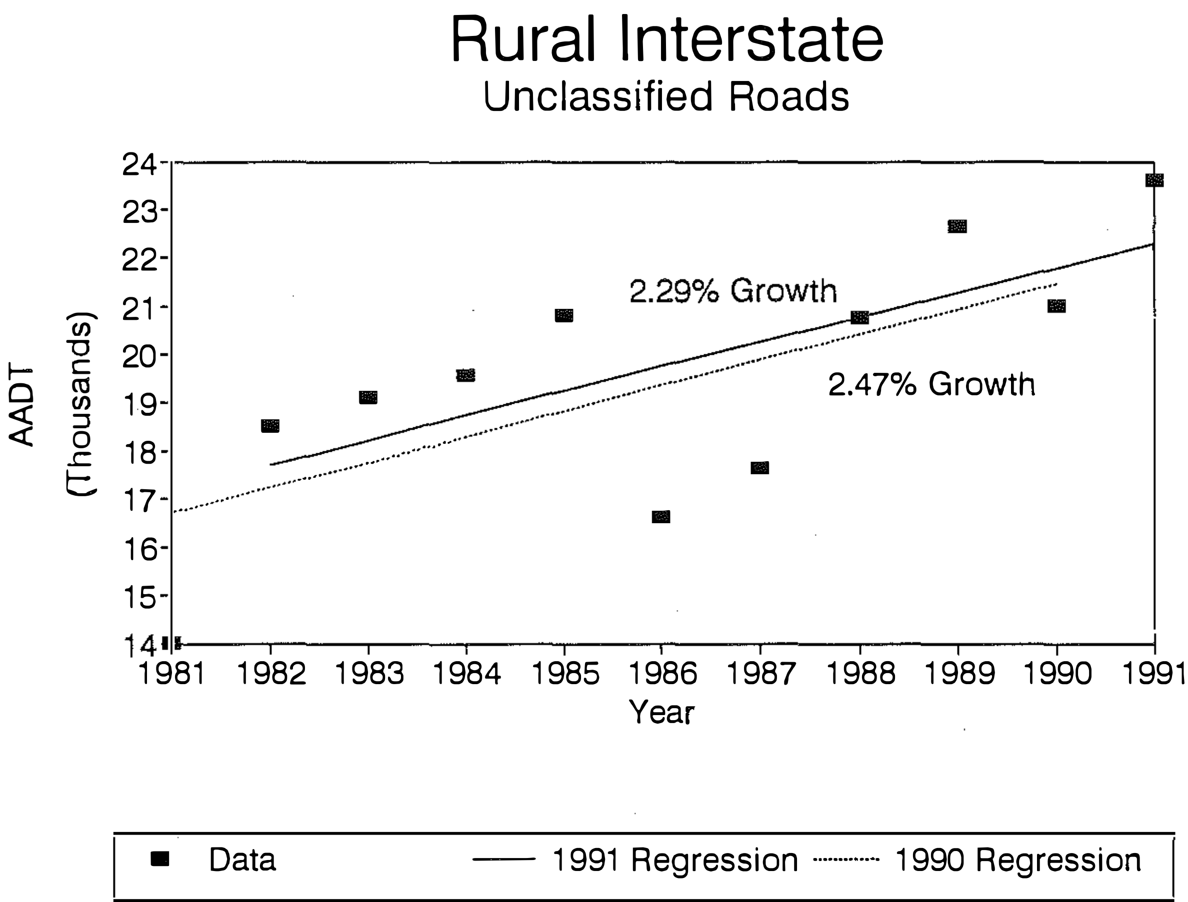

Figure 4. Growth of AADT on Unclassified Roads of the Rural Interstate System

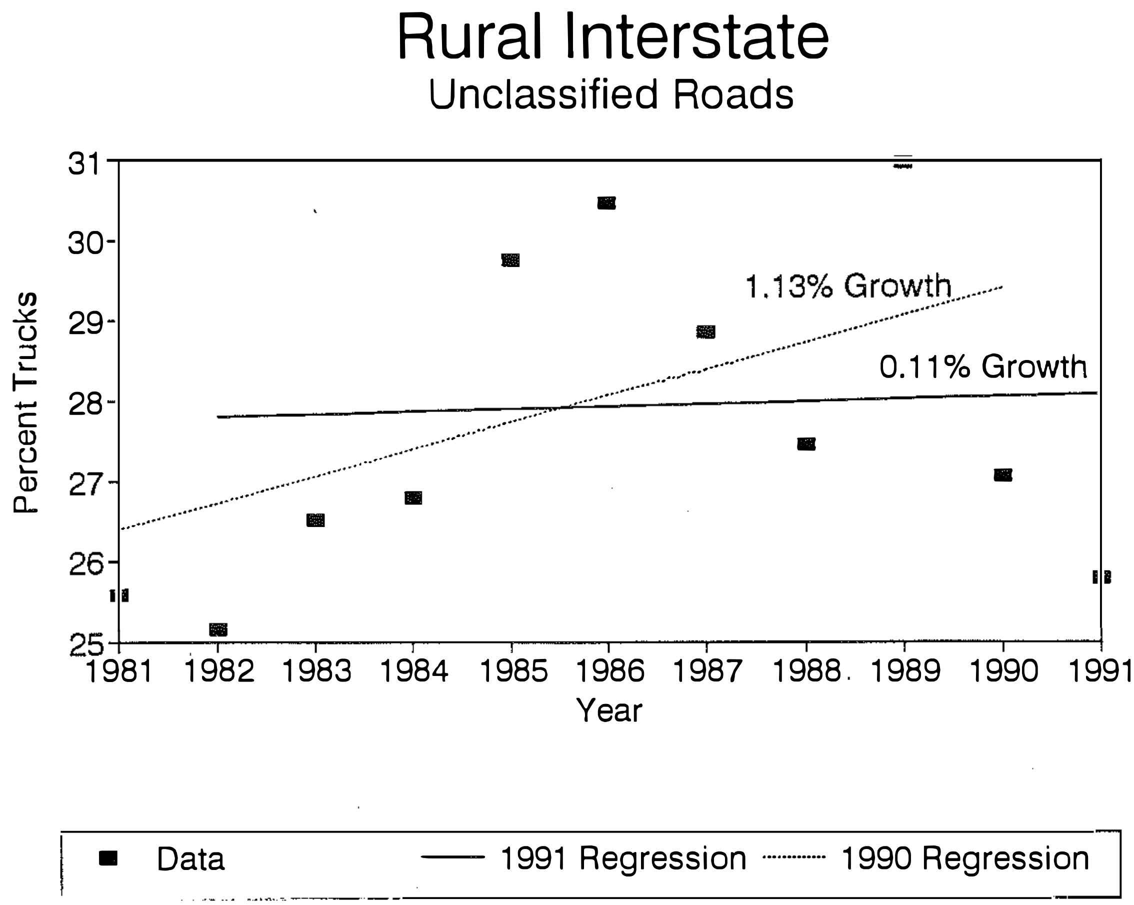

Figure 5. Growth of Percent Trucks on Unclassified Roads of the Rural Interstate System

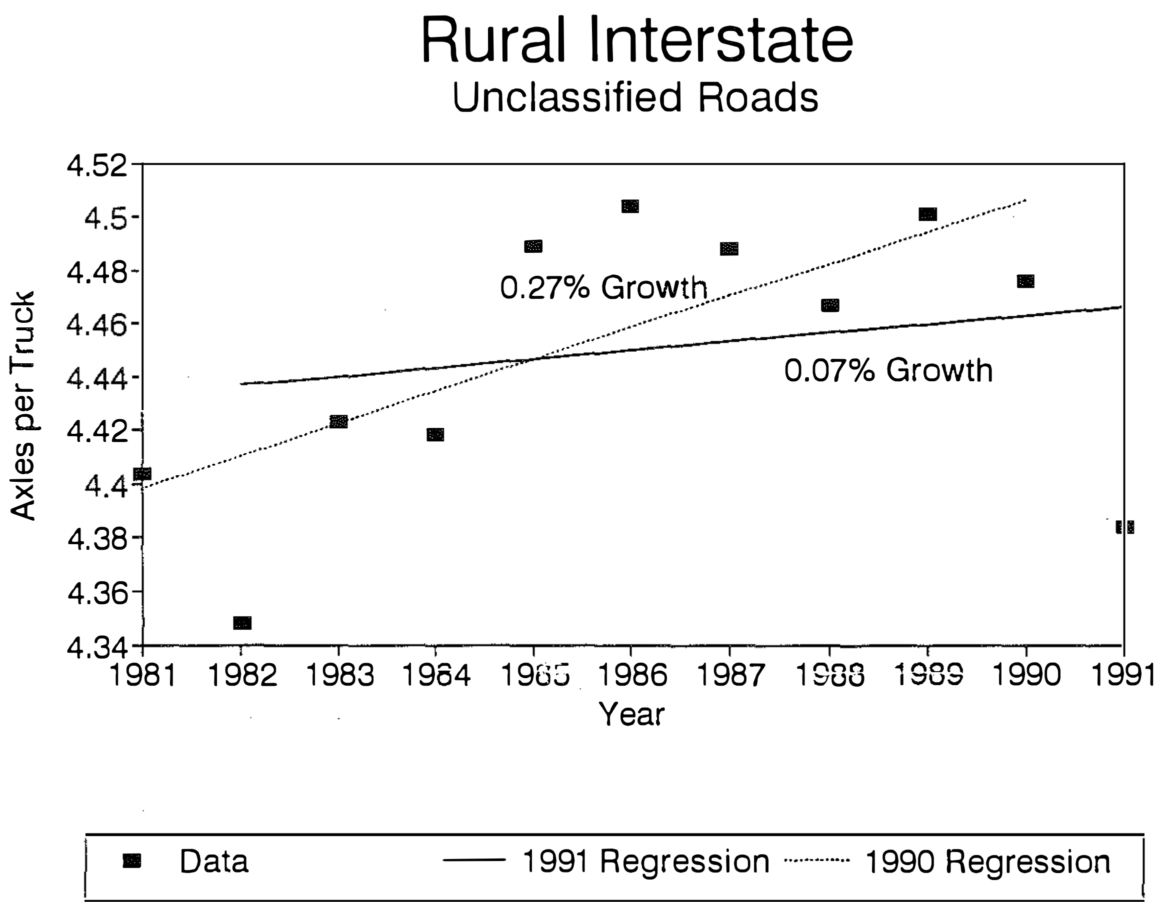

Figure 6. Growth of Axles per Truck on Unclassified Roads of the Rural Interstate System

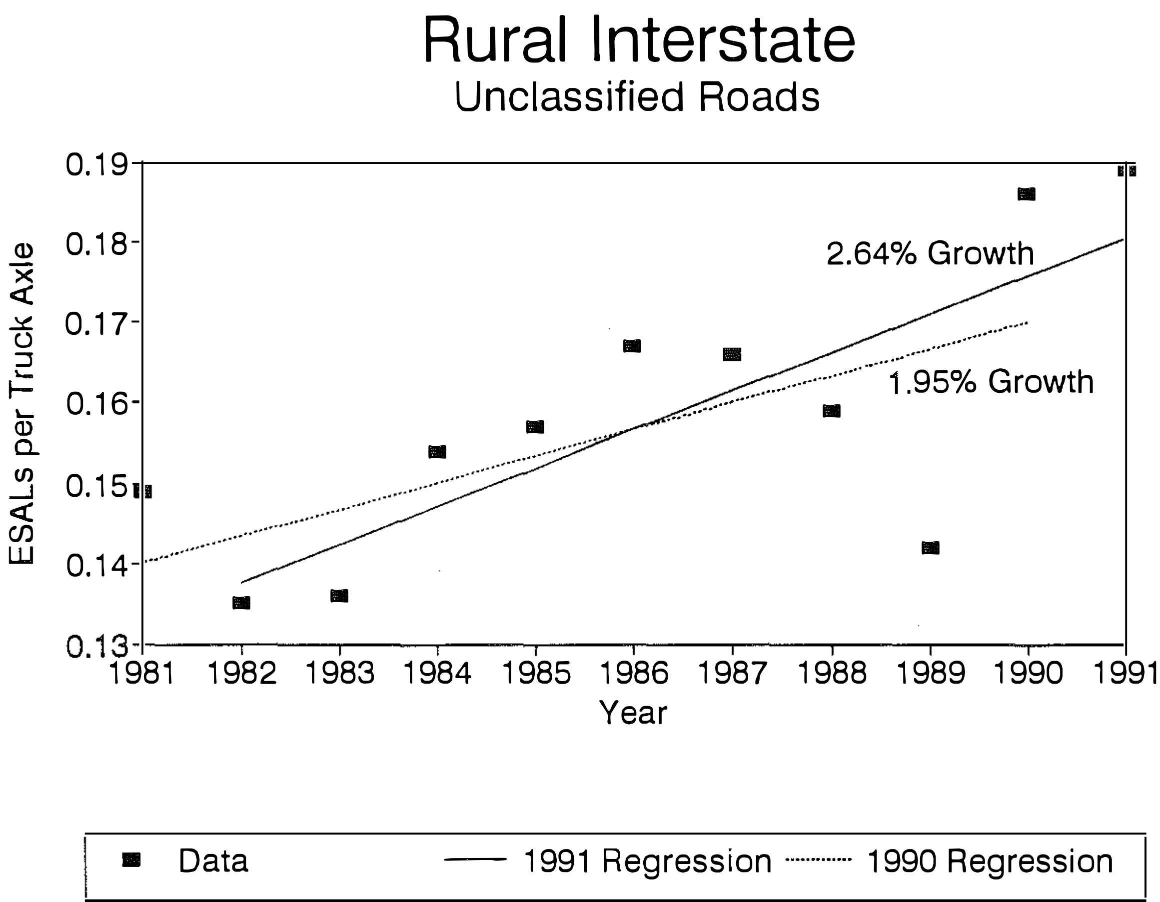

Figure 7. Growth of ESALs per Truck Axle on Unclassified Roads of the Rural Interstate System

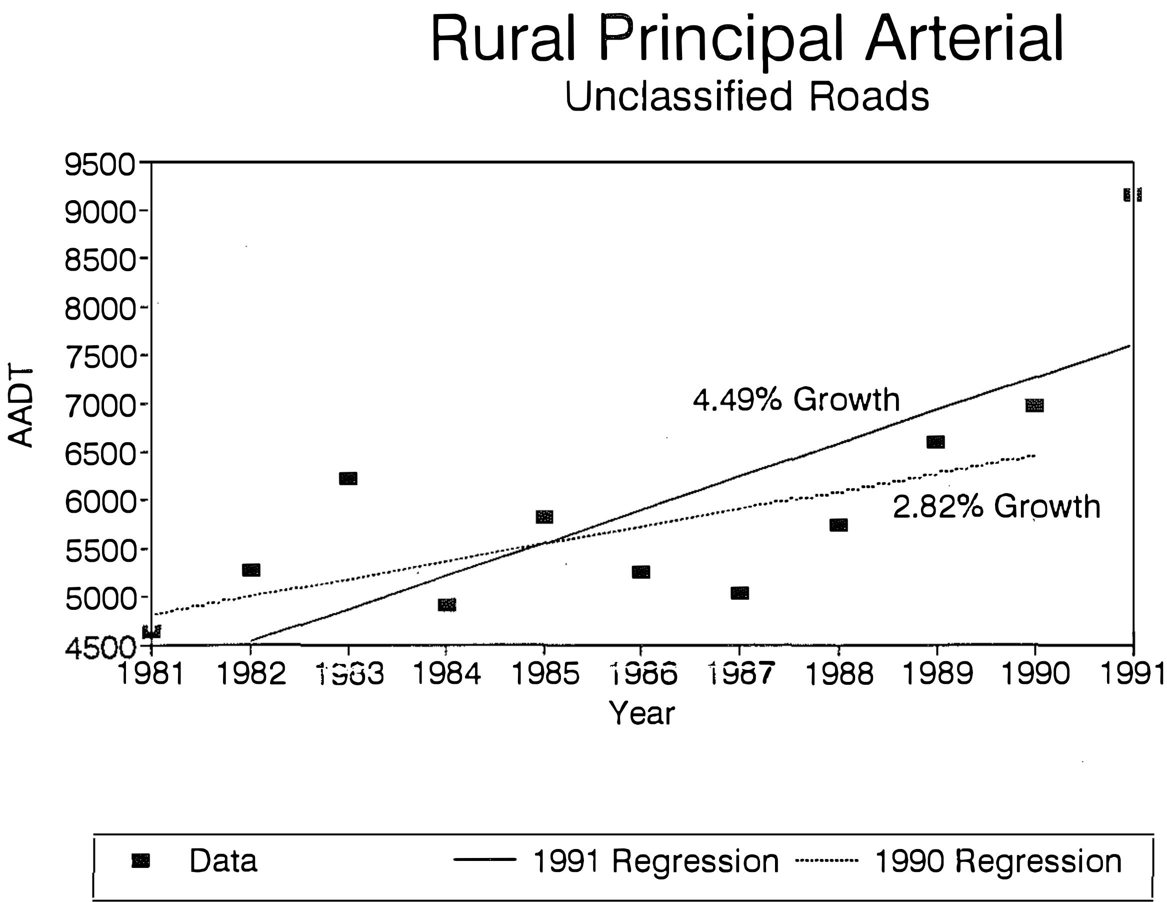

Figure 8. Growth of AADT on Unclassified Roads of the Rural Principal Arterial System

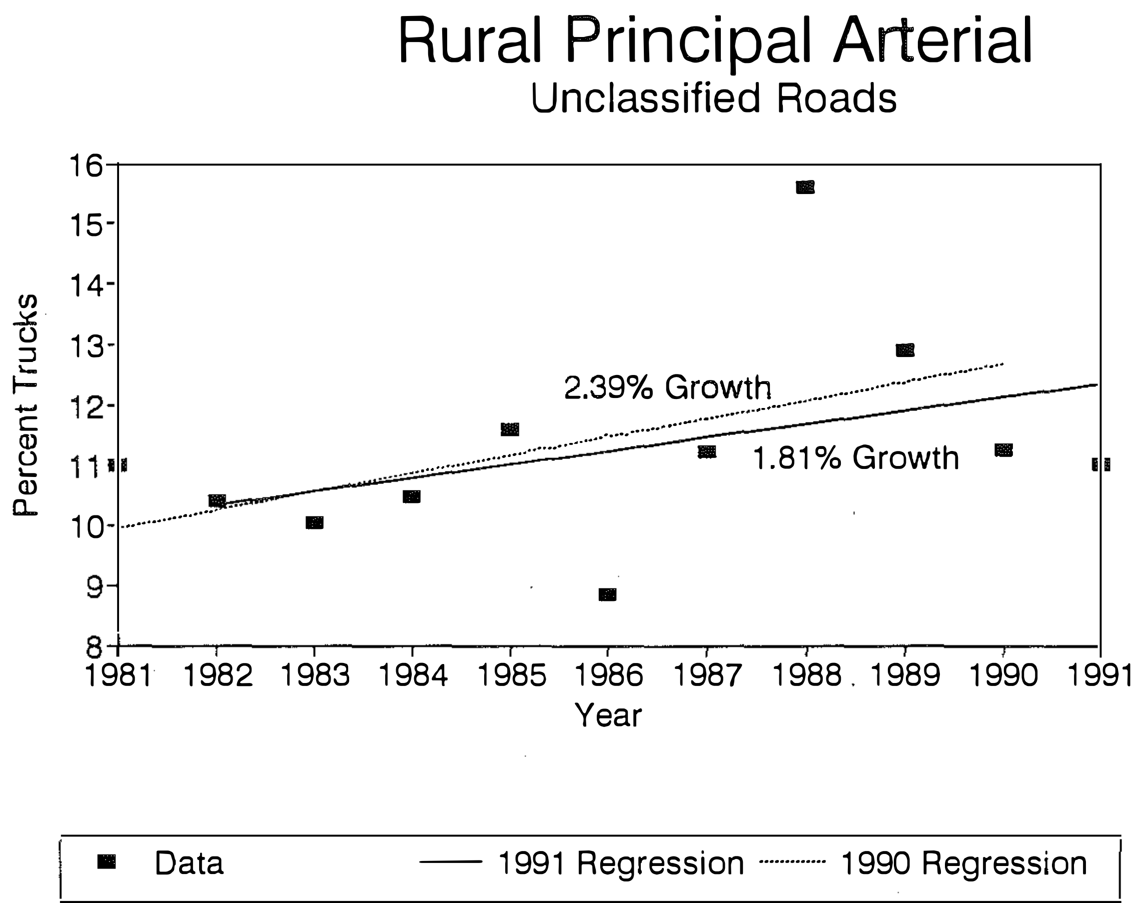

Figure 9. Growth of Percent Trucks on Unclassified Roads of the Rural Principal Arterial System

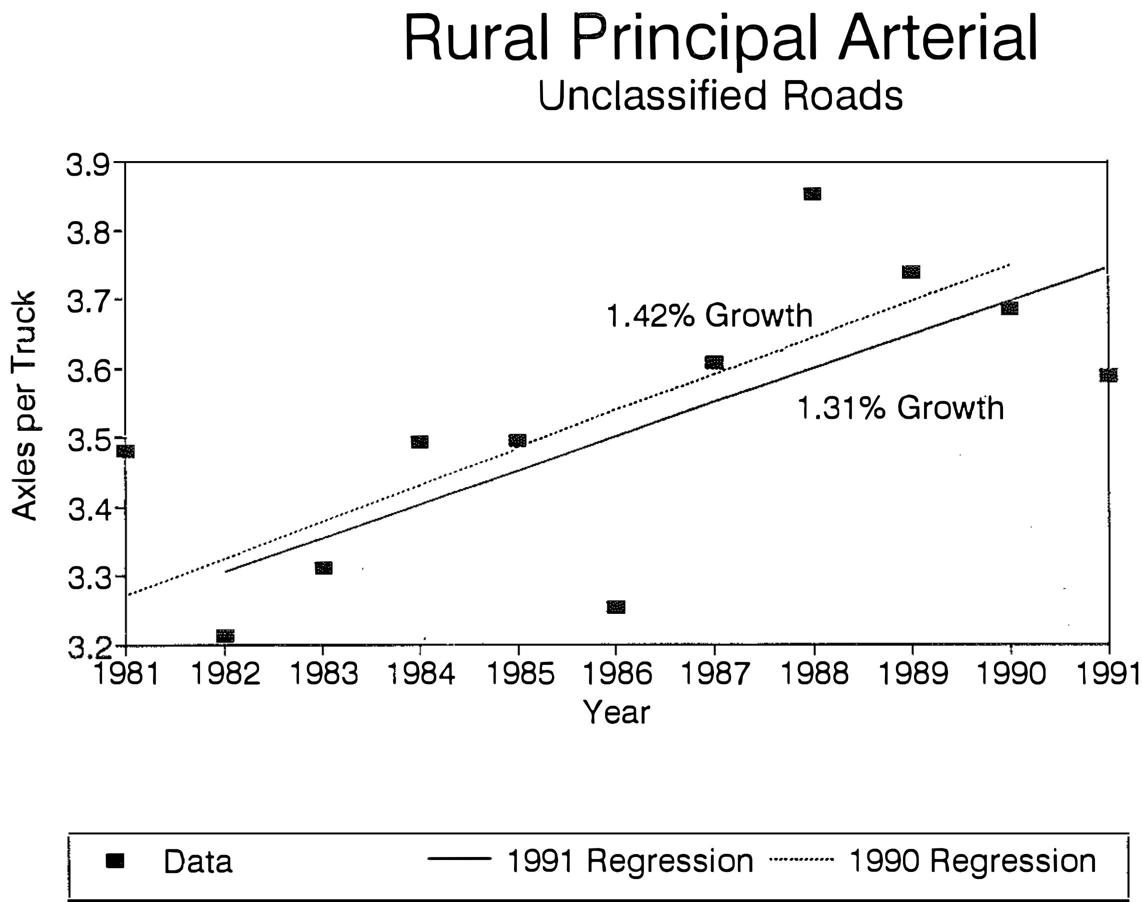

Figure 10. Growth of Axles per Truck on Unclassified Roads of the Rural Principal Arterial System

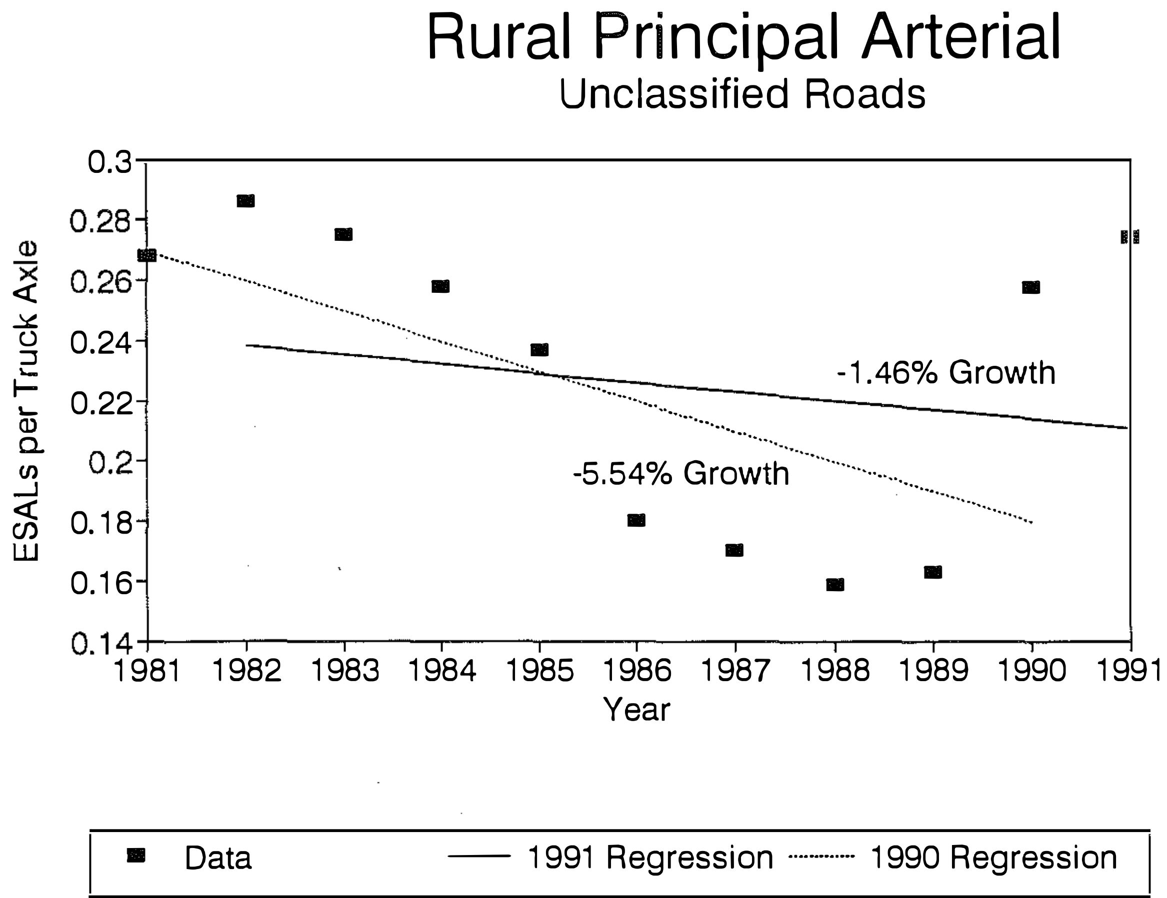

Figure 11. Growth of ESALs per Truck Axle on Unclassified Roads of the Rural Principal Arterial System

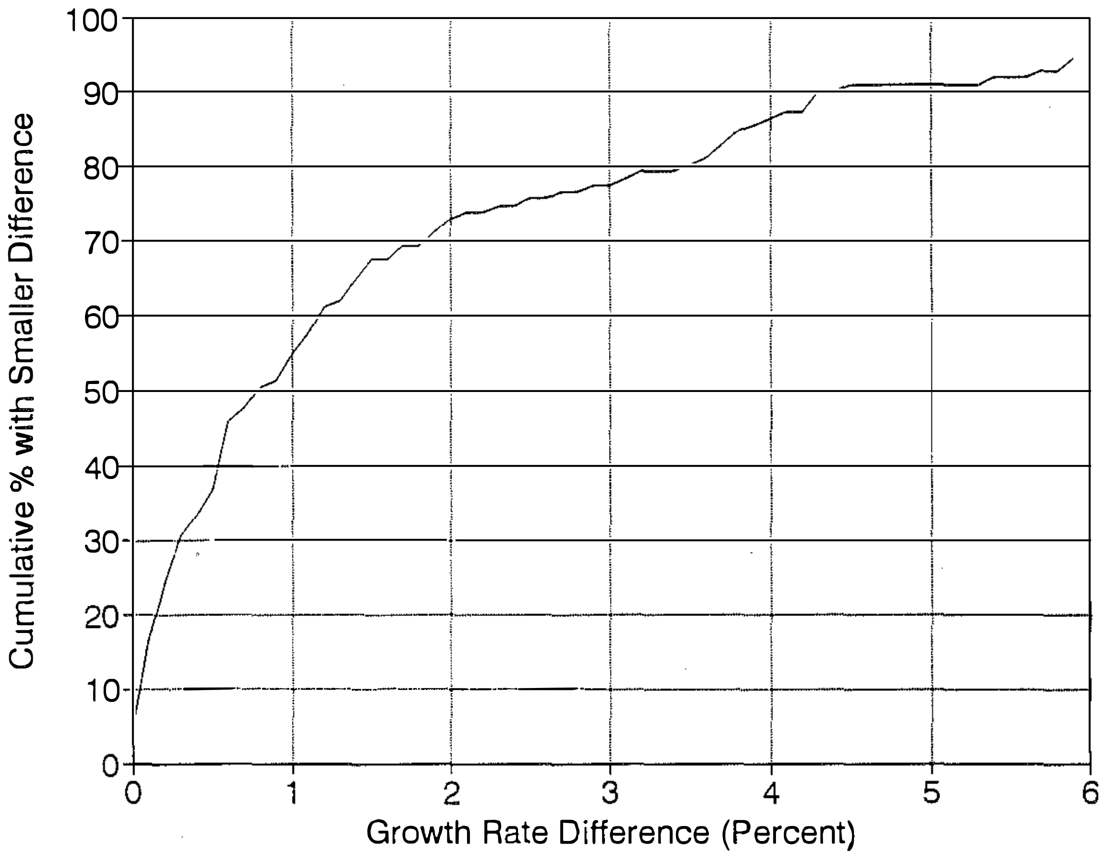

Figure 12. Cumulative Frequency Distribution of Difference Between 1990 and 1991 ESAL-Parameter Growth Rates

1990 AND 1991 MODEL CALIBRATIONS

The aforedescribed activities encompass the major new developments of this quick-response study. In addition, the 1990 ESAL model was recalibrated, and the 1991 model was calibrated for the first time. The primary purpose for the 1990 recalibration was to determine the probable effect of the changes that had been made on the stability of the incremental percentages from year to year. The calibration process also uncovered several minor errors in the historical files that provide input to the SUMMARY program. These errors were easily corrected. When the calibration work was completed, a special effort was made to secure and document the files, programs, and procedures in order to minimize possible errors in future years.

POSSIBLE FUTURE REVISIONS

As mentioned earlier, the many modifications that have been made to EALCALC, the principal computer program of the ESAL series, have dimmed the prospects for its continued successful evolution. In addition, other approaches seem to offer more flexibility to the ESAL-modeling and other related efforts and may offer a more accurate reflection of coal-movement effects. Accordingly, the current study undertook the task of outlining one promising alternative.

dBASE

The proposed alternative, outlined in detail in Appendix D, is based on the replacement of EALCALC by a new dBASE program. The WIM and UNITEAL programs would essentially be retained but would be combined into a single program with the purposes of preliminary processing of vehicle weight records and the development of a new summary weight database of greatly reduced size. The CLASSUM program would be retained essentially intact since it currently produces the necessary kind of database. A new dBASE routine would process these data and produce the necessary output which would be similar in nature to the current output from the EALCALC and SMOOTH programs. The current LOADMTR, EALCALC, and SMOOTH programs would no longer be needed.

The current approach to ESAL forecasting would be retained including the current ESAL parameters (AADT, percent trucks, percent of trucks classified as heavy/coal, axles per normal truck, axles per heavy/coal truck, ESALs per normal truck axle, and ESALs per heavy/coal truck axle). The most significant change in methodology would be to use data from Kentucky's annual coal-haulage statistics to identify the roads over which significant coal haulage occurs. Such a procedure is expected to produce more accurate ESAL estimates for both non-coal-haul and coal-haul roads alike.

A proposal to implement this approach will be considered among possible new FY 1994 initiatives.

OTHER MATTERS

Other changes which were considered during the current investigation but which were tabled for the present include the following:

- Stability in the estimates would be enhanced by limiting analysis to a fixed and constant group of locations (such as HPMS classification counts). However, the aforedescribed changes will yield more stable estimates than in the past, and potentially valuable data should not be excluded unless absolutely necessary. Change can be made later if desired.

- Putting lower and/or upper bounds on each parameter average would assure that, even with an inadequate sample, ESAL estimates would be reasonable. Currently the only constraint is a lower bound of two on the number of axles per truck. The most compelling current need may be for a minimum value for ESALs per heavy/coal truck axle on rural non-interstate roads. However, it was generally agreed that current programs should not be modified to include bounded averages until the current sample size is shown to be too small to yield reliable estimates for some functional classifications. However, it is certainly appropriate to set such limits manually in making forecasts for specific locations.

- The desirability of collecting more WIM and classification data as may be necessary to assure statistical reliability of the estimates for some functional classes was also considered. Although this is a key issue and one that has been raised in prior years, it was considered to be beyond the scope of the current investigation.

- Forecasters should limit incremental percentages for all parameters to some reasonable range (currently 0 to 5 percent). The refinements described herein should reduce the incidence of excessively large or excessively small increments and, hence, the frequency with which such restrictions would be necessary. Future study may define the reasonable ranges in the parameter estimates for the different functional classes of highways.

- The possibility for adding a future graphical display to the annual output which shows the growth incremental percentages as a time series could be explored.

REFERENCES

1. Southgate, Herbert F., "Relationship Between Weights Measured by Permanent Truck Scales and Golden River Weigh-In-Motion Scales," Research Report KTC-89-31, Kentucky Transportation Center, University of Kentucky, Lexington, Kentucky, May 1989.

APPENDIX A

RESEARCH PROPOSAL

PROPOSAL FOR RESEARCH STUDY

ENHANCEMENTS TO PROCEDURE FOR ESTIMATING ESALS

The current procedure for estimating equivalent single axleloads was updated in 1990 to incorporate traffic data categorized by functional class rather than statewide averages. This change resulted from the influx of data generated by automatic equipment to classify and weigh vehicles in motion. A much wider range of geographic conditions and road conditions were sampled and the expectation was a more accurate representation of the data used in the estimation procedure. Significant revisions in the computer programs were required which produced data currently being used by the Division of Planning to estimate ESALs for the design of pavements. Use of the data during the past two years has resulted in requests from the Division of Planning for refinements which would make the data more usable for specific needs. In addition, the continuing analysis of weight and classification data for coal trucks has identified potential improvements that could produce more representative data for these types of vehicles.

Overall, there are several enhancements to the procedure for estimating ESALs which offer the potential for improved accuracy and optimization of use. Included are the following: 1) improvements in the procedures for processing weight and classification data for coal trucks; 2) analysis to determine how coal trucks could be identified when using automatic vehicle classification equipment; 3) production of graphs to visually display trends in parameters used to estimate ESALs; and 4) restructuring the format of the tabular output to enable prompt access to the data.

It is anticipated that approximately six months will be required to complete the tasks described.

Estimated Cost, FY 1992 $25,000

APPENDIX B

PROGRAM LISTING FOR WIM CONVERSION

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SET "IERR" TO DESIRED CODE TO PRINT REJECTED RECORDS
C       1 = AXLES WITHOUT WEIGHT
C       2 = AXLE SPACINGS LESS THAN 1.5 FEET
C       3 = MISMATCH BETWEEN NUMBERS OF AXLE WEIGHTS AND SPACINGS
C       4 = >10% DIFFERENCE BETWEEN GROSS WEIGHT AND SUM OF AXLE WEIGHT
C       5 = >10% DIFFERENCE BETWEEN WHEELBASE AND SUM OF AXLE SPACINGS
C       9 = ALL REJECTED RECORDS
C       0 = NO REJECTED RECORDS
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      BLOCK DATA
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      DATA (FAEQ(I),I=1,7)/3557.2423,3076.3085,2939.1856,4890.9914,
     1   2926.9924,38041.8808,4947.4293/
      DATA (FBEQ(I),I=1,7)/0.8932135,0.735836,0.720307,0.7463315,
     1   0.8693042,0.2044157,0.8634991/
      DATA (RAEQ(I),I=1,7)/3864.2167,2972.9051,3503.9253,4055.0787,
     1   1611.5815,37421.673,15454.4434/
      DATA (RBEQ(I),I=1,7)/0.8619752,0.7179301,0.7393744,0.8017253,
     1   0.9220579,0.2192902,0.6378718/
      DATA C5A,C5B,C6A,C6B/01.0549759,0.0432813,0.85158123,0.05135361/
      DATA C5NA,C5NB,C6NA,C6NB/0.0760435,0.0354375,0.7438038,0.0312612/
      DATA C5S,C6S,C5NS,C6NS/.4368124,1.3110553,.1334733,.9876060/
      END
      CHARACTER*64 FNAME,DATANM,FOUT2,A*80,AA*80
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IERR=9
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     VARIABLE NAME DEFINITIONS
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ITYPE IS USED TO IDENTIFY PAVEMENT SURFACE MATERIAL, "1"=FLEXIBLE
C       AMD "2"=RIGID.
C     FAEQ AND FBEQ ARE CONSTANTS OF STRAIGHT LINE EQUATIONS TO CONVERT
C       WIM AXLELOADS TO STATIC AXLELOADS FOR FLEXIBLE PAVEMENTS.
C     RAEQ AND RBEQ ARE CONSTANTS OF STRAIGHT LINE EQUATIONS TO CONVERT
C       WIM AXLELOADS TO STATIC AXLELOADS FOR RIGID PAVEMENTS.
C     IAXS(1),IAXS(2),...IAXS(70), ARE AXLE SPACINGS.
C     IAW(1),IAW(2),..IAW(70), ARE DYNAMIC WIM AXLELOADS.
C     ISW(1),ISW(2),..ISW(70), ARE CALCULATED STATIC AXLELOADS.
C     I=1 CORRESPONDS TO A STEERING AXLE.
C     I=2 CORRESPONDS TO A DRIVE SINGLE AXLE.
C     I=3 CORRESPONDS TO A TRAILER SINGLE AXLE.
C     I=4 CORRESPONDS TO A DRIVE TANDEM AXLE GROUP IF AXLE SPACING IS
C       LESS THAN OR EQUAL TO 5.0 FEET.
C     I=5 CORRESPONDS TO A TRAILER TANDEM AXLE GROUP IF AXLE SPACING IS
C       LESS THAN OR EQUAL TO 5.0 FEET.
C     I=6 CORRESPONDS TO A DRIVE TRIDEM AXLE GROUP IF FIRST TO THIRD
C       AXLE-SPACING IS LESS THAN OR EQUAL TO 10.0 FEET.
C     I=7 CORRESPONDS TO A TRAILER TRIDEM AXLE GROUP IF FIRST TO THIRD
C       AXLE-SPACING IS LESS THAN OR EQUAL TO 10.0 FEET.
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ITYPE=1
      IFORM=1
      NERRORS1=0
      NERRORS2=0
      NERRORS3=0
      NERRORS4=0
      NERRORS5=0
      NGOOD=0
      IAFC=0
   22 CALL INIT
```

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ DATA FROM WIM FILE
C * *  * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      READ(5,4,END=1234) AA
    4 FORMAT(A80)
      IF(AA(1:1).EQ.'2') THEN
         WRITE(8,4)AA
C        WRITE(6,4)AA
         GOTO 22
      ELSE IF(IFORM.EQ.1) THEN
      READ(AA,1000)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1 (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2 (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1000 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I6,I4,9I3,I4,I3,I1)
      ELSE
      READ(AA,1002)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1 (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2 (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1002 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I7,I4,9I3,I3,I3,I1)
      END IF
      IF(IAXS(2).EQ.O.AND.IAXS(1).LE.120) GOTO 22
   16 IF(ICONTN(JCONT).EQ.O.OR.ICONTN(JCONT).EQ.9) GOTO 19
      IF(ICONTN(JCONT).GE.1.AND.ICONTN(JCONT).LE.8) THEN
      LRT=JCONT*8-2
      LOP=JCONT*8+5
      MRT=JCONT*8-3
      MOP=JCONT*8+4
      JCONT=JCONT+1
   15 READ(5,1001)ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,(BD(J),J=1,6),
     1 (IAW(L),L=LRT,LOP),(IAXS(M),M=MRT,MOP),IRSN,ICONTN(JCONT)
 1001 FORMAT(I1,I2,A2,A3,1X,4I2,6A1,5X,17I3,I1)
      IF(ICONTN(JCONT).GE.1.AND.ICONTN(JCONT).LE.8) GOTO 16
      ENDIF
   19 CONTINUE
      IAFC=IAFC+1
      IF (IAFC.EQ.1) WRITE (8,2222) FC,IYR
 2222 FORMAT (5X,'FUNCTIONAL CLASS= ',A2,5X,'YEAR= 19',I2)
C     ***************************************************************
C     THE FOLLOWING SECTION PERFORMS EDIT CHECKS ON THE WEIGHT DATA
C     AND REPORTS THE RESULTS.                               1/93
C     ***************************************************************
C
C     NERRORS1 = NO. OF ERRORS:  AXLE WITHOUT WEIGHT
C     NERRORS2 = NO. OF ERRORS:  AXLE SPACING LESS THAT 1.5 FEET
C     NERRORS3 = NO. OF ERRORS:  MISMATCH BETWEEN NUMBERS OF AXLE
C       WEIGHTS AND SPACINGS
C     NERRORS4 = NO. OF ERRORS:  10+ PERCENT DIFFERENCE BETWEEN GROSS
C       WEIGHT AND SUM OF AXLE WEIGHTS
C     NERRORS5 = NO. OF ERRORS:  10+ PERCENT DIFFERENCE BETWEEN
C       WHEELBASE AND SUM OF AXLE SPACINGS
C
      SUMW=O.O
      SUMS=O.O
      NA=O
      NB=O
      IWMAX=O
      ISMAX=O
      DO 11 I=1,15
      IF (IAW(I).GT.O) NA=NA+1
      IF (IAW(I).GT.O) IWMAX=I
      SUMW=SUMW+IAW(I)
      IF (IAXS(I).GT.O) NB=NB+1
      IF (IAXS(I).GT.O) ISMAX=I
      SUMS=SUMS+IAXS(I)
```

```
   11 CONTINUE
      DO 121 I=1,IWMAX
      IF (IAW(I).EQ.0) THEN
          NERRORS1=NERRORS1+1
      IF (IERR.EQ.1.OR.IERR.EQ.9)
     *    WRITE(8,1111)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1    (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2    (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1111 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I6,I4,9I3,I4,I3,I1,2X,'E1')
          GO TO 22
      ELSE
      END IF
  121 CONTINUE
      DO 141 I=1,ISMAX
      IF (IAXS(I).LT.15) THEN
          NERRORS2=NERRORS2+1
      IF (IERR.EQ.2.OR.IERR.EQ.9)
     *    WRITE(8,1112)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1    (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2    (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1112 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I6,I4,9I3,I4,I3,I1,2X,'E2')
          GO TO 22
      ELSE
      END IF
  141 CONTINUE
      IF (NA.NE.(NB+1)) THEN
          NERRORS3=NERRORS3+1
      IF (IERR.EQ.3.OR.IERR.EQ.9)
     *    WRITE(8,1113)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1    (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2    (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1113 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I6,I4,9I3,I4,I3,I1,2X,'E3')
          GO TO 22
      ELSE
      END IF
      TWT=ITWT
      AXTOT=IAXTOT
      IF (SUMW.LT.(TWT*.9).OR.SUMW.GT.(TWT*1.1)) THEN
          NERRORS4=NERRORS4+1
      IF (IERR.EQ.4.OR.IERR.EQ.9)
     *    WRITE(8,1114)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1    (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2    (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1114 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I6,I4,9I3,I4,I3,I1,2X,'E4')
          GO TO 22
      ELSE
      END IF
      IF (SUMS.LT.(AXTOT*.9).OR.SUMS.GT.(AXTOT*1.1)) THEN
          NERRORS5=NERRORS5+1
      IF (IERR.EQ.5.OR.IERR.EQ.9)
     *    WRITE(8,1115)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,
     1    (BD(J),J=1,9),(ICH(J),J=1,6),ICMOD,ITWT,(IAW(L),L=1,5),
     2    (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(JCONT)
 1115 FORMAT(I1,I2,A2,A3,I1,4I2,9A1,6I1,3X,I6,I4,9I3,I4,I3,I1,2X,'E5')
          GO TO 22
      ELSE
      END IF
      NGOOD=NGOOD+1
C     ***********************************************************
C     END OF DATA EDIT SECTION
C     ***********************************************************
      IF(IAXS(1).GT.55) GOTO 245
C     ***********************************************************
C     THE FOLLOWING SECTION TO 245 APPLIES TO TRUCKS HAVING 2 STEERING
```

```
C     AXLES (EX: CRANES, DRILL RIGS, TRANSIT MIXERS, EUROPEAN DUMPS)
C     ***************************************************************
      IF(IAXS(1).LE.55.AND.IAXS(2).LE.55.AND.IAXS(3).GT.55) GOTO 241
      IF(IAXS(1).LE.55.AND.IAXS(2).GT.55) GOTO 243
  241 DO 242 J=4,13
      IF(IAXS(J).EQ.O) GOTO 244
  242 CONTINUE
  244 ICT(1)=2
      ICT(2)=J
      CALL TRTRI(1,2,3)
      IF(J.EQ.5) CALL DRTAN(4,5)
      IF(J.EQ.6) CALL DRTRI(4,5,6)
      IF(J.EQ.7) CALL DQUAD(4,5,6,7)
      GOTO 100
  243 DO 246 J=3,13
      IF(IAXS(J).EQ.O) GOTO 247
  246 CONTINUE
  247 ICT(1)=2
      ICT(2)=J
      CALL TRTAN(1,2)
      IF(J.EQ.4) CALL DRTAN(3,4)
      IF(J.EQ.5) CALL DRTRI(3,4,5)
      IF(J.EQ.6) CALL DQUAD(3,4,5,6)
      GOTO 100
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C *   CHECK AXLE SPACING TO DETERMINE AXLE CONFIGURATION   *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  245 AS=O
      DO 24 I=1,13
      IF (IAXS(I).EQ.O) GOTO 25
      AS=AS+0.1*IAXS(I)
      AI=I
      AV=AS/AI
   24 CONTINUE
   25 J=I-1
      IF (J.EQ.O) GOTO 28
      ACK=12.5*AV
      ICK=ACK
      DO 29 K=1,J
      IF (IAXS(K).GT.ICK) GOTO 28
   29 CONTINUE
      IF(IAXS(2).EQ.O) GOTO 28
      ICT(1)=9
      ICT(2)=2
      CALL STEER
      CALL DRSING
      DO 26 L=3,I
      M=L
      ICT(M)=O
      CALL TRSING(M)
   26 CONTINUE
      IF (I.LE.4) THEN
      ICT(1)=3
      ELSE
      ICT(1)=5
      END IF
      IF (I.EQ.3.OR.I.EQ.5) THEN
      ICT(3)=1
      ELSE
      ICT(3)=2
      END IF
      IF (I.GE.5) THEN
      ICT(4)=2
```

```
      END IF
      GOTO 100
   28 IF(IAXS(2).EQ.O) GOTO 21
      IF(IAXS(3).EQ.O) GOTO 31
      IF(IAXS(4).EQ.O) GOTO 41
      IF(IAXS(5).EQ.O) GOTO 51
      IF(IAXS(6).EQ.O) GOTO 61
      IF(IAXS(7).EQ.O) GOTO 71
      IF(IAXS(8).EQ.O) GOTO 81
      GOTO 91
C  * * * * * * * 2-AXLE TRUCK * * *
   21 CALL STEER
      CALL DRSING
      IWT=ISW(1)+ISW(2)
      IF(IAXS(1).GE.200) THEN
      ICT(1)=1
      ICT(2)=9
      GOTO 100
      ELSE IF(IAXS(2).EQ.O.AND.IAXS(1).GE.88.AND.IAXS(1).LE.145.AND.
     1   IWT.GT.30.AND.IWT.LE.66) THEN
      ICT(1)=2
      ICT(2)=0
      GOTO 100
      ELSE IF(IAXS(2).EQ.O.AND.IAXS(1).LT.115.AND.IWT.GT.66.AND.
     1    IWT.LE.90) THEN
      ICT(1)=2
      ICT(2)=1
      GOTO 100
      ELSE IF(IWT.GT.90) THEN
      ICT(1)=2
      ICT(2)=2
      GOTO 100
      ELSE
      ICT(1)=2
      ICT(2)=2
      GOTO 100
      END IF
C  * * * * * * * 3-AXLE TRUCK * * *
   31 CALL STEER
      IF(IAXS(1).GE.190.AND.IAXS(2).LE.50) THEN
      ICT(1)=1
      ICT(2)=9
      CALL STEER
      CALL DRTAN(2,3)
      GOTO 100
      ELSE
      END IF
      IF(IAXS(2).LE.50) GOTO 33
      ICT(1)=3
      ICT(2)=2
      ICT(3)=1
      CALL DRSING
      CALL TRSING(3)
      GOTO 100
   33 CALL DRTAN(2,3)
      ICT(1)=2
      ICT(2)=3
      GOTO 100
C  * * * * * * * 4-AXLE TRUCK * * *
   41 CALL STEER
      IF((IAXS(2)+IAXS(3)).LE.100) GOTO 43
      IF(IAXS(2).LE.55) GOTO 42
      CALL DRSING
      ICT(1)=3
```

```
      ICT(2)=2
      IF(IAXS(3).LE.50) THEN
            CALL TRTAN(3,4)
            ICT(3)=2
            GOTO 100
         ELSE
            CALL TRSING(3)
            CALL TRSING(4)
            IF (IAXS(3).LE.80) THEN
               ICT(3)=2
               ELSE
               ICT(3)=7
            END IF
        GOTO 100
      END IF
   42 CALL DRTAN(2,3)
      CALL TRSING(4)
      ICT(1)=3
      ICT(2)=3
      ICT(3)=1
      GOTO 100
   43 CALL DRTRI(2,3,4)
      ICT(1)=2
      ICT(2)=4
      GOTO 100
C  * * * * * * * 5-AXLE TRUCK * * *
   51 CALL STEER
      IAX23=IAXS(2)+IAXS(3)
      IAX24=IAXS(2)+IAXS(3)+IAXS(4)
      IAX34=IAXS(3)+IAXS(4)
      IF(IAX24.LE.150) GOTO 52
      IF(IAX23.LE.100) GOTO 58
      IF(IAXS(2).LE.55.AND.IAXS(4).LE.55) GOTO 53
      IF(IAXS(2).LE.55.AND.IAXS(4).GT.55) GOTO 55
      IF(IAXS(2).GT.55.AND.IAX34.LE.100) GOTO 56
C     IF ABOVE 5 IF STATEMENTS ARE FALSE, THEN HAVE 5-AXLE DOUBLE BOTTOM
      ICT(1)=5
      ICT(2)=2
      ICT(3)=1
      ICT(4)=2
      CALL DRSING
      CALL TRSING(3)
      CALL TRSING(4)
      CALL TRSING(5)
      GOTO 100
   52 CALL DQUAD(2,3,4,5)
      ICT(1)=2
      ICT(2)=5
      GOTO 100
   53 CALL DRTAN(2,3)
      CALL TRTAN(4,5)
      ICT(1)=3
      ICT(2)=3
      ICT(3)=2
      GOTO 100
   55 CALL DRTAN(2,3)
      CALL TRSING(4)
      CALL TRSING(5)
      ICT(1)=3
      ICT(2)=3
            IF (IAXS(4).LE.80) THEN
               ICT(3)=2
               ELSE
               ICT(3)=7
```

```
      GOTO 100
      END IF
   56 CALL DRSING
      CALL TRTRI(3,4,5)
      ICT(1)=3
      ICT(2)=2
      ICT(3)=3
      GOTO 100
   58 CALL DRTRI(2,3,4)
      CALL TRSING(5)
      ICT(1)=3
      ICT(2)=4
      ICT(3)=1
      GOTO 100
C  * * * * * * * 6-AXLE TRUCK * * *
   61 CALL STEER
      IAXTR=IAXS(4)+IAXS(5)
      IAXD=IAXS(2)+IAXS(3)+IAXS(4)
      IF(IAXS(2).GT.55) GOTO 66
      IF(IAXD.LE.100.AND.IAXS(5).LE.55) GOTO 64
      IF(IAXD.LE.100.AND.IAXS(5).GT.55.AND.IAXS(5).LT.120) GOTO 65
      IF(IAXS(2).LE.55.AND.IAXTR.LE.100) GOTO 62
      IF(IAXS(2).LE.50.AND.IAXS(3).GT.50) GOTO 63
C     IF ALL OF ABOVE ARE FALSE, THEN HAVE 5-AXLE TRACTOR + TRL. SNGL.
      CALL TRSING(6)
      CALL DQUAD(2,3,4,5)
      ICT(1)=3
      ICT(2)=5
      ICT(3)=1
      GOTO 100
   62 CALL DRTAN(2,3)
      CALL TRTRI(4,5,6)
      ICT(1)=3
      ICT(2)=3
      ICT(3)=3
      GOTO 100
   63 CALL DRTAN(2,3)
      CALL TRSING(4)
      CALL TRSING(5)
      CALL TRSING(6)
      ICT(1)=5
      ICT(2)=3
      ICT(3)=1
      ICT(4)=2
      GOTO 100
   64 CALL DRTRI(2,3,4)
      CALL TRTAN(5,6)
      ICT(1)=3
      ICT(2)=4
      ICT(3)=2
      GOTO 100
   65 CALL DRTRI(2,3,4)
      CALL TRSING(5)
      CALL TRSING(6)
      ICT(1)=3
      ICT(2)=4
      ICT(3)=7
      GOTO 100
   66 CALL DRSING
      IF(IAXS(4).GT.55.AND.IAXS(4).LT.100) THEN
        CALL TRTAN(3,4)
        CALL TRSING(5)
        CALL TRSING(6)
        ICT(1)=5
```

```
        ICT(2)=2
        ICT(3)=2
        ICT(4)=2
        GOTO 100
      ELSE
        CALL TQUAD(3,4,5,6)
        ICT(1)=3
        ICT(2)=2
        ICT(3)=4
        GOTO 100
      END IF
C   * * * * * * * 7-AXLE TRUCK * * *
   71 CALL STEER
      IAXD23=IAXS(2)+IAXS(3)
      IAXD24=IAXS(2)+IAXS(3)+IAXS(4)
      IAXD26=IAXS(2)+IAXS(3)+IAXS(4)+IAXS(5)+IAXS(6)
      IAXT46=IAXS(4)+IAXS(5)+IAXS(6)
      IAXT56=IAXS(5)+IAXS(6)
      IF(IAXS(2).GT.55.AND.IAXD26.LE.250) THEN
        CALL SEXTET(2,3,4,5,6,7)
        ICT(1)=2
        ICT(2)=7
        GOTO 100
      ELSE IF(IAXS(2).GT.55.AND.IAXS(3).GT.50.AND.IAXS(4).GT.50.AND.
     1  IAXS(5).GT.50.AND.IAXS(6).GT.50) THEN
        CALL DRSING
        CALL TRSING(3)
        CALL TRSING(4)
        CALL TRSING(5)
        CALL TRSING(6)
        CALL TRSING(7)
        ICT(1)=7
        ICT(2)=2
        ICT(3)=1
        ICT(4)=2
        ICT(5)=2
        GOTO 100
      ELSE IF(IAXS(2).GT.55.AND.IAXS(3).LE.50.AND.IAXS(4).GT.50.AND.
     1  IAXS(5).GT.50.AND.IAXS(6).GT.50) THEN
        CALL TRTAN(3,4)
        CALL TRSING(5)
        CALL TRSING(6)
        CALL TRSING(7)
        ICT(1)=7
        ICT(2)=2
        ICT(3)=2
        ICT(4)=2
        GOTO 100
      ELSE IF(IAXS(2).GT.55.AND.IAXS(3).GT.50.AND.IAXS(4).GT.50.AND.
     1  IAXS(5).GT.50.AND.IAXS(6).LE.50) THEN
        CALL TRSING(3)
        CALL TRSING(4)
        CALL TRSING(5)
        CALL TRTAN(6,7)
        ICT(1)=7
        ICT(2)=2
        ICT(3)=1
        ICT(4)=2
        ICT(5)=2
        GOTO 100
      ELSE
      END IF
      IF(IAXS(2).LE.55.AND.IAXS(3).GT.50) GOTO 72
      IF(IAXD23.LE.100.AND.IAXS(4).GT.50) GOTO 73
```

```
      IF(IAXD24.LE.150) GOTO 74
   72 CALL DRTAN(2,3)
        ICT(1)=3
        ICT(2)=3
      IF(IAXT46.LE.150) GOTO 77
      IF(IAXT46.GT.150.AND.IAXT56.LT.100) THEN
        CALL TRTRI(5,6,7)
        CALL TRSING(4)
        ICT(3)=9
        GOTO 100
      ELSE IF (IAXT46.GT.150.AND.IAXS(6).LE.80) THEN
        CALL TRTRI(4,5,6)
        CALL TRSING(7)
        ICT(3)=9
        GOTO 100
      ELSE IF(IAXT46.GT.150.AND.IAXS(6).GT.80) THEN
          CALL TRTRI(4,5,6)
          CALL TRSING(7)
          ICT(1)=5
          ICT(2)=3
          ICT(3)=3
          ICT(4)=1
          GOTO 100
      ELSE
          CALL TRTAN(4,5)
          CALL TRSING(6)
          CALL TRSING(7)
          ICT(1)=5
          ICT(2)=3
          ICT(3)=2
          ICT(4)=2
      GOTO 100
      ENDIF
   77 CALL TQUAD(4,5,6,7)
      ICT(1)=3
      ICT(2)=3
      ICT(3)=4
      GOTO 100
   73 CALL DRTRI(2,3,4)
      IF(IAXT56.GT.100) GOTO 731
      CALL TRTRI(5,6,7)
      ICT(1)=3
      ICT(2)=4
      ICT(3)=3
      GOTO 100
  731 CALL TRSING(5)
      CALL TRSING(6)
      CALL TRSING(7)
      ICT(1)=5
      ICT(2)=4
      ICT(3)=1
      ICT(3)=2
      GOTO 100
   74 CALL DQUAD(2,3,4,5)
      IF(IAXS(6).LE.50) GOTO 741
      CALL TRSING(6)
      CALL TRSING(7)
      IF(IAXS(6).GT.80) THEN
         ICT(1)=3
         ICT(2)=5
         ICT(3)=7
         GOTO 100
      ELSE
         ICT(1)=4
```

```
          ICT(2)=5
          ICT(3)=7
          GOTO 100
       END IF
  741  CALL TRTAN(6,7)
       ICT(1)=3
       ICT(2)=5
       ICT(3)=2
       GOTO 100
C  * * * * * * * 8-AXLE TRUCK * * *
   81  CALL STEER
       IAXD23=IAXS(2)+IAXS(3)
       IAXD24=IAXS(2)+IAXS(3)+IAXS(4)
       IAXD37=IAXS(3)+IAXS(4)+IAXS(5)+IAXS(6)+IAXS(7)
       IAXD27=IAXS(2)+IAXS(3)+IAXS(4)+IAXS(5)+IAXS(6)+IAXS(7)
       IAXT34=IAXS(3)+IAXS(4)
       IAXT57=IAXS(5)+IAXS(6)+IAXS(7)
       IAXT67=IAXS(6)+IAXS(7)
       IF(IAXD27.LE.300) THEN
       CALL DRTAN(2,3)
       CALL DRTAN(4,5)
       CALL DRTRI(6,7,8)
       ICT(1)=2
       ICT(2)=8
       GOTO 100
       ELSE
       END IF
       IF(IAXS(2).GT.55) GOTO 815
       IF(IAXS(2).LE.55.AND.IAXS(3).GT.50) GOTO 82
       IF(IAXD23.LE.100.AND.IAXS(4).GT.50) GOTO 83
       IF(IAXD24.LE.150) GOTO 84
  815  CALL DRSING
       IF(IAXD37.LE.300) THEN
       CALL TRTAN(3,4)
       CALL TRTAN(5,6)
       CALL TRTAN(7,8)
       ICT(1)=3
       ICT(2)=2
       ICT(3)=6
       GOTO 100
       ELSE IF(IAXT34.LE.100.AND.IAXT67.LE.100.AND.IAXS(5).GT.80) THEN
       CALL TRTRI(3,4,5)
       CALL TRTRI(6,7,8)
       ICT(1)=5
       ICT(2)=2
       ICT(3)=3
       ICT(4)=3
       GOTO 100
       ELSE IF(IAXS(2).GT.80.AND.IAXS(4).GT.80.AND.IAXS(6).GT.80.AND.
      1    IAXS(3).LE.50.AND.IAXS(5).LE.50.AND.IAXS(7).LE.50) THEN
       CALL TRTAN(3,4)
       CALL TRTAN(5,6)
       CALL TRTAN(7,8)
       ICT(1)=5
       ICT(2)=2
       ICT(3)=2
       ICT(4)=4
       GOTO 100
       ELSE
       END IF
   82  CALL DRTAN(2,3)
       IF(IAXS(4).LE.50.AND.IAXS(5).GT.50.AND.IAXS(6).GT.80.AND.
      1    IAXS(7).LE.50) GOTO 88
       IF(IAXS(4).LE.50.AND.IAXS(5).LE.50.AND.IAXS(6).GT.80) GOTO 89
```

```
      CALL TQUINT(4,5,6,7,8)
      ICT(1)=3
      ICT(2)=3
      ICT(3)=5
      GOTO 100
   89 CALL TRTRI(4,5,6)
      IF(IAXS(7).GT.50) GOTO 891
      CALL TRTAN(7,8)
      ICT(1)=5
      ICT(2)=3
      ICT(3)=3
      ICT(4)=2
      GOTO 100
  891 CALL TRSING(7)
      CALL TRSING(8)
      ICT(1)=5
      ICT(2)=3
      ICT(3)=3
      ICT(4)=7
      GOTO 100
   88 CALL TRTAN(4,5)
      CALL TRSING(6)
      CALL TRTAN(7,8)
      ICT(1)=5
      ICT(2)=3
      ICT(3)=2
      ICT(4)=3
      GOTO 100
   83 CALL DRTRI(2,3,4)
      IF(IAXT57.LT.150) THEN
      CALL TQUAD(5,6,7,8)
      ICT(1)=3
      ICT(2)=4
      ICT(3)=4
      GOTO 100
      ELSE IF(IAXT67.LE.100.AND.IAXS(7).GT.50) THEN
      CALL TRTRI(5,6,7)
      CALL TRSING(8)
      ICT(1)=5
      ICT(2)=4
      ICT(3)=3
      ICT(4)=1
      GOTO 100
      ELSE
  832 CALL TRTAN(5,6)
      IF(IAXS(7).LE.50) THEN
      CALL TRTAN(7,8)
      ICT(1)=5
      ICT(2)=4
      ICT(3)=2
      ICT(4)=2
      GOTO 100
      ELSE
      CALL TRSING(7)
      CALL TRSING(8)
      ICT(1)=5
      ICT(2)=4
      ICT(3)=2
      ICT(4)=7
      GOTO 100
      END IF
      END IF
   84 CALL DQUAD(2,3,4,5)
      IF(IAXT67.LE.100) GOTO 841
```

```
      IF(IAXS(6).GT.80.AND.IAXS(7).LE.50) THEN
      CALL TRSING(6)
      CALL TRTAN(7,8)
      ICT(1)=4
      ICT(2)=5
      ICT(3)=3
      GOTO 100
      ELSE
      CALL TRSING(6)
      CALL TRSING(7)
      CALL TRSING(8)
      ICT(1)=5
      ICT(2)=5
      ICT(3)=1
      ICT(4)=2
      GOTO 100
      END IF
  841 CALL TRTRI(6,7,8)
      ICT(1)=3
      ICT(2)=5
      ICT(3)=3
      GOTO 100
C  * * * * * * * 9-AXLE TRUCK * * *
   91 CALL STEER
      IAX23=IAXS(2)+IAXS(3)
      IAX24=IAXS(2)+IAXS(3)+IAXS(4)
      IAX25=IAXS(2)+IAXS(3)+IAXS(4)+IAXS(5)
      IAXT45=IAXS(4)+IAXS(5)
      IAXT56=IAXS(5)+IAXS(6)
      IAXT67=IAXS(6)+IAXS(7)
      IAXT78=IAXS(7)+IAXS(8)
      IAXT68=IAXS(6)+IAXS(7)+IAXS(8)
      IAXT58=IAXS(5)+IAXS(6)+IAXS(7)+IAXS(8)
      IAXT48=IAXS(4)+IAXS(5)+IAXS(6)+IAXS(7)+IAXS(8)
      IF(IAXS(2).LE.55.AND.IAXS(3).GT.80.AND.IAXT48.LE.250) THEN
          CALL DRTAN(2,3)
          CALL TRTRI(4,5,6)
          CALL TRTRI(7,8,9)
          ICT(1)=3
          ICT(2)=3
          ICT(3)=6
          GOTO 100
      ELSE IF(IAXS(2).LE.55.AND.IAXS(3).GT.80.AND.IAXT45.LE.100.AND.
     1    IAXS(6).GT.80.AND.IAXS(7).GT.80.AND.IAXS(8).LE.50) THEN
          CALL DRTAN(2,3)
          CALL TRTRI(4,5,6)
          CALL TRSING(7)
          CALL TRTAN(8,9)
          ICT(1)=5
          ICT(2)=3
          ICT(3)=3
          ICT(4)=8
          GOTO 100
      ELSE
          CALL DRTAN(2,3)
          CALL TRTAN(4,5)
          IF(IAXS(5).GT.80.AND.IAXS(6).GT.80) THEN
                CALL TRSING(6)
                CALL TRSING(7)
                CALL TRSING(8)
                CALL TRSING(9)
                ICT(1)=7
                ICT(2)=3
                ICT(3)=2
```

```
              ICT(4)=2
              ICT(5)=2
              GOTO 100
         ELSE IF(IAXS(5).GT.80.AND.IAXS(6).LE.50.AND.IAXS(7).GT.80.AND.
     1        IAXS(8).LE.50) THEN
              CALL TRTAN(6,7)
              CALL TRTAN(8,9)
              ICT(1)=5
              ICT(2)=3
              ICT(3)=2
              ICT(4)=9
              GOTO 100
         END IF
      END IF
      IF(IAX23.LE.100.AND.IAXS(4).GT.80.AND.IAXT58.LE.200) THEN
         CALL DRTRI(2,3,4)
         CALL TRTAN(5,6)
         CALL TRTRI(7,8,9)
         ICT(1)=3
         ICT(2)=4
         ICT(3)=5
         GOTO 100
      ELSE IF(IAX23.LE.100.AND.IAXS(4).GT.80.AND.IAXT56.LE.100.AND.
     1   IAXS(7).GT.80.AND.IAXS(8).GT.80) THEN
              CALL DRTRI(2,3,4)
              CALL TRTAN(5,6)
              CALL TRSING(7)
              CALL TRSING(8)
              CALL TRSING(9)
              ICT(1)=7
              ICT(2)=4
              ICT(3)=2
              ICT(4)=1
              ICT(5)=7
              GOTO 100
      ELSE IF(IAX23.LE.100.AND.IAXS(4).GT.80.AND.IAXT56.LE.100.AND.
     1   IAXS(7).GT.80.AND.IAXS(8).LE.80) THEN
              CALL DRTRI(2,3,4)
              CALL TRTRI(5,6,7)
              CALL TRTAN(8,9)
              ICT(1)=5
              ICT(2)=4
              ICT(3)=3
              ICT(4)=7
              GOTO 100
      ELSE IF(IAX23.LE.100.AND.IAXS(4).GT.80.AND.IAXS(5).LE.50.AND.
     1   IAXS(6).GT.80.AND.IAXS(7).GT.80.AND.IAXS(8).LE.80) THEN
              CALL DRTRI(2,3,4)
              CALL TRTAN(5,6)
              CALL TRSING(7)
              CALL TRTAN(8,9)
              ICT(1)=5
              ICT(2)=4
              ICT(3)=2
              ICT(4)=8
              GOTO 100
      ELSE
      ENDIF
      IF(IAX24.LE.150.AND.IAXS(5).GT.80.AND.IAXT68.LE.150) THEN
         CALL DQUAD(2,3,4,5)
         CALL TQUAD(6,7,8,9)
         ICT(1)=3
         ICT(2)=5
         ICT(3)=4
```

```
            GOTO 100
        ELSE IF(IAX24.LE.150.AND.IAXS(5).GT.80.AND.IAXT67.LE.100.AND.
       1    IAXS(8).GT.80) THEN
                  CALL DQUAD(2,3,4,5)
                  CALL TRTRI(6,7,8)
                  CALL TRSING(9)
                  ICT(1)=5
                  ICT(2)=5
                  ICT(3)=3
                  ICT(4)=1
                  GOTO 100
        ELSE IF(IAX24.LE.150.AND.IAXS(5).GT.80.AND.IAXS(6).LE.50.AND.
       1    IAXS(7).GT.80.AND.IAXS(8).GT.80) THEN
                  CALL DQUAD(2,3,4,5)
                  CALL TRTAN(6,7)
                  CALL TRSING(8)
                  CALL TRSING(9)
                  ICT(1)=5
                  ICT(2)=5
                  ICT(3)=2
                  ICT(4)=7
                  GOTO 100
        ELSE
        ENDIF
        IF(IAX25.LE.200.AND.IAXT78.LE.100) THEN
            CALL DQUINT(2,3,4,5,6)
            CALL TRTRI(7,8,9)
            ICT(1)=3
            ICT(2)=6
            ICT(3)=3
            GOTO 100
        ELSE
        ENDIF
  100   CALL SUM
        CALL COAL
  200   CALL PRN
        GOTO 22
 1234   CONTINUE
        WRITE (8,551) NERRORS1
        WRITE (8,552) NERRORS2
        WRITE (8,553) NERRORS3
        WRITE (8,554) NERRORS4
        WRITE (8,555) NERRORS5
        WRITE (8,556) NGOOD
  551   FORMAT (I6,5X,'AXLE(S) WITHOUT WEIGHT')
  552   FORMAT (I6,5X,'AXLE SPACING(S) LESS THAT 1.5 FEET')
  553   FORMAT (I6,5X,'MISMATCHES BETWEEN NUMBERS OF AXLE WEIGHTS AND SPAC
       *INGS')
  554   FORMAT (I6,5X,'>10 PERCENT DIFFERENCE BETWEEN GROSS WEIGHT AND SUM
       * OF AXLE WEIGHT')
  555   FORMAT (I6,5X,'>10 PERCENT DIFFERENCE BETWEEN WHEELBASE AND SUM OF
       * AXLE SPACINGS')
  556   FORMAT (I6,5X,'SUCCESSFUL ENTRIES')
        RETURN
        END
C   * * * * END OF MAIN PROGRAM * * * * * * * *
        SUBROUTINE INIT
        COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
        COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
        COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
        COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
        COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
        DO 10 I=1,13
        IAW(I)=0
```

```
      ISW(I)=0
   10 CONTINUE
      DO 11 J=1,12
      IAXS(J)=0
   11 CONTINUE
      DO 12 K=1,6
      ICT(K)=O
   12 CONTINUE
      DO 14 IK=1,10
      ICONTN(IK)=O
   14 CONTINUE
      ITWRC=O
      AV=O.
      ICK=O
      ISC=O
      FC=O
      SIN=O
      IYR=O
      IDT=O
      IAMO=O
      IDAY=O
      IHOUR=O
      IVTC=O
      ITWT=O
      ITOTWT=O
      IAXTOT=O
      IRSN=O
      ICMOD=99999
      CNVEQ=O
      CV=O
      IAX2=O
      IAX3=O
      IAXD=O
      IAXTR=O
      IAXD4=O
      IAXT4=O
      IAXD3=O
      IAXT3=O
      IAXT3=O
      IAXT6=O
      IAXT5=O
      IAXT31=O
      IAXT32=O
      IAXTOT=O
      JCONT=1
      IAFC=O
      RETURN
      END
      SUBROUTINE STEER
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(1)
      B=FBEQ(1)
      GOTO 25
   20 A=RAEQ(1)
      B=RBEQ(1)
   25 CONTINUE
      W=100*IAW(1)
      S=O.O1*(A+B*W)
      ISW(1)=S
```

```
      RETURN
      END
      SUBROUTINE DRSING
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(2)
      B=FBEQ(2)
      GOTO 25
   20 A=RAEQ(2)
      B=RBEQ(2)
   25 CONTINUE
      W=100*IAW(2)
      S=0.01*(A+B*W)
      ISW(2)=S
      RETURN
      END
      SUBROUTINE TRSING(I1)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(3)
      B=FBEQ(3)
      GOTO 25
   20 A=RAEQ(3)
      B=RBEQ(3)
   25 CONTINUE
C   3 FORMAT(' TRSING =',I5,' IAW(I1)=',I5,' ISW(I1)=',I5)
      W=100*IAW(I1)
      S=0.01*(A+B*W)
      ISW(I1)=S
C     WRITE(*,3)I1,IAW(I1),ISW(I1)
      RETURN
      END
      SUBROUTINE DRTAN(I1,I2)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(4)
      B=FBEQ(4)
      GOTO 25
   20 A=RAEQ(4)
      B=RBEQ(4)
   25 CONTINUE
      P1=100*IAW(I1)
      P2=100*IAW(I2)
      W=P1+P2
      WT=(A+B*W)
      ISW(I1)=0.01*WT*P1/W
      ISW(I2)=0.01*WT*P2/W
      RETURN
      END
      SUBROUTINE TRTAN(I1,I2)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
```

```
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(5)
      B=FBEQ(5)
      GOTO 25
   20 A=RAEQ(5)
      B=RBEQ(5)
   25 CONTINUE
      P1=100*IAW(I1)
      P2=100*IAW(I2)
      W=P1+P2
      WT=(A+B*W)
      ISW(I1)=0.01*WT*P1/W
      ISW(I2)=0.01*WT*P2/W
      RETURN
      END
      SUBROUTINE DRTRI(I1,I2,I3)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(6)
      B=FBEQ(6)
      GOTO 25
   20 A=RAEQ(6)
      B=RBEQ(6)
   25 CONTINUE
      P1=100*IAW(I1)
      P2=100*IAW(I2)
      P3=100*IAW(I3)
      W=P1+P2+P3
      WT=(A+B*W)
      ISW(I1)=0.01*WT*P1/W
      ISW(I2)=0.01*WT*P2/W
      ISW(I3)=0.01*WT*P3/W
      RETURN
      END
      SUBROUTINE TRTRI(I1,I2,I3)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      IF(ITYPE.EQ.2) GOTO 20
      A=FAEQ(7)
      B=FBEQ(7)
      GOTO 25
   20 A=RAEQ(7)
      B=RBEQ(7)
   25 CONTINUE
      P1=100*IAW(I1)
      P2=100*IAW(I2)
      P3=100*IAW(I3)
      W=P1+P2+P3
      WT=(A+B*W)
      ISW(I1)=0.01*WT*P1/W
      ISW(I2)=0.01*WT*P2/W
      ISW(I3)=0.01*WT*P3/W
      RETURN
      END
```

```
      SUBROUTINE DQUAD(I1,I2,I3,I4)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      CALL DRTAN(I1,I2)
      CALL DRTAN(I3,I4)
      RETURN
      END
      SUBROUTINE TQUAD(I1,I2,I3,I4)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      CALL TRTAN(I1,I2)
      CALL TRTAN(I3,I4)
      RETURN
      END
      SUBROUTINE DQUINT(I1,I2,I3,I4,I5)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      CALL DRTAN(I1,I2)
      CALL DRTRI(I3,I4,I5)
      RETURN
      END
      SUBROUTINE TQUINT(I1,I2,I3,I4,I5)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      CALL TRTAN(I1,I2)
      CALL TRTRI(I3,I4,I5)
      RETURN
      END
      SUBROUTINE SEXTET(I1,I2,I3,I4,I5,I6)
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      CALL TRTRI(I1,I2,I3)
      CALL TRTRI(I4,I5,I6)
      RETURN
      END
      SUBROUTINE SUM
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      ITOTWT=O
      DO 30 J=1,13
      ITOTWT=ITOTWT+ISW(J)
   30 CONTINUE
      RETURN
      END
C  SUBROUTINE COAL FOLLOWS
      SUBROUTINE COAL
```

```
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
      ICMOD=99999
      ATOTWT=ITOTWT*.1
      AX3=IAXS(3)*0.1
      IF(ICT(1).EQ.3.AND.ICT(2).EQ.3.AND.ICT(3).EQ.2.AND.ITOTWT
     1 .GT.800) THEN
          CNVEQ=C5NA+C5NB*ATOTWT+C5NS
          GOTO 10
      ELSE IF(ICT(1).EQ.3.AND.ICT(2).EQ.3.AND.ICT(3).EQ.3.AND.ITOTWT
     1 .GT.900) THEN
          CNVEQ=C6NA+C6NB*ATOTWT+C6NS
          GOTO 10
      ELSE IF(ICT(1).EQ.2.AND.ICT(2).EQ.3.AND.ICT(3).EQ.0
     1       .AND.ITOTWT.GT.600) THEN
          ICMOD=11200
          GO TO 15
      ELSE IF(ICT(1).EQ.2.AND.ICT(2).EQ.4.AND.ICT(3).EQ.0
     1       .AND.ITOTWT.GT.750) THEN
          ICMOD=11200
          GO TO 15
      ELSE
      END IF
      GO TO 15
   10 CONTINUE
      CV=ATOTWT/AX3
      IF(CV.GE.CNVEQ) THEN
   11 ICMOD=11200
      ELSE
      END IF
   15 CONTINUE
      RETURN
      END
      SUBROUTINE PRN
      COMMON/DAN/FAEQ(7),FBEQ(7),RAEQ(7),RBEQ(7)
      COMMON/COL/C5A,C5B,C6A,C6B,C5NA,C5NB,C6NA,C6NB,C5S,C6S,C5NS,C6NS
      COMMON/DAT1/ITWRC,ISC,FC,SIN,IYR,IAMO,IDAY,IHOUR,ICMOD,ITWT,JCONT
      COMMON/DAT2/IAXS(70),IAXTOT,IRSN,ICONTN(10),ITYPE,ICT(6)
      COMMON/DAT3/IAW(70),ISW(70),BD(9),ICH(6),IDT,ITOTWT,CNVEQ,CV
 1103 WRITE(6,1100)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,(ICT(I),
     1 I=1,6),(BD(J),J=7,9),(ICH(J),J=1,6),ICMOD,ITOTWT,(ISW(L),L=1,5),
     2 (IAXS(M),M=1,4),IAXTOT,IRSN,ICONTN(1)
      IF(ICONTN(1).EQ.0) GOTO 20
      DO 55 KK=1,JCONT
      LRT=KK*8-2
      LOP=KK*8+5
      MRT=KK*8-3
      MOP=KK*8+4
 1102 WRITE(6,1101)ITWRC,ISC,FC,SIN,IDT,IYR,IAMO,IDAY,IHOUR,(ICT(I),
     1  I=1,6),(ISW(L),L=LRT,LOP),(IAXS(M),M=MRT,MOP),IRSN,ICONTN(KK+1)
      IF(ICONTN(KK+1).EQ.9) GOTO 20
   55 CONTINUE
 1100 FORMAT(I1,I2,A2,A3,I1,4I2,6I1,3A1,6I1,3X,I5,1X,I4,9I3,I4,I3,I1)
 1101 FORMAT(I1,I2,A2,A3,I1,4I2,6I1,5X,17I3,I1)
   20 RETURN
      END
```

APPENDIX C

PROGRAM LISTING FOR UNIT ESALS

```
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     SET "IERR" TO DESIRED CODE TO PRINT REJECTED RECORDS
C       1 = EXCESSIVE GROSS WEIGHT
C       2 = NEGLIGIBLE GROSS WEIGHT
C       3 = EXCESSIVE WHELLBASE
C       4 = NEGLIGIBLE WHEELBASE
C       9 = ALL REJECTED RECORDS
C       0 = NO REJECTED RECORDS
C  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER NERRORS,NN,VEHTYPE,CARDC,NCT,NA,NB,IVEH(3,14),
     +NVT,NAR,COMMOD,VEHT
      CHARACTER*64 ALL*80
      DIMENSION SEAL(3,14),EALPAXLE(3,14),EALPVEH(3,14),AXLEPVEH(3,14),
     +W(20),S(20),AXLE(3,14),VEH(3,14)
      IERR=9
      CST1 = -3.540112
      CST2 = 2.72886
      CST3 = .289133
      CSI1 = -3.439501
      CSI2 = .423747
      CSI3 = 1.846657
      CTA1 = -2.979479
      CTA2 = -1.265144
      CTA3 = 2.007989
      CTR1 = -2.740987
      CTR2 = -1.873428
      CTR3 = 1.964442
      CQU1 = -2.589482
      CQU2 = -2.224981
      CQU1 =  1.923512
      NERRORS1=0
      NERRORS2=0
      NERRORS3=0
      NERRORS4=0
      NERRORS5=0
      NGOOD=0
      NN=0
      DO 20 I=1,3
      DO 10 J=1,14
      SEAL(I,J)=0.0
      AXLE(I,J)=0.0
      VEH(I,J)=0.0
      EALPAXLE(I,J)=0.0
      EALPVEH(I,J)=0.0
      AXLEPVEH(I,J)=0.0
      IFC=1
   10 CONTINUE
   20 CONTINUE
  100 CONTINUE
      DO 30 K=1,20
      W(K)=0.0
      S(K)=0.0
   30 CONTINUE
      READ(5,1000,END=888) FC,VEHTYPE,COMMOD,TW,(W(L),L=1,5),(S(L),L=1,4
     +),TS,CARDC,ALL,IYR
 1000 FORMAT(T4,I2,T18,I4,T36,I5,T42,F4.1,T46,9F3.1,T73,F4.1,T80,I1,T1,
     +A80,T10,I2)
      IF (IFC.EQ.1) WRITE (8,99) FC,IYR
   99 FORMAT (5X,'FUNCTIONAL CLASS= ',I2,5X,'YEAR= 19',I2,/)
      IFC=IFC+1
      IF (CARDC.EQ.9) GO TO 100
      NVT=0
      NN=NN+1
      IF (VEHTYPE.EQ.1900) NVT=4
```

```
      IF (VEHTYPE.EQ.2200) NVT=5
      IF (VEHTYPE.EQ.2300) NVT=6
      IF (VEHTYPE.EQ.2400) NVT=7
      IF (VEHTYPE.EQ.2500) NVT=7
      IF (VEHTYPE.EQ.3210) NVT=8
      IF (VEHTYPE.EQ.3310) NVT=8
      IF (VEHTYPE.EQ.3220) NVT=8
      IF (VEHTYPE.EQ.3270) NVT=8
      IF (VEHTYPE.EQ.3230) NVT=9
      IF (VEHTYPE.EQ.3320) NVT=9
      IF (VEHTYPE.EQ.3370) NVT=9
      IF (VEHTYPE.EQ.3410) NVT=9
      IF (VEHTYPE.EQ.3330) NVT=10
      IF (VEHTYPE.EQ.3350) NVT=10
      IF (VEHTYPE.EQ.3510) NVT=10
      IF (VEHTYPE.EQ.3340) NVT=10
      IF (VEHTYPE.EQ.3240) NVT=10
      IF (VEHTYPE.EQ.3430) NVT=10
      IF (VEHTYPE.EQ.3440) NVT=10
      IF (VEHTYPE.EQ.3450) NVT=10
      IF (VEHTYPE.EQ.3390) NVT=10
      IF (VEHTYPE.EQ.3520) NVT=10
      IF (VEHTYPE.EQ.3540) NVT=10
      IF (VEHTYPE.EQ.5212) NVT=11
      IF (VEHTYPE.EQ.5222) NVT=12
      IF (VEHTYPE.EQ.5312) NVT=12
      IF (VEHTYPE.EQ.5329) NVT=13
      IF (VEHTYPE.GE.5400.AND.VEHTYPE.LE.5499) NVT=13
      IF (VEHTYPE.EQ.5531) NVT=13
      VEHT=VEHTYPE/100
      IF (VEHT.EQ.92.AND.CARDC.NE.9)  NERRORS5=NERRORS5+1
      IF (NVT.NE.0) GO TO 110
      IF (NVT.EQ.0) GO TO 100
  110 CONTINUE
      IF (COMMOD.EQ.99999) THEN
      NCT=1
      ELSE
      NCT=2
      END IF
      IF (CARDC.EQ.1) READ (5,1100) (W(L),L=6,13),(S(L),L=5,12),ALL2
 1100 FORMAT(T29,8F3.1,T53,8F3.1,T1,A80)
      SUMW=0.0
      SUMS=0.0
      NA=0
      NB=0
      DO 40 I=1,15
      IF (W(I).GT.0.0) NA=NA+1
      SUMW=SUMW+W(I)
      IF (S(I).GT.0.0) NB=NB+1
      SUMS=SUMS+S(I)
   40 CONTINUE
C*********************************************************************
C   THE FOLLOWING SECTION CHECKS FOR ERRORS
C*********************************************************************
C     NERRORS1 = NO. OF ERRORS:  EXCESSIVE GROSS WEIGHT
C     NERRORS2 = NO. OF ERRORS:  NEGLIGIBLE GROSS WEIGHT
C     NERRORS3 = NO. OF ERRORS:  EXCESSIVE WHEELBASE
C     NERRORS4 = NO. OF ERRORS:  NEGLIGIBLE WHEELBASE
C     NERRORS5 = NO. OF ERRORS:  VEHICLE TYPE "921"
C*********************************************************************
      ALW=(20+(NVT-4)*45/9)/10.0
      UW=(660+(NVT-4)*1000/6)/10.0
      ALS=(20+(NVT-4)*104/9)/10.0
      US=(460+(NVT-4)*540/9)/10.0
```

```
      IF (TW.GT.UW.OR.TW.GT.180) THEN
          NERRORS1=NERRORS1+1
      IF (IERR.NE.1.AND.IERR.NE.9) GO TO 100
      WRITE (8,1111) ALL
 1111 FORMAT(T1,A80)
      WRITE (8,11111) NVT,TW,UW,NERRORS1
11111 FORMAT(5X,'NVT=',I2,5X,'TW= ',F5.1,5X,'UW= ',F5.1,5X,'EXCESSIVE GR
     +OSS WT  ERROR NO.',I5)
          GO TO 100
      ELSE
      END IF
      IF (TW.LT.ALW) THEN
          NERRORS2=NERRORS2+1
      IF (IERR.NE.2.AND.IERR.NE.9) GO TO 100
      WRITE (8,1112) ALL
 1112 FORMAT(T1,A80)
      WRITE (8,2222) NVT,TW,ALW,NERRORS2
 2222 FORMAT(5X,'NVT=',I2,5X,'TW= ',F5.1,5X,'LW= ',F4.1,5X,'NEGLIGIBLE G
     +ROSS WT  ERROR NO.',I5)
          GO TO 100
      ELSE
      END IF
      IF (TS.GT.US.OR.TS.GT.100) THEN
          NERRORS3=NERRORS3+1
      IF (IERR.NE.3.AND.IERR.NE.9) GO TO 100
          WRITE (8,1113) ALL
 1113 FORMAT(T1,A80)
          WRITE (8,3333) NVT,TS,US,NERRORS3
 3333 FORMAT(5X,'NVT=',I2,5X,'TS= ',F5.1,5X,'US= ',F4.1,5X,'EXCESSIVE WH
     +EELBASE   ERROR NO.',I5)
          GO TO 100
      ELSE
      END IF
      IF (TS.LT.ALS) THEN
          NERRORS4=NERRORS4+1
      IF (IERR.NE.4.AND.IERR.NE.9) GO TO 100
          WRITE (8,1114) ALL
 1114 FORMAT(T1,A80)
          WRITE (8,4444) NVT,TS,ALS,NERRORS4
 4444 FORMAT(5X,'NVT=',I2,5X,'TS= ',F5.1,5X,'LS= ',F4.1,5X,'NEGLIGIBLE W
     +HEELBASE   ERROR NO.',I5)
          GO TO 100
      ELSE
      END IF
      IF (CARDC.NE.9.) NGOOD=NGOOD+1
      NAR=NA
      EAL=0.0
C***********************************************************************
C     THE NEXT STATMENT LIMITS THE EALS FOR STEERING AXLE LOADS IN
C     EXCESS OF 22,500 LBS.
C***********************************************************************
      IF (W(1).GT.22.5) THEN
      EAL=EAL+5.39
      ELSE
      EAL=EAL+10.0**(CST1+CST2*ALOG10(W(1))+CST3*ALOG10(W(1))**2)
      END IF
      NAR=NAR-1
      DO 50 I=1,14
      W(I)=W(I+1)
      S(I)=S(I+1)
   50 CONTINUE
  200 CONTINUE
      IF (NAR.EQ.0) GO TO 300
C***********************************************************************
```

```
C     THE NEXT STATMENT LIMITS THE EALS FOR QUAD AXLE GROUPS WITH LOADS
C     IN EXCESS OF 150,000 LBS.
C*********************************************************************
      IF (S(1).GT.0.0.AND.S(2).GT.0.0.AND.S(3).GT.0.0.AND.(S(1)+S(2)+S(3
     +)).LE.15.0.AND.(W(1)+W(2)+W(3)+W(4)).LE.150.0) THEN
      EAL=EAL+10.0**(CQ01+CQ02*ALOG10(W(1)+W(2)+W(3)+W(4))+CQ03*
     +ALOG10(W(1)+W(2)+W(3)+W(4))**2)
      ELSE IF (S(1).GT.0.0.AND.S(2).GT.0.0.AND.S(3).GT.0.0.AND.(S(1)+S(2
     +)+S(3)).LE.15.0.AND.(W(1)+W(2)+W(3)+W(4)).GT.150.0) THEN
      EAL=EAL+58.3
      NAR=NAR-4
      DO 55 I=1,14
      W(I)=W(I+4)
      S(I)=S(I+4)
   55 CONTINUE
      GO TO 200
      END IF
C*********************************************************************
C     THE NEXT STATMENT LIMITS THE EALS FOR TRIDEM AXLE GROUPS WITH
C     LOADS IN EXCESS OF 112,500 LBS.
C*********************************************************************
      IF (S(1).GT.0.0.AND.S(2).GT.0.0.AND.(S(1)+S(2)).LE.10.0.AND.
     +(W(1)+W(2)+W(3)).LE.112.5) THEN
      EAL=EAL+10.0**(CTR1+CTR2*ALOG10(W(1)+W(2)+W(3))+CTR3*
     +ALOG10(W(1)+W(2)+W(3))**2)
      ELSE IF (S(1).GT.0.0.AND.S(2).GT.0.0.AND.(S(1)+S(2)).LE.10.0.AND.
     +(W(1)+W(2)+W(3)).GT.112.5) THEN
      EAL=EAL+59.0
      NAR=NAR-3
      DO 60 I=1,14
      W(I)=W(I+3)
      S(I)=S(I+3)
   60 CONTINUE
      GO TO 200
      END IF
C*********************************************************************
C     THE NEXT STATMENT LIMITS THE EALS FOR TANDEM AXLE GROUPS WITH
C     LOADS IN EXCESS OF 75,000 LBS.
C*********************************************************************
      IF (S(1).GT.0.0.AND.S(1).LE.5.0.AND.(W(1)+W(2)).GT.75.0) THEN
      EAL=EAL+63.0
      ELSE IF (S(1).GT.0.0.AND.S(1).LE.5.0.AND.(W(1)+W(2)).LE.75.0)
     +THEN
      EAL=EAL+10.0**(CTA1+CTA2*ALOG10(W(1)+W(2))+CTA3*ALOG10(W(1)+
     +W(2))**2)
      NAR=NAR-2
      DO 70 I=1,14
      W(I)=W(I+2)
      S(I)=S(I+2)
   70 CONTINUE
      GO TO 200
      END IF
C     IF (W(1).EQ.0.0)  WRITE (7,1170) NN
C1170 FORMAT (T5,'NN=',I5)
C*********************************************************************
C     THE NEXT STATMENT LIMITS THE EALS FOR OTHER SINGLE AXLES WITH
C     LOADS IN EXCESS OF 37,500 LBS.
C*********************************************************************
      IF (W(1).GT.37.5) THEN
      EAL=EAL+78.6
      ELSE
      EAL=EAL+10.0**(CSI1+CSI2*ALOG10(W(1))+CSI3*(ALOG10(W(1))**2))
      END IF
      NAR=NAR-1
```

```
      DO 80 I=1,14
      W(I)=W(I+1)
      S(I)=S(I+1)
   80 CONTINUE
      GO TO 200
  300 CONTINUE
      SEAL(NCT,NVT)=SEAL(NCT,NVT)+EAL
      AXLE(NCT,NVT)=AXLE(NCT,NVT)+NA
      VEH(NCT,NVT)=VEH(NCT,NVT)+1
      GO TO 100
  888 CONTINUE
      DO 90 J=1,13
      SEAL(3,J)=SEAL(1,J)+SEAL(2,J)
      AXLE(3,J)=AXLE(1,J)+AXLE(2,J)
      VEH(3,J)=VEH(1,J)+VEH(2,J)
   90 CONTINUE
      DO 92 I=1,3
      DO 91 J=1,13
      SEAL(I,14)=SEAL(I,14)+SEAL(I,J)
      AXLE(I,14)=AXLE(I,14)+AXLE(I,J)
      VEH(I,14)=VEH(I,14)+VEH(I,J)
   91 CONTINUE
   92 CONTINUE
      DO 94 I=1,3
      DO 93 J=1,14
      IF (AXLE(I,J).GT.0.) THEN
      EALPAXLE(I,J)=SEAL(I,J)/AXLE(I,J)
      ELSE
      EALPAXLE(I,J)=0.0
      ENDIF
      IF (VEH(I,J).GT.0.0) THEN
      EALPVEH(I,J)=SEAL(I,J)/VEH(I,J)
      ELSE
      EALPVEH(I,J)=0.0
      ENDIF
      IF (VEH(I,J).GT.0.0) THEN
      AXLEPVEH(I,J)=AXLE(I,J)/VEH(I,J)
      ELSE
      AXLEPVEH(I,J)=0.0
      ENDIF
   93 CONTINUE
   94 CONTINUE
      WRITE (6,1172) FC,IYR
 1172 FORMAT (T5,'FUNCTIONAL CLASS ',I2,5X,'YEAR 19',I2)
      WRITE (6,1173)
 1173 FORMAT (T5,'NUMBER OF VEHICLES WEIGHED')
      WRITE (6,1174)
 1174 FORMAT (T5,'VEH TYPE',5X,'NON-COAL',5X,'COAL',5X,'ALL')
      DO 95 J=4,14
      IVEH(1,J)=VEH(1,J)
      IVEH(2,J)=VEH(2,J)
      IVEH(3,J)=VEH(3,J)
      WRITE (6,1180) J,IVEH(1,J),IVEH(2,J),IVEH(3,J)
 1180 FORMAT (T8,I2,T19,I6,T29,I6,T37,I6)
   95 CONTINUE
      WRITE (6,1200)
 1200 FORMAT (T5,'EALS PER AXLE')
      WRITE (6,1174)
      DO 96 J=4,14
      WRITE (6,1210) J,EALPAXLE(1,J),EALPAXLE(2,J),EALPAXLE(3,J)
 1210 FORMAT (T8,I2,T18,F8.5,T27,F8.5,T36,F8.5)
   96 CONTINUE
      WRITE (6,1300)
 1300 FORMAT (T5,'EALS PER VEHICLE')
```

```
      WRITE (6,1174)
      DO 97 J=4,14
      WRITE (6,1210) J,EALPVEH(1,J),EALPVEH(2,J),EALPVEH(3,J)
   97 CONTINUE
      WRITE (6,1400)
 1400 FORMAT (T5,'AXLES PER VEHICLE')
      WRITE (6,1174)
      DO 98 J=4,14
      WRITE (6,1210) J,AXLEPVEH(1,J),AXLEPVEH(2,J),AXLEPVEH(3,J)
   98 CONTINUE
      WRITE (8,556) NERRORS1
      WRITE (8,557) NERRORS2
      WRITE (8,558) NERRORS3
      WRITE (8,559) NERRORS4
      WRITE (8,555) NERRORS5
      WRITE (8,560) NGOOD
  556 FORMAT (//I6,5X,'EXCESSIVE GROSS WEIGHT(S)')
  557 FORMAT (I6,5X,'NEGLIGIBLE GROSS WEIGHT(S)')
  558 FORMAT (I6,5X,'EXCESSIVE WHEELBASE(S)')
  559 FORMAT (I6,5X,'NEGLIGIBLE WHEELBASE(S)')
  555 FORMAT (I6,5X,'VEHICLE TYPE "921"')
  560 FORMAT (/I6,5X,'SUCCESSFUL ENTRIES')
      RETURN
      END
```

APPENDIX D

POSSIBLE FUTURE REVISION USING dBASE

POSSIBLE REVISION OF ESAL SUMMARIES

November 11, 1992
Revised: February 5, 1993

The process for annually calibrating Kentucky ESAL models has lost some of its original convenience and reliability as modifications have been made to accommodate changes to the traffic monitoring program and, especially, its data-collection component. An alternative is proposed herein as a relatively inexpensive way to modify the current process to restore its convenience and reliability as well as to enhance its capabilities. Modified versions of current programs, LOADMTR and CLASSUM, would continue to be used[4]. For weight data, each station would be processed independently (functional class summations would be eliminated), and output data would be stored by station. The current WIM and UNITEAL programs would be added to LOADMTR. CLASSUM processing and output would remain essentially unchanged. EALCALC and SMOOTH would be replaced by a dBASE program which would merge the necessary databases and produce the output reports.

REVISIONS TO MORE ACCURATELY TREAT COAL MOVEMENTS

Because coal movements place such a large burden on Kentucky's highway system, special emphasis is placed on them in the ESAL models. Current procedures suffer, however, in two very important respects. First, a highway segment can be identified as a coal-haul highway only when a manual classification count has been taken there. Segments not included within the classification count program and those for which automatic classifiers are used cannot be identified as coal-haul highways and, hence, cannot be treated as such in the ESAL estimations. Second, the separation of weight data into normal and heavy/coal categories (without regard for the type of highway on which it was collected) has screened overloaded and coal trucks from the traffic loading applied to non-coal highways. This results in an underestimate of the ESAL accumulations on these facilities.

Proposed herein is a modification that uses annual coal-haul-road data to identify coal-haul highways. Weight data would be summarized 1) for both classified vehicles (normal and heavy/coal) and unclassified vehicles and 2) for both non-coal-haul highways and coal-haul highways. Its use in the ESAL models would be as follows:

---

[4]These two programs remain both functional as well as convenient to use.

Type of Highway	Type of Classification Data	
	Manual	Automatic
Non-Coal	Unclassified Weight Data for Non-Coal Highways	Unclassified Weight Data for Non-Coal Highways
Coal	Classified Weight Data for Coal Highways	Unclassified Weight Data for Coal Highways

Heavy/coal trucks would be identified as they are at present. In the WIM data, gross weight is the primary factor. In the manual classification data, truck body style and commodity are used. Coal trucks are not identified at automatic classifier sites.

FILE STRUCTURE AND CONTENTS

The following files, on PC media, provide the necessary input to a new dBASE program:

File 1 - Weight Data

Source: Produced by LOADMTR

Structure: Read-only file with input data processed and output records grouped by station

Contents:
- Year
- Station
- County
- Route
- Milepoint
- ESALs per axle by vehicle type by commodity type (normal, heavy/coal, and unclassified)
- Number of axles per vehicle by vehicle type by commodity type
- Number of vehicles weighed by vehicle type by commodity type

Code: LOADMTR code must be changed to 1) add WIM processing, 2) add/modify edit specifications, 3) add ESAL computations, 4) produce new output file, and 5) remove old file output if desired

Comments: Would no longer have full capacity to obtain alternate axle load distributions by backward searches and alternate axle types. However, the dBASE program could merge as much historical data as required. A separate program--inputting LOADMTR output and producing required output--has already been written (last year).

File 2 - Classification Data

Source:	Produced by CLASSUM
Structure:	Read-only file with output records grouped by station
Contents:	Year Station County Route Milepoint AADT (as supplied by Kentucky Transportation Cabinet) Percentages by vehicle type Percentage of coal trucks (manual counts only)
Code:	Code must be changed only to produce file output and possibly to remove any previous file output
Comments:	None

File 3 - Volume Data (TVS)

Comments:	The separate volume file could be accessed later. For now, volumes would be taken from classification data.

File 4 - Station Identification

Source:	New file created and maintained manually
Structure:	One record for each WIM or classification station
Contents:	Station County Route Milepoint Functional classification Geographical area ???

File 5 - Coal-Haul Road System (YR_COAL.SEG)

Source:	File produced annually from coal-haul report database
Structure:	???
Contents:	County Route Start milepoint End milepoint

Annual coal tonnage - cardinal direction
Annual coal tonnage - non-cardinal direction
Annual coal tonnage - both directions

File 6 - Summary File

A read-only summary file--similar to that currently used--might be desirable for the purpose of conveniently developing "smoothed" estimates.

dBASE PROGRAM

1. We need to address how coal trucks (identified on classification records) would be distributed to the various truck types. In the past we have used weight data and could continue to do so. However, the algorithm may require modification.
2. Need to be able to process weight and classification data as averages or moving averages
3. Consideration must be given to how best to classify roads. Certainly one element will be functional classification. Another must relate to the extent of coal haulage. (Will this be determined from weight data, classification data, coal-haul road designation, a combination of all, or ...?) The Division of Planning needs great flexibility for classifying roads. For example, the capacity should be provided to generate reports on various ESAL parameters for volume groups or geographical areas within a functional class (with the appropriate statistical level of confidence specified).
4. The process would continue to be driven largely by classification data. Average weight data (for each highway category) would be integrated with site-specific vehicle classification data to produce site-specific traffic parameters.
5. Output (similar to current output)
    a. Traffic parameters for classification station within most recent three years (weight data accumulated and averaged at level of highway class)
    b. Traffic parameters by highway class (averaged for three most recent years) without time series data
    c. Traffic parameter regression by highway class with current incremental percentages without time series data
    d. Growth increments as a time series

OVERALL PROCESS

1. Run modified versions of current programs LOADMTR and CLASSUM with hard-copy output similar to present. NOTE: LOADMTR has been run as a summary program adding together data for stations of the same functional class.

Future runs would handle each station individually: the printed output would be excessive unless some of it was suppressed.

2. Edit data using those modifications currently under development

COMMENTS AND QUESTIONS

1. Thus far no consideration has been given to such questions as: 1) is available data being fully exploited, 2) should the data collection program be modified, 3) how reliable is the data and the estimates, 4) whether truck classification data could or should be developed from the weight database, etc.

   Response from Division of Planning: Classification data is developed by KYTC from the WIM data.

2. What kind of documentation would be necessary if this revision were made?

   Response from Division of Planning: A report would be needed.

3. How much reprocessing of prior-year data would be necessary?

   Response from Division of Planning: No reprocessing of data should be needed prior to 1989. Our WIM program started then and our classification data was significantly upgraded due to the implementation of the three-year HPMS cycle.

4. Should classification data be processed by functional class? In any case must CLASSUM be changed with respect to the sample factoring process (page 40)?

   Response from Division of Planning: Sample factoring will ultimately be tied to the new Traffic Monitoring Standards recommendations. ASTM E 1442-91 (attached) calls for use of permanent classification data to factor short counts.

5. Is additional investigation of adequacy of WIM-to-static conversions necessary?

   Response from Division of Planning: WIM data has been rightfully alluded to in every ESAL revision discussion. As you know, we have four different types of WIM devices (Golden River, PAT, IRD, Toledo Scale) which have not been linked exhaustively to a static standard or each other. This is a fertile area for improvement.

6. It is assumed that all subsequent weight data will require WIM-to-static conversions.

   Response from Division of Planning: WIM-to-static conversions are necessary until more accuracy is shown by WIM equipment.

7. Recent data seem to be inflating "winter" volumes. Should we reexamine this situation, perhaps considering better use of multi-seasonal classification counts to develop the expansion factors?

   Response from Division of Planning: Please explain how winter volumes are inflated since we are unaware of this trend. As mentioned above, multi-season and continuous data must be used in the future.

8. Should dBASE output be hard copy, computer accessible, or both?

   Response from Division of Planning: dBASE output should be both computer accessible and hard copy.