TRAIN PERFORMANCE CALCULATOR FOR THE IBM PC

A Thesis
Presented to
the Department of Civil Engineering
The Faculty of Engineering
The University of Manitoba

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in Civil Engineering

by
Robert Richard Girling
April 1986

[redacted]

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author would like to thank Professor A. Clayton for his encouragement and support of this project.

A special thanks goes out to Mr. S.C. Rawal and Mr. T.S. Wee of Canadian Pacific's Planning and Research department for their assistance, and who provided valuable information and sample runs of CP's train performance calculator.

ABSTRACT

A train performance calculator program was written in Basic for the IBM Personal Computer. The train performance calculator simulates the operation of a single train along a specific route. Velocity, time, and fuel consumption of a train at any point along a route is predicted by the program. The train may be of any configuration and the route may have any combination of grades and curvature.

The algorithm used in this train performance calculator finds the minimum travel time of a train over a route, subject to power, braking and route constraints.

The program and the models used to predict train performance are described, followed by sample runs.

Chapter I

INTRODUCTION

A train performance calculator (TPC) is a computer program which simulates a single train of any configuration operating over a route with any combination of grades and curvature. This TPC program finds the minimum travel time for a given train and a given route through maximum allowable acceleration and deceleration. Deceleration is based on a full service application of train air brakes.

Train performance calculators are used in railroad operations and research to (1):

1. Determine costs of railway operations.
2. Determine fuel consumption.
3. Determine the tonnage rating of locomotives based on the required minimum speed over the ruling grade.
4. Compare running a specific train over different routes.
5. Study effects of track relocation, grade and curvature reduction, or new construction.
6. Generate output needed for route capacity models.
7. To study the sensitivity of train operations to variables such as speed, gradient, and curvature.

The TPC can also be used as a tool to teach students vehicle behavior.

The TPC program described in this paper is different from most other TPC programs in that it runs on an IBM personal computer, rather than a mainframe computer system. The IBM PC increases the accessibility of a TPC program to more people, including middle level transportation managers. This increased accessibility may facilitate optimization of train operations to be done more often at the regional and divisional levels, which should result in increased productivity and efficiency.

Other advantages of this particular TPC program include its ease of use, the graphical display of output data, and the use of a realistic braking model.

The TPC program can be divided into three sections, input, output, and the main program itself. Input consists of track, consist, and locomotive data. Output includes time and velocity of a train at any point along its route, as well as an estimation of fuel consumption. The main program uses train resistance, motive power, braking, and train operations models to simulate train operations over a route. The following pages describe the components of the program in more detail, followed by a discussion of verification studies and other improvements which could be made. Appendix A contains a User's Guide describing the operation of the program.

Chapter II

TRAIN PERFORMANCE SIMULATION

The operation of a train over a route is simulated by using four models. The models are used to calculate train resistance, tractive effort, braking force, and, in combination, train handling behavior. By using the predictions of these four models, the velocity of the train and its position along the track can be determined at any time. Train handling is done with the objective of minimizing travel time over the given route. Trains accelerate at maximum horsepower where possible and decelerate by using a full service brake application.

2.1 TRAIN RESISTANCE MODEL

Train resistance is the resistance of a train to motion and is made up of three components, inherent resistance of a train on level tangent track, grade resistance and curve resistance.

2.1.1 Inherent Resistance

Resistance to forward motion by a train on level, tangent track, in still air, is modelled by using the following general expression (5):

$$TRI=A+BV+CV*V$$

where TRI = inherent train resistance in pounds

V = train speed in mph

A,B,C = empirical coefficients which are partially a function of train weight, number of axles, and streamlining

This equation incorporates resistance of the train as a result of rolling friction, bearing resistance, flange nosing and air resistance. It is empirically derived through actual in-field dynamometer tests on trains. Commonly used coefficients for this equation are given in Table 1. In the program, CP Rail's coefficients are used for default values. (A default value is a typical "built in" value that is used by a computer program in the absence of a value specified by the user). Plots of three train resistance equations for a typical 100 ton capacity freight car are shown in Figure 1, using the coefficients given in Table 1. Figure 1 illustrates the parabolic behavior of train resistance due to air resistance.

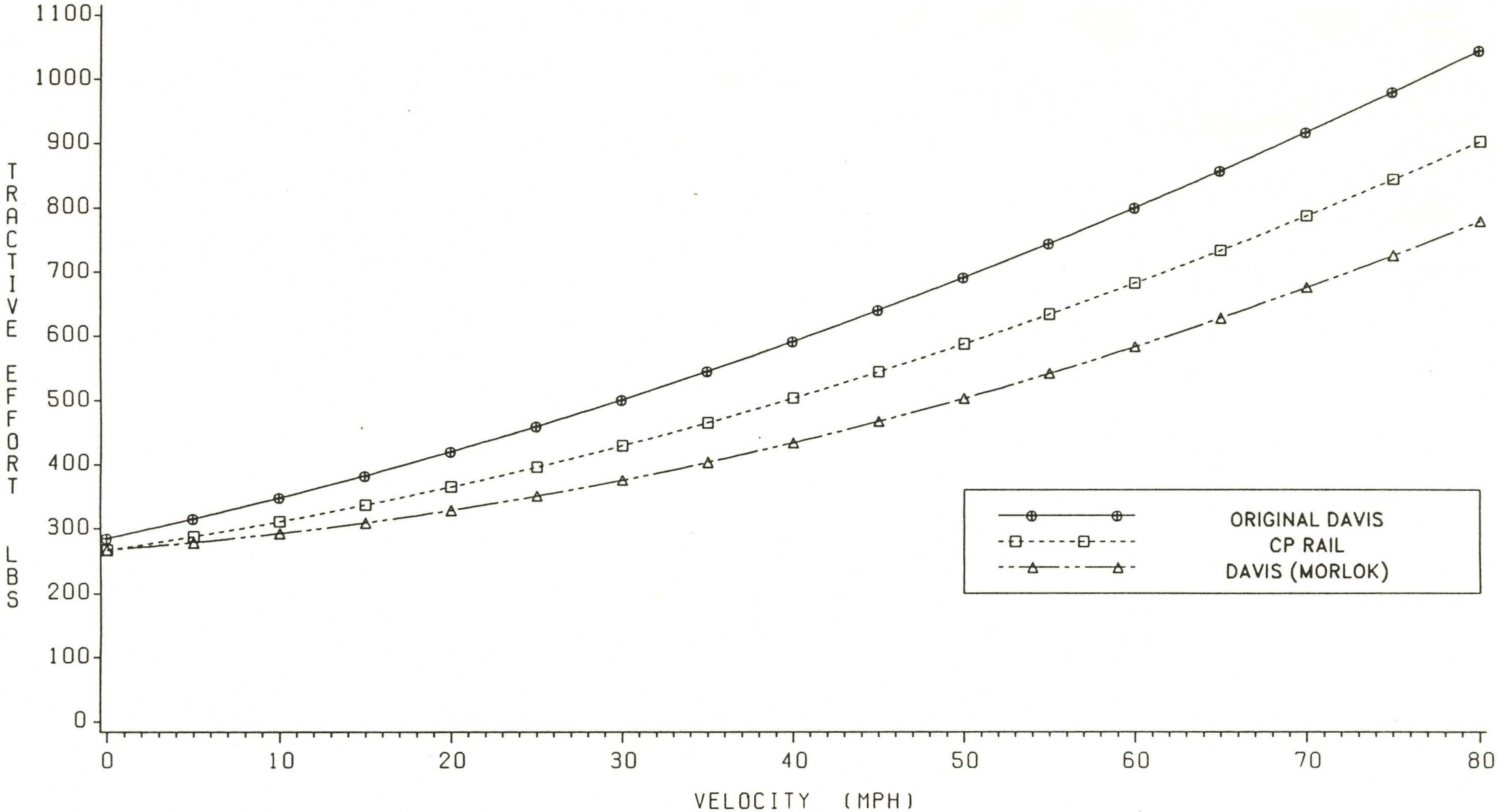


FIG.1 TRAIN RESISTANCE CURVES
100 TON CAPACITY FREIGHT CAR

TABLE 1

Resistance Coefficients

Source	Type	A	B	C
Davis (3)	Diesel	1.3T+29N	0.03T	.288
	Std. Freight	1.5T+18.125N	0.015T	.055
	Piggyback	0.6T+20N	0.01T	.200
CN (4)	Diesel	1.5T+18N	0.03T	.066
	Std. Freight	1.5T+18N	0.03T	.050
	Piggyback	1.5T+18N	0.03T	.102
Davis (2)	Diesel	1.3T+29N	0.03T	.288
	Std. Freight	1.3T+29N	0.045T	.045
	Piggyback	-	-	-

N is number of axles on vehicle
T is vehicle weight in tons

2.1.2 Grade Resistance

Grade resistance is the component of train weight tending to roll a car down an incline. Grade resistance can be positive or negative, depending on the grade. The expression used to model grade resistance is:

$$TRG=T*\sin(\arctan(g/100))*2000$$

where TRG = grade resistance in pounds

T = train weight in tons

g = grade in %

However, for the small grades (<2%) used on railroads, grade resistance can be approximated by TRG=g*T*20.

2.1.3 Curvature Resistance

Curvature resistance is the added resistance on a train due to rounding a curve and represents the work expended in overcoming the flange/rail friction (2). Curvature resistance is given by Hay (2) as approximately 0.8 pounds per ton per degree curve. This resistance applies only to that part of a train which is fully into the curve. This value is also empirically derived through field tests. In the program, curvature resistance is converted to an equivalent grade resistance using 0.04% grade per degree curve (2) as follows:

GEC=.04*D

Where GEC = curvature grade equivalent in %

D = degree of curve

This is added to the existing grade and calculated in the grade resistance equation to produce an effective grade resistance.

2.1.4 Total Train Resistance

Total train resistance is modelled as the sum of the inherent, grade, and curvature resistances as shown below:

TRT=TRI+TRGE

where TRT = total train resistance

TRI = inherent train resistance

TRGE = effective grade resistance which combines the resistance due to the actual grade and the curvature grade equivalent

2.2 MOTIVE POWER MODEL

The motive power model simulates the pulling effort of diesel-electric locomotives. From mechanics, Power=Force * Velocity. Rearranging this equation by solving for force and adjusting for units, the tractive effort of a diesel electric locomotive is modelled as (2):

$$TE=375*E*HP/(V*100)$$

where TE = tractive effort available at the rail/wheel interface in pounds

E = locomotive efficiency in %

HP = rated horsepower of the locomotive

V = locomotive speed in mph

375 = conversion factor for units

This equation is used to approximate the actual tractive effort-velocity curves. Exceptions to this approximation occur for high horsepower four axle units where performance control circuitry reduces horsepower at low speeds (<23 mph) to prevent slipping of the four axle unit when in multiple unit operation with six axle units (5).

Locomotive efficiency is the percentage of rated horsepower of the diesel reaching the driver wheel rims. The reduction of rated horsepower is due to parasitic loads, such as cooling fans, as well as generator and traction motor inefficiencies. A typical value for locomotive efficiency of 83% is used as the default value.

Tractive effort is limited by adhesion (ie. locomotive wheels slip if too much power is applied). Therefore, tractive effort is always less than or equal to locomotive weight times the coefficient of adhesion. Tractive effort of a locomotive is constrained by:

$$TE <= f*LWT/100$$

where TE = tractive effort of the locomotive at the rail/wheel interface in pounds

f = coefficient of adhesion

LWT = locomotive weight in pounds

Adhesion values range from 5% up to 35% depending on velocity, rail conditions, and locomotive adhesion control systems. Some modern locomotives (GM SD60's) have adhesion values of up to 25%, due to wheel slip control systems (5). The maximum reliable adhesion value for most trains is about 20 percent. The default value used in this program is 18.5 percent.

Tractive effort is also limited by the maximum coupler knuckle strength, which is specified as 350,000 pounds (5). It was assumed that the train engineer would not risk a train separation by exceeding this value. Therefore, tractive effort is also constrained by the following:

$$TE <= 350{,}000 \text{ lbs}$$

The maximum speed of a locomotive depends on the gear ratio between the traction motors and axles. The most common freight gearing is 62:15, with a top speed of 65 mph. Going over the maximum speed can cause damage to the armature windings in the traction motors, due to centrifugal forces.

The mimimum continuous speed of a locomotive is based on gear ratio as well as the thermal capacity of the traction motors. Going below the minimum continuous speed at full horsepower causes overheating of the traction motors, which results in damage if continued for too long.

The tractive effort curve for a 3000 HP, GM SD40-2 is plotted in Figure 2, using the above equations. This figure shows the drop in tractive effort with increased velocity.

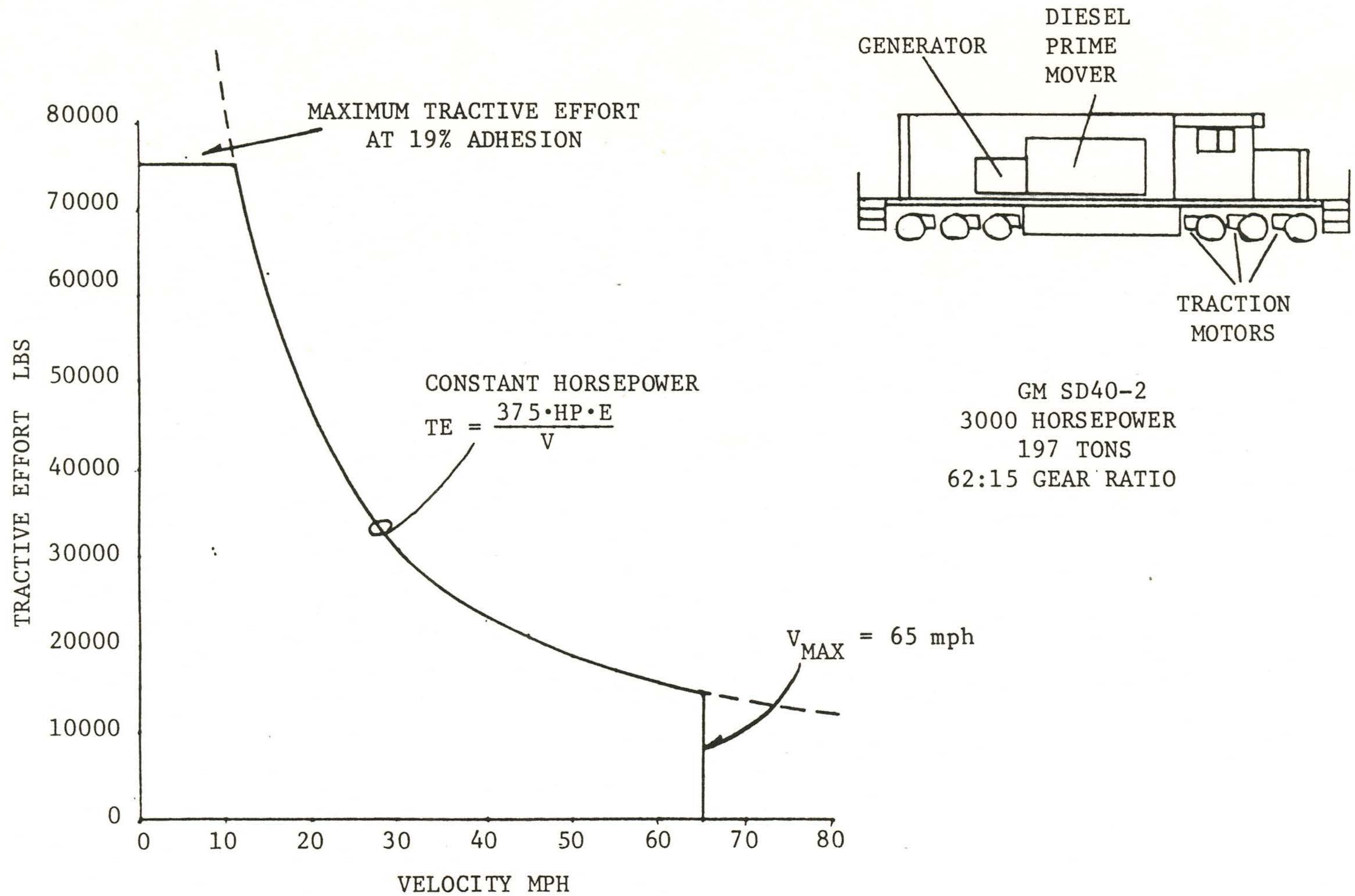


FIG. 2 TRACTIVE EFFORT CURVE
DIESEL ELECTRIC LOCOMOTIVE

2.2.1 Fuel Consumption Estimation

Fuel consumption is estimated by multiplying the work done by the locomotive in foot pounds by a fuel use coefficient given as 0.0324 U.S. gallons per million foot pounds (3,6). Fuel consumption due to idling, power braking by train engineers and trainline air recharge is not taken into account. Power braking is the practice of working the diesels against the train brakes in order to keep slack stretched. Poor correlation between TPC model prediction and actual fuel use have been found in the past, with variations of up to 50% (7). Therefore, not much weight should be placed on fuel consumption figures given by this model until it is verified.

CP Rail uses a more realistic algorithm based on eight different fuel rates for each throttle position of the locomotive (4). Future improvements to the program should incorporate CP Rail's method of calculating fuel consumption.

2.3 BRAKING MODEL

The friction force of a brake shoe acting on the wheel tread of a railway wheel is the coefficient of friction between the brake shoe and wheel times the force on the brake shoe. The maximum frictional resistance at the rail is the coefficient of adhesion times the weight of the car. If the friction force due to the brake shoe is greater than the frictional resistance developed at the rail, the wheel will slide, resulting in longer stopping distances and wheel damage (8,14).

In general, the coefficient of adhesion at the rail is greater than coefficient of friction at the brake shoe, because static or rolling friction is greater than kinetic or sliding friction. However, on freight cars with cast iron brake shoes, the brake shoe force is not allowed by FRA regulation to exceed 60% of the car lightweight for a full service brake application, as a safety factor against the wheels sliding (2,5,14). If the car was to be braked at 60% of its loaded weight, the car wheels would slide when the car was empty. This 60% ratio of force applied to the brake shoe over the car weight is called the braking ratio. The actual or Golden Shoe ratio, is determined from actual tests, and is usually about 65% of the design braking ratio (5). Therefore, the actual braking ratio is about 0.65*(0.60)=0.39. Thus braking resistance for a train is (2):

$$FB=RWef$$

Where FB = braking force in pounds

R = braking ratio for full service brake application (cast iron brake shoes=0.39)

e = brake lever efficiency (0.90)

f = coefficient of brake shoe friction

W = train lightweignt in pounds

Some modern empty/load brake equipment adjusts for car weight changes by sensing the distance from the car body to top of the truck side frame (5,11). These cars are braked at a higher braking ratio. However, in the interest of a conservative estimate of braking force, a 60% braking ratio is assumed.

The coefficient of brake shoe friction is not constant, but changes with train speed and brake application time. It has been hypothesized that the change is due to the heating up of the metals during high speed braking, resulting in a lower coefficient of friction (11).

Hay (2) presents a graph of speed versus the coefficient of brake shoe friction for various brake shoe types. Expressions were found by trial and error to fit Hay's cast iron, medium braking curve, as well as the Cobra composition, medium braking curve. The expressions are:

f=0.5-0.07*ln((V+0.3)/0.3) (Cast Iron brake shoe)

f=0.49-0.055*ln((V+2)/2) (Cobra composition brake shoe)

where f = coefficient of brake shoe friction

V = train speed in mph (33" diameter steel wheels assumed)

Figure 3 plots the equations of the two brake shoe types, illustrating the decrease in brake shoe friction with increased velocity.

The use of Cast iron shoes is being phased out, with approximately 70% of modern rail car fleets now using composition brake shoes (8). Composition brake shoes have a higher coefficient of friction than cast iron shoes, allowing braking ratios of cars equipped with them to be reduced from 60% to 30% of the car lightweight (5). The cast iron curve is used as the default expression in the program.

The final expression for modelling the full service braking force is:

FB=0.39*(0.9)*W*(0.5-0.07*ln((V+0.3)/0.3)) (Cast Iron)

where FB = braking force acting on the train in pounds

W = train lightweight in pounds

V = train speed in mph

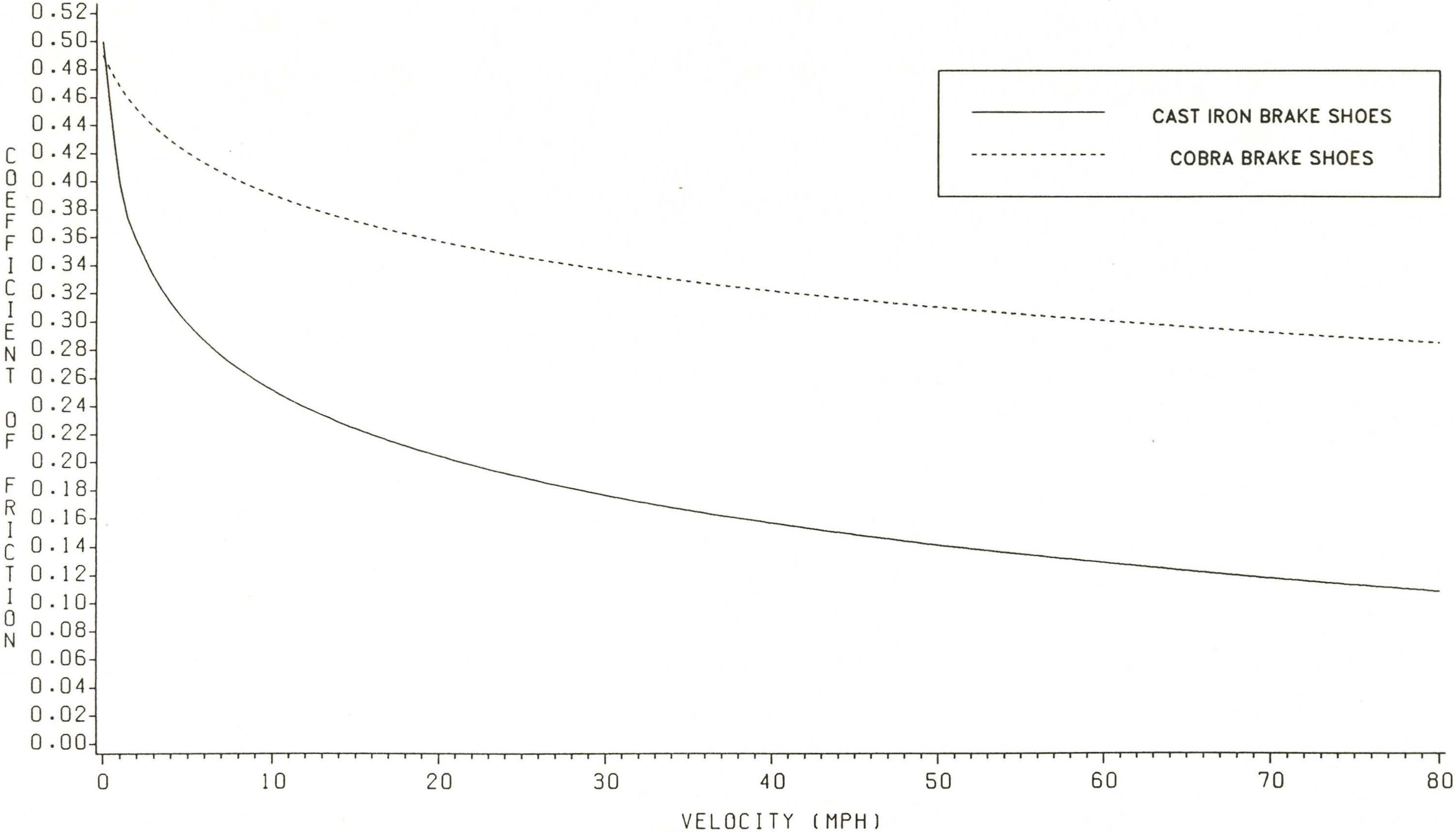


FIG.3 BRAKE SHOE FRICTION CURVES (HAY)

Service braking (10 lb. reduction) force would be approximately one half of the full service braking force (5).

With this braking model, runaway trains are possible once the train reaches a critical speed on a downgrade. A rough estimate for maximum speeds on a given downgrade can be found by the program by increasing speed limits on a downgrade until a runaway occurs. Maximum speed on a given downgrade can also be solved directly by setting FB=TRT. The braking model also allows proper estimates of braking distances and time for both loaded and empty trains on any grade, which is not possible with a constant deceleration model.

Dynamic braking is not modelled. Dynamic braking is the retarding action on a train created from changing kinetic energy to electrical energy by converting a Diesel-electric's traction motors into generators. Electrical energy generated from the traction motors is then dissipated as heat from resistance grids on the locomotive. Dynamic braking should be included in future improvements of the program for proper modelling of train behavior.

2.4 TRAIN OPERATIONS MODEL

The train operations model combines the train resistance, tractive effort, and braking force models to predict the velocity, time, and fuel consumption of a train at any point along the a track. Train handling by the operations model is done with the goal of shortest possible travel time over a given route, subject to power and speed limitations.

The train operation is simulated in the model using one mph speed increments. An average accelerating force is found by taking the difference between train resistance and tractive effort forces obtained by averaging each force over each increment in speed. The average accelerating force, acceleration, change in time, distance and work done is found over each one mph increment. The program goes through a numerical integration, calculating the changes in the trains state corresponding to each speed increment. Figure 4 illustrates the methods used in the calculation of train motion.

The algorithm increments the trains velocity, trying to reach the speed limit at all times. The train will always try to match the speed limit set by accelerating if possible, if it is below the speed limit, braking if it is above the speed limit.

As the train moves down the track, its location is compared at each iteration with the track data array. Once the train reaches a point of change in either degree of curve, grade, or speed limit, the information is updated from the track data array and the train continues with the appropriate change in effect. As the train approaches balancing speed, acceleration decreases and distance increments get larger for each constant increase in velocity. The algorithm ensures that the train does not increment over track changes (e.g. grade changes). If the train passes over a change, the algorithm "backs up" and recalculates to the point of change. A new velocity at the point of change is calculated by using the distance from the previous position of the train to the point of change. Thus, the algorithm ensures all grades and curves are accounted for in calculations.

F_a = AVERAGE TRACTIVE EFFORT - AVERAGE TRAIN RESISTANCE

$Acc = F_a/Mass$

$\Delta T = \Delta V/Acc$

$\Delta X = V_0 \cdot \Delta T + \frac{1}{2} Acc \cdot \Delta T^2$

ΔW = AVRG. TRACTIVE EFFORT $\cdot \Delta X$

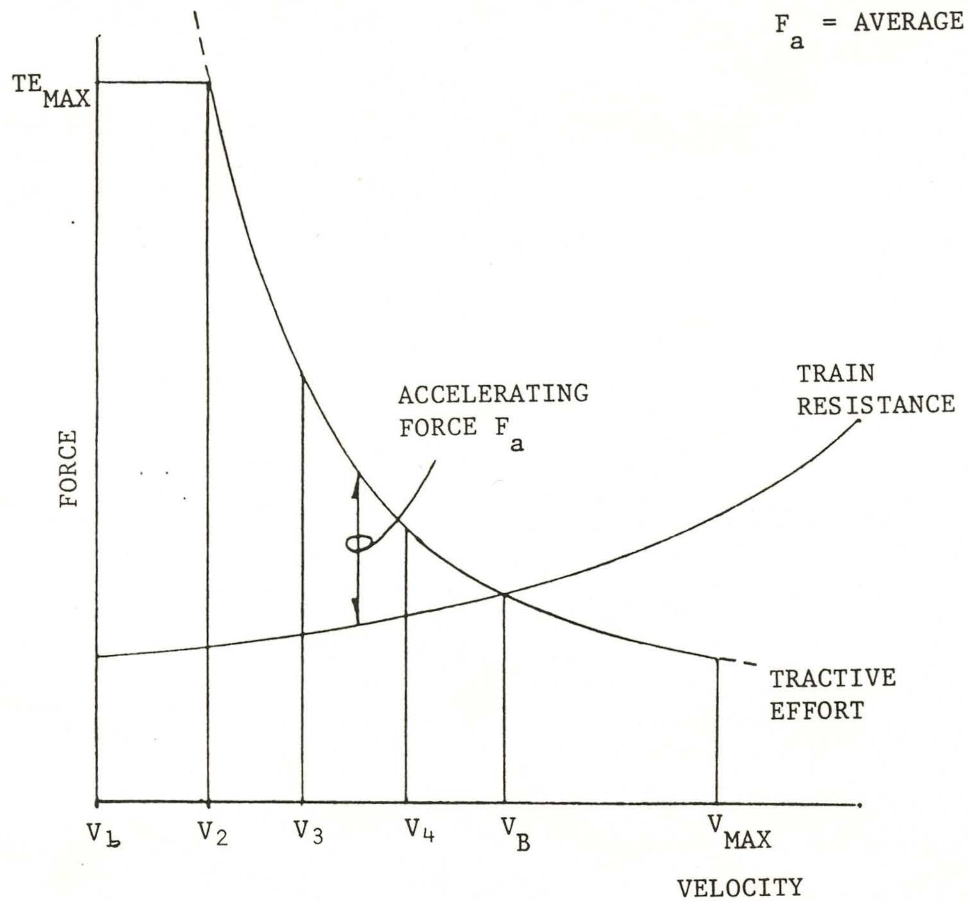


FIG. 4 CALCULATION OF TRAIN MOTION

Once the train reaches the speed limit, the distance to the next change in track conditions is calculated. Using this distance, time and work to the point of change is calculated at constant velocity. At the point of change in track conditions, it is calculated whether the train can maintain constant velocity. If it can, it progresses to the next change in track condition. If it can not, the velocity is adjusted up or down as appropriate. The above procedure is also used if the train reaches its balancing speed. The balancing speed is calculated to within 1/10 of a mile per hour, after which the train is advanced to the next change in track conditions. The simulation ends when either train speed reaches zero, or the train comes to the end of the route.

2.4.1 Assumptions

When the train is under the speed limit, engines are assumed to be at full throttle with the brakes off. When over the speed limit, engines are assumed to be idling with brakes applied in full service. These assumptions do not always reflect actual operating practice, where power braking is used to keep slack stretched throughout the train.

When the train is at the speed limit, it is assumed that tractive effort or braking force will just match train resistance, if possible. Practically, such fine control of tractive effort or braking is difficult. This is especially true of braking. If a reduction of braking effort is needed in order for the train to match the speed limit, the train brakes must be released, and then reapplied at the required level. This cannot be done too often, because it will deplete the air reservoirs on the cars, resulting in the loss of air brakes.

The train is modelled as a point mass, which introduces some errors over grade and curvature changes. These errors, however, are small, especially for short trains and long distances (10). Future development of the model could include modifications to the point mass assumption.

Chapter III

DATA INPUT/OUTPUT

3.1 DATA INPUT

The data input for the TPC is in four main sections: simulation parameters; route input; consist input; and power input. Defaults are given to all input variables to mimimize data entry.

3.1.1 Specification and Simulation Parameters

This section allows the user to specify certain operating conditions for the running of a train. Default parameters can be changed to simulate a particular run. Parameters included are train direction, initial speed, operational speed limit, percent adhesion, locomotive efficiency, percentage of cars having composition brake shoes and Davis equation resistance coefficients.

Train direction controls whether the train moves from lowest to highest milepost (forward), or in the opposite direction (backward). The program is defaulted in the forward direction. Initial speed is used to start the train at a desired speed (default= 0 mph). Operational speed limit is the maximum allowable speed for a particular train (default=65 mph). An example of this would be rail or hazardous goods trains which are not allowed to reach full track speed. Per cent adhesion is the coefficient of friction at the wheel-rail interface (default=18.5%). The coefficient of friction will vary somewhat with speed, rail conditions and locomotive wheel slip control systems. Locomotive efficiency is the percentage of the rated horsepower of the diesel reaching the driver wheel rims (default=83%). Percentage of cars with composition brake shoes affects braking distances due to the higher coefficient of friction of this type of brake shoe (default= 0%). The Davis resistance coefficients are used in the Davis equation to predict train resistance. CP Rail's coefficients are used for defaults.

3.1.2 Route Data Input

Information on track grades, speed limits and curvature is entered through this portion of the program. If the track route is not specified by the user, the program will use the first track route file stored on disk. Track data files are available from the disk, or new track data files can be entered and stored on disk for future use.

When route data is entered for the first time, information on grades, speed limits and curvature is entered separately, sorted in order of milepost by the program, and then stored on disk. The route is defined by milepost and a change or any combination of changes in either grade, curvature or speed limit. This results in a track route with segments of constant grade, curvature, and speed limits. Track data can be displayed on the screen to check for errors.

3.1.3 Consist Data Input

Car information obtained from the Car and Locomotive Cyclopedia (11) is displayed on the screen listing car type, car length, car lightweight and capacity. A running total of train weight, train length and total number of cars are displayed at the bottom of the screen. The type of car, number of cars and whether they are loaded or empty are entered in the fields presented on the screen. Loaded cars are assumed to be loaded to capacity. The default is arbitrarily chosen as 25 loaded boxcars and 50 empty boxcars.

3.1.4 Motive Power Input

A list of commonly used diesels and their horsepower, weight, length, and number of axles are presented on the screen (12,13). Running totals are displayed on the screen for total horsepower and total unit weight. The number of each type of diesel wanted is entered in the fields presented on the screen. Three SD40-2's are arbitrarily selected as defaults.

3.2 DATA OUTPUT

After the train performance simulation has been completed, output is displayed in trip summaries, data tables or graphs. An output menu appears in which the selection of the type of data wanted is made. Hardcopy output is available for all output data.

3.2.1 Trip Summary

The trip summary presents output on train information such as power, weight, power to weight ratios, length, and payload, as well as trip statistics on fuel consumption, average velocity, trip time and ton miles to the gallon.

3.2.2 Data Tables

Data tables on the screen present output of milepost, grade, speed limit, train speed, train resistance, tractive effort, braking force, time and work for the train at any point along the entire route. On the screen, the data tables are presented as pages which can be moved up or down, or to a specific page. The printer prints the degree of curve, as well as acceleration in addition to the output described above.

3.2.3 Graphs

A velocity-distance graph, along with a route profile, is presented when graphical output is selected from the output menu. The graphs are plots of values in the data tables allowing the user to more easily visualize the trains behavior. The scale can be changed on the graphs to "zoom" in for greater detail.

Chapter IV

COMMENTARY

Unfortunately, the program was not completed in time for verification of the output. Attempts at verification will include comparison of results with CP Rail's TPC. To that end, sample runs of a 100 car grain train over the Bredenbury and Minnedosa subdivisions were obtained courtesy of CP Rail. Output from CP Rail's TPC has been compared to actual trains with results within 5% to 10% of actual train trip times (4).

Dynamometer car drawbar data would be useful in verification of the inherent resistance equations. The coefficients used in this program are not based on 100% roller bearing consists, which are becoming more common for unit trains. In 1980, 80% of all freight car mileage in the U.S. was on cars with roller bearings (1).

Empirical results from braking tests are also needed to verify the braking model used in this program. At this time, it is unknown how realistic the braking distances and times predicted by the model are.

Besides verification, important extensions to the program are needed. These include:

1. Braking in advance of speed restrictions.
2. Modelling dynamic braking.

3. A more realistic fuel consumption algorithm based on fuel rates of diesels in each notch.
4. Accounting for vertical curves through grade change vertexes.
5. Allowing station stops.

REFERENCES

1. Train Performance Calculator: A Survey and Assesment. SRI International, Report No. FRA/ORD-81/02 (April 1981).

2. Hay, William W. Railroad Engineering, 2nd ed. John Wiley and Sons, Inc., New York (1982).

3. Morlok, Edward K. Introduction to Transportation Engineering and Planning. McGraw Hill, New York (1978).

4. Wee, T.S. Correspondence with CP Rail, Planning and Research Dept. (Nov., 1985).

5. Armstrong, J.H. The Railroad - What it is, What it does, 2nd ed. Simmons Boardman Publishing Corporation, Omaha (1984).

6. Poole. Costing, A Tool for Railroad Management. Simmons Boardman Books, Omaha (1961).

7. Resistance of a Freight Train to Forward Motion, Volume III, Economic Analysis and Correlation of Predictions with Field Data. MITRE Corp., Metrek Division, Report No. FRA/ORD-78/04.III.

8. Harding, J.W. Passenger and Freight Car Brake Equipments Air Brake, Vol. II. International Textbook Company,Scranton (1928).

9. Conversation with CN car shop manager, CN Symington Yards, Winnipeg (Jan., 1986).

10. Lim, S.C. A Digital Computer Simulation Model for Capacity Analysis on a Single Track Railway line with Passing Loops. Msc. Thesis, University of Manitoba, Winnipeg (1983).

11. Ellsworth, G.K. Car and Locomotive Cyclopedia of American Practices, 4th ed. Simmons Boardman Publishing Corporation, Omaha (1978).

12. CP Rail Mechanical Department publication (1974).

13. Dean, Murray W. and Hanna, David B. Canadian Pacific Diesel Locomotives. Railfare Enterprises, Don Mills (1981).

14. Drennan, C.M. ABC's of Air Brakes. Simmons Boardman Books, Omaha (1961).

15. Resistance of a Freight Train to Forward Motion, Volume IV, Users Manual for Freight Train Fuel Consumption Program. MITRE Corp., Metrek Division, Report No. FRA/ORD-78/04.IV.

16. Graham, Neill _Programming the IBM Personal Computer_: _Basic_. Holt, Rinehart and Wilson, New York (1982).

Appendix A

USER'S GUIDE

TRAIN PERFORMANCE CALCULATOR FOR THE IBM PC

A.1 USING THE TPC PROGRAM

A.2 HARDWARE REQUIREMENTS

The TPC program will run on a IBM personal computer and will run on most computers compatible with the IBM PC. A 128K RAM computer with a single disk drive and graphics card is required to take full advantage of the program. The program will still work without the graphics card, however graphs will not be plotted. If the TPC program is developed further, two disk drives will be required, one for the program disk, the other for the track data files disk.

Computers with 512K RAM available can use a RAM disk program (eg. Superdrive) to speed up data access. A RAM disk program loads data and programs from a floppy disk into a computers memory. When data is requested from a program, the data is accessed from the RAM disk, rather than the floppy disk. Data transfer using the RAM disk program is much faster than data transfer from the floppy disk drive, resulting in faster program operation.

A.3 ABOUT THE SOFTWARE

The program was written in IBM BASICA and then compiled for faster execution and to free up more variable space for data storage. It is written in a semi-structured format making good use of subroutines. Error trapping is included in the program to allow recovery from mistakes, such as not having the printer turned on when trying to print.

The main difficulty of using a microprocessor for a TPC program is the lack of memory for storage of track and output data. Thus, for long runs, some time is spent accessing new track data files and loading them into the computer. However, this time is not long. The program as it stands simulates train runs one subdivision at a time.

A.3.1 General

The TPC program for the IBM PC is designed to be straight forward and easy to use. Data entry and editing is simple and quick, with defaults for all input variables. The program is currently defaulted to run a train with 3 SD40-2 diesels, 25 loaded boxcars and 50 empty boxcars over the first subdivision available on the track data file.

The program is driven by a main menu, which accesses the five major portions of the program. The program is divided into simulation parameters, route input, consist input, power input and simulation run. Figure A1 shows the relationships between the different program sections

Program control and data entry are done through the use of the cursor, return and escape (esc) keys. Making choices on the menu screens

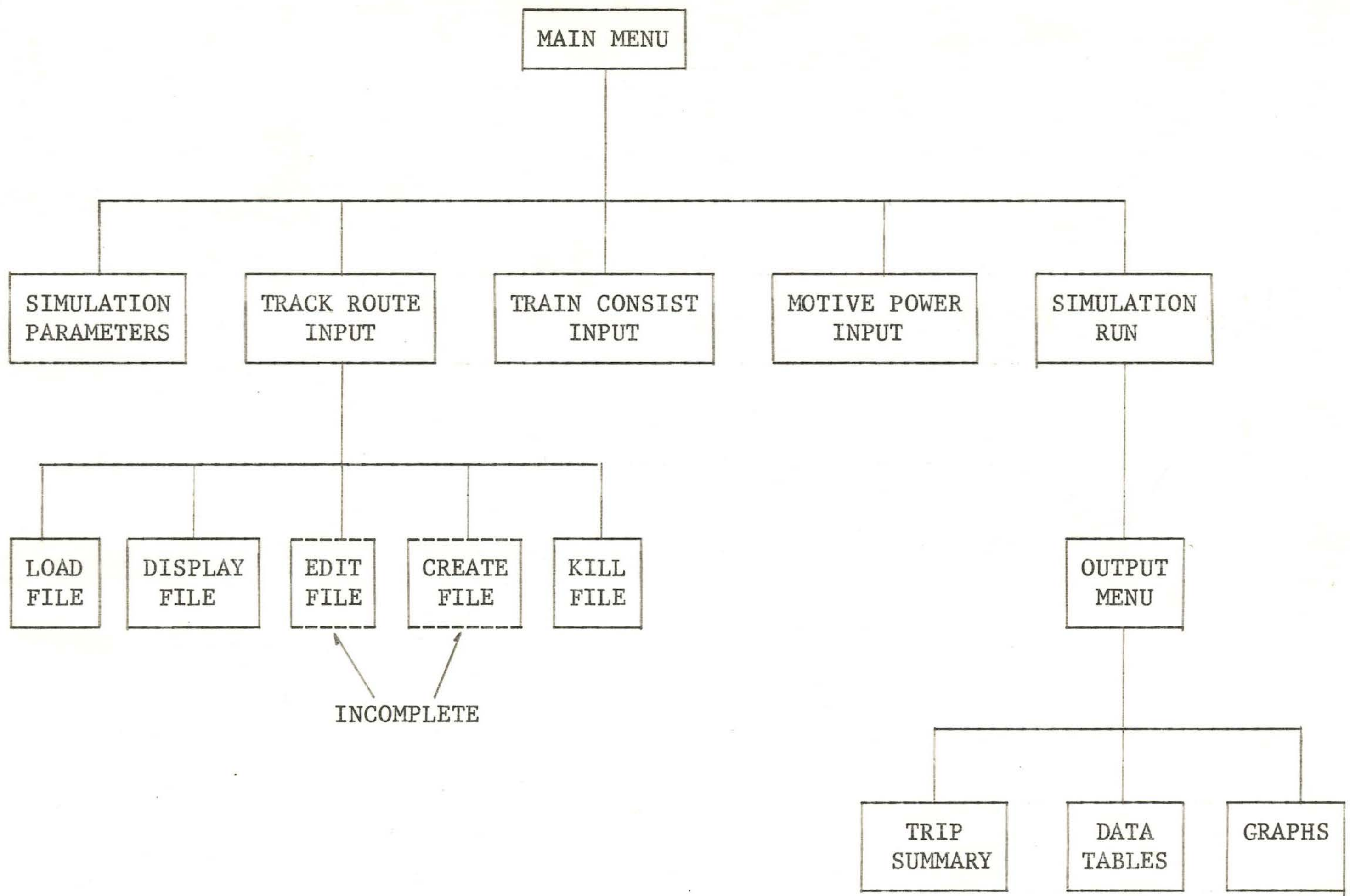


FIG. A1. PROGRAM SCHEMATIC.

is accomplished by using the cursor keys and pressing return after the selection is made.

Data entry and editing are done through fields presented on the screen. By using the up and down cursor keys, the cursor can be moved from field to field on the screen. To change a value in the field on the screen, type over it.

A.3.2 Sample Run

This section describes how to use the TPC program. Illustrations of the numbered screens referred to in the text are located at the end of the Appendix. The illustrations are actual reproductions of screens that appear while the program is operating.

A.3.2.1 Starting the Program

To start the program, put the program disk in the left disk drive and turn the computer on.

A.3.2.2 Main Menu

Eventually, after the disk drive is finished, the Main Menu Screen (1) will appear. The five sections of the program are shown on the screen. To move to a section, use the cursor arrows to highlight a selection, then press return. It is possible to immediately select the run simulation section, run the train, and see the results of the simulation. The program will automatically run a 75 car train of box-cars, with 3 SD40-2 diesels over a track file called the La Riviere Subdivision. The track file is not representative of CP Rail's La Riviere Subdivision at this time.

To change the train, track route or parameters for a particular run requires the use of the following four sections:

1. Simulation Parameters
2. Track Route Selection
3. Train Consist Selection
4. Motive Power Selection

A.3.2.3 Simulation Parameters

The first section is the Simulation Parameters Screen (2). Changes are made by using the cursor arrow keys to move to the value(s) you want changed. Type in the appropriate value(s), then press [esc] to return to the main menu.

A.3.2.4 Track Route Selection

The next section is the Track Route Selection Screen (3). To choose the desired track file, highlight a filename with the up/down cursor arrow keys. The appropriate action to be done to a file is highlighted by using the left/right cursor arrow keys. When satisfied, press return to initiate the action desired. A track file must be loaded to return to the main menu.

A.3.2.5 Train Consist Selection

The Train Consist Screen (4) is the third section. The number and type of cars on the train, and whether they are empty or loaded are specified here. Again, the values are changed by using the up/down cursor arrow keys, and typing in the desired value. Return to the main menu by pressing [esc].

A.3.2.6 Motive Power Selection

The fourth section is the Motive Power Screen (5). The type and number of diesels desired for the train is entered by using the up/down cursor arrow keys to move to a choice, and type in the number of diesels wanted. Press [esc] to return to the main menu.

A.3.2.7 Run Simulation

The parameters, track, train, and power necessary for the simulation have been entered. Move to the Run Simulation choice on the main menu and press return. **Running Simulation** will flash on the screen, with the milepost of the calculated location of the train shown in the lower right corner.

A.3.2.8 Output Menu

When the simulation is complete, the computer will "beep", and present the Output Selection Screen (6). A trip summary, data tables or graphs can be selected from the output menu. Selections are made by using the up/down cursor arrow keys and pressing return. To return to the main menu for another simulation, press [esc]. The information presented in each output format is described in the following sections.

A.3.2.9 Trip Summary

The Trip Summary Screen (7) presents summary information about the train as well as the time and fuel consumed by the train over the route. Return to the ouput menu by pressing [esc].

A.3.2.10 Data Tables

A detailed description of the train's velocity and time along the route is given on the Data Tables Screens (8,9). Moving from page to page is done by using the [pgup] and [pgdn] keys. To move to a specific page, press [F1], the page wanted, then return. To print the trip summary and data tables on the printer, press [P]. Returning to the output menu is done by pressing [esc].

A.3.2.11 Graphs

The Graphs Screen (10) plots velocity versus distance as well as the profile of the track route. The velocity distance graph plots the values in the data tables, which makes it easier to visualize the trains behavior along the track.

To change the scale of the graphs, press [F2]. Enter the starting and ending milepost of the new graph desired in the pop up menu (11). Press [esc] to change the graphs (12). The scale of the graph can be changed as many times as desired. To return to the output menu, press [esc] again.

Train Performance Calculator
Main Menu

Select simulation parameters

Select track route

Select train consist

Select motive power

Run simulation

1

Simulation Parameters

Train Direction (Fwd/Bkwd)	[0]	Assumed Adhesion (%)	[18.5]
Initial Speed (mph)	[0]	Locomotive Eff. (%)	[83]
Operational Speed Limit (mph)	[65]	Composition Brk Shoes (%)	[0]

Resistance Coefficients

Type	A		B	C
Freight Car	[1.5]	[18]	[.03]	[.05]
Piggyback	[1.5]	[18]	[.03]	[.102]
Locomotive	[1.5]	[18]	[.03]	[.066]

[Esc] Exit

2

Track Data Files Available

LARIVIERE SUBDIVISION
KEEWATIN SUBDIVISION

LOAD DISPLAY EDIT KILL CREATE

3

Train Consist

Car Type	Length (Ft)	Ltwt (Lbs)	Capy (Lbs)	Cars Loaded	Cars Empty
Boxcar	55	62100	154000	[25]	[50]
Cvrd Hopper	64	63300	199000	[0]	[0]
Tank Car	54	71000	192000	[0]	[0]
Piggyback	94	68500	149000	[0]	[0]
Hopper Car	49	59900	200000	[0]	[0]

Total Number of Cars: 75
Total Length (Ft): 4125
Total Weight (Tons): 4254

[Esc] Exit

4

Motive Power

Model	Horsepower	Weight (Tons)	No. of Axles	No. of Units
SD40-2	3000	197	6	[3]
GP38	2020	131	4	[0]
GP9	1750	130	4	[0]
M636	3312	196	6	[0]
M630	3040	196	6	[0]
C424	2400	129	4	[0]

Number of Units: 3
Total Horsepower: 9000
Total Weight (Tons): 591

[Esc] Exit

5

Output Selection

Trip Summary

Data Tables

Graphs

[Esc] Exit

6

TRAIN INFORMATION
LARIVIERE SUBDIVISION

NUMBER OF UNITS 3
TOTAL HORSEPOWER 9000
TOTAL UNIT WEIGHT (TONS) 591
ASSUMED ADHESION .185

NUMBER OF CARS 75
TRAIN LENGTH (FT) 4125
TRAIN GROSS WEIGHT (TONS) 4253.75
TRAIN LIGHTWEIGHT (TONS) 2328.75
TRAIN PAYLOAD (TONS) 1925

HP/TON GROSS WEIGHT 2.1
HP/TON PAYLOAD 4.7

TOTAL TIME (H:M:S) 1:25:43
AVERAGE VELOCITY (MPH) 45.8

DIESEL FUEL CONSUMED (GALS) 412
TON MILES/GALLON (GROSS WEIGHT) 676.3
TON MILES/GALLON (PAYLOAD) 306.0

[Esc] Exit

7

LARIVIERE SUBDIVISION

PAGE 1 OF 13

MILEPOST	GRADE (%)	SPEED LIMIT (MPH)	TRAIN SPEED (MPH)	TRAIN RESISTANCE (LBS)	TRACTIVE EFFORT (LBS)	BRAKING FORCE (LBS)	TIME (H:M:S)	WORK (FT LBS) (X 1E6)
0.00	+0.00	50	1.0	+12816	218670	0	0:00:01	0
0.00	+0.00	50	2.0	+12974	218670	0	0:00:03	1
0.00	+0.00	50	3.0	+13139	218670	0	0:00:05	3
0.00	+0.00	50	4.0	+13312	218670	0	0:00:07	5
0.01	+0.00	50	5.0	+13493	218670	0	0:00:09	7
0.01	+0.00	50	6.0	+13681	218670	0	0:00:11	11
0.01	+0.00	50	7.0	+13878	218670	0	0:00:13	15
0.02	+0.00	50	8.0	+14083	218670	0	0:00:15	19
0.02	+0.00	50	9.0	+14295	218670	0	0:00:16	24
0.03	+0.00	50	10.0	+14515	218670	0	0:00:18	30
0.03	+0.00	50	11.0	+14744	218670	0	0:00:20	37
0.04	+0.00	50	12.0	+14980	218670	0	0:00:22	44
0.04	+0.00	50	13.0	+15224	215481	0	0:00:24	51
0.05	+0.00	50	14.0	+15476	200089	0	0:00:26	59
0.06	+0.00	50	15.0	+15736	186750	0	0:00:28	68

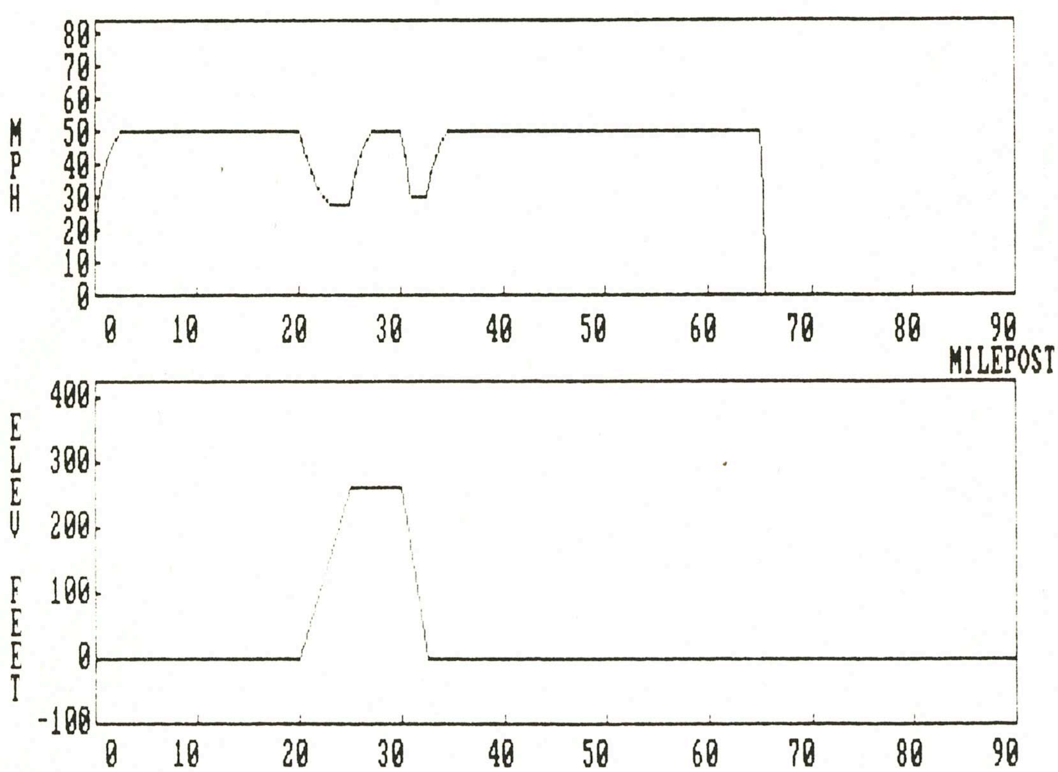
[PG UP] [PG DN] [F1] GOTO PAGE [ESC] EXIT [P] PRINT

8

MILEPOST	GRADE (%)	SPEED LIMIT (MPH)	TRAIN SPEED (MPH)	TRAIN RESISTANCE (LBS)	TRACTIVE EFFORT (LBS)	BRAKING FORCE (LBS)	TIME (H:M:S)	WORK (FT LBS) (X 1E6)
65.45	+0.00	0	10.0	+14515	0	396508	1:25:35	12708
65.45	+0.00	0	9.0	+14295	0	407924	1:25:36	12708
65.46	+0.00	0	8.0	+14083	0	420640	1:25:37	12708
65.46	+0.00	0	7.0	+13878	0	434990	1:25:37	12708
65.46	+0.00	0	6.0	+13681	0	451458	1:25:38	12708
65.46	+0.00	0	5.0	+13493	0	470779	1:25:39	12708
65.46	+0.00	0	4.0	+13312	0	494151	1:25:40	12708
65.46	+0.00	0	3.0	+13139	0	523738	1:25:41	12708
65.46	+0.00	0	2.0	+12974	0	564092	1:25:41	12708
65.46	+0.00	0	1.0	+12816	0	627868	1:25:42	12708
65.46	+0.00	0	0.0	+12667	0	791775	1:25:43	12708

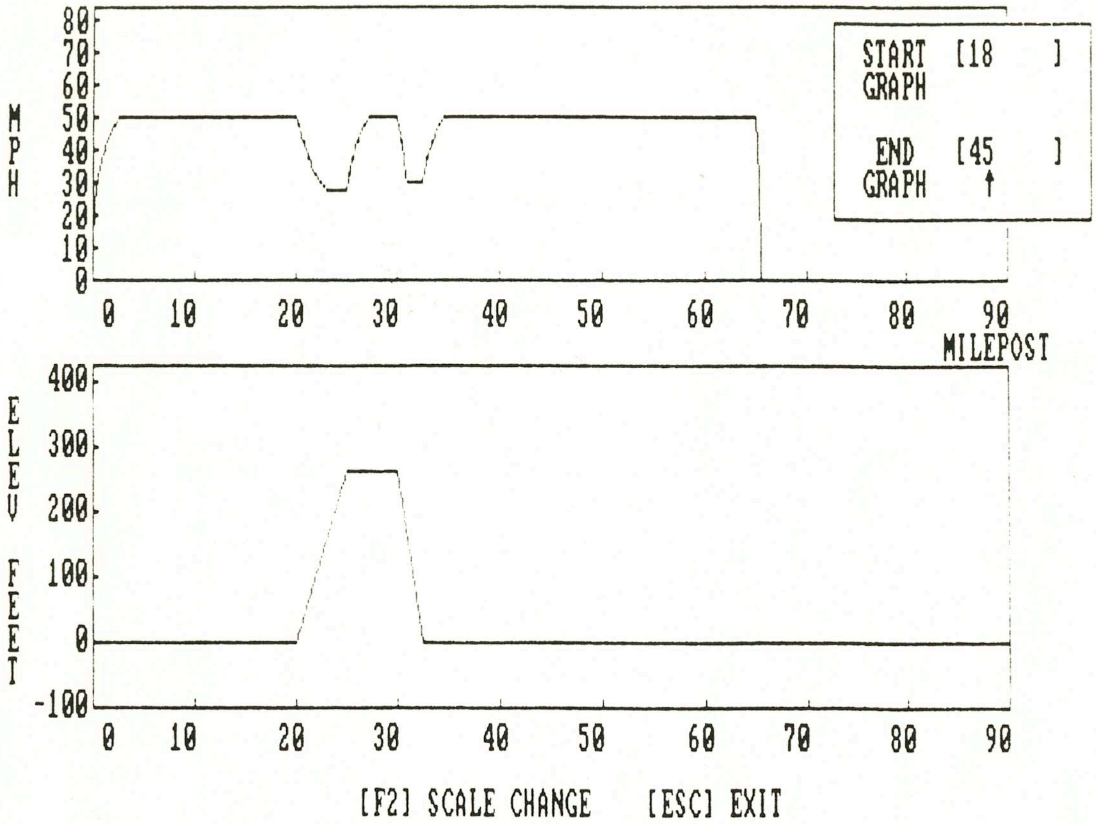
[PG UP] [PG DN] [F1] GOTO PAGE [ESC] EXIT [P] PRINT

9



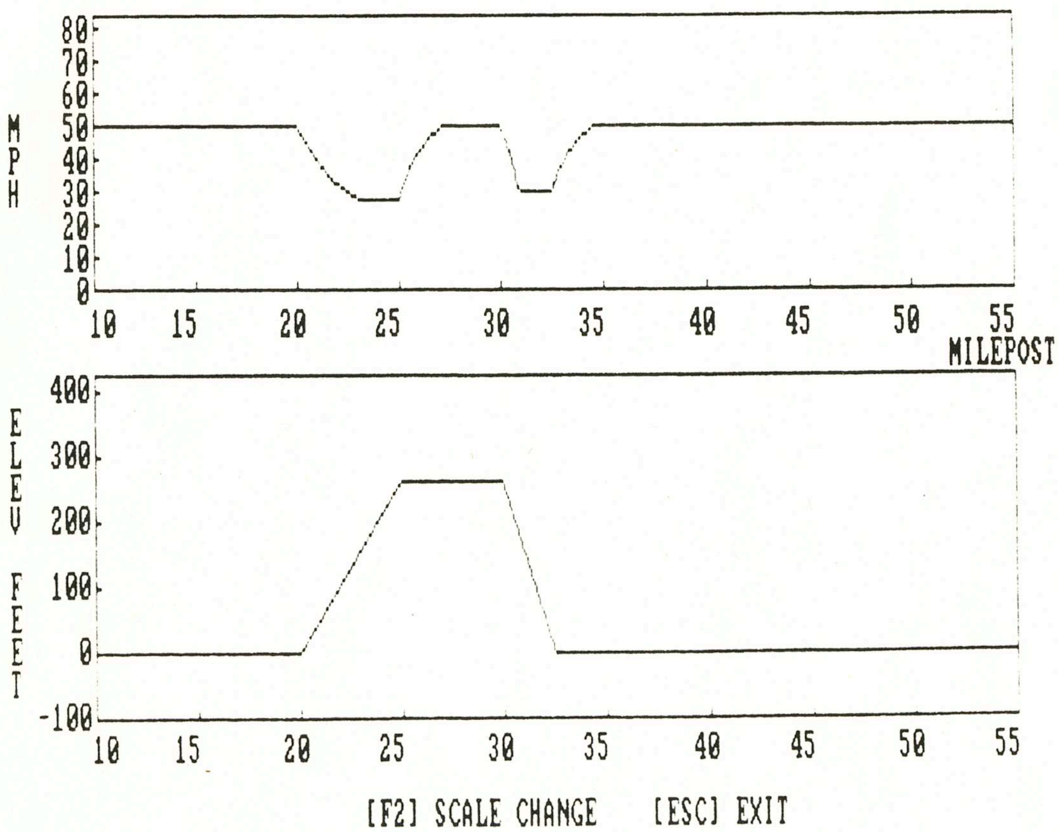
[F2] SCALE CHANGE [ESC] EXIT

10



START [18]
GRAPH
END [45]
GRAPH ↑
MPH
80
70
60
50
40
30
20
10
0
0 10 20 30 40 50 60 70 80 90
MILEPOST
ELEV FEET
400
300
200
100
0
-100
0 10 20 30 40 50 60 70 80 90
[F2] SCALE CHANGE [ESC] EXIT

11



MPH
80
70
60
50
40
30
20
10
0
10 15 20 25 30 35 40 45 50 55
MILEPOST
ELEV FEET
400
300
200
100
0
-100
10 15 20 25 30 35 40 45 50 55
[F2] SCALE CHANGE [ESC] EXIT

12

Appendix B

PROGRAM LISTING

```
1 '---------------------------------------------------------------------------
2 '               Train Performance Calculator for the IBM PC
3 '                          Transport Group
4 '                   Department of Civil Engineering
5 '                       University of Manitoba
6 '
7 '                Written by: Rob Girling, IV Civil Eng.
8 '                     51 Henday Bay, Winnipeg  MB
9 '                         Oct/84 to March/86
10 '---------------------------------------------------------------------------
12 KEY OFF:KEY 1,"":KEY 2,"":DEF SEG:POKE 106,0:SCREEN 0
15 DEFINT P,J,K:ON ERROR GOTO 19000
20 DIM TRK(75,6),A$(25),GR(25,4),SP(25,4),CV(20,4),CAR(10,4),LOCO(6,4)
30 DIM MP(400),G(400),GE(400),DC(400),MS(400),TS(400),TR(400),BF(400),AC(400)
40 DIM T(400),WK(400),TE(400),P(1000),COEFF$(36),COEFF(36)
50 '
60 GOSUB 16000:GOSUB 17000:GOSUB 18000
62 FOR X=1 TO 36:COEFF$(X)=STR$(COEFF(X))
63 COEFF$(X)=RIGHT$(COEFF$(X),LEN(COEFF$(X))-1):NEXT
65 I1=1:NY=1:NY3=1:TE(1)=1:GOSUB 1885:GOSUB 1900
67 GOSUB 2600:GOSUB 2900:GOSUB 3500:GOSUB 3300
100 '
110 CLS:A$="A":COLOR 15:LOCATE 2,27,0:PRINT "Train Performance Calculator"
120 LOCATE 3,38:PRINT "Main Menu":LOCATE 4,1:COLOR 7:PRINT STRING$(80,"_")
130 GOSUB 332:GOSUB 342:GOSUB 352:GOSUB 362:GOSUB 372
140 WHILE ASC(A$)<>13
160 COLOR 0,7:INV1=0:INV2=0:ON NY GOSUB 330,340,350,360,370:COLOR 7,0
170 A$=INKEY$:IF A$="" THEN 170 ELSE IF ASC(A$)=13 THEN 220
180 IF LEN(A$)=2 THEN B$=RIGHT$(A$,1) ELSE 170
185 IF B$="" THEN 170
190 IF ASC(B$)=72 THEN GOSUB 315:NY=NY-1
200 IF ASC(B$)=80 THEN GOSUB 315:NY=NY+1
210 IF NY>5 THEN NY=1 ELSE IF NY<1 THEN NY=5
220 WEND
300 ON NY GOSUB 500,1000,2000,3000,4000
305 NY=NY+1:IF NY>5 THEN NY=1
306 GOTO 110
310 '
315 INV1=223:INV2=220:ON NY GOSUB 330,340,350,360,370:RETURN
320 '
330 LOCATE 7,26:PRINT STRING$(30,223-INV1)
331 LOCATE 9,26:PRINT STRING$(30,220-INV2)
332 LOCATE 8,26:PRINT " Select simulation parameters ":RETURN
340 LOCATE 9,31:PRINT STRING$(20,223-INV1)
341 LOCATE 11,31:PRINT STRING$(20,220-INV2)
342 LOCATE 10,31:PRINT " Select track route ":RETURN
350 LOCATE 11,30:PRINT STRING$(22,223-INV1)
351 LOCATE 13,30:PRINT STRING$(22,220-INV2)
352 LOCATE 12,30:PRINT " Select train consist ":RETURN
360 LOCATE 13,30:PRINT STRING$(21,223-INV1)
361 LOCATE 15,30:PRINT STRING$(21,220-INV2)
362 LOCATE 14,30:PRINT " Select motive power ":RETURN
370 LOCATE 15,33:PRINT STRING$(16,223-INV1)
371 LOCATE 17,33:PRINT STRING$(16,220-INV2)
372 LOCATE 16,33:PRINT " Run simulation ":RETURN
380 '
390 '
500 CLS:LOCATE 1,30:COLOR 15:PRINT "Simulation Parameters":COLOR 7
510 LOCATE 2,1:PRINT STRING$(80,"_"):LOCATE 4,3
520 PRINT USING "Train Direction (Fwd/Bkwd)    [\   \]       Assumed Adhesion (%)       [\   \]";COEFF$(1);COEFF$(4):LOCATE 6,3
540 PRINT USING "Initial Speed (mph)           [\   \]       Locomotive Eff. (%)        [\   \]";COEFF$(2);COEFF$(5)
550 LOCATE 8,3:PRINT USING "Operational Speed Limit (mph) [\   \]";COEFF$(3);
555 PRINT USING "       Composition Brk Shoes (%) [\   \]";COEFF$(6)
560 LOCATE 11,28:COLOR 15:PRINT "Resistance Coefficients":COLOR 7
570 LOCATE 13,8:PRINT "Type" TAB(32) "A" TAB(52) "B" TAB(70) "C"
575 PRINT STRING$(80,"_")
590 LOCATE 16,5:PRINT USING "Freight Car         [\   \]   [\   \]";COEFF$(7);COEFF$(8);
600 PRINT USING "         [\   \]          [\   \]";COEFF$(9),COEFF$(10)
610 LOCATE 18,5:PRINT USING "Piggyback           [\   \]   [\   \]";COEFF$(11);COEFF$(12);
620 PRINT USING "         [\   \]          [\   \]";COEFF$(13);COEFF$(14)
630 LOCATE 20,5:PRINT USING "Locomotive          [\   \]   [\   \]";COEFF$(15);COEFF$(16);
640 PRINT USING "         [\   \]          [\   \]";COEFF$(17),COEFF$(18)
650 LOCATE 24,35:COLOR 15:PRINT "[Esc] Exit";:COLOR 7
655 '
657 '
658 RN=4:CN=1:NY4=1:DATUM=33:MAX=18:MIN=1:FMT2$="[\   \]":GOTO 730
660 A$=INKEY$:IF A$="" THEN 660
670 IF ASC(A$)=27 THEN GOSUB 785:RETURN
685 IF LEN(A$)=2 THEN GOSUB 900
690 IF ASC(A$)>47 AND ASC(A$)<58 OR A$="." OR ASC(A$)=32 THEN GOSUB 770
695 IF ASC(A$)=13 THEN GOSUB 785
720 IF CN<1 THEN CN=1 ELSE IF CN>5 THEN CN=5:BEEP
725 GOSUB 800:GOSUB 840
735 PRINT USING "\   \";COEFF$(NY4):COLOR 7
740 LOCATE RN,CN+DATUM,1
750 GOTO 660
760 '
770 IF CN>LEN(COEFF$(NY4)) THEN COEFF$(NY4)=COEFF$(NY4)+A$:GOTO 776
775 MID$(COEFF$(NY4),CN,1)=A$
776 CN=CN+1:RETURN
777 '
785 COEFF(NY4)=VAL(COEFF$(NY4)):COEFF$(NY4)=STR$(COEFF(NY4))
787 COEFF$(NY4)=RIGHT$(COEFF$(NY4),LEN(COEFF$(NY4))-1):RETURN
788 '
800 IF NY4<4 THEN DATUM=33:RETURN
805 IF NY4>3 AND NY4<7 THEN DATUM=73:RETURN
810 IF NY4=7 OR NY4=11 OR NY4=15 THEN DATUM=24:RETURN
815 IF NY4=8 OR NY4=12 OR NY4=16 THEN DATUM=34:RETURN
820 IF NY4=9 OR NY4=13 OR NY4=17 THEN DATUM=49:RETURN
825 IF NY4=10 OR NY4=14 OR NY4=18 THEN DATUM=67:RETURN
830 '
835 '
840 IF NY4=1 OR NY4=4 THEN RN=4:RETURN
845 IF NY4=2 OR NY4=5 THEN RN=6:RETURN
850 IF NY4=3 OR NY4=6 THEN RN=8:RETURN
855 IF NY4>6 AND NY4<11 THEN RN=16:RETURN
860 IF NY4>10 AND NY4<15 THEN RN=18:RETURN
865 IF NY4>14 AND NY4<19 THEN RN=20:RETURN
867 '
900 B$=RIGHT$(A$,1)
910 IF ASC(B$)=72 THEN GOSUB 785:LOCATE RN,DATUM:PRINT USING FMT2$;COEFF$(NY4):NY4=NY4-1:CN=1
920 IF ASC(B$)=80 THEN GOSUB 785:LOCATE RN,DATUM:PRINT USING FMT2$;COEFF$(NY4):NY4=NY4+1:CN=1
930 IF ASC(B$)=77 THEN CN=CN+1
940 IF ASC(B$)=75 THEN CN=CN-1
950 IF NY4>MAX THEN NY4=MIN ELSE IF NY4<MIN THEN NY4=MAX
960 RETURN
970 '
980 '
1000 NY3=1:NX3=1:I2=0:A$="A":INV1=223:INV2=220
1020 CLS:COLOR 15:LOCATE 1,30:PRINT "Track Data Files Available":COLOR 7
```

```
1025 LOCATE 2,1:PRINT STRING$(80,"_")
1030 GOSUB 1140:GOSUB 1144:GOSUB 1148:GOSUB 1152:GOSUB 1156
1035 '
1080 GOSUB 1885
1090 FOR I1=1 TO I2:LOCATE 4+I1,40-LEN(A$(I1))/2:PRINT A$(I1):NEXT I1
1092 WHILE ASC(A$)<>13
1097 COLOR 0,7:INV1=0:INV2=0:LOCATE 4+NY3,40-LEN(A$(NY3))/2:PRINT A$(NY3)
1098 ON NX3 GOSUB 1140,1144,1148,1152,1156:COLOR 7,0
1102 A$=INKEY$:IF A$="" THEN 1102 ELSE IF ASC(A$)=13 THEN 1118
1104 IF LEN(A$)=2 THEN B$=RIGHT$(A$,1) ELSE 1102
1106 IF ASC(B$)=72 THEN GOSUB 1125:NY3=NY3-1
1108 IF ASC(B$)=80 THEN GOSUB 1125:NY3=NY3+1
1110 IF ASC(B$)=77 THEN GOSUB 1125:NX3=NX3+1
1112 IF ASC(B$)=75 THEN GOSUB 1125:NX3=NX3-1
1114 IF NY3>I2 THEN NY3=1 ELSE IF NY3<1 THEN NY3=I2
1116 IF NX3>5  THEN NX3=1 ELSE IF NX3<1 THEN NX3=5
1118 WEND
1120 ON NX3 GOSUB 1900,10000,10000,1800,1187
1122 IF NX3=1 OR NX3=2 THEN RETURN ELSE 1000
1124 '
1125 INV1=223:INV2=220:LOCATE 4+NY3,40-LEN(A$(NY3))/2:PRINT A$(NY3)
1126 ON NX3 GOSUB 1140,1144,1148,1152,1156:RETURN
1127 '
1140 LOCATE 20,11:PRINT STRING$(6,223-INV1)
1141 LOCATE 22,11:PRINT STRING$(6,220-INV2)
1143 LOCATE 21,11:PRINT " LOAD ":RETURN
1144 LOCATE 20,23:PRINT STRING$(9,223-INV1)
1145 LOCATE 22,23:PRINT STRING$(9,220-INV2)
1147 LOCATE 21,23:PRINT " DISPLAY ":RETURN
1148 LOCATE 20,38:PRINT STRING$(6,223-INV1)
1149 LOCATE 22,38:PRINT STRING$(6,220-INV2)
1151 LOCATE 21,38:PRINT " EDIT ":RETURN
1152 LOCATE 20,50:PRINT STRING$(6,223-INV1)
1154 LOCATE 22,50:PRINT STRING$(6,220-INV2)
1155 LOCATE 21,50:PRINT " KILL ":RETURN
1156 LOCATE 20,62:PRINT STRING$(8,223-INV1)
1157 LOCATE 22,62:PRINT STRING$(8,220-INV2)
1159 LOCATE 21,62:PRINT " CREATE ":RETURN
1185 '
1187 CLS:X=0:LOCATE 10,20:INPUT "NAME OF FILE ";A$(I2+1)
1190 N=0:CLS
1200 N=N+1:A2$=""
1210 PRINT"MILEPOST";GR(N,1)
1220 PRINT "ELEVATION (FT)";:PRINT USING"######";GR(N,2)
1230 INPUT"GRADE (%)";A2$
1240 IF A2$="" THEN 1290 ELSE GR(N,3)=VAL(A2$)
1250 INPUT"DISTANCE (MILES)";GR(N,4):PRINT:PRINT
1260 GR(N+1,2)=SIN(ATN(GR(N,3)/100))*GR(N,4)*5280+GR(N,2)
1270 GR(N+1,1)=GR(N,1)+GR(N,4):GOTO 1200
1280 '
1290 K=0:CLS
1300 K=K+1:A2$=""
1305 IF SP(K,1)>GR(N,1) THEN SP(K,1)=GR(N,1):GOTO 1370
1310 PRINT"MILEPOST";SP(K,1)
1320 INPUT"SPEED LIMIT (MPH)";A2$
1330 IF A2$=""THEN 1370 ELSE SP(K,2)=VAL(A2$)
1335 IF A2$="0" THEN 1370
1340 INPUT"DISTANCE (MILES)";SP(K,3):PRINT:PRINT
1345 IF SP(K,3)=0 THEN BEEP:GOTO 1340
1350 SP(K+1,1)=SP(K,1)+SP(K,3):GOTO 1300
1360 '
1370 I=0:CLS
1380 I=I+1:A2$=""
1385 IF CV(I,1)>SP(K,1) THEN CV(I,1)=SP(K,1):GOTO 1460
1390 PRINT"MILEPOST";CV(I,1)
1400 INPUT"DEGREE CURVE";A2$
1410 IF A2$="" THEN 1460 ELSE CV(I,2)=VAL(A2$)
1420 INPUT"LENGTH (MILES)";CV(I,3):PRINT:PRINT
1430 CV(I,4)=CV(I,2)*.04
1440 CV(I+1,1)=CV(I,1)+CV(I,3):GOTO 1380
1450 '
1460 G=1:S=1:C=1
1470 X=X+1
1480 IF GR(G,1)<SP(S,1) AND GR(G,1)<CV(C,1) GOTO 1600
1490 IF SP(S,1)<GR(G,1) AND SP(S,1)<CV(C,1) GOTO 1550
1500 IF CV(C,1)<GR(G,1) AND CV(C,1)<SP(S,1) GOTO 1560
1510 IF GR(G,1)=SP(S,1) AND SP(S,1)=CV(C,1)AND CV(C,1)=GR(G,1) GOTO 1570
1520 IF GR(G,1)=SP(S,1) AND GR(G,1)<CV(C,1) GOTO 1580
1530 IF CV(C,1)=GR(G,1) AND GR(G,1)<SP(S,1) GOTO 1590
1540 OS=S:OC=C:S=S+1:C=C+1:TRK(X,1)=SP(OS,1):GOTO 1610
1550 OS=S:S=S+1:TRK(X,1)=SP(OS,1):GOTO 1610
1560 OC=C:C=C+1:TRK(X,1)=CV(OC,1):GOTO 1610
1570 OS=S:OC=C:S=S+1:C=C+1: GOTO 1600
1580 OS=S:S=S+1:GOTO 1600
1590 OC=C:C=C+1
1600 OG=G:G=G+1:TRK(X,1)=GR(OG,1)
1610 TRK(X,2)=SP(OS,2):TRK(X,3)=GR(OG,3):TRK(X,4)=CV(OC,2):TRK(X,5)=GR(OG,3)+CV(OC,4):TRK(X,6)=TRK(X-1,6)+(TRK(X,1)-TRK(X-1,1))*5
1623 IF X<>1 AND TRK(X,1)=0 OR TRK(X,2)=0 THEN X=X-1:GOTO 1640
1625 GOTO 1470
1630 '
1640 IF A=4 THEN 1670
1650 OPEN "FILENAME" FOR OUTPUT AS #1
1660 FOR I1=1 TO I2+1:WRITE #1,A$(I1):NEXT I1:CLOSE:A$=A$(I2+1)
1665 '
1670 OPEN A$ FOR OUTPUT AS #1
1680 FOR I=1 TO X
1690 WRITE #1,TRK(I,1),TRK(I,2),TRK(I,3),TRK(I,4),TRK(I,5),TRK(I,6)
1700 NEXT
1704 CLOSE
1706 I=I-1:RETURN
1710 '
1720 '
1800 CLS:PRINT"KILL FILENAME  ";A$(NY3);:INPUT "  (Y/N)";A$
1830 IF A$="Y" OR A$="y" THEN KILL A$(NY3) ELSE RETURN
1840 FOR N=0 TO I2-F1:A$(NY3+N)=A$(NY3+1+N):NEXT N
1850 OPEN "FILENAME" FOR OUTPUT AS #1
1860 FOR N=1 TO I2-1:WRITE #1,A$(N):NEXT N:CLOSE:RETURN
1870 '
1880 '
1885 OPEN "FILENAME" FOR INPUT AS #1:I2=0
1887 IF EOF(1) THEN LOCATE 10,30:CLOSE:PRINT "No Track Data Files Available":NY3=5
1889 WHILE NOT EOF(1)
1891 I2=I2+1:INPUT #1,A$(I2)
1893 WEND:CLOSE:RETURN
1900 I=0
1902 Z$=LEFT$(A$(NY3),8)
1904 OPEN Z$ FOR INPUT AS #2
1906 WHILE NOT EOF(2)
1908 I=I+1
1910 INPUT #2,TRK(I,1),TRK(I,2),TRK(I,3),TRK(I,4),TRK(I,5),TRK(I,6)
```

```
1912 WEND
1914 Z=LOC(2)
1916 CLOSE
1917 ELMAX=-9999:ELMIN=9999
1918 FOR X=1 TO I
1920 IF TRK(X,6)>ELMAX THEN ELMAX=TRK(X,6)
1922 IF TRK(X,6)<ELMIN THEN ELMIN=TRK(X,6)
1924 NEXT
1926 RETURN
1940 '
1950 '
2000 RN=9:CN=1:NY4=21:DATUM=58:MAX=30:MIN=21:FMT2$="[\ \]":GOSUB 17000
2060 CLS:COLOR 15:LOCATE 1,33:PRINT"Train Consist":COLOR 7
2065 LOCATE 2,1:PRINT STRING$(80,"_")
2070 LOCATE 5,7:COLOR 15:PRINT"Car Type" TAB(25) "Length" TAB(37) "Ltwt" TAB(48) "Capy" TAB(58) "Cars" TAB(71) "Cars"
2080 LOCATE 6,26:PRINT"(Ft)" TAB(37) "(Lbs)" TAB(48) "(Lbs)" TAB(57) "Loaded" TAB(70) "Empty":COLOR 7
2090 LOCATE 9,8:PRINT"Boxcar" TAB(26) CAR(1,1) TAB(36) CAR(1,2) TAB(47) CAR(1,3) TAB(58);:PRINT USING "[\ \]         [\ \]";COEFF$(
2100 LOCATE 11,5:PRINT"Cvrd Hopper" TAB(26) CAR(2,1) TAB(36) CAR(2,2) TAB(47) CAR(2,3) TAB(58);:PRINT USING "[\ \]         [\ \]";C
2110 LOCATE 13,7:PRINT"Tank Car" TAB(26) CAR(3,1) TAB(36) CAR(3,2) TAB(47) CAR(3,3) TAB(58);:PRINT USING "[\ \]         [\ \]";COEF
2120 LOCATE 15,7:PRINT"Piggyback" TAB(26) CAR(4,1) TAB(36) CAR(4,2) TAB(47) CAR(4,3) TAB(58);:PRINT USING "[\ \]         [\ \]";COE
2130 LOCATE 17,7:PRINT"Hopper Car" TAB(26) CAR(5,1) TAB(36) CAR(5,2) TAB(47) CAR(5,3) TAB(58);:PRINT USING "[\ \]         [\ \]";CO
2135 LOCATE 25,33:COLOR 15:PRINT "[Esc] Exit";:COLOR 7:GOSUB 2630:GOTO 2230
2150 '
2160 A$=INKEY$:IF A$="" THEN 2160
2170 IF ASC(A$)=27 THEN GOSUB 785:GOSUB 2900:RETURN
2180 IF LEN(A$)=2 THEN GOSUB 900
2190 IF ASC(A$)>47 AND ASC(A$)<58 OR A$="." OR ASC(A$)=32 THEN GOSUB 770
2195 IF ASC(B$)=72 OR ASC(B$)=80 THEN GOSUB 785:GOSUB 2600:GOSUB 2630:B$="a"
2200 IF ASC(A$)=13 THEN GOSUB 785:GOSUB 2600:GOSUB 2630
2210 IF CN<1 THEN CN=1 ELSE IF CN>3 THEN CN=3:BEEP
2220 GOSUB 2400
2230 LOCATE RN,DATUM:PRINT "[  ]" :LOCATE RN,DATUM+1:COLOR 15
2240 PRINT USING "\ \";COEFF$(NY4):COLOR 7
2250 LOCATE RN,CN+DATUM,1
2260 GOTO 2160
2280 '
2290 '
2400 IF NY4=21 OR NY4=22 THEN RN=9
2420 IF NY4=23 OR NY4=24 THEN RN=11
2430 IF NY4=25 OR NY4=26 THEN RN=13
2440 IF NY4=27 OR NY4=28 THEN RN=15
2450 IF NY4=29 OR NY4=30 THEN RN=17
2455 DATUM=70
2460 IF NY4=21 OR NY4=23 OR NY4=25 OR NY4=27 OR NY4=29 THEN DATUM=58
2465 RETURN
2470 '
2600 SN=0:LT=0:WT=0:LTWT=0:X1=0
2603 FOR X=1 TO 5
2605 COUNT=COEFF(X+X1+20)+COEFF(X+X1+21)
2607 SN=SN+COUNT
2610 LT=LT+CAR(X,1)*COUNT
2620 LTWT=LTWT+CAR(X,2)*COUNT
2622 WT=WT+CAR(X,3)*COEFF(X+X1+20):X1=X1+1:NEXT
2625 WT=(WT+LTWT)/2000:LTWT=LTWT/2000:RETURN
2627 '
2630 LOCATE 20,27:PRINT "Total Number of Cars: ";:COLOR 15
2632 PRINT USING "  ###";SN:COLOR 7
2635 LOCATE 21,30:PRINT "Total Length (Ft): ";:COLOR 15
2637 PRINT USING "#####";LT:COLOR 7
2640 LOCATE 22,28:PRINT "Total Weight (Tons): ";:COLOR 15
2642 PRINT USING "#####";WT:COLOR 7:RETURN
2645 '
2900 NP=COEFF(27)+COEFF(28):NB=SN-NP
2910 NPWT=(CAR(4,2)*COEFF(28)+(CAR(4,2)+CAR(4,3))*COEFF(27))/2000
2920 NBWT=WT-NPWT
2930 A1=COEFF(7)*NBWT+COEFF(8)*4*NB+COEFF(11)*NPWT+COEFF(12)*4*NP
2940 B1=COEFF(9)*NBWT+COEFF(13)*NPWT
2950 C1=COEFF(10)*NB+COEFF(14)*NP:RETURN
2955 '
3000 MAX=36:MIN=31:NY4=31:DATUM=68:RN=7:CN=1:FMT2$="[\\]":GOSUB 18000
3050 CLS:COLOR 15:LOCATE 1,34:PRINT"Motive Power":COLOR 7
3060 LOCATE 2,1:PRINT STRING$(80,"_")
3065 LOCATE 4,8:COLOR 15:PRINT "Model" TAB(21) "Horsepower" TAB(37) "Weight" TAB(48) "No. of Axles" TAB(63) "No. of Units":LOCATE
3070 LOCATE 7,8:PRINT "SD40-2" TAB(23) LOCO(1,1) TAB(38) LOCO(1,2) TAB(53) LOCO(1,3) TAB(68):PRINT USING "[\\]";COEFF$(31)
3080 LOCATE 9,8:PRINT "GP38" TAB(23) LOCO(2,1) TAB(38) LOCO(2,2) TAB(53) LOCO(2,3) TAB(68);:PRINT USING "[\\]";COEFF$(32)
3090 LOCATE 11,8:PRINT "GP9" TAB(23) LOCO(3,1) TAB(38) LOCO(3,2) TAB(53) LOCO(3,3) TAB(68);:PRINT USING "[\\]";COEFF$(33)
3100 LOCATE 13,8:PRINT "M636" TAB(23) LOCO(4,1) TAB(38) LOCO(4,2) TAB(53) LOCO(4,3) TAB(68);:PRINT USING "[\\]";COEFF$(34)
3110 LOCATE 15,8:PRINT "M630" TAB(23) LOCO(5,1) TAB(38) LOCO(5,2) TAB(53) LOCO(5,3) TAB(68);:PRINT USING "[\\]";COEFF$(35)
3120 LOCATE 17,8:PRINT "C424" TAB(23) LOCO(6,1) TAB(38) LOCO(6,2) TAB(53) LOCO(6,3) TAB(68);:PRINT USING "[\\]";COEFF$(36):LOCATE
3125 PRINT "[Esc]  Exit";:COLOR 7:GOSUB 3525:GOTO 3230
3157 '
3160 A$=INKEY$:IF A$="" THEN 3160
3170 IF ASC(A$)=27 THEN GOSUB 785:GOSUB 3300:RETURN
3180 IF LEN(A$)=2 THEN GOSUB 900
3190 IF ASC(A$)>47 AND ASC(A$)<58 OR A$="." OR ASC(A$)=32 THEN GOSUB 770
3200 IF ASC(A$)=13 THEN GOSUB 785:GOSUB 3500:GOSUB 3525
3205 IF ASC(B$)=72 OR ASC(B$)=80 THEN GOSUB 785:GOSUB 3500:GOSUB 3525:B$="a"
3210 CN=1
3220 FOR X=1 TO NY4-30:RN=NY4+X-25:NEXT
3230 LOCATE RN,DATUM:PRINT "[ ]":LOCATE RN,DATUM+1:COLOR 15
3240 PRINT USING "\\";COEFF$(NY4):COLOR 7
3250 LOCATE RN,CN+DATUM,1:GOTO 3160
3260 '
3300 A2=COEFF(15)*LWT+COEFF(16)*LA
3310 B2=COEFF(17)*LWT
3320 C2=COEFF(18)*NS1:RETURN
3330 '
3340 '
3500 NS1=0:LWT=0:HP=0:LA=0
3505 FOR X=1 TO 6
3510 NS1=NS1+COEFF(X+30):LWT=LWT+COEFF(X+30)*LOCO(X,2):LA=LA+COEFF(X+30)*LOCO(X,3):HP=HP+COEFF(X+30)*LOCO(X,1):NEXT:RETURN
3520 '
3525 LOCATE 20,33:PRINT "Number of Units: ";:COLOR 15:PRINT USING "    ##";NS1:COLOR 7
3530 LOCATE 21,32:PRINT "Total Horsepower: ";:COLOR 15:PRINT USING "#####";HP:COLOR 7
3540 LOCATE 22,29:PRINT "Total Weight (Tons): ";:COLOR 15:PRINT USING " ####";LWT:COLOR 7:RETURN
3550 '
```

```
1025 LOCATE 2,1:PRINT STRING$(80,"_")
1030 GOSUB 1140:GOSUB 1144:GOSUB 1148:GOSUB 1152:GOSUB 1156
1035 '
1080 GOSUB 1885
1090 FOR I1=1 TO I2:LOCATE 4+I1,40-LEN(A$(I1))/2:PRINT A$(I1):NEXT I1
1092 WHILE ASC(A$)<>13
1097 COLOR 0,7:INV1=0:INV2=0:LOCATE 4+NY3,40-LEN(A$(NY3))/2:PRINT A$(NY3)
1098 ON NX3 GOSUB 1140,1144,1148,1152,1156:COLOR 7,0
1102 A$=INKEY$:IF A$="" THEN 1102 ELSE IF ASC(A$)=13 THEN 1118
1104 IF LEN(A$)=2 THEN B$=RIGHT$(A$,1) ELSE 1102
1106 IF ASC(B$)=72 THEN GOSUB 1125:NY3=NY3-1
1108 IF ASC(B$)=80 THEN GOSUB 1125:NY3=NY3+1
1110 IF ASC(B$)=77 THEN GOSUB 1125:NX3=NX3+1
1112 IF ASC(B$)=75 THEN GOSUB 1125:NX3=NX3-1
1114 IF NY3>I2 THEN NY3=1 ELSE IF NY3<1 THEN NY3=I2
1116 IF NX3>5  THEN NX3=1 ELSE IF NX3<1 THEN NX3=5
1118 WEND
1120 ON NX3 GOSUB 1900,10000,10000,1800,1187
1122 IF NX3=1 OR NX3=2 THEN RETURN ELSE 1000
1124 '
1125 INV1=223:INV2=220:LOCATE 4+NY3,40-LEN(A$(NY3))/2:PRINT A$(NY3)
1126 ON NX3 GOSUB 1140,1144,1148,1152,1156:RETURN
1127 '
1140 LOCATE 20,11:PRINT STRING$(6,223-INV1)
1141 LOCATE 22,11:PRINT STRING$(6,220-INV2)
1143 LOCATE 21,11:PRINT " LOAD ":RETURN
1144 LOCATE 20,23:PRINT STRING$(9,223-INV1)
1145 LOCATE 22,23:PRINT STRING$(9,220-INV2)
1147 LOCATE 21,23:PRINT " DISPLAY ":RETURN
1148 LOCATE 20,38:PRINT STRING$(6,223-INV1)
1149 LOCATE 22,38:PRINT STRING$(6,220-INV2)
1151 LOCATE 21,38:PRINT " EDIT ":RETURN
1152 LOCATE 20,50:PRINT STRING$(6,223-INV1)
1154 LOCATE 22,50:PRINT STRING$(6,220-INV2)
1155 LOCATE 21,50:PRINT " KILL ":RETURN
1156 LOCATE 20,62:PRINT STRING$(8,223-INV1)
1157 LOCATE 22,62:PRINT STRING$(8,220-INV2)
1159 LOCATE 21,62:PRINT " CREATE ":RETURN
1185 '
1187 CLS:X=0:LOCATE 10,20:INPUT "NAME OF FILE ";A$(I2+1)
1190 N=0:CLS
1200 N=N+1:A2$=""
1210 PRINT"MILEPOST";GR(N,1)
1220 PRINT "ELEVATION (FT)";:PRINT USING"######";GR(N,2)
1230 INPUT"GRADE (%)";A2$
1240 IF A2$="" THEN 1290 ELSE GR(N,3)=VAL(A2$)
1250 INPUT"DISTANCE (MILES)";GR(N,4):PRINT:PRINT
1260 GR(N+1,2)=SIN(ATN(GR(N,3)/100))*GR(N,4)*5280+GR(N,2)
1270 GR(N+1,1)=GR(N,1)+GR(N,4):GOTO 1200
1280 '
1290 K=0:CLS
1300 K=K+1:A2$=""
1305 IF SP(K,1)>GR(N,1) THEN SP(K,1)=GR(N,1):GOTO 1370
1310 PRINT"MILEPOST";SP(K,1)
1320 INPUT"SPEED LIMIT (MPH)";A2$
1330 IF A2$=""THEN 1370 ELSE SP(K,2)=VAL(A2$)
1335 IF A2$="0" THEN 1370
1340 INPUT"DISTANCE (MILES)";SP(K,3):PRINT:PRINT
1345 IF SP(K,3)=0 THEN BEEP:GOTO 1340
1350 SP(K+1,1)=SP(K,1)+SP(K,3):GOTO 1300
1360 '
1370 I=0:CLS
1380 I=I+1:A2$=""
1385 IF CV(I,1)>SP(K,1) THEN CV(I,1)=SP(K,1):GOTO 1460
1390 PRINT"MILEPOST";CV(I,1)
1400 INPUT"DEGREE CURVE";A2$
1410 IF A2$="" THEN 1460 ELSE CV(I,2)=VAL(A2$)
1420 INPUT"LENGTH (MILES)";CV(I,3):PRINT:PRINT
1430 CV(I,4)=CV(I,2)*.04
1440 CV(I+1,1)=CV(I,1)+CV(I,3):GOTO 1380
1450 '
1460 G=1:S=1:C=1
1470 X=X+1
1480 IF GR(G,1)<SP(S,1) AND GR(G,1)<CV(C,1) GOTO 1600
1490 IF SP(S,1)<GR(G,1) AND SP(S,1)<CV(C,1) GOTO 1550
1500 IF CV(C,1)<GR(G,1) AND CV(C,1)<SP(S,1) GOTO 1560
1510 IF GR(G,1)=SP(S,1) AND SP(S,1)=CV(C,1)AND CV(C,1)=GR(G,1) GOTO 1570
1520 IF GR(G,1)=SP(S,1) AND GR(G,1)<CV(C,1) GOTO 1580
1530 IF CV(C,1)=GR(G,1) AND GR(G,1)<SP(S,1) GOTO 1590
1540 OS=S:OC=C:S=S+1:C=C+1:TRK(X,1)=SP(OS,1):GOTO 1610
1550 OS=S:S=S+1:TRK(X,1)=SP(OS,1):GOTO 1610
1560 OC=C:C=C+1:TRK(X,1)=CV(OC,1):GOTO 1610
1570 OS=S:OC=C:S=S+1:C=C+1: GOTO 1600
1580 OS=S:S=S+1:GOTO 1600
1590 OC=C:C=C+1
1600 OG=G:G=G+1:TRK(X,1)=GR(OG,1)
1610 TRK(X,2)=SP(OS,2):TRK(X,3)=GR(OG,3):TRK(X,4)=CV(OC,2):TRK(X,5)=GR(OG,3)+CV(OC,4):TRK(X,6)=TRK(X-1,6)+(TRK(X,1)-TRK(X-1,1))*5
1623 IF X<>1 AND TRK(X,1)=0 OR TRK(X,2)=0 THEN X=X-1:GOTO 1640
1625 GOTO 1470
1630 '
1640 IF A=4 THEN 1670
1650 OPEN "FILENAME" FOR OUTPUT AS #1
1660 FOR I1=1 TO I2+1:WRITE #1,A$(I1):NEXT I1:CLOSE:A$=A$(I2+1)
1665 '
1670 OPEN A$ FOR OUTPUT AS #1
1680 FOR I=1 TO X
1690 WRITE #1,TRK(I,1),TRK(I,2),TRK(I,3),TRK(I,4),TRK(I,5),TRK(I,6)
1700 NEXT
1704 CLOSE
1706 I=I-1:RETURN
1710 '
1720 '
1800 CLS:PRINT"KILL FILENAME  ";A$(NY3);:INPUT "  (Y/N)";A$
1830 IF A$="Y" OR A$="y" THEN KILL A$(NY3) ELSE RETURN
1840 FOR N=0 TO I2-F1:A$(NY3+N)=A$(NY3+1+N):NEXT N
1850 OPEN "FILENAME" FOR OUTPUT AS #1
1860 FOR N=1 TO I2-1:WRITE #1,A$(N):NEXT N:CLOSE:RETURN
1870 '
1880 '
1885 OPEN "FILENAME" FOR INPUT AS #1:I2=0
1887 IF EOF(1) THEN LOCATE 10,30:CLOSE:PRINT "No Track Data Files Available":NY3=5
1889 WHILE NOT EOF(1)
1891 I2=I2+1:INPUT #1,A$(I2)
1893 WEND:CLOSE:RETURN
1900 I=0
1902 Z$=LEFT$(A$(NY3),8)
1904 OPEN Z$ FOR INPUT AS #2
1906 WHILE NOT EOF(2)
1908 I=I+1
1910 INPUT #2,TRK(I,1),TRK(I,2),TRK(I,3),TRK(I,4),TRK(I,5),TRK(I,6)
```

```
1912 WEND
1914 Z=LOC(2)
1916 CLOSE
1917 ELMAX=-9999:ELMIN=9999
1918 FOR X=1 TO I
1920 IF TRK(X,6)>ELMAX THEN ELMAX=TRK(X,6)
1922 IF TRK(X,6)<ELMIN THEN ELMIN=TRK(X,6)
1924 NEXT
1926 RETURN
1940 '
1950 '
2000 RN=9:CN=1:NY4=21:DATUM=58:MAX=30:MIN=21:FMT2$="[\ \]":GOSUB 17000
2060 CLS:COLOR 15:LOCATE 1,33:PRINT"Train Consist":COLOR 7
2065 LOCATE 2,1:PRINT STRING$(80,"_")
2070 LOCATE 5,7:COLOR 15:PRINT"Car Type" TAB(25) "Length" TAB(37) "Ltwt" TAB(48) "Capy" TAB(58) "Cars" TAB(71) "Cars"
2080 LOCATE 6,26:PRINT"(Ft)" TAB(37) "(Lbs)" TAB(48) "(Lbs)" TAB(57) "Loaded" TAB(70) "Empty":COLOR 7
2090 LOCATE 9,8:PRINT"Boxcar" TAB(26) CAR(1,1) TAB(36) CAR(1,2) TAB(47) CAR(1,3) TAB(58);:PRINT USING "[\ \]          [\ \]";COEFF$(
2100 LOCATE 11,5:PRINT"Cvrd Hopper" TAB(26) CAR(2,1) TAB(36) CAR(2,2) TAB(47) CAR(2,3) TAB(58);:PRINT USING "[\ \]          [\ \]";C
2110 LOCATE 13,7:PRINT"Tank Car" TAB(26) CAR(3,1) TAB(36) CAR(3,2) TAB(47) CAR(3,3) TAB(58);:PRINT USING "[\ \]          [\ \]";COEF
2120 LOCATE 15,7:PRINT"Piggyback" TAB(26) CAR(4,1) TAB(36) CAR(4,2) TAB(47) CAR(4,3) TAB(58);:PRINT USING "[\ \]          [\ \]";COE
2130 LOCATE 17,7:PRINT"Hopper Car" TAB(26) CAR(5,1) TAB(36) CAR(5,2) TAB(47) CAR(5,3) TAB(58);:PRINT USING "[\ \]          [\ \]";CO
2135 LOCATE 25,33:COLOR 15:PRINT "[Esc] Exit";:COLOR 7:GOSUB 2630:GOTO 2230
2150 '
2160 A$=INKEY$:IF A$="" THEN 2160
2170 IF ASC(A$)=27 THEN GOSUB 785:GOSUB 2900:RETURN
2180 IF LEN(A$)=2 THEN GOSUB 900
2190 IF ASC(A$)>47 AND ASC(A$)<58 OR A$="." OR ASC(A$)=32 THEN GOSUB 770
2195 IF ASC(B$)=72 OR ASC(B$)=80 THEN GOSUB 785:GOSUB 2600:GOSUB 2630:B$="a"
2200 IF ASC(A$)=13 THEN GOSUB 785:GOSUB 2600:GOSUB 2630
2210 IF CN<1 THEN CN=1 ELSE IF CN>3 THEN CN=3:BEEP
2220 GOSUB 2400
2230 LOCATE RN,DATUM:PRINT "[  ]" :LOCATE RN,DATUM+1:COLOR 15
2240 PRINT USING "\ \";COEFF$(NY4):COLOR 7
2250 LOCATE RN,CN+DATUM,1
2260 GOTO 2160
2280 '
2290 '
2400 IF NY4=21 OR NY4=22 THEN RN=9
2420 IF NY4=23 OR NY4=24 THEN RN=11
2430 IF NY4=25 OR NY4=26 THEN RN=13
2440 IF NY4=27 OR NY4=28 THEN RN=15
2450 IF NY4=29 OR NY4=30 THEN RN=17
2455 DATUM=70
2460 IF NY4=21 OR NY4=23 OR NY4=25 OR NY4=27 OR NY4=29 THEN DATUM=58
2465 RETURN
2470 '
2600 SN=0:LT=0:WT=0:LTWT=0:X1=0
2603 FOR X=1 TO 5
2605 COUNT=COEFF(X+X1+20)+COEFF(X+X1+21)
2607 SN=SN+COUNT
2610 LT=LT+CAR(X,1)*COUNT
2620 LTWT=LTWT+CAR(X,2)*COUNT
2622 WT=WT+CAR(X,3)*COEFF(X+X1+20):X1=X1+1:NEXT
2625 WT=(WT+LTWT)/2000:LTWT=LTWT/2000:RETURN
2627 '
2630 LOCATE 20,27:PRINT "Total Number of Cars: ";:COLOR 15
2632 PRINT USING "  ###";SN:COLOR 7
2635 LOCATE 21,30:PRINT "Total Length (Ft): ";:COLOR 15
2637 PRINT USING "#####";LT:COLOR 7
2640 LOCATE 22,28:PRINT "Total Weight (Tons): ";:COLOR 15
2642 PRINT USING "#####";WT:COLOR 7:RETURN
2645 '
2900 NP=COEFF(27)+COEFF(28):NB=SN-NP
2910 NPWT=(CAR(4,2)*COEFF(28)+(CAR(4,2)+CAR(4,3))*COEFF(27))/2000
2920 NBWT=WT-NPWT
2930 A1=COEFF(7)*NBWT+COEFF(8)*4*NB+COEFF(11)*NPWT+COEFF(12)*4*NP
2940 B1=COEFF(9)*NBWT+COEFF(13)*NPWT
2950 C1=COEFF(10)*NB+COEFF(14)*NP:RETURN
2955 '
3000 MAX=36:MIN=31:NY4=31:DATUM=68:RN=7:CN=1:FMT2$="[\\]":GOSUB 18000
3050 CLS:COLOR 15:LOCATE 1,34:PRINT"Motive Power":COLOR 7
3060 LOCATE 2,1:PRINT STRING$(80,"_")
3065 LOCATE 4,8:COLOR 15:PRINT "Model" TAB(21) "Horsepower" TAB(37) "Weight" TAB(48) "No. of Axles" TAB(63) "No. of Units":LOCATE
3070 LOCATE 7,8:PRINT "SD40-2" TAB(23) LOCO(1,1) TAB(38) LOCO(1,2) TAB(53) LOCO(1,3) TAB(68):PRINT USING "[\\]";COEFF$(31)
3080 LOCATE 9,8:PRINT "GP38" TAB(23) LOCO(2,1) TAB(38) LOCO(2,2) TAB(53) LOCO(2,3) TAB(68);:PRINT USING "[\\]";COEFF$(32)
3090 LOCATE 11,8:PRINT "GP9" TAB(23) LOCO(3,1) TAB(38) LOCO(3,2) TAB(53) LOCO(3,3) TAB(68);:PRINT USING "[\\]";COEFF$(33)
3100 LOCATE 13,8:PRINT "M636" TAB(23) LOCO(4,1) TAB(38) LOCO(4,2) TAB(53) LOCO(4,3) TAB(68);:PRINT USING "[\\]";COEFF$(34)
3110 LOCATE 15,8:PRINT "M630" TAB(23) LOCO(5,1) TAB(38) LOCO(5,2) TAB(53) LOCO(5,3) TAB(68);:PRINT USING "[\\]";COEFF$(35)
3120 LOCATE 17,8:PRINT "C424" TAB(23) LOCO(6,1) TAB(38) LOCO(6,2) TAB(53) LOCO(6,3) TAB(68);:PRINT USING "[\\]";COEFF$(36):LOCATE
3125 PRINT "[Esc]  Exit";:COLOR 7:GOSUB 3525:GOTO 3230
3157 '
3160 A$=INKEY$:IF A$="" THEN 3160
3170 IF ASC(A$)=27 THEN GOSUB 785:GOSUB 3300:RETURN
3180 IF LEN(A$)=2 THEN GOSUB 900
3190 IF ASC(A$)>47 AND ASC(A$)<58 OR A$="." OR ASC(A$)=32 THEN GOSUB 770
3200 IF ASC(A$)=13 THEN GOSUB 785:GOSUB 3500:GOSUB 3525
3205 IF ASC(B$)=72 OR ASC(B$)=80 THEN GOSUB 785:GOSUB 3500:GOSUB 3525:B$="a"
3210 CN=1
3220 FOR X=1 TO NY4-30:RN=NY4+X-25:NEXT
3230 LOCATE RN,DATUM:PRINT "[ ]":LOCATE RN,DATUM+1:COLOR 15
3240 PRINT USING "\\";COEFF$(NY4):COLOR 7
3250 LOCATE RN,CN+DATUM,1:GOTO 3160
3260 '
3300 A2=COEFF(15)*LWT+COEFF(16)*LA
3310 B2=COEFF(17)*LWT
3320 C2=COEFF(18)*NS1:RETURN
3330 '
3340 '
3500 NS1=0:LWT=0:HP=0:LA=0
3505 FOR X=1 TO 6
3510 NS1=NS1+COEFF(X+30):LWT=LWT+COEFF(X+30)*LOCO(X,2):LA=LA+COEFF(X+30)*LOCO(X,3):HP=HP+COEFF(X+30)*LOCO(X,1):NEXT:RETURN
3520 '
3525 LOCATE 20,33:PRINT "Number of Units: ";:COLOR 15:PRINT USING "    ##";NS1:COLOR 7
3530 LOCATE 21,32:PRINT "Total Horsepower: ";:COLOR 15:PRINT USING "#####";HP:COLOR 7
3540 LOCATE 22,29:PRINT "Total Weight (Tons): ";:COLOR 15:PRINT USING " ####";LWT:COLOR 7:RETURN
3550 '
```

Main Program

```
4000 CLS:SCREEN 0:Y=0:Z=0:A3=A1+A2:B3=B1+B2:C3=C1+C2:MPB=9999
4010 LOCATE 9,31:COLOR 31:PRINT "RUNNING SIMULATION":COLOR 7
4020 TEM=COEFF(4)*20*LWT:IF TEM>350000! THEN TEM=350000!
4030 TS(1)=COEFF(2):N=1:J=1:GOSUB 4500:GOSUB 4700:N=1:J=2:VL=TRK(1,2)
4060 IF TS(1)=0 THEN N=0:I1=0
4065 '
4066 '
4070 IF N<2 THEN 4075 ELSE IF TS(N)<=0 OR J=I+2 THEN GOSUB 4950:BEEP:GOSUB 5000:RETURN
4075 N=N+1:GOSUB 4400:TS(N)=TS(N-1)+I1
4080 COLOR 15:LOCATE 23,62:PRINT USING "MILEPOST:####.##";MP(N-1):COLOR 7
4090 GOSUB 4500:OVL=VL:VL=MS(N):IF VL>COEFF(3) THEN VL=COEFF(3)
4095 '
4100 ON (SGN(TS(N)-VL)+2) GOSUB 4110,4200,4300:GOTO 4070
4110 BF(N)=0:IF TS(N-1)=OVL AND ABS(TR(N-1))=(BF(N-1)+TE(N-1)) THEN GOSUB 4600:GOSUB 4910:RETURN
4111 IF(TE(N-1)-TR(N-1)-BF(N-1))*(TE(N)-TR(N)-BF(N))<=0 THEN GOSUB 4965:BF(N)=0:RETURN
4113 GOSUB 4700:GOSUB 4800:BF(N)=0:RETURN
4114 '
4200 ON (SGN(TR(N))+2) GOSUB 4210,4230,4260:RETURN
4210 TE(N)=0
4212 IF MBF<ABS(TR(N)) THEN BF(N)=MBF:GOSUB 4960:I1=1:GOSUB 4800:RETURN
4214 IF TS(N-1)+1=VL OR TS(N-1)-1=VL THEN 4218
4216 BF(N)=ABS(TR(N)):AC(N)=0:I1=0:GOSUB 4600:J=J+1:RETURN
4218 GOSUB 4700:IF MP(N)>TRK(J,1) THEN GOSUB 4810:RETURN
4219 BF(N)=ABS(TR(N)):AC(N)=0:I1=0:J=J+1:RETURN
4220 '
4230 TE(N)=0:BF(N)=0:GOSUB 4600:J=J+1:I1=0:RETURN
4232 '
4260 BF(N)=0:IF TE(N)<TR(N) THEN GOSUB 4960:I1=-1:GOSUB 4800:RETURN
4265 IF TS(N-1)+1=VL OR TS(N-1)-1=VL THEN 4275
4270 TE(N)=TR(N):AC(N)=0:I1=0:GOSUB 4600:J=J+1:RETURN
4275 GOSUB 4700:IF MP(N)>TRK(J,1) THEN GOSUB 4810:RETURN
4276 TE(N)=TR(N):AC(N)=0:I1=0:J=J+1:RETURN
4277 '
4279 '
4300 BF(N)=MBF:TE(N)=0
4302 IF TS(N-1)=OVL AND ABS(TR(N-1))=(BF(N-1)+TE(N-1)) THEN GOSUB 4600:GOSUB 4910:RETURN
4303 IF(TE(N-1)-TR(N-1)-BF(N-1))*(TE(N)-TR(N)-BF(N))<=0 THEN GOSUB 4965:TE(N)=0:RETURN
4305 GOSUB 4700:GOSUB 4800:TE(N)=0:RETURN
4310 '
4320 '
4400 G(N)=TRK(J-1,3):GE(N)=TRK(J-1,5)
4410 DC(N)=TRK(J-1,4):MS(N)=TRK(J-1,2):RETURN
4420 '
4430 '
4500 TR(N)=A3+B3*TS(N)+C3*TS(N)^2+WT*2000*SIN(ATN(GE(N)/100))
4505 MBF=LTWT*(340-48*LOG((TS(N)+.3)/.3))
4510 IF TS(N)=0 THEN TE(N)=TEM:RETURN
4520 TE(N)=3.75*COEFF(5)*HP/TS(N)
4530 IF TE(N)>TEM THEN TE(N)=TEM
4540 RETURN
4550 '
4560 '
4600 D3=TRK(J-1,1)
4610 D1=D3-MP(N-1)
4625 IF TS(N-1)=OVL THEN T1=D1*3600/OVL ELSE T1=D1*3600/VL
4630 T(N)=T(N-1)+T1
4640 WK(N)=WK(N-1)+TE(N-1)*D1*.00528
4650 MP(N)=D3
4660 AC(N)=(TE(N)-TR(N)-BF(N))*.011/WT:RETURN
4670 '
4700 ABF=(BF(N)+BF(N-1))*.5
4710 ATE=(TE(N)+TE(N-1))*.5
4720 AC(N)=(ATE-((TR(N)+TR(N-1))*.5+ABF))*.011/WT
4722 IF MP(N-1)=TRK(J-1,1) THEN AC(N)=(TE(N)-(TR(N)+BF(N)))*.011/WT
4730 T1=ABS(I1/AC(N))
4740 T(N)=T(N-1)+T1
4750 D1=TS(N-1)*T1/3600+AC(N)*T1^2/7200
4760 MP(N)=MP(N-1)+D1
4770 WK(N)=WK(N-1)+ATE*D1*.00528
4780 RETURN
4790 '
4800 IF MP(N)<TRK(J,1) THEN I1=SGN(AC(N)):RETURN
4810 TS(N)=(TS(N-1)^2+7200*AC(N)*(TRK(J,1)-MP(N-1)))^.5
4815 I1=TS(N)-TS(N-1)
4820 GOSUB 4500
4822 IF TS(N)>VL THEN BF(N)=MBF:TE(N)=0
4825 GOSUB 4700
4830 MP(N)=TRK(J,1):J=J+1
4831 I1=INT(TS(N)+ABS(SGN(TE(N)-TR(N)-BF(N))))-TS(N):RETURN
4840 '
4910 I1=SGN(AC(N))
4914 IF TS(N)<>FIX(TS(N)) THEN I1=INT(TS(N)+ABS(SGN(TE(N)-TR(N)-BF(N))))-TS(N)
4925 RETURN
4950 GOSUB 4500:I1=1:IF BF(N-1)>0 THEN BF(N)=MBF:TE(N)=0:GOSUB 4700:RETURN
4960 IF TS(N-1)=OVL AND ABS(TR(N-1))=(BF(N-1)+TE(N-1)) THEN GOSUB 4600 ELSE GOSUB 4700
4963 RETURN
4964 '
4965 IF MP(N-1)=TRK(J-1,1) THEN GOSUB 4980:RETURN
4967 TTS=TS(N):IF TE(N)>TR(N) THEN 4970
4968 TS(N)=TS(N)-.01:GOSUB 4500:IF ABS(TS(N)-(TTS-1))<.01 THEN I1=TS(N)-TS(N-1):GOSUB 4700:GOSUB 4800:RETURN
4969 IF (TR(N)-TE(N))>(.001*TE(N)) THEN 4968 ELSE 4972
4970 TS(N)=TS(N)+.01:GOSUB 4500:IF ABS(TS(N)-(TTS+1))<.01 THEN I1=TS(N)-TS(N-1):GOSUB 4700:GOSUB 4800:RETURN
4971 IF (TE(N)-TR(N))>(.001*TE(N)) THEN 4970
4972 I1=TS(N)-TS(N-1):GOSUB 4700:GOSUB 4800
4973 IF MP(N)=TRK(J-1,1) THEN RETURN
4974 AC(N)=0:I1=0:J=J+1:VL=TS(N):TE(N)=TR(N):RETURN
4977 '
4980 IF (TE(N)-TR(N)-BF(N))<0 THEN I1=I1-1 ELSE I1=I1+1
4985 TS(N)=TS(N-1)+I1:GOSUB 4500:GOSUB 4700:GOSUB 4800:RETURN
4990 '
4995 '
```

```
5000 NY2=1
5002 CLS:A$="a":COLOR 15:LOCATE 2,32,0:PRINT "Output Selection"
5003 LOCATE 24,35:PRINT "[Esc]  Exit":COLOR 7
5004 LOCATE 3,1:PRINT STRING$(80,"_")
5005 GOSUB 5064:GOSUB 5070:GOSUB 5076
5006 WHILE ASC(A$)<>13
5008 COLOR 0,7:INV1=0:INV2=0:ON NY2 GOSUB 5060,5066,5072:COLOR 7,0
5010 A$=INKEY$:IF A$="" THEN 5010 ELSE IF ASC(A$)=13 THEN 5050
5011 IF ASC(A$)=27 THEN RETURN
5012 IF LEN(A$)=2 THEN B$=RIGHT$(A$,1) ELSE 5010
5014 IF ASC(B$)=72 THEN GOSUB 5055:NY2=NY2-1
5016 IF ASC(B$)=80 THEN GOSUB 5055:NY2=NY2+1
5018 IF NY2>3 THEN NY2=1 ELSE IF NY2<1 THEN NY2=3
5020 WEND
5050 ON NY2 GOSUB 5100,5200,7500
5051 NY2=NY2+1:IF NY2>3 THEN NY2=1:GOTO 5002
5052 GOTO 5002
5053 '
5055 INV1=223:INV2=220:ON NY2 GOSUB 5060,5066,5072:RETURN
5057 '
5060 LOCATE 7,34:PRINT STRING$(14,223-INV1)
5062 LOCATE 9,34:PRINT STRING$(14,220-INV2)
5064 LOCATE 8,34:PRINT " Trip Summary ":RETURN
5066 LOCATE 9,35:PRINT STRING$(13,223-INV1)
5068 LOCATE 11,35:PRINT STRING$(13,220-INV2)
5070 LOCATE 10,35:PRINT " Data Tables ":RETURN
5072 LOCATE 11,37:PRINT STRING$(8,223-INV1)
5074 LOCATE 13,37:PRINT STRING$(8,220-INV2)
5076 LOCATE 12,37:PRINT " Graphs ":RETURN
5078 '
5100 OQ=1:CLS:COLOR 15:PRINT"TRAIN INFORMATION":PRINT A$(NX3)
5102 COLOR 7:LOCATE 5,1:PRINT"NUMBER OF UNITS  ";NS1 TAB(40)
5103 PRINT"NUMBER OF CARS  ";SN :PRINT"TOTAL HORSEPOWER  ";HP TAB(40)
5104 PRINT"TRAIN LENGTH (FT)  ";LT:PRINT"TOTAL UNIT WEIGHT (TONS)  ";LWT TAB(40)
5105 PRINT"TRAIN GROSS WEIGHT (TONS)  ";WT
5106 PRINT "ASSUMED ADHESION  ";COEFF(4)/100 TAB(40)
5107 PRINT"TRAIN LIGHTWEIGHT (TONS)  ";LTWT:PRINT TAB(40) "TRAIN PAYLOAD (TONS)  ";WT-LTWT
5110 PRINT:PRINT:PRINT "HP/TON GROSS WEIGHT";:PRINT USING "   ##.#";(HP/WT);
5111 PRINT TAB(40);"TOTAL TIME (H:M:S)  ";
5112 T(R)=T(N):GOSUB 9500:PRINT T$
5114 PRINT"HP/TON PAYLOAD  ";
5115 IF WT=LTWT THEN PRINT " EMPTY TRAIN"; ELSE PRINT USING " ##.#";HP/(WT-LTWT);
5117 PRINT TAB(40);"AVERAGE VELOCITY (MPH)";:PRINT USING "####.#";MP(N)*3600/T(N)
5120 PRINT:PRINT:PRINT"DIESEL FUEL CONSUMED (GALS)";:PRINT USING "  ####";WK(N)*.0324
5130 PRINT"TON MILES/GALLON (GROSS WEIGHT)";:PRINT USING "  ####.#";WT*MP(N)/(WK(N)*.0324)
5140 PRINT"TON MILES/GALLON (PAYLOAD)";:IF WT=LTWT THEN PRINT "  EMPTY TRAIN" ELSE PRINT USING "  ###.#";(WT-LTWT)*MP(N)/(WK(N)*.
5150 LOCATE 24,35:COLOR 15:PRINT"[Esc]  Exit";:COLOR 7
5160 INPT$=INKEY$:IF INPT$="" THEN 5160 ELSE IF ASC(INPT$)=27 THEN RETURN
5170 GOTO 5160
5180 '
5200 TP=INT((N-1)/15)+1:R1=TP*15-14:TNL=N:NL=15
5210 GOSUB 5500:CLS:GOSUB 6000:FOR R=OQ TO Q:GOSUB 7000:NEXT R
5220 INPT$=INKEY$:IF INPT$="" THEN 5220
5225 IF ASC(INPT$)=27 THEN RETURN
5230 IF ASC(INPT$)=112 OR ASC(INPT$)=80 THEN GOSUB 5320:GOTO 5220
5235 IF LEN(INPT$)=2 THEN INPT$=RIGHT$(INPT$,1) ELSE 5220
5237 IF ASC(INPT$)=59 THEN GOSUB 5300:GOTO 5210
5240 IF ASC(INPT$)=81 THEN OQ=OQ+15:GOTO 5210
5250 IF ASC(INPT$)=73 THEN OQ=OQ-15:GOTO 5210
5260 GOTO 5220
5290 '
5300 LOCATE 25,30:INPUT "GOTO PAGE NUMBER";PGN$:PN=FIX(VAL(PGN$))
5310 OQ=(PN-1)*NL+1:RETURN
5315 '
5320 WIDTH "LPT1:",132:LPRINT CHR$(15):GOSUB 8000
5325 FOR R=1 TO N:GOSUB 9000:INPT$=INKEY$:IF ASC(INPT$)=27 THEN RETURN:NEXT
5330 WIDTH "LPT1:",80:LPRINT CHR$(18):RETURN
5340 '
5350 '
5500 IF OQ>R1 THEN OQ=1
5510 IF OQ<1 THEN OQ=R1
5520 Q=OQ+(NL-1):IF Q>TNL THEN Q=TNL
5530 PN=INT(OQ/NL)+1:RETURN
5535 '
6000 COLOR 15:PRINT A$(NY3);:PRINT TAB(65) "PAGE";PN;:PRINT "OF";TP:PRINT:PRINT"MILEPOST" TAB(11) "GRADE" TAB(18) "SPEED" TAB(25)
6005 PRINT "TRAIN" TAB(34) "TRAIN" TAB(43) "TRACTIVE" TAB(52) "BRAKING" TAB(63) "TIME" TAB(71) "WORK"
6010 PRINT TAB(12) "(%)" TAB(18) "LIMIT" TAB(25) "SPEED" TAB(32) "RESISTANCE" TAB(44) "EFFORT" TAB(53) "FORCE" TAB(61) "(H:M:S)"
6020 PRINT TAB(18) "(MPH)" TAB(25) "(MPH)" TAB(34) "(LBS)" TAB(45) "(LBS)" TAB(53) "(LBS)" TAB(70) "(X 1E6)":PRINT
6030 LOCATE 23,6:PRINT "[PG UP]     [PG DN]     [F1] GOTO PAGE     [ESC] EXIT     [P] PRINT":COLOR 7:LOCATE 7,1:RETURN
6040 '
7000 GOSUB 9500:PRINT USING " ###.##";MP(R);:PRINT USING "   +#.##";G(R);
7010 PRINT USING "   ###";MS(R);:PRINT USING "   ###.#";TS(R);
7020 PRINT USING "   +######";TR(R);:PRINT USING "    ######";TE(R);
7030 PRINT USING "  #######";BF(R);:PRINT "  ";T$;
7040 PRINT USING "  ######";WK(R):RETURN
7050 '
7060 '
7500 SG=0:EG=MP(N-1):FMT2$="[\    \]"
7510 GOSUB 7549:GOSUB 7950:GOSUB 7900:GOSUB 7850:GOSUB 7800
7520 IF ASC(B$)=60 THEN GOSUB 7808:GOTO 7510
7530 SCREEN 0:RETURN
7535 '
7549 CLS:SCREEN 2:LK=1:LOCATE 25,27:PRINT "[F2] SCALE CHANGE    [ESC] EXIT"
7550 LOCATE 4,1:PRINT"M":PRINT"P":PRINT"H"
7551 LOCATE 11,70:PRINT"MILEPOST"
7552 LOCATE 13,1:PRINT"E":PRINT"L":PRINT"E":PRINT"V":PRINT:PRINT"F"
7553 PRINT"E":PRINT"E":PRINT"T"
7555 FOR YY=80 TO 0 STEP -10
7556 LOCATE LK,5:PRINT USING "##";YY
7557 LK=LK+1:NEXT
7560 LINE(50,0)-(590,67),,B
7570 FOR YY=3 TO 59 STEP 8
7580 LINE (50,YY)-(53,YY):NEXT
7590 FOR XX=110 TO 590 STEP 60
7600 LINE(XX,67)-(XX,65):NEXT
7690 '
7700 LK=12
7701 TEMP=1.2*ELMAX-.8*ELMIN
7702 IF TEMP<10000 THEN DIV=1000
7703 IF TEMP<1000 THEN DIV=100
7704 IF TEMP<100  THEN DIV=10
7705 TP=(FIX(1.2*ELMAX/DIV)+1)*DIV:BT=(FIX(.8*ELMIN/DIV)-1)*DIV
7706 IN=ABS(TP-BT)/5
7710 FOR IYY=TP TO BT STEP -1*IN
7720 LOCATE LK,3:PRINT USING "####";IYY
7730 LK=LK+2:NEXT
7740 LINE (50,88)-(590,172),,B
7750 FOR YY=92 TO 172 STEP 16
7760 LINE (50,YY)-(53,YY):NEXT
7770 FOR XX=110 TO 590 STEP 60
```

```
7780 LINE (XX,172)-(XX,170):NEXT:RETURN
7787 '
7789 '
7800 A$=INKEY$:IF A$="" THEN 7800 ELSE IF ASC(A$)=27 THEN RETURN
7804 B$="a":IF LEN(A$)=2 THEN B$=RIGHT$(A$,1) ELSE 7800:RETURN
7805 RETURN
7806 '
7808 RN=2:CN=1:NY4=19:DATUM=71:MAX=20:MIN=19:COEFF$(19)=STR$(SG):COEFF$(20)=STR$(EG):OCN=1:ORN=2
7809 FOR X=19 TO 20:COEFF$(X)=RIGHT$(COEFF$(X),LEN(COEFF$(X))-1):NEXT
7810 GET(487,3)-(639,53),P
7811 PUT (487,3),P,XOR:LINE (486,4)-(638,52),,B
7812 LOCATE 2,64:PRINT USING "START  [\     \]";COEFF$(19)
7814 LOCATE 3,64:PRINT "GRAPH"
7815 LOCATE 5,64:PRINT USING " END   [\     \]";COEFF$(20)
7816 LOCATE 6,64:PRINT "GRAPH":LOCATE 3,72:PRINT CHR$(24):GET(568,16)-(576,23),P
7818 A$=INKEY$:IF A$="" THEN 7818 ELSE IF ASC(A$)=27 THEN RETURN
7820 IF LEN(A$)=2 THEN GOSUB 900
7822 IF ASC(A$)>47 AND ASC(A$)<58 OR A$="." OR ASC(A$)=32 THEN GOSUB 770
7824 IF CN<1 THEN CN=1 ELSE IF CN>6 THEN CN=6
7826 IF NY4=19 THEN RN=2 ELSE RN=5
7828 LOCATE RN,DATUM:PRINT USING "[\     \]";COEFF$(NY4)
7829 PUT((71+CN-1)*8,RN*8),P,XOR:PUT((71+OCN-1)*8,ORN*8),P,XOR
7830 OCN=CN:ORN=RN
7832 SG=VAL(COEFF$(19)):EG=VAL(COEFF$(20)):GOTO 7818
7840 '
7850 FOR KK=1 TO N
7855 IF MP(KK)<=EG THEN KKSTOP=KK
7856 IF MP(KK)<=SG THEN KKSTART=KK
7857 NEXT
7860 FOR KK=1 TO I
7865 IF TRK(KK,1)<=EG THEN KKSTOP2=KK
7866 IF TRK(KK,1)<=SG THEN KKSTART2=KK
7867 NEXT
7870 SGEL=TRK(KKSTART2,6)+52.8*(SG-TRK(KKSTART2,1))*TRK(KKSTART2,3)
7875 EGEL=TRK(KKSTOP2,6)+52.8*(EG-TRK(KKSTOP2,1))*TRK(KKSTOP2,3)
7877 '
7880 PSET(50,67-TS(KKSTART)*.8)
7885 FOR KK=KKSTART+1 TO KKSTOP
7887 LINE-((50+(MP(KK)-SG)*60/INC),(67-TS(KK)*.8)):NEXT
7890 LINE -(590,67-TS(KKSTOP+1)*.8):PSET(50,172-(SGEL-BT)*16/IN)
7892 FOR KK=KKSTART2+1 TO KKSTOP2
7894 LINE-((50+(TRK(KK,1)-SG)*60/INC),(172-((TRK(KK,6)-BT)*16/IN))):NEXT
7896 LINE-(590,172-(EGEL-BT)*16/IN):RETURN
7897 '
7898 '
7900 FOR RC=10 TO 23 STEP 13
7905 LK=11:LOCATE RC,4:PRINT STRING$(75,0)
7910 IF FACTOR=1 THEN LOCATE RC,6:PRINT USING "###                                                                  ###";SG,EG:LK=12
7915 FOR IXX=SG+INC TO EG-INC*(2-FACTOR) STEP INC*FACTOR
7920 LOCATE RC,LK:PRINT USING FMT$;IXX
7925 LK=LK+7.5*FACTOR:NEXT:NEXT:RETURN
7927 '
7929 '
7950 DIFF=ABS(EG-SG):INC=100:IF DIFF<675 THEN INC=75
7955 IF DIFF<450 THEN INC=50:IF DIFF<225 THEN INC=25
7960 IF DIFF<180 THEN INC=20:IF DIFF<135 THEN INC=15
7965 IF DIFF<90  THEN INC=10:IF DIFF<45  THEN INC=5
7970 IF DIFF<18  THEN INC=2 :IF DIFF<9   THEN INC=1
7975 IF DIFF<4.5 THEN INC=.5:IF DIFF<1.8 THEN INC=.2
7980 IF DIFF<.9  THEN INC=.1:IF DIFF<.45 THEN INC=.05
7985 SG=(FIX(SG/INC)-1)*INC:IF SG<0 THEN SG=0
7990 EG=SG+9*INC:FMT$="###":FACTOR=1
7995 IF INC<1 THEN FMT$="###.##":FACTOR=2
7996 RETURN
7998 '
7999 '
8000 LPRINT"TRAIN INFORMATION":LPRINT:LPRINT
8010 LPRINT"NUMBER OF UNITS";NS1 TAB(40)
8020 LPRINT"NUMBER OF CARS";SN:LPRINT"TOTAL HORSEPOWER";HP TAB(40)
8030 LPRINT"TRAIN LENGTH (FT)";LT:LPRINT"TOTAL UNIT WEIGHT (TONS)";LWT TAB(40)
8040 LPRINT"TRAIN GROSS WEIGHT (TONS)";WT:LPRINT"ASSUMED ADHESION";F TAB(40)
8050 LPRINT"TRAIN LIGHTWEIGHT (TONS)";LTWT:LPRINT TAB(40) "TRAIN PAYLOAD (TONS)";WT-LTWT
8060 LPRINT:LPRINT:LPRINT "HP/TON GROSS WEIGHT";:LPRINT USING "##.#";HP/WT;
8070 LPRINT TAB(40) "DIESEL FUEL CONSUMED (GALS)";:LPRINT USING "  ####";WK(N)*3.248065E-02
8080 LPRINT "HP/TON PAYLOAD";:IF WT=LTWT THEN LPRINT "    EMPTY TRAIN";
8087 IF WT><LTWT THEN LPRINT USING "##.#";HP/(WT-LTWT);
8090 LPRINT TAB(40) "TON MILES/GALLON (GROSS WEIGHT)";:LPRINT USING "  ###.#";WT*MP(N)/(WK(N)*.0324)
8100 LPRINT TAB(40) "TON MILES/GALLON (PAYLOAD)";:IF WT=LTWT THEN LPRINT "  EMPTY TRAIN"
8105 IF WT><LTWT THEN LPRINT USING "  ###.#";(WT-LTWT)*MP(N)/(WK(N)*.0324)
8107 LPRINT:LPRINT:LPRINT
8110 LPRINT"       MILEPOST" TAB(19) "GRADE" TAB(30) "DEGREE" TAB(40)
8120 LPRINT"SPEED" TAB(50) "TRAIN" TAB(62) "TRAIN" TAB(73) "TRACTIVE" TAB(85)
8130 LPRINT"BRAKING" TAB(95) "ACCELERATION"  TAB(111) "TIME" TAB(122) "WORK"
8140 LPRINT TAB(20) "(%)" TAB(31) "CURVE" TAB(40) "LIMIT" TAB(50)
8150 LPRINT"SPEED" TAB(59) "RESISTANCE" TAB(74) "EFFORT" TAB(86)
8160 LPRINT"FORCE" TAB(98) "(MPH/S)" TAB(109) "(H:M:S)" TAB(120) "(FT LBS)"
8170 LPRINT TAB(40)"(MPH)" TAB(50) "(MPH)" TAB(62) "(LBS)" TAB(75) "(LBS)" TAB(86) "(LBS)" TAB(120) "(X 1E6)":LPRINT:RETURN
8190 '
8200 '
9000 GOSUB 9500:LPRINT USING "      ###.##";MP(R);
9010 LPRINT USING "      +#.##";G(R);
9020 LPRINT USING "      ##.##";DC(R);
9030 LPRINT USING "      ###";MS(R);
9040 LPRINT USING "      ###.#";TS(R);
9050 LPRINT USING "      +######";TR(R);
9060 LPRINT USING "      ######";TE(R);
9070 LPRINT USING "     #######";BF(R);
9080 LPRINT USING "       +#.##";AC(R);
9090 LPRINT "   ";T$;
9100 LPRINT USING "      ######";WK(R):RETURN
9110 '
9120 '
9500 HR=T(R)/3600:IHR=INT(HR)
9510 MN=(HR-IHR)*60:IMN=INT(MN)
9520 SEC=(MN-IMN)*60:ISEC=INT(SEC)
9531 T1$=RIGHT$(STR$(IHR)+":",3)
9532 T2$=RIGHT$("0"+RIGHT$(STR$(IMN),LEN(STR$(IMN))-1)+":",3)
9533 T3$=RIGHT$("0"+RIGHT$(STR$(ISEC),LEN(STR$(ISEC))-1),2)
9544 T$=T1$+T2$+T3$:RETURN
9545 '
10000 GOSUB 1900:OQ=1
10010 TP=INT((I-1)/12)+1:R1=TP*12-11:TNL=I:NL=12
10020 GOSUB 5500:CLS:GOSUB 15000
10030 LOCATE 20,20:PRINT "(E)DIT      PG UP      PG DN       (Q)UIT":LOCATE 7,1
10040 FOR R=OQ TO Q:GOSUB 15100:NEXT R
10050 INPT$=INKEY$:IF INPT$="" THEN 10050
10055 IF INPT$="Q" OR INPT$="q" THEN RETURN
10060 IF INPT$="E" OR INPT$="e" THEN 10110
10065 IF LEN(INPT$)=2 THEN INPT$=RIGHT$(INPT$,1) ELSE 10050
10070 IF ASC(INPT$)=81 THEN OQ=OQ+12
```

```
10080 IF ASC(INPT$)=73 THEN OQ=OQ-12
10100 GOTO 10020
10110 '
10120 COLOR 0:LOCATE 20,20:PRINT"                                                  ":COLOR 15
10130 LOCATE 20,6:PRINT "1.EDIT" TAB(19) "2. DELETE" TAB(34) "3. INSERT" TAB(47) "4. APPEND" TAB(64) "5. QUIT"
10140 LOCATE 22,30:INPUT "ENTER CHOICE (1-5)";A:COLOR 7
10150 IF A<1 OR A>5 THEN BEEP:GOTO 10140
10160 IF A=5 THEN 10020
10170 ON A GOSUB 11000,12000,13000,14000
10180 GOTO 10010
10200 '
11000 COLOR 15:LOCATE 23,32:INPUT "EDIT LINE NUMBER";A1:COLOR 7
11005 IF A1>I OR A1<1 THEN BEEP:GOTO 11000
11010 CLS:GOSUB 15000:PRINT:PRINT
11015 RR=A1-2:IF RR<1 THEN RR=1
11020 FOR R=RR TO A1:GOSUB 15100:NEXT R
11030 PRINT:PRINT:COLOR 15:PRINT"EDIT LINE";A1:PRINT:PRINT:COLOR 7
11040 GOSUB 13500:GOSUB 13700
11140 PRINT:PRINT:PRINT TAB(31):INPUT "DATA CHANGES OK (Y/N)";A1$
11150 IF A1$="N" THEN 11010
11160 A$=A$(F1):X=1:GOSUB 1670:RETURN
11170 '
11180 '
12000 COLOR 15:LOCATE 23,30:INPUT "DELETE LINE NUMBER";A1
12010 IF A1<1 OR A1>I THEN BEEP:GOTO 12000
12020 PRINT TAB(25) "LINE";A1;:INPUT "TO BE DELETED (Y/N)";A1$:COLOR 7
12030 IF A1$="N" THEN RETURN
12040 I=I-1:FOR R=A1 TO I
12050 TRK(R,1)=TRK(R+1,1):TRK(R,6)=TRK(R+1,6):TRK(R,3)=TRK(R+1,3)
12060 TRK(R,4)=TRK(R+1,4):TRK(R,5)=TRK(R+1,5):TRK(R,2)=TRK(R+1,2):NEXT R
12070 GOSUB 13700:A$=A$(F1):X=1:GOSUB 1670:RETURN
12080 '
12090 '
13000 COLOR 15:LOCATE 23,30:INPUT "INSERT AFTER LINE NUMBER";A1:COLOR 7
13010 IF A1>I OR A1<1 THEN BEEP:GOTO 13000
13020 I=I+1:FOR R=I TO A1+2 STEP -1
13030 TRK(R,1)=TRK(R-1,1):TRK(R,6)=TRK(R-1,6):TRK(R,3)=TRK(R-1,3)
13040 TRK(R,4)=TRK(R-1,4):TRK(R,5)=TRK(R-1,5):TRK(R,2)=TRK(R-1,2):NEXT R
13050 CLS:GOSUB 15000:PRINT:PRINT
13060 RR=A1-2:IF RR<1 THEN RR=1
13070 FOR R=RR TO A1:GOSUB 15100:NEXT R
13080 PRINT:PRINT:COLOR 15:PRINT TAB(30) "INSERT AFTER LINE NUMBER";A1:COLOR 7:PRINT:PRINT:A1=A1+1
13090 GOSUB 13500:GOSUB 13700:A$=A$(F1):X=I:GOSUB 1670:RETURN
13100 '
13110 '
13500 INPUT "MILEPOST";TRIAL
13510 IF A1=I THEN 13530:IF TRIAL>=TRK(A1+1,1) THEN BEEP:GOTO 13500
13520 IF A1=1 THEN 13540
13530 IF TRIAL<=TRK(A1-1,1) THEN BEEP:GOTO 13500
13540 TRK(A1,1)=TRIAL:INPUT "GRADE (%)";TRK(A1,3)
13550 INPUT "DEGREE OF CURVE";TRK(A1,4)
13560 INPUT "SPEED LIMIT";TRK(A1,2):TRK(A1,5)=TRK(A1,3)+TRK(A1,4)*.04:RETURN
13570 '
13580 '
13700 FOR R=A1 TO I
13710 TRK(R,6)=TRK(R-1,6)+(TRK(R,1)-TRK(R-1,1))*5280*SIN(ATN(TRK(R-1,3)/100))
13720 NEXT R:RETURN
13730 '
13780 '
14000 SP(1,1)=TRK(I,1):CV(1,1)=TRK(I,1):GR(1,1)=TRK(I,1)
14010 X=I-1:A$=A$(F1):GOSUB 1190:RETURN
14050 '
14060 '
15000 PRINT "FILENAME:  ";A$(NY3);:PRINT TAB(65) "PAGE";PN;:PRINT"OF";TP:PRINT
15010 PRINT TAB(7) "MILEPOST" TAB(19) "ELEVATION" TAB(32) "% GRADE";
15020 PRINT TAB(44) "DEGREE" TAB(56) "% GRADE" TAB(70) "SPEED"
15030 PRINT TAB(22) "(FT)" TAB(45) "CURVE" TAB(55) "EQUIVALENT" TAB(70) "LIMIT"
15040 PRINT TAB(70) "(MPH)":RETURN
15100 PRINT USING "####";R;
15110 PRINT USING "  ####.##";TRK(R,1);
15120 PRINT USING "      +#####";TRK(R,6);
15130 PRINT USING "       +#.##";TRK(R,3);
15140 PRINT USING "       ##.##";TRK(R,4);
15150 PRINT USING "       +##.##";TRK(R,5);
15160 PRINT USING "       ###";TRK(R,2):RETURN
15170 '
16000 COEFF$(1)="F":COEFF(2)=0:COEFF(3)=65:COEFF(4)=18.5:COEFF(5)=83:COEFF(6)=0
16010 COEFF$(1)="F":COEFF(2)=0:COEFF(3)=65:COEFF(4)=18.5:COEFF(5)=83:COEFF(6)=0
16020 COEFF(7)=1.5:COEFF(8)=18:COEFF(9)=.03:COEFF(10)=.05
16030 COEFF(11)=1.5:COEFF(12)=18:COEFF(13)=.03:COEFF(14)=.102
16040 COEFF(15)=1.5:COEFF(16)=18:COEFF(17)=.03:COEFF(18)=.066
16050 COEFF(21)=25:COEFF(22)=50:COEFF(31)=3:RETURN
16060 '
17000 CAR(1,1)=55:CAR(1,2)=62100!:CAR(1,3)=154000!
17010 CAR(2,1)=64:CAR(2,2)=63300!:CAR(2,3)=199000!
17030 CAR(3,1)=54:CAR(3,2)=71000!:CAR(3,3)=192000!
17040 CAR(4,1)=94:CAR(4,2)=68500!:CAR(4,3)=149000!
17050 CAR(5,1)=49:CAR(5,2)=59900!:CAR(5,3)=200000!:RETURN
17090 '
18000 LOCO(1,1)=3000:LOCO(1,2)=197:LOCO(4,3)=6
18010 LOCO(2,1)=2020:LOCO(2,2)=131:LOCO(2,3)=4
18020 LOCO(3,1)=1750:LOCO(3,2)=130:LOCO(3,3)=4
18030 LOCO(4,1)=3312:LOCO(4,2)=196:LOCO(4,3)=6
18035 LOCO(5,1)=3040:LOCO(5,2)=196:LOCO(5,3)=6
18040 LOCO(6,1)=2400:LOCO(6,2)=129:LOCO(6,3)=4:RETURN
19000 IF NOT(ERR=24 OR ERR=27) THEN RESUME 19040
19010 CLS:LOCATE 10,31:PRINT "Printer not ready"
19020 LOCATE 14,20:PRINT "Put printer on line, then press any key"
19030 IF INKEY$="" THEN 19030 ELSE RESUME 5210
19040 CLS:PRINT "Error Number";ERR;
19045 PRINT "  at line number";ERL
19050 PRINT:PRINT:PRINT "Number of records read from file #2=";Z
19060 INPUT Q$:CLOSE:RETURN
```