TARADCOM

R and D LABORATORY

TECHNICAL REPORT

NO. 12503

U.S. ARMY TARADCOM

NATO REFERENCE MOBILITY MODEL, EDITION I

USERS GUIDE
VOLUME II

CONTRACT NO. DAAK30-77-C-0027
OCT 1979

by PETER W. HALEY
TARADCOM
M. PETER JURKAT
PETER M. BRADY, JR.
STEVENS INSTITUTE OF TECHNOLOGY
DAVIDSON LABORATORY
HOBOKEN, NJ 07030

U.S. ARMY TANK-AUTOMOTIVE
RESEARCH AND DEVELOPMENT COMMAND
Warren, Michigan 48090

Technical Report 12503

NATO REFERENCE MOBILITY MODEL, EDITION I
USERS GUIDE

VOLUME II

OBSTACLE MODULE

DA Project 1L162601AH91

Prepared by

Stevens Institute of Technology
Davidson Laboratory
Castle Point Station
Hoboken, NJ 07030

Contract DAAK30-77-C-0027

OCT 1979

for

US Army Tank-Automotive Research & Development Command
Warren, Michigan 48090

ABSTRACT

Instructions in the organization and use of the computer programs which implement the Initial NATO Reference Mobility Model (INRMM) are presented. Volume II is devoted to the INRMM Obstacle-Crossing Module. A brief description of the mathematical equations and computing algorithms which predict the speed of a vehicle over a variety of terrain, the input data required, and the outputs generated is included. Some aid to the interpretation of various output variables is given.

KEY WORDS

Mobility
Mobility Modeling
Computerized Simulation
Vehicle Performance
Terrain
Obstacle Crossing

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

I INTRODUCTION AND OVERVIEW*

The NATO Reference Mobility Model (NRMM) is a collection of equations and algorithms designed to simulate the cross-country movement of vehicles. It was developed from several predecessor models, principally AMC-74 (Jurkat, Nuttall and Haley (1975)). This report, in several volumes, provides some background and motivation for most aspects of the Model, and presents documentation for the coded version now available through the U. S. Army Tank-Automotive Research and Development Command (TARADCOM).

A. Background

Rational design and selection of military ground vehicles requires objective evaluation of an ever-increasing number of vehicle system options. Technology, threat, operational requirements, and cost constraints change with time. Current postures must be reexamined, new options evaluated, and new trade-offs and decisions made. In the single area of combat vehicles, for example, changes in one or another influencing factor might require trade-offs that run the gamut from opting for an air or ground system, through choosing wheels, tracks or air cushions, to designating a new tire.

The former Mobility Systems Laboratory of the then U. S. Army Tank-Automotive Command (TACOM) and the U. S. Army Engineer Waterways Experiment Station (WES) are the Army agencies responsible for

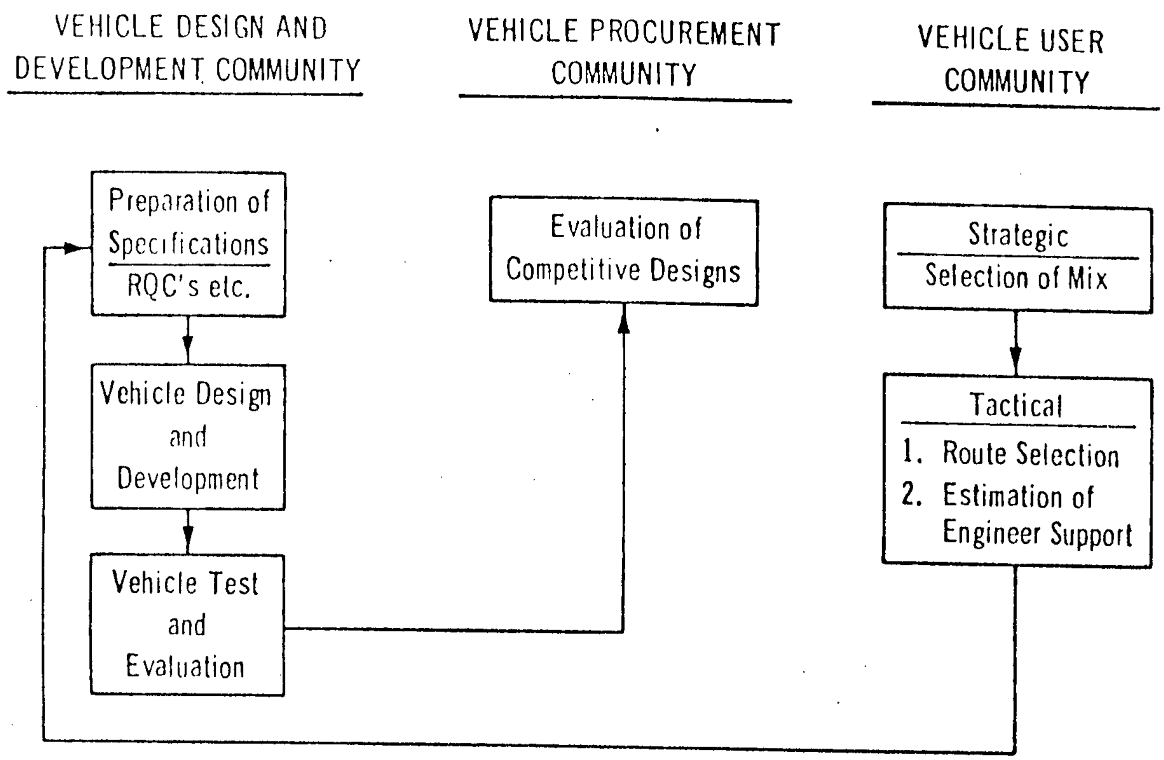
* This chapter is adapted from Jurkat, Nuttall and Haley (1975).

conducting ground mobility research. In 1971, a unified U. S. ground mobility program, under the direction of the then Army Materiel Command (AMC), was implemented that specifically geared the capabilities of both laboratories to achieve common goals.

As a first step in the unified program, a detailed review was made of existing vehicle mobility technology and of the problems and requirements of the various engineering practitioners associated with the military vehicle life cycle. One basic requirement was identified as common to all practitioners surveyed: the need for an objective analytical procedure for quantitatively assessing the performance of a vehicle in a specified operational environment. This is the need that is addressed to a substantial extent by the INRMM and its predecessors.

In theory, a single methodology can serve some of the needs of all major practitioners, provided it relates vehicle performance to basic characteristics of the vehicle-driver-terrain system at appropriate levels of detail.

Three principal categories of potential users of the methodology were identified: the vehicle development community, the vehicle procurement community, and the vehicle user community (Figure I.A.1). The greatest level of detail is needed by the design and development engineer (vehicle design and development community) who is interested in subtle engineering details--for example, wheel geometry, sprung masses, spring rates, track widths, etc.--and their



PROSPECTIVE USERS OF VEHICLE PERFORMANCE PREDICTION METHODOLOGY

FIGURE I-A-1

interactions with soil strength, tree stems of various sizes and spacings, approach angles in ditches and streams, etc. At the other end of the spectrum is the strategic planner (user community), who is interested in such highly aggregated characteristics as the average cross-country speed of a given vehicle throughout a specified region--the net result of many interactions of the engineering details with features of the total operational environment. Between these two extremes, is the person responsible for selection of the vehicles who must evaluate the effect of changes of major subsystems or choose from concepts of early design stages. To be responsive to the needs of all three user communities, the methodology must be flexible enough to provide compatible results at many levels and in an appropriate variety of formats.

Interest in a single, unified methodology applicable to the needs of these three principal users led to the creation of a cross-country vehicle computer simulation combining the best available knowledge and models of the day. Much of this knowledge was collected in Rula and Nuttall (1971). The first realization of the simulation was a series of computer programs known as the AMC-71 Mobility Model, called AMC-71 for short (US ATAC(1973)). This model first became operational in 1971; it was published in 1973. It was conceived as the first generation of a family whose descendants, under the evolutionary pressures of subsequent research and validation testing results, application experiences, and growing user requirements, would be characterized by greater accuracy and applicability. A relatively current status report may be found in Nuttall, Rula and Dugoff (1974).

The first descendant, known as AMC-74, is the basis for the INRMM. It is documented in Jurkat, Nuttall and Haley (1975). The following is a description of this model.

B. Modeling Off-Road Vehicle Mobility

In undertaking mobility modeling, the first question to be answered was the seemingly easy one: What is mobility? The answer had been elusive for many years. Semantic reasons can be traced to the beginnings of mobility research, but there was also a pervasive reluctance to accept the simple fact that even intuitive notions about a vehicle's mobility depend greatly on the conditions under which it is operating. By the mid-1960s, however, a consensus had emerged that the maximum feasible speed-made-good* by a vehicle between two points in a given terrain was a suitable measure of its intrinsic mobility in that situation.

This definition not only identified the engineering measure of mobility, but also its dependence on both terrain and mission. When, at a suitably high resolution, the terrain involved presents the identical set of impediments to vehicle travel throughout its extent, mobility in that terrain (ignoring edge effects) is the vehicle's maximum straight-line speed as limited only by those impediments. But when, as is typically the case, the terrain is not so homogeneous, the problem immediately becomes more complex. Maximum speed-made-good then becomes an interactive function of terrain variations, end points specified, and the path selected. (Note that the last two constitute at least part of a detailed mission statement.) As a way to achieve a useful simulation in this complicated situation the INRMM deliberately

*Speed-made-good between two points is the straight-line distance between the points divided by total travel time, irrespective of path.

simplifies the real areal terrain into a mosaic of terrain units within each of which the terrain characteristics are considered sufficiently uniform to permit use of the simple, maximum straight-line speed of the vehicle to define its mobility in, along, or across that terrain unit. A terrain unit or segment specified for a road or trail is, similarly, considered to have uniform characteristics throughout its extent.

Maximum speed predictions are made for each terrain unit without concern for whether or not distances within the unit are adequate to permit the vehicle to reach the predicted maximum. This vehicle and terrain-specific speed prediction is the basic output of the model. The model, in addition, generates data that may be used to predict operational vibration levels, mission fuel consumption, etc., and can provide diagnostic information as to the factors limiting speed performance in the terrain unit.

The speed and other performance predictions for all terrain units in an area can be incorporated into maps that specify feasible levels of performance that a given vehicle might achieve at all points in the area. At this point, the output is reasonably general and is essentially independent of mission and operational scenario influences. The basic data constituting the maps must usually be further processed to meet the needs of specific users. These needs vary from relatively simple statistics or indices reflecting overall vehicle compatibility with the terrain, to extensive analyses involving detailed or generalized missions. None of these so called post-processors is included as part of the INRMM.

C. Overall Structure of the INRMM

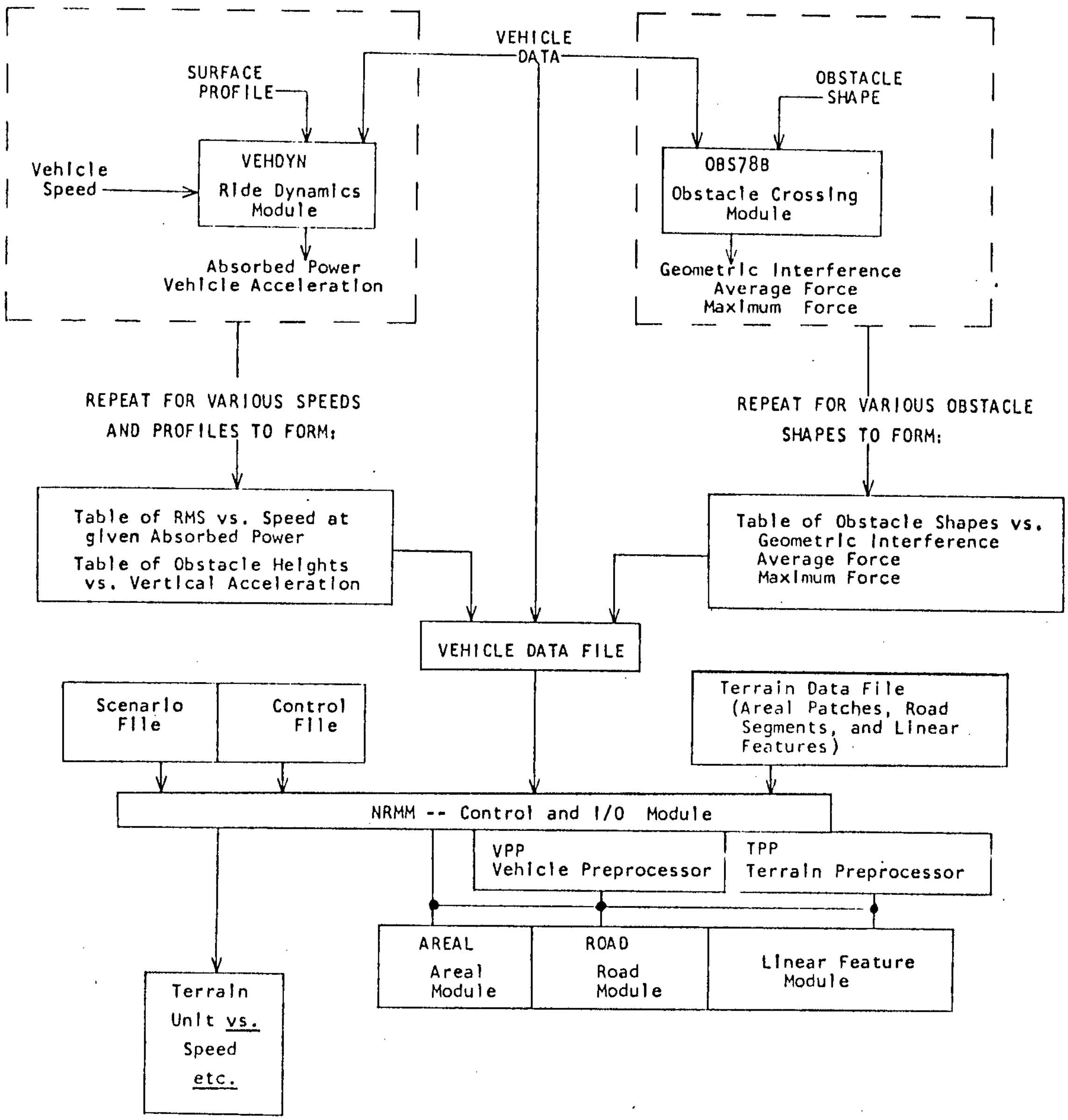
In formulating AMC-71, it was recognized that its ultimate usefulness to decision makers in the vehicle development, procurement, and user communities would depend upon its realism and credibility. (See Nuttall and Dugoff (1973).) These perceived requirements led to several more concrete objectives related to the overall structure of the model. It was determined that the model should be designed to:

1. Allow validation by parts and as a whole.
2. Make a clear distinction between engineering predictions and any whose outcome depends significantly upon human judgment, with the latter kept visible and accessible to the model user.
3. Be updated readily in response to new vehicle and vehicle-terrain technology.
4. Use measured subsystem performance data in place of analytical predictions when and as available and desired.

These objectives, plus the primary goal of supporting decision making relating to vehicle performance at the several levels, clearly dictated a highly modular structure that could both provide and accept data at the subsystem level, as well as make predictions for the vehicle as a whole. The resulting gross structure of the model is illustrated in Figure I.C.1.

At the heart of the model are three independent computational modules, each comprised of analytical relations derived from laboratory and field research, suitably coupled in the particular type of operation. These are:

FIGURE I.C.1 - GENERAL ORGANIZATION OF THE INITIAL NATO REFERENCE MOBILITY MODEL



1. The Areal Module, which computes the maximum feasible speed for a single vehicle in a single areal terrain unit (patch).

2. The Linear Feature Module, which computes the minimum feasible time for a single vehicle, aided or unaided, to cross a uniform segment of a significant linear terrain

feature such as a stream, ditch, or embankment (not currently available).

3. The Road Module, which computes the maximum feasible speed of a single vehicle traveling along a uniform segment of a road or trail.

These Modules and the Terrain and Vehicle Preprocessors are collected in a computer program called NRMM and are described in Volume I.

These three Modules may be used separately or together. Alternately, INRMM has the ability to simulate travel from terrain unit to terrain unit in the sequence given by the terrain input file. In this mode, known as the traverse mode, sufficient output data can be provided so that the user may calculate acceleration and deceleration times and distances between and across terrain unit boundaries, and thereby determine actual travel time and speed-made-good over a chosen route.

All three modules draw from a common data base that describes quantitatively the vehicle, the driver, and the terrain to be examined in the simulation. The general content of the data base is shown in Table I.C.1.

TABLE I.C.1

Terrain, Vehicle, Driver Attributes Characterized in INRMM Data Base

Terrain	Vehicle	Driver
Surface Composition Type Strength	Geometric characteristics	Reaction Times
		Recognition distance
Surface Geometry Slope Altitude Discrete Obstacles Roughness Road Curvature Road Width Road Superelevation	Inertial characteristics Mechanical characteristics	Acceleration and impact tolerances Minimum acceptable speeds
Vegetation Stem Size Stem Spacing		
Linear Geometry Stream cross section Water velocity Water depth		

D. Model Inputs and Preprocessors

1. Terrain

For the purposes of the model, each terrain unit is described at any given time by values for a series of 22 mathematically independent terrain factors for an areal unit (including lake and marsh factors), 10 for the cross section of a linear feature to be negotiated, and 9 to quantify a road segment. General-purpose terrain data also include separate values for several terrain factor values that vary during the year. For example, at present such general data for areal terrain include four values for soil strength (dry, average, wet, and wet-wet seasons) and four seasonal values for recognition distances in vegetated areas. Similar variations in effective ground roughness, resulting from seasonal changes in soil moisture (including freezing) and in the cultivation of farm land, can be envisioned for the future. Further details on the terrain factors used are given in Rula and Nuttall (1975).

As discussed earlier, the basic approach to representing a complex terrain is to subdivide it into areal patches, linear feature segments, or road segments, each of which can be considered to be uniform within its bounds. Besides supplying actual values for the terrain factors, this concept may be implemented by dividing the range of each individual terrain factor value into a number of class intervals, based upon considerations of vehicle response sensitivity and practical measurement and mapping resolution problems. A patch or a segment is then defined by the condition that the class interval designator for each factor involved is the same throughout. A new patch or segment is defined whenever one or more factors fall into a new class interval.

Before being used in the three computational Modules, the basic terrain data are passed through a Terrain Data Preprocessor, called TPP in the Computer Program NRMM. This preprocessor does three things:

1. Converts as necessary all data from the units in which they are stored to inches, pounds, seconds and radians, which are used throughout the subsequent performance calculations.
2. Selects prestored soil strengths and visibility distances according to run specifications, which are supplied as part of the scenario data (see below).
3. Calculates from the terrain measurements in the basic terrain data a small number of mathematically dependent terrain variables used repeatedly in the computational modules.

2. Vehicle

The vehicle is specified in the vehicle data base in terms of its basic geometric, inertial, and mechanical characteristics. The complete vehicle characterization as used by the performance computation modules includes measures of dynamic response to ground roughness and obstacle impact, and the clearance and traction requirements of the vehicle while it is negotiating a parametric series of discrete obstacles.

The model structure permits use at these points of appropriate data derived either from experiments or from supporting stand-alone simulations used as preprocessors. One supporting two-dimensional ride and obstacle crossing Dynamics Module for obtaining requisite dynamics responses(currently called VEHDYN and described in Volume III) and a second supporting Module for computing obstacle crossing traction requirements and interferences (currently called OBS78B and described in this Volume) are available as elements of the INRMM. Both derive some required information from the basic vehicle data base, and both, when used, constitute stand-alone vehicle data preprocessors.

There is also a Vehicle Data Preprocessor called VPP (integral to NRMM) which, like the Terrain Data Preprocessor, has three functions:

1. Conversion of vehicle input data to uniform inches, pounds, seconds, and radians.
2. Calculation, from the input data, of controlling soil performance parameters and other simpler dependent vehicle variables subsequently used by the computational modules, but usually not readily measured on a vehicle or available in its engineering specifications.
3. Computation of the basic steady-state traction versus speed characteristics of the vehicle power train, from engine and power train characteristics.

As in the case of dynamic responses and obstacle capabilities, the last item, the steady-state tractive force-speed relation, may be input directly from proving ground data, when available and desired.

3. Driver

The driver attributes used in the model characterize the driver in terms of his limiting tolerance to shock and vibration and his ability to perceive and react to visual stimuli affecting his behaviour as a vehicle controller. While these attributes are identified in Figure I.C.1 and Table I.C.1 as part of the data base INRMM provides for their specific identification and user control so that the effects of various levels of driver motivation, associated with combat or tactical missions, for example, can be considered.

4. Scenario

Several optional features are available to the user of the INRMM (weather, presumed driver motivation, operational variations in tire inflation pressure) which allow the user to match the model predictions to features or assumptions of the full operational scenario for which predictions are required. Model instructions which select and control these options are referred to as scenario inputs.

The scenario options include the specification of:

1. Season, which, when seasonal differences in soil strength constitute a part of the terrain data, allows selection of the soil strength according to the variations in soil moisture with seasonal rainfall, and
2. Weather, which affects soil slipperiness and driving visibility, (including dry snow over frozen ground and associated conditions).
3. Several levels of operational influences on driver tolerances to ride vibrations and shock, and on driver strategy in

negotiating vegetation and using brakes.

4. Reasonable play of tire pressure variations to suit the mode of operation--on-road, cross-country, and in sand.

E. Stand-Alone Simulation Modules

As indicated above, the Model is implemented by a series of independent Modules. The Terrain and Vehicle Preprocessors, already described, form two of these. Two further major stand-alone simulation Modules will now be outlined.

1. Obstacle-crossing Module-OBS78B

This Module determines interferences and traction requirements when vehicles are crossing the kind of minor ditches and mounds characterized as part of the areal terrain; it is described fully in this Volume. It is used as a stand-alone Preprocessor Module to the Areal Module of INRMM.

The Obstacle-crossing Module simulates the inclination and position, interferences, and traction requirements of a two-dimensional (vertical center-line plane) vehicle crossing a single obstacle in a trapezoidal shape as a mound or a ditch. The module determines a series of static equilibrium positions of the vehicle as it progresses across the obstacle profile. Extent of interference is determined by comparison of the obstacle profile and the displaced vehicle bottom profile. Traction demand at each position is determined by the forces on driven running gear elements, tangential to the obstacle surface, required to maintain the vehicle's static position. Pitch compliance of suspension elements is not accounted for but frame articulation (as at pitch joints, trailer hitches, etc) is permitted.

The Obstacle-crossing Module produces a table of minimum clearances (or maximum interferences) and average and maximum force required to cross a representative sample of obstacles defined by combinations of obstacle dimensions varied over the ranges appropriate for features included in the areal terrain description. This simulation is done only once for each vehicle. Included in the INRMM Areal Module is a three-dimensional linear interpolation routine which, for any given set of obstacle parameters, approximates from the derived table the corresponding vehicle clearance (or interference) and associated traction requrements. Obviously, the more entries there are in the table, the more precise will be the determination.

2. Ride Dynamics Module- VEHDYN

The Areal Module examines as possible vehicle speed limits in a given terrain situation two limits which are functions of vehicle dynamic perceptions: speed as limited by the driver's tolerance to his vibrational environment when the vehicle is operating over continuously rough ground, and speed as limited by the driver's tolerance to impact received while the vehicle is crossing discrete obstacles. It is assumed that the driver will adjust his speed to ensure that his tolerance levels will not be exceeded.

The Ride Dynamics Module of INRMM, called VEHDYN and described in Volume III, computes accelerations and motions at the driver's station (and other locations, if desired) while the vehicle is operating at a given speed over a specific terrain profile. The profile may be continuously, randomly rough, may consist solely of a single discrete obstacle, uniformly spaced obstacles of a specific height or may be anything in between. From the computed motions, associated with driver modeling and specified tolerance criteria, simple relations are developed for a given vehicle between relevant terrain measurements and maximum tolerable speed. The terrain measurement to which ride speed is related is the root mean square (rms) elevation of the ground profile (with terrain slopes and long-wavelength components removed). The terrain descriptors for obstacles are obstacle height and obstacle spacing.

The terrain parameters involved, rms elevation and obstacle height and spacing, are factors quantified in each patch description, and rms elevation is specified for each road segment. Preprocessing of the vehicle data in the ride dynamics module provides an expedient means of predicting dynamics-based speed in the patch and road segment modules via a simple, rapid table-lookup process.

The currently implemented Ride Dynamics Module is a digital simulation that treats vehicle motions in the vertical center-line plane only (two dimensions). It is a generalized model that will handle any rigid-frame vehicle on tracks and/or tires, with any suspension. Tires are modeled using a segmented wheel representation, (see Lessem (1968)) and a variation of this representation is used to introduce first-order coupling of the road wheels on a tracked vehicle by its tracks.

a) Driver model and tolerance criteria.

It has been shown empirically that, in the continuous roughness situation, driver tolerance is a function of the vibrational power being absorbed by the body. (See Pradko, Lee and Kaluza (1966).) The same work showed that the tolerance limit for representative young American males is approximately 6 watts of continuously absorbed power, and the research resulted in a relatively simple model for power absorption by the body. The body power absorption model, based upon shaping filters applied to the decomposed acceleration spectrum at the driver's station, is an integral part of the INRMM two-dimensional dynamics simulation.

In the past, only the 6 watt criterion was used to determine a given vehicle's speed as limited by rms roughness. More recent measurements in the field have shown that with sufficient motivation young military drivers will tolerate more than 6 watts for periods of many minutes. Accordingly, INRMM will accept as vehicle data a series of ride speed versus rms elevation relations, each corresponding to a different absorbed power level, and will use these to select ride-speed limits according to the operationally related level called for by the scenario. The Ride Dynamics Module will, of course, produce the required additional data, but some increased running time is involved.

The criterion limiting the speed of a vehicle crossing a single discrete obstacle, or a series of closely, regularly spaced obstacles, is a peak acceleration at the driver's seat of 2.5-g passing a 30-Hz. filter. Data relating the 2.5-g speed limit to obstacle height and spacing can be developed in the ride dynamics module by inputting appropriate obstacle profiles.

INRMM requires two obstacle impact relations: the first, speed versus obstacle height for a single obstacle (spacing very great); and the second, speed versus regular obstacle spacing for that single obstacle height (from the single obstacle relation) which limits vehicle speed to a maximum of 15 mph. For obstacles spaced at greater than two vehicle lengths, the single-obstacle speed versus obstacle height relation is used. For closer spacings, the least speed allowable by either relation is selected.

3. Main Computational Modules - NRMM

The highly iterative computations required to predict vehicle performance in each of the many terrain units needed to describe even limited geographic areas are carried out in the three main computational modules. Each of these involve only direct arithmetic algorithms which are rapidly processed in modern computers. In INRMM, even the integrations required to compute acceleration and deceleration between obstacles within an areal patch are expressed in closed, algebraic form.

Terrain input data include a flag, which signifies to the model whether the data describes an areal patch, a linear feature segment, or a road segment. This flag calls up the appropriate computational Module.

a) Areal Terrain Unit Module

This Module calculates the maximum average speed a vehicle could achieve and maintain while crossing an areal terrain unit. The speed is limited by one or a combination of the following factors:

1. Traction available to overcome the combined resistances of soil, slope, obstacles, and vegetation.
2. Driver discomfort in negotiating rough terrain (ride comfort) and his tolerance to vegetation and obstacle impacts.
3. Driver reluctance to proceed faster than the speed at which the vehicle could decelerate to a stop within the, possibly limited, visibility distance prevailing in the areal unit (braking-visibility limit).
4. Maneuvering to avoid trees and/or obstacles.
5. Acceleration and deceleration between obstacles if they are to be overriden.
6. Damage to tires.

Figure I.E.1 shows a general flow chart of how the calculations of the Areal Module are organized.

After determination of some vehicle and terrain - dependent factors used repetitively in the patch computation (1),* the Module is entered with the relation between vehicle steady-state speed and theoretical tractive force and with the minimum soil strength that the vehicle requires to maintain headway on level, weak soils. These data

* Numbers in parentheses correspond to numbers in Figure I.E.1.

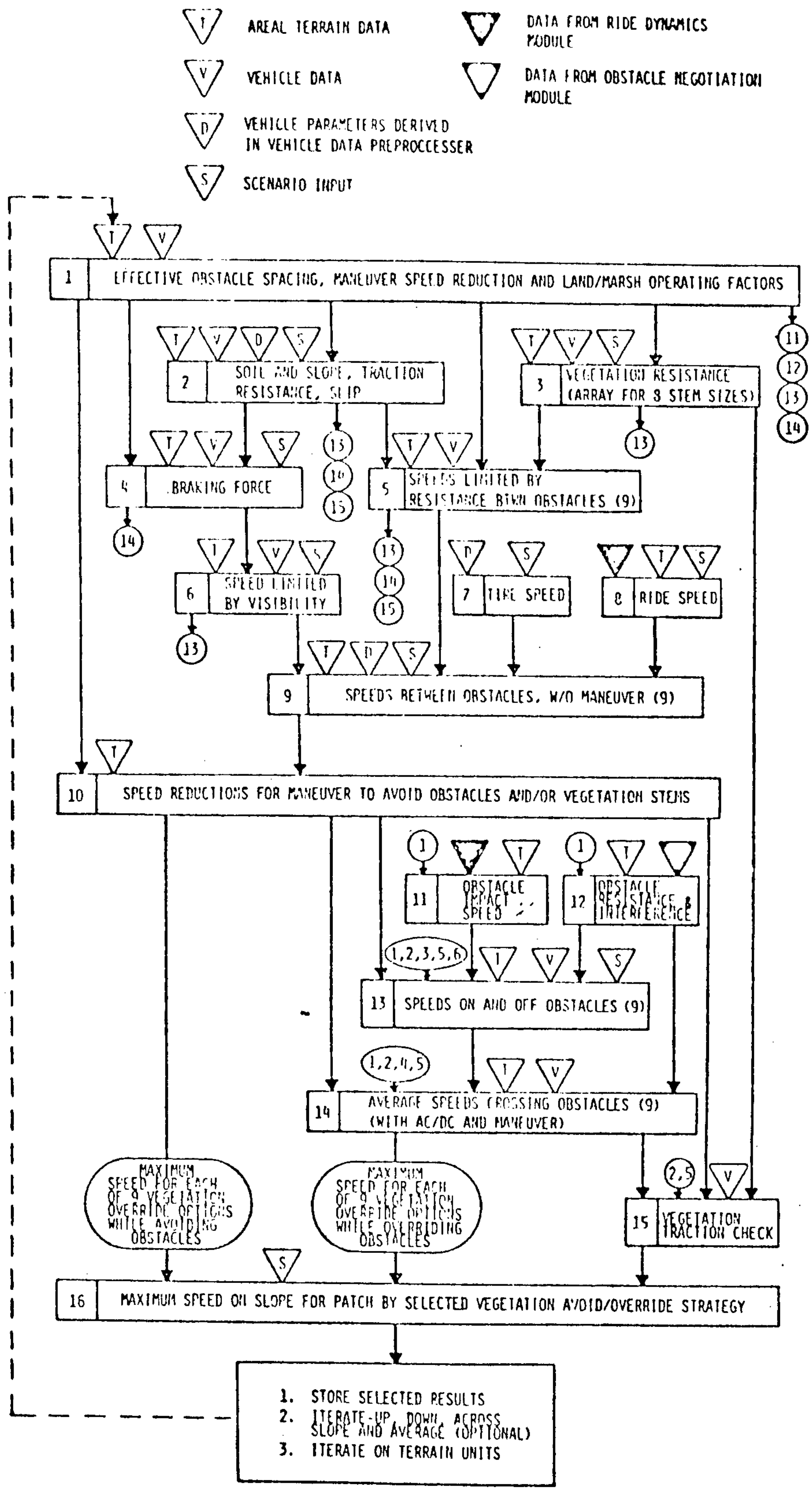


FIGURE I.E.1 -- GENERAL FLOW OF INRMM AREAL MODULE are provided by the vehicle data preprocessor. Soil and slope resistances (2) and braking force limits (4) are computed, and the basic tractive force-speed relation is modified to account for soil-limited traction, soil and slope resistances, and resulting tire or track slip. Forces required to override prevailing tree stems are calculated for eight cases (3): first, overriding only the smallest stems, then overriding the next largest class of stems as well, etc., until in the eighth case all stems are being overridden.

Stem override resistances are combined with the modified tractive force-speed relation to predict nine speeds as limited by basic resistances (5). (The ninth speed corresponds to avoiding all tree stems.)

Maximum braking force and recognition distance are combined to compute a visibility-limited speed (6). Resistance and visibility-limited speeds are compared to the speed limited by tire loading and inflation (7), if applicable, and to the speed limit imposed by driver tolerance to vehicle motions resulting from ground roughness (8). The least of these speeds for each tree override-and-avoid option becomes the maximum speed possible between obstacles by that option, except for degradation due to maneuvering (9).

Obstacle avoidance and/or the tree avoidance implied by limited stem override requires the vehicle to maneuver (or may be impossible).

Using speed reduction factors (derived in 1) associated with avoiding all obstacles (if possible) and avoiding the appropriate classes of tree stems, a series of nine possible speeds (possibly including zero, or NOGO) is computed (10).

A similar set of nine speed predictions is made for the vehicle maneuvering to avoid tree stems only (10). These are further modified by several obstacle crossing considerations.

Possible NOGO interference between the vehicle and the obstacle is checked (12). If obstacle crossing proves to be NOGO, all associated vegetation override and avoid options are also NOGO. If there are no critical interferences, the increase in traction required to negotiate the obstacle is determined (12).

Next, obstacle approach speed and the speed at which the vehicle will depart the obstacle, as a result of the momentarily added resistance encountered, are computed (13). Obstacle approach speed is taken as the lesser of the speed between obstacles, reduced for maneuver required by each stem override and avoid option, and the speed limited by the driver to control his crossing impact (11). Speeds off the obstacle are computed on the basis solely of the soil-and slope-modified tractive force-speed relation (22), i.e. before the tractive force speed relation is modified to account for vegetation override forces, the traction increment required for obstacle negotiation, or any kinetic energy available as a result of the associated obstacle approach speed (13).

Final average speed in the patch for each of the nine tree stem override and avoid options, while the vehicle is overriding patch obstacles, is computed from the speed profile resulting, in general, from considering the vehicle to accelerate from the assigned speed off the obstacle to the allowable speed between obstacles (or to a lesser speed if obstacle spacing is insufficient), to brake to the allowable obstacle approach speed, and to cross the obstacle per se at the computed crossing speed.

Following a final check to ensure that traction and kinetic energy are sufficient for single-tree overrides required (and possible resetting of speeds for some options to NOGO) a single maximum in-patch speed (for the direction of travel being considered relative to the in-unit slope) is selected from among the nine available values associated with obstacle avoidance and the nine for the obstacle override cases. If all 18 options are NOGO, the patch is NOGO for the direction of travel. If several speeds are given, selection is made by one of two logics according to scenario input instructions.

In the past the driver was assumed to be both omniscient and somewhat mad. Accordingly, the maximum speed possible by any of the 18 strategies was selected as the final speed prediction for the terrain unit (and slope direction). Field tests have shown, however, that a driver does not often behave in this ideal manner when driving among trees. Rather, he will take heroic measures to reach some reasonable minimum speed, but will not continue such efforts when those measures involve knocking down trees that he judges it imprudent to attack, even though by doing so he could go still faster. In INRMM, either assignment of maximum speed may be made: the absolute maximum which addresses the vehicle's ultimate potential, or a lesser value which in effect more precisely models actual driver behavior.

If the scenario data specify a traverse prediction, the in-unit speed and other predictions are complete at this point, and the model stores those results specified by the user and goes on to consider the next terrain unit (or next vehicle, condition, etc). When a full areal prediction is called for, the entire computation is repeated three times: once for the vehicle operating up the in-unit slope, once across the slope, and once down the slope. Desired data are stored from each such run prior to the next, and at the conclusion of the third run, the three speeds are averaged. Averaging is done on the assumption that one-third of the distance* will be travelled in each direction, resulting in an omnidirectional mean.

* the average speed, V_{av}, is the harmonic average of the three speeds, i.e.

$$V_{av} = 3/[(1/V_{up}) + (1/V_{across}) + (1/V_{down})]$$

b) Road Module

The Road Module calculates the maximum average speed a vehicle can be expected to attain traveling along a nominally uniform stretch of road, termed a road unit. Travel on super highways, primary and secondary roads, and trails is distinguished by specifying a road type and a surface condition factor. From these characteristics, values of tractive and rolling resistance coefficients for wheeled and tracked vehicles on hard surfaced roads are determined by a table look-up. For trails, surface condition is specified in terms of cone index (CI) or rating cone index (RCI). Traction, motion resistance, and slip are computed using the soil submodel of the Areal Module, with scenario weather factors used in the same way as in making off-road predictions.

The relations used for computing vehicle performance on smooth, hard pavements are taken from the literature (Smith (1970) and Taborek (1957)).

The structure of the Road Module, while much simpler, parallels that of the Areal Module. Separate speeds are computed as limited by available traction and countervailing resistances (rolling, aerodynamic, grade, and curvature), by ride dynamics (absorbed power), by visibility and braking, by tire load, inflation and construction, and by road curvature per se (a feature not directly considered in the Areal Module). The least of these five speeds is assigned as the maximum for the road unit (for the assumed direction relative to the specified grade).

The basic curvature speed limits are derived from American Association of State Highway Officials (AASHO) experience data for the four classes of roads (AASHO (1975)) under dry conditions and are not vehicle dependent. These are appropriately reduced for reduced traction conditions, and vehicle dependent checks are made for tipping or sliding while the vehicle is in the curve.

At the end of a computation, data required by the user are stored. If the model is run in the traverse mode, the model returns to compute values for the next unit; if in the areal mode, it automatically computes performance for both the up-grade and down-grade situations and at the conclusion computes the bidirectional (harmonic) average speed. Scenario options are similar to those for the Areal Module.

F. Acknowledgments

As with any comprehensive compendium covering knowledge in a particular subject area, the results are due to the combined effort of all workers in the discipline. The authors, in this case, are somewhat akin to the scribes of ancient days, recording and organizing the wisdom and folly of those around them.

There are those, however, whose contributions stand out as related to the creation of the Mobility Model itself. The authors wish to acknowledge these people explicitly.

Clifford J. Nuttall, Jr., currently with the Mobility Systems Division, Geotechnical Laboratory at the U. S. Army Engineer Waterways Experiment Station (WES) provided the inspiration for many of the submodels, guided the evolution of the content of the entire model, and provided the wisdom and judgement which hopefully kept the various portions in proportion with each other. Additional experience in use of this and predecessor models came from many studies conducted by Donald Randolph at WES. During the model development period, general direction and supervision at WES came from W. G. Schockley, A. A. Rula, E. S. Rush and J. L. Smith.

Peter Haley, from the Tank Automotive Concepts Laboratory, USA TARADCOM and, also the manager of the NATO Reference Mobility Model, in addition to providing overall guidance and judgment

did much of the seemingly endless detailed design and testing of the algorithms and code. He was aided in the coding by Thomas Washburn. Direct supervision of the model development at TARADCOM came from Zoltan J. Janosi, who also now serves as Chairman of the Technical Management Committee of the NATO Reference Mobility Model. General supervision during the project was provided by J. G. Parks, O. Renius, and Lt. Col. T. H. Huber. Dr. E. N. Petrick, Chief Scientist of USA TARADCOM, the moving force of the NATO RSI effort in the U. S. Army vehicle community, provided overall guidance and support for this activity. He has been aided in this by Edward Lowe, NATO Standardization and Metrication Officer at TARADCOM.

Newell Murphy, of the Mobility Systems Division, WES provided the driving force behind the current version of the Ride Dynamics Module, supervising its conception, creation, and testing as well as guiding the field work supporting it. Richard Ahlvin of WES and Jeff Wilson of Mississippi State University bore primary responsibility for the production of the sequence of computer programs which have implemented this Module.

The authors also wish to acknowledge the contributions of their colleagues at Stevens Institute of Technology. Jan Nazalewicz was responsible for much of the Obstacle Module. Supervision and guidance during the project came from I. Robert Ehrlich and Irmin O. Kamm.

The arduous task of entering and formatting the text of this report was performed by M. Raihan Ali and Gabriel Totino. Graphics and charts were prepared by Mary Ann McGuire and Christopher McLaughlin. The authors benefited from a careful review of the first draft by Peter Haley. Finally each of the authors notes than any errors are the fault of the other author.

II ALGORITHMS AND EQUATIONS

A. Introduction

The Obstacle Module, OBS78B, is a stand alone program which simulates the placement of the vehicle at a sequence of positions across the obstacle and for each position calculates

1. the tractive forces under the running gear to maintain that position,

and

2. the clearances/interferences between the frame of the vehicle and the obstacle at that position,

and then

3. selects the maximum interference, CLRMIN, (or minimum clearance if there is no interference) and the maximum tractive effort, FOOMAX, and calculates the average tractive effort, FOO, across the various positions.

Figure II.A.1 gives an overall view of the structure of the Obstacle Module.

The obstacles are restricted to the "standard" trapezoidal shape used throughout the INRMM. The effect of the predominant slope may be included in OBS78B, but there are currently no provisions for incorporating the predominant slope in combination with obstacle crossing in the Operational Modules. Thus, for the Obstacle Module the terrain input may be characterized as illustrated in Figure II.A.2.

There is a restriction in OBS78B that the combination of slope and obstacle approach angle may not exceed the vertical for any obstacle flank on which the vehicle may rest.

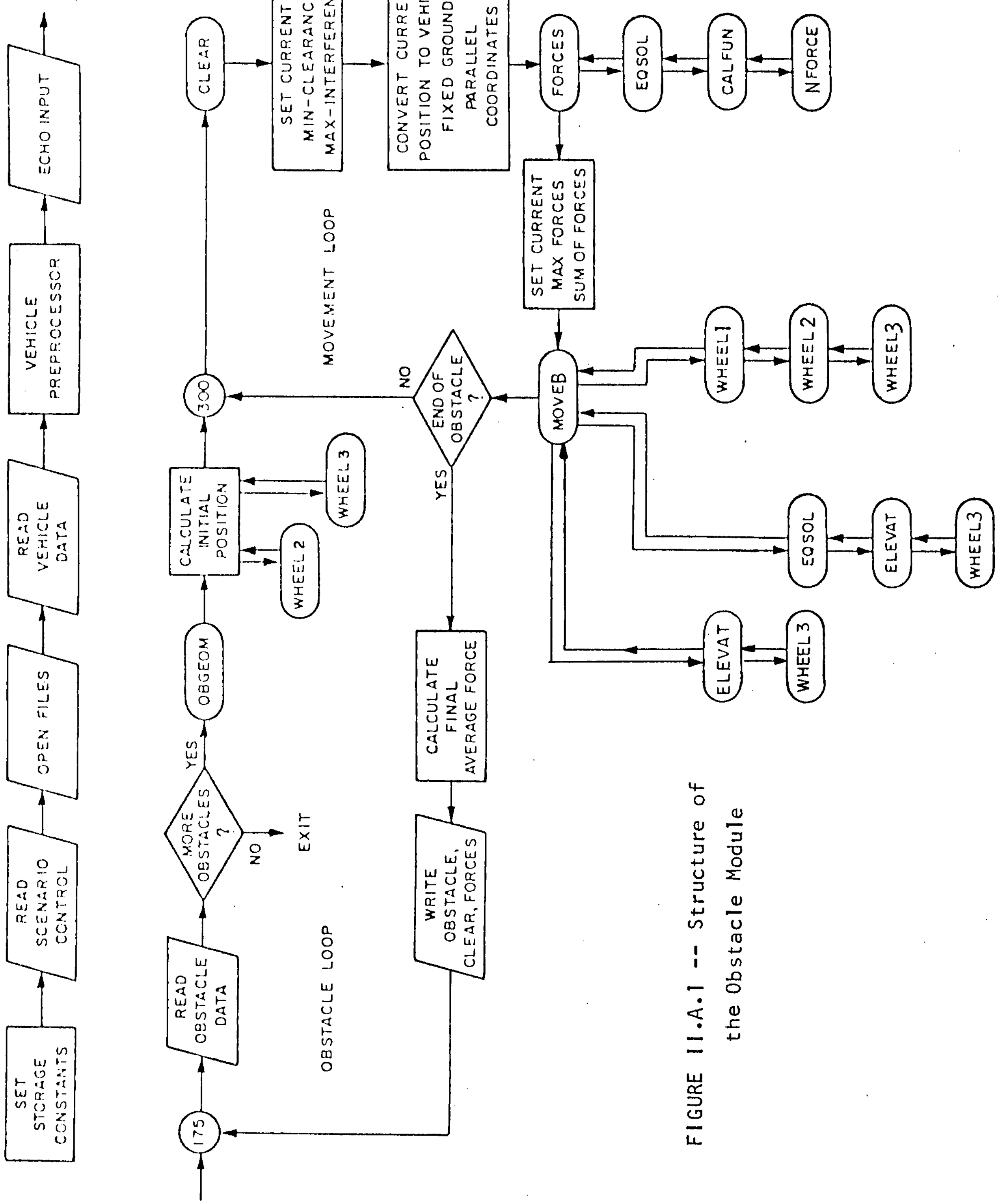


FIGURE II.A.1 -- Structure of the Obstacle Module

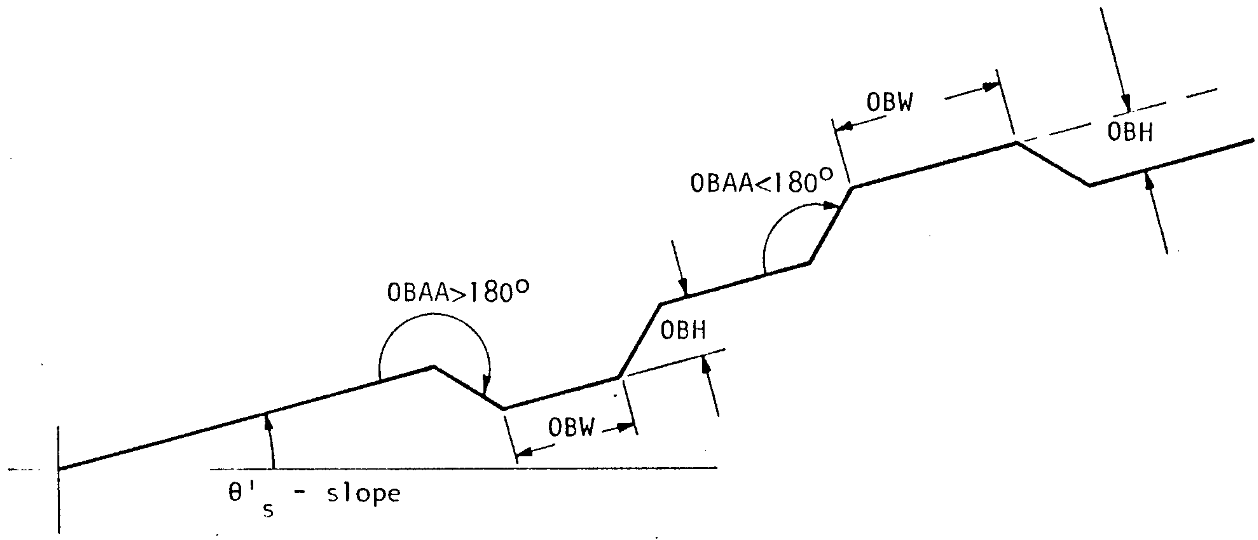


FIGURE II.A.2 - Obstacle Geometry

The vehicle is restricted to two units, a prime mover, supported by suspension assemblies at two points, and a trailer, supported by a suspension assembly at one point with a hitch rigidly attached to the prime mover about which the trailer may pivot. The suspension assemblies are rigid (no springs or dampers) and may be single wheeled or "bogied", which for the purposes of OBS78B means two wheels attached to a rigid member which pivots about its center at the suspension support point. This motion is restricted by, possibly different, pitch up and down limits with respect to the frame of the vehicle. Any mix of single wheeled or bogie suspensions may exist on the prime mover-trailer combination. The wheels are also assumed rigid but need not have the same radii for all suspension assemblies.

However, both wheels on a bogie have the same radius.

Tracked vehicles may be simulated by a double bogie wheeled vehicle where the wheel radius is the road wheel radius plus the thickness of the track. The bogie centers may be located anywhere the user wishes; reasonable results have been obtained by using the location of the second and second-from-last roadwheel centers. The width of the bogie, defined as the distance between the centers of the two wheels on the bogie, is also at the discretion of the user; reasonable results have been obtained by choosing the distance between two road wheels. When the bogie center and width have been chosen, the bogie angular limits should then be set to reflect the actual road wheel displaced as if the track were present at its normal tension. This will result in a large pitch up angular limit for the front bogie and a smaller pitch down angular limit. The rear bogie will have the reverse angular limits.

When the vehicle data has been read by the program, some initial calculations are done. These are described more fully below. The program then reads the obstacle shape and calculates hub profiles. These profiles are intended to simulate the path taken by the wheel centers across the obstacle, assuming a rigid wheel and uninterrupted contact. The program will use one of these two possible hub profiles across a mound:

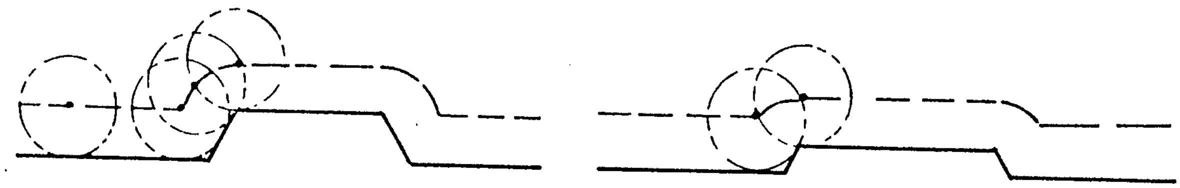


FIGURE II.A.3 - Hub Profiles Across Mounds

or one of these four possible hub profiles across a ditch:

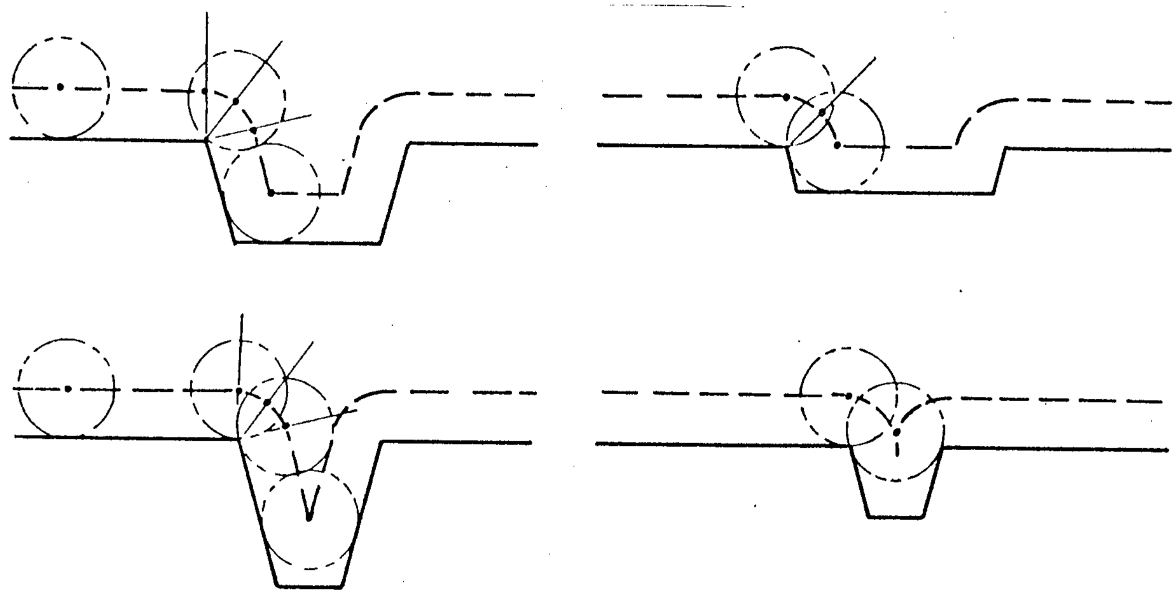


FIGURE II.A.4 - Hub Profiles Across Ditches

It may be observed that the vertical variation of the hub profile may be attenuated when compared to that of the obstacle profile; this effect may occur both for the net change in elevation and/or the rate of that change. This attenuation increases as the radius of the wheel increases with respect to the obstacle dimensions.

Tracked vehicles, in effect, attenuate obstacles as if they were equipped with very large wheels. The exact equivalent wheel diameter which attenuates an obstacle as does the tracked suspension element is not readily calculated, and for any one vehicle may not be constant for all obstacles. In the Obstacle Module, two different wheel sizes are used to simulate tracked vehicles:

1. for a flexible track the radius of the wheel used to calculate the hub profile is set at one-half the distance between suspension element support points, and
2. for a non-flexible (girderized) track the radius of the wheel used to calculate the hub profile is set at the full distance between suspension element support points.

Figure II.A.5 shows the vehicle parameters used in the module and indicates the vehicle configurations which can be simulated. Tracked vehicles pulling trailers are not simulated.

All horizontal dimensions are positive to the right of the hitch and negative to the left. All vertical dimensions are measured with respect to the ground when the vehicle is empty and at rest on level, hard ground. Vehicle motion is assumed from left to right.

N.B.: Either or both of the suspension elements of the prime mover may be single wheel or bogie supports. The hitch may be located before the second axle to possibly simulate a fifth wheel.

The wheels of a suspension element may be powered, braked, both or neither. Suspension types may be mixed in any combination but both wheels of a bogie suspension are assumed to have the same radius and ability to be powered and braked. During execution of the program, however, at any position on the obstacle either all braked wheels are braked or all powered wheels are powered.

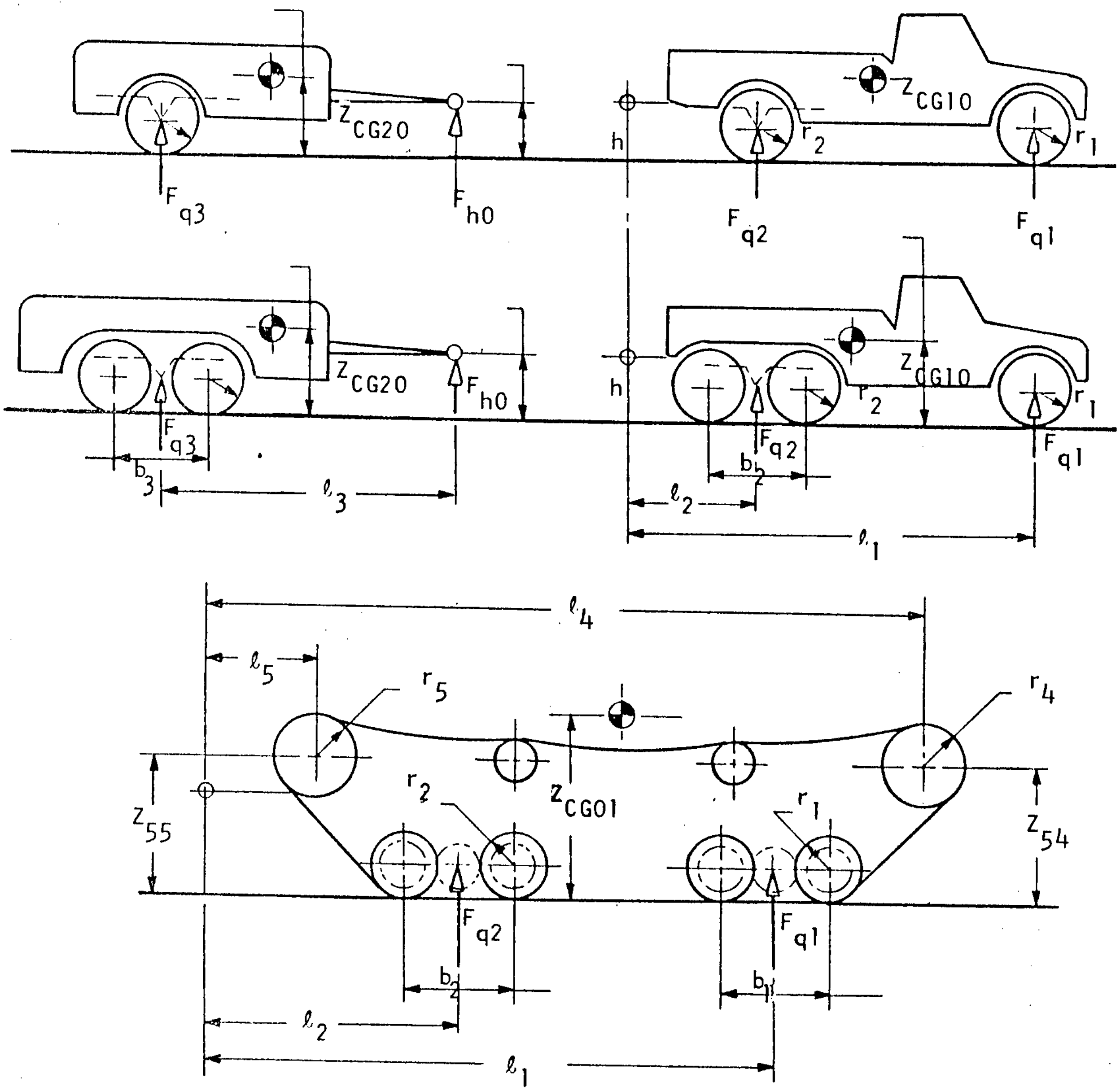


FIGURE II.A.5 -- Vehicle Parameters

B. Coordinate Systems

Four separate coordinate systems are used in OBS78B, vehicle input data coordinates, vehicle coordinates, ground fixed coordinates and vehicle/ground coordinates. Each system is specified below.

1. Vehicle Input Data Coordinates

This coordinate system (Figure II.B.1) is centered at a point on the ground directly under the hitch when the vehicle is resting on a hard, flat surface and facing toward the right of the observer.

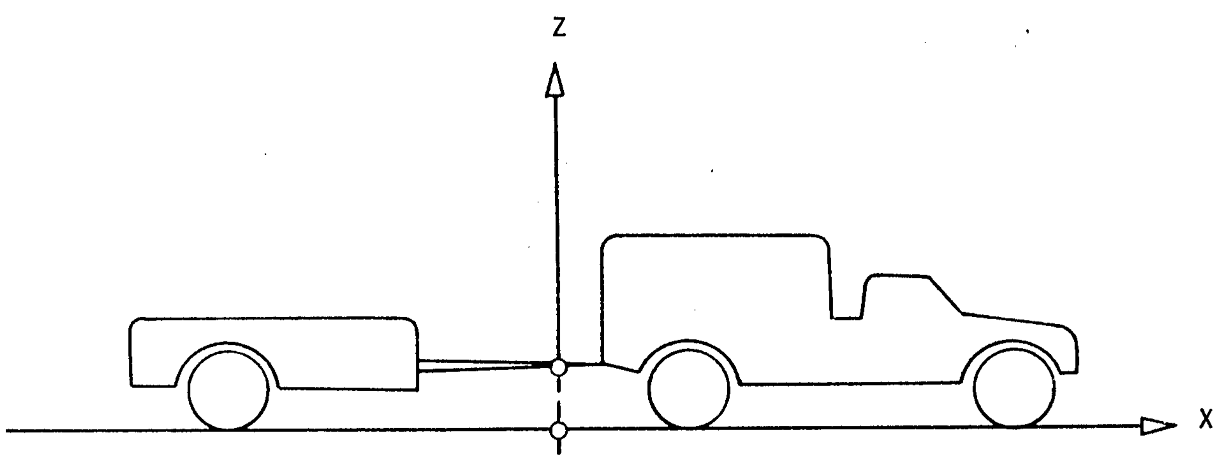


FIGURE II.B.1 -- Vehicle Input Data Coordinates

All vehicle input data is given with respect to this coordinate system. It is used only for the convenience of the investigator; all data is immediately transferred to the Vehicle Coordinates.

2. Vehicle Coordinates

This coordinate system is centered at the hitch and moves with the prime mover. See Figure II.B.2.

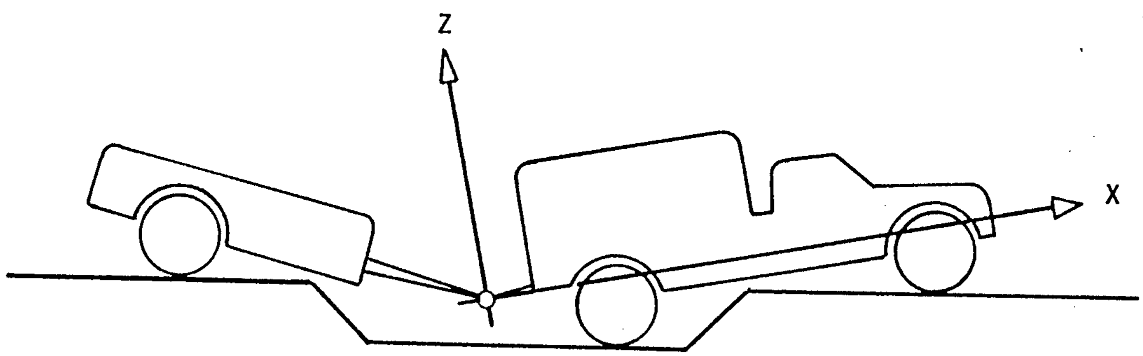


FIGURE II.B.2 -- Vehicle Coordinates

The x-axis is horizontal and fixed to the vehicle when the vehicle is at rest on hard, flat ground. Thus the Vehicle Coordinates are initially parallel to the Input Data Coordinates translated vertically a distance of the height of the hitch for an empty vehicle. The pitch angle of the vehicle, θ_1 , is in effect the angle the vehicle x-axis makes with the Ground Fixed Coordinate System.

3. Ground Fixed Coordinate System

This coordinate system remains fixed to the ground and is centered at the first obstacle profile break point. Its coordinates are designated with primed quantities. The z'-axis is positive up, along the negative gravity vector, and the x'-axis is positive to the right. See Figure II.B.3.

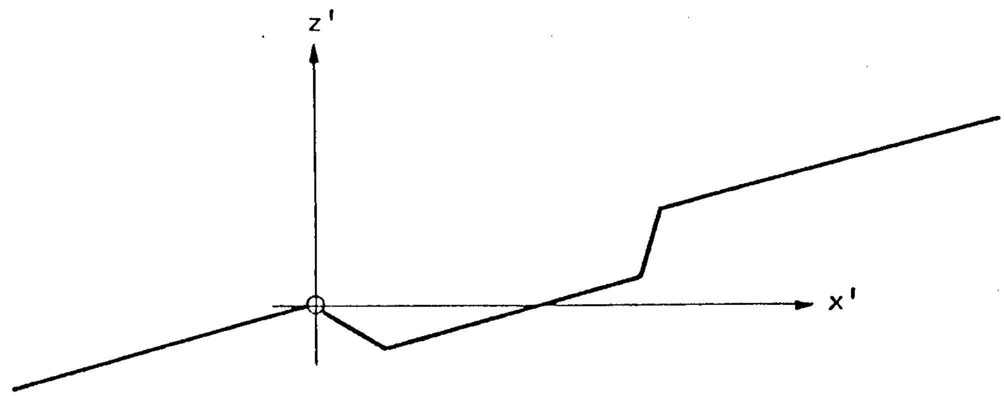


FIGURE II.B.3 -- Ground Fixed Coordinates

4. Vehicle Fixed-Ground Parallel Coordinate System

This coordinate system is centered at the hitch and moves with the vehicle; however it remains parallel to the Ground Fixed Coordinate System. Initially it coincides with the Vehicle Coordinates when the vehicle is at rest on hard, flat ground. Its coordinates are designated by a superscript F.

The relationship between the three program coordinate systems is illustrated in Figure II.B.4.

C. OBS78B Vehicle Preprocessor

After the vehicle data is read, several derived vehicle descriptors are calculated. These descriptors are given in terms of the vehicle coordinates.

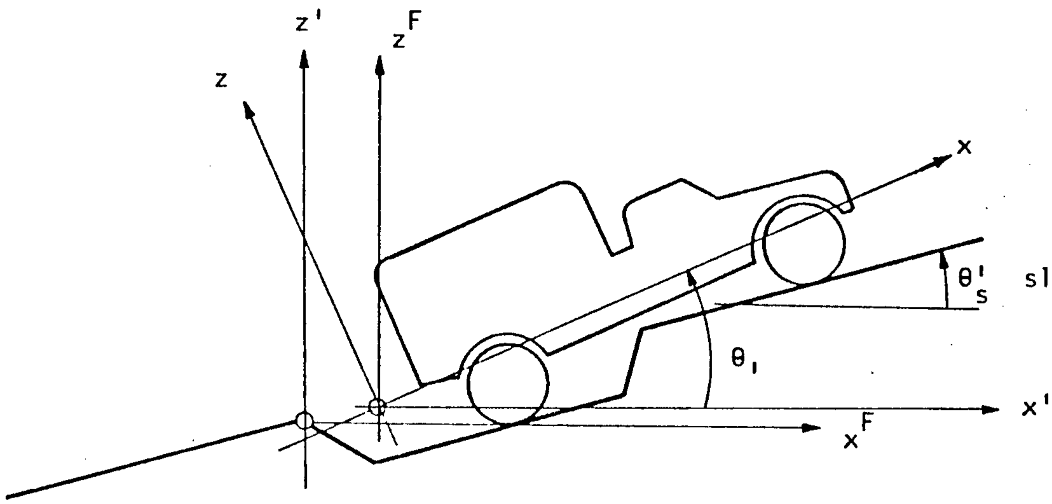


FIGURE II.B.4 -- Relation of Three Coordinate Systems

Since the vehicle load distribution is given for an empty vehicle, a combined vehicle-load CG is calculated (superscript e means empty vehicle).

The empty vehicle weight at the vehicle CG:

$$F^e_{CG1} = -F_{q1} - F_{q2}$$

The x-coordinate of the empty vehicle CG:

$$x^e_{CG1} = -(F_{q1}l_1 + F_{q2}l_2)/F^e_{CG1}$$

The empty trailer weight at the trailer CG:

$$F^e_{CG2} = -F_{q3} - F_{h0}$$

The x-coordinate of the empty trailer CG:

$$x^e_{CG2} = -F_{q3}l_3/F^e_{CG2}$$

The loaded weights at the combined CG:

$$F_{CG1} = F^e_{CG1} - \Delta W_1$$

$$F_{CG2} = F^e_{CG2} - \Delta W_2$$

The coordinates of the combined vehicle/load CG:

$$x_{CGi} = (F^e_{CGi}\ x^e_{CGi} - \Delta W_i d_i)/\ F_{CGi}$$

$$z_{CGi} = (F^e_{CGi}\ z^e_{CGi} - \Delta W_i e_i)\ /F_{CGi}$$

where i1 for the vehicle, 2 for the trailer.

From now on these coordinates of the loaded vehicle will be called the vehicle and trailer CG coordinates.

The radius vector from the CG to the hitch in polar coordinates:

$$R_{hi} = [x^2_{CGi} + z^2_{CGi}]^{1/2}$$

$$\theta_{ohi} = \arctan(z_{CGi}/x_{CGi}) \pm \pi$$

where i=1 for the vehicle, 2 for the trailer.

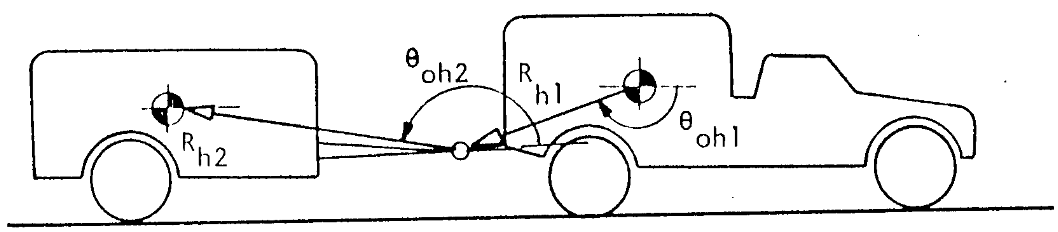


FIGURE II.C.1 -- Hitch and Trailer CG Location

N.B.: Radius vector is from vehicle CG to hitch and from hitch to trailer CG.

θ_{ohi} is adjusted to lie in the interval $[-\pi, \pi]$.

The polar coordinates of the vehicle suspension support points:

$$r_{BCi} = [\ (l_i - x_{CG1})^2 + (r_i - h - z_{CGi})^2\ ]^{1/2}, \quad i=1,2$$

$$\theta_{BCi} = \arctan[(r_i - h - z_{CG1})/ (l_i - x_{CG1})] , i=1,2$$

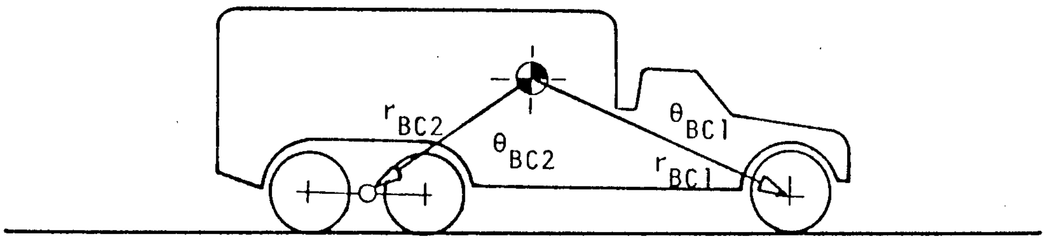


FIGURE II.C.2 -- Vehicle Suspension Support Point Locations

The following are calculated for each suspension element which is represented by a bogie:

The polar coordinates of the wheel centers when they are at their limit position closest to the vehicle:

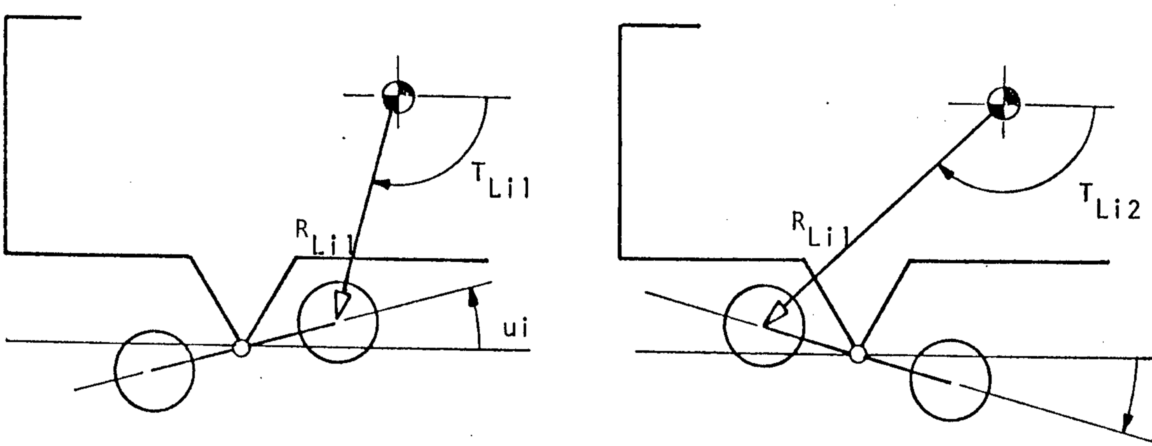


FIGURE II.C.3 -- Wheel Center Locations at Bogie Limits

(x_B, z_B) are the coordinates of the suspension support center with respect to the first unit CG.

$$R_{Li1} = [(x_B + (b_i/2)\cos\beta_{ui} - x_{CG1})^2 + (z_B + (b_i/2)\sin\beta_{ui} - z_{CGi})^2]^{1/2}$$

$$R_{Li2} = [(x_B - (b_i/2)\cos\beta_{di} - x_{CG1})^2 + (z_B - (b_i/2)\sin\beta_{di} - z_{CG1})^2]^{1/2}$$

$$T_{Li1} = \arctan[(z_B + (b_i/2)\sin\beta_{ui} - z_{CG1}) / (x_B + (b_i/2)\cos\beta_{ui} - x_{CG1})]$$

$$T_{Li2} = \arctan[(z_B - (b_i/2)\sin\beta_{di} - z_{CG2}) / (x_B - (b_i/2)\cos\beta_{di} - x_{CG2})]$$

For the trailer, these polar coordinates are given with respect to the hitch:

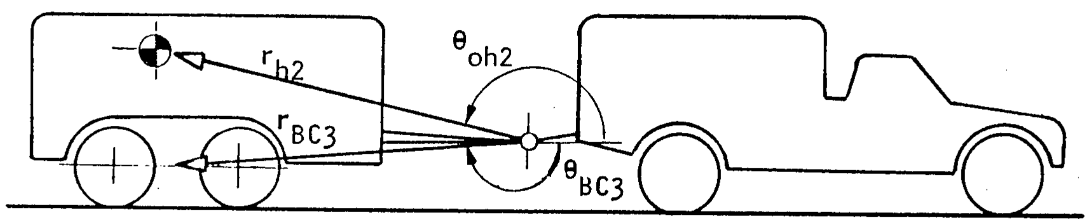


FIGURE II.C.4 -- Trailer CG and Suspension Support Location

$$r_{h2} = [x_{CG2}^2 + z_{CG2}^2]^{1/2}$$

$$\theta_{Oh2} = \arctan (z_{CG2} / x_{CG2})$$

$$r_{BC3} = [l_3^2 + (r_3 - h)^2]^{1/2}$$

$$\theta_{BC3} = \arctan [(r_3 - h)/l_3]$$

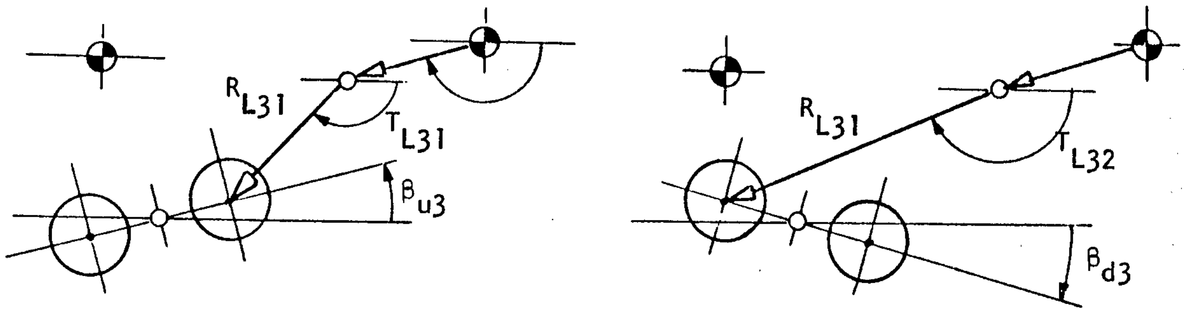


FIGURE II.C.5 -- Trailer Bogie Wheel Locations at Bogie Limits

(x_{hB}, z_{hB}) are the coordinates of the trailer suspension support point in vehicle coordinates.

$$R_{L31} = [(x_{hB} + (b_3/2)\cos\beta_{u3})^2 + (z_{hB} + (b_3/2)\sin\beta_{u3})^2]^{1/2}$$

$$T_{L31} = \arctan[(z_{hB} + (b_3/2)\sin\beta_{u3})/(x_{hB} + (b_3/2)\cos\beta_{u3})]$$

$$R_{L32} = [(x_{hB} - (b_3/2)\cos\beta_{d3})^2 + (z_{hB} - (b_3/2)\sin\beta_{d3})^2]^{1/2}$$

$$T_{L32} = \arctan[(z_{hB} - (b_3/2)\sin\beta_{d3})/(x_{hB} - (b_3/2)\cos\beta_{d3})]$$

The effective radius of the wheels to be used in the hub profile calculations is set to

$r_{ti} = r_i$ for wheeled vehicle unit

$r_{ti} = 1/2(l_1 - l_2)$ for tracked unit with flexible track

$r_{ti} = r_{ti} - r_i$ for tracked unit with girderized track.

Since the use of r_{ti} may have the effect of raising the entire vehicle far above the ground level, the result may be that no interference between vehicle bottom and the ground will be recorded when, in fact, it would actually occur. To avoid this difficulty, the difference between the hub profile effective radius and the normal radius

$$BPRFDL = r_{ti} - r_i$$

is used to lower the vehicle bottom profile.

The vehicle bottom profile itself is specified in the input data as the location of breakpoints given in the vehicle input coordinates. These breakpoints are then shifted to the vehicle coordinates. The preprocessor calculates the length and direction of the radius vector to each of these breakpoints. The radius vector originates at the hitch joint for both the prime mover and the trailer.

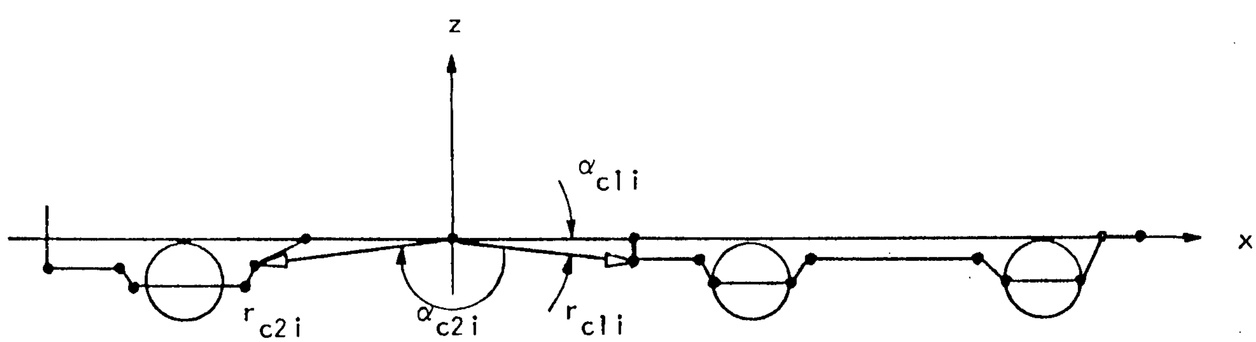


FIGURE II.C.6 -- Specification of Vehicle Bottom Profile Breakpoints

In Figure II.C.6, the bottom profile points are marked with heavy dots and calculated as follows:

$$r_{cki} = [\, x_{cki}^2 + (y_{cki} - BPRFDL)^2 \,]^{1/2}$$

$$\alpha_{cki} = \arctan [(y_{cki} - BPRFDL) / x_{cki}]$$

where $k = 1$ denotes the prime mover

$k = 2$ denotes the trailer

and

for $i = 1, \ldots, N_{ck}$

where N_{ck} is the number of bottom profile breakpoints on unit k. The hitch may, but need not be, included as a bottom profile breakpoint.

This completes the calculations of the OBS78B vehicle preprocessor. The predominant slope, θ_s', is read and then the program enters the obstacle loop. The set of three descriptors for each obstacle is read; these are OBH, OBAA, and OBW as defined in section III.B. The program then transfers to subroutine OBGEOM where the hub profiles and the step size are calculated.

Before transfer to OBGEOM, a check is made to determine if the sum of the predominant slope and the obstacle approach slope exceeds the vertical. If it does, an error message is printed, calculations for the obstacle are skipped and the next obstacle is read.

D. Subroutine OBGEOM

This subroutine introduces the obstacle and hub profile index scheme used throughout the program. For an obstacle/wheel combination such that all hub profile flanks are present it is illustrated in Figure II.D.1.

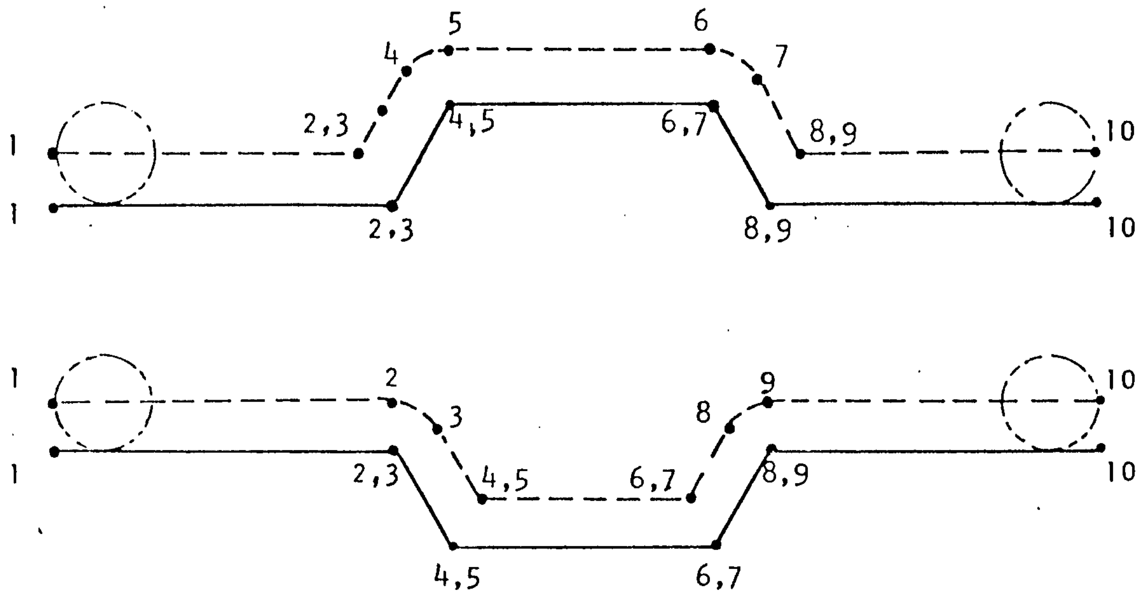


FIGURE II.D.1 -- Obstacle and Hub Profile Breakpoint Indices

Observe that all obstacle breakpoints except 1 and 10 have two indices. This is to accomodate the hub profile breakpoint numbering which may result in two profile elements for each obstacle breakpoint. The obstacle and hub profile flanks are given the number of their left end breakpoint index as shown in Figure II.D.2. For obstacle/wheel combinations that give rise to hub profiles of fewer elements, some hub profile breakpoints may have up to six indices.

The ground fixed coordinate system always has its origin at the obstacle breakpoint 2.

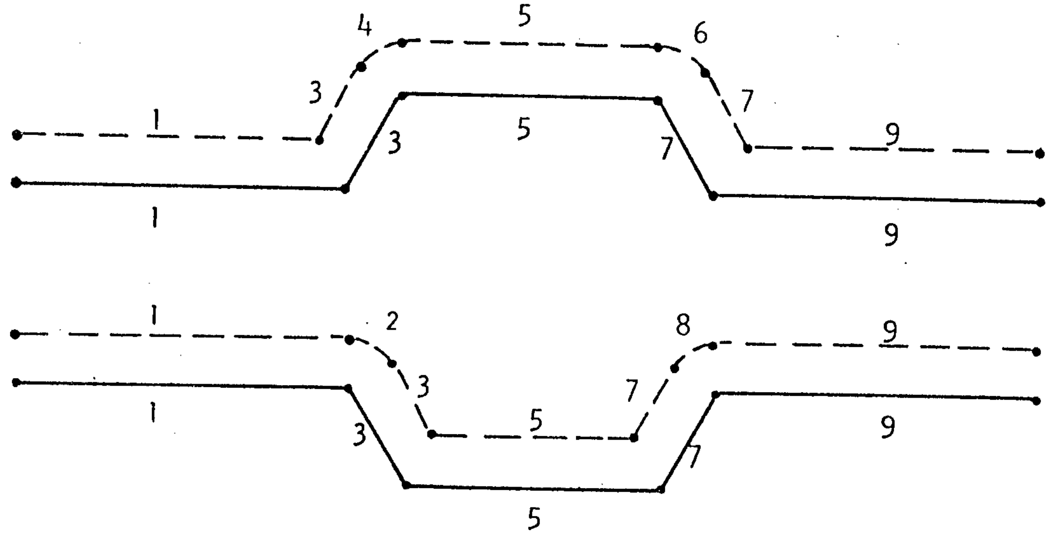


FIGURE II.D.2 -- Obstacle and Hub Profile Flank Indices

The approach and departure flanks, numbered 1 and 9 respectively, are set so that their slope is the predominant slope, θ_s', and their length is sufficient to accomodate all suspension elements simultaneously plus 1 inch. The vehicle is started on the approach slope .1 inches from initial contact with a mound or with its front wheel contact point .1 inches from hub profile element number 2 for a ditch.

Subroutine OBGEOM first calculates the x',z'-coordinates of the obstacle and hub profile breakpoints for zero predominant slope. It then rotates the location of these points about obstacle breakpoint 2 (the x'z' origin) through angle θ_s'. The length of each of the obstacle and hub profile elements is calculated. In addition, for each obstacle element, the angle with respect to the x'-axis is also given. For the hub profile elements, the coefficients of the general quadratic

$$A_{ij}x^2 + B_{ij}xz + C_{ij}z^2 + D_{ij}x + E_{ij}z + F_{ij} = 0$$

are calculated. Here the subscript j refers to the hub profile element number and i refers to the suspension element whose wheels generate it. Since hub profile elements are always either points, lines, or arcs, $B_{ij} = 0$ and $A_{ij} = C_{ij} = 1$ for arcs whereas $A_{ij} = B_{ij} = C_{ij} = 0$ for lines and points.

Finally, OBGEOM calculates STEP, the distance the first unit CG will be moved from position to position across the obstacle. For this version of the Obstacle Module, STEP is constant for a vehicle/obstacle combination and is set to 49% of the shortest hub profile element length or 1 inch, whichever is greater.

E. Initial Values and Position

When the vehicle and obstacle have been completely defined, the initial position of the vehicle on the approach slope is calculated. Also, initial values for the solutions of the force balance equations are set. These variables (the solution variables for the force balance equations) are defined as

XN(1) = overall traction coefficient
XN(2) = normal force on first suspension element
XN(3) = normal force on second suspension element
XN(4) = normal force on third suspension element
XN(5) = horizontal hitch force applied to vehicle
XN(6) = vertical hitch force applied to vehicle

For initialization, XN(1) = RTOW(1), the resistance over weight coefficient of the first suspension element (an input number); XN(2), XN(3), and XN(4) are set to the normal load on those suspension elements when the vehicle is at rest on level ground; XN(5) $= F_{hx'} = 0$, and XN(6) $= F_{hz'o}$ the initial hitch load when the trailer is at rest on level ground.

To position the vehicle, the following calculations are performed:

a) the first wheel is positioned 1/10 inches before its second hub profile breakpoint

$$x'_{w11} = x'_{h12} - .1 \cos(\theta'_s)$$
$$z'_{w11} = z'_{h12} - .1 \sin(\theta'_s)$$

b) for a single wheel first suspension element the bogie center is set equal to the first wheel center

$$x'_{BC1} = x'_{w11}$$
$$z'_{BC1} = z'_{w11}$$

for a bogie first suspension element, the second wheel is located one bogie width behind the first and the bogie center is set between the two wheels

$$x'_{w12} = x'_{w11} - b_1 \cos(\theta'_s)$$
$$z'_{w12} = z'_{w11} - b_1 \sin(\theta'_s)$$
$$x'_{BC1} = (x'_{w11} + x'_{w12})/2$$
$$z'_{BC1} = (z'_{w11} + x'_{w12})/2$$

$$\beta_1 = \arctan((z'_{w11} - z'_{w12})/ (x'_{w11} - x'_{w12}))$$

c) the vehicle pitch angle is set parallel to the approach slope angle

$$\theta_1' = \arctan(D_{11}/ -E_{11})$$

the vehicle CG location is determined

$$x_{CG1}' = x_{BC1}' - r_{BC1} \cos(\theta_{BC1} + \theta_1')$$
$$z_{CG1}' = z_{BC1}' - r_{BC1} \sin(\theta_{BC1} + \theta_1')$$

and the location of the second suspension bogie center is calculated

$$x_{BC2}' = x_{CG1}' + r_{BC2} \cos(\theta_{BC2} + \theta_1')$$
$$z_{BC2}' = z_{CG1}' + r_{BC2} \sin(\theta_{BC2} + \theta_1')$$

d) for a single wheel second suspension, the location of the wheel center is set equal to the location of the bogie center

$$x_{W21}' = x_{BC2}'$$
$$z_{W21}' = z_{BC2}'$$

for a bogie second suspension element, the bogie angle is assumed equal to the pitch angle of the vehicle and the two wheel centers are located by

$$x'_{w21} = x'_{BC2} + (b_2/2)\cos(\theta'_1)$$
$$z'_{w21} = z'_{BC2} + (b_2/2)\sin(\theta'_1)$$
$$x'_{w22} = x'_{BC2} - (b_2/2)\cos(\theta'_1)$$
$$z'_{w22} = z'_{BC2} - (b_2/2)\sin(\theta'_1)$$

e) the hitch is then located by

$$x'_h = x'_{CG1} + R_{h1}\cos(\theta_{oh1} + \theta'_1)$$
$$z'_h = z'_{CG1} + R_{h1}\sin(\theta_{oh1} + \theta'_1)$$

For the simulation of tracked vehicles there is included, as suspension elements 4 and 5, the front and rear spridlers, respectively. In simulating a tracked vehicle, front spridler/obstacle interference is checked after step c) above. If interference is found, the vehicle is moved away from the obstacle along the approach slope until no interference is found. Thus the front spridler is located by

$$x'_s = x'_{CG1} + r_{BC4}\cos(\theta_{BC4} + \theta'_1)$$
$$z'_s = z'_{CG1} + r_{BC4}\sin(\theta_{BC4} + \theta'_1)$$

These two coordinates are passed to subroutine WHEEL3 to calculate how far above or below the front spridler hub profile the point (x'_s, z'_s) is located.

If the result of WHEEL3 is negative the spridler is below its hub profile which indicates interference. The vehicle is moved backwards on the obstacle approach slope to the point where hub profile element 3 intersects hub profile element 1 of the front spridler. The slope of hub profile element 3 is given by

$$(z'_{04} - z'_{02})/(x'_{04} - x'_{02}) = s_2.$$

The slope of the front spridler hub profile element 1 is given by $s_1 = \tan\theta'_s$. The coordinates of the point to which the front spridler center must be moved in order to just touch the obstacle is given by the solution of the following two equations

$$(z - z'_s)/(x - x'_s) = s_1$$

$$(z - z'_{h42})/(x - x'_{h42}) = s_2$$

The distance the vehicle has to be moved back to just clear the obstacle is

$$R = [(x'_s - x)^2 + (z'_s - z)^2]^{1/2} .$$

The new value of the initial coordinates of the first wheel are replaced by $(x'_{w11} - R\cos\theta'_s,\ z'_{w11} - R\sin\theta'_s)$.

The calculations from b) on are then repeated.

f) once all the values describing the vehicle's initial position have been calculated, the trailer (if there is one) is located. Given the location of the hitch (x'_h, z'_h) and the length, r_{BC3}, of the radius vector from the hitch to the trailer suspension support point, the subroutine WHEEL2 locates the trailer suspension support point (x'_{BC3}, z'_{BC3}) on the hub profile of the trailer wheels. For single wheel trailer suspension, the wheel center is set to the suspension support point

$$x'_{w13} = x'_{BC3} \qquad \text{single wheel}$$
$$z'_{w13} = z'_{BC3}$$

For trailer with bogie suspension, the wheels are located half a bogie arm before and behind the support point by

$$x'_{w13} = x'_{BC3} + (b_3/2)\cos(\theta'_2)$$
$$z'_{w13} = z'_{BC3} + (b_3/2)\sin(\theta'_2)$$

$$x'_{w23} = x'_{BC3} - (b_3/2)\cos(\theta'_2)$$

$$z'_{w23} = x'_{BC3} - (b_3/2)\sin(\theta'_2)$$

where $\theta'_2 = \theta'_1$.

g) The trailer CG is located by

$$x'_{CG2} = x'_h + R_{h2}\cos(\theta_{oh2} + \theta'_2)$$

$$z'_{CG2} = z'_h + R_{h2}\sin(\theta_{oh2} + \theta'_2)$$

h) and the angle under the wheels is set to the approach slope

$\alpha_{ij} = \theta'_s$ for wheel j of suspension element i.

F. Vehicle Movement Loop

This portion of the program calculates the clearance or interference between the bottom frame of the vehicle/trailer and the obstacle; calculates the forces between the wheels and the surface of the approach slope/obstacle/departure slope required to maintain the vehicle at the given position; and then moves the vehicle to a new position on the approach slope/obstacle/departure slope such that the distance of the CG at the new position from the CG at the previous position is equal to STEP. The program then returns to the clearance/interference calculations.

The movement loop is organized around three major subroutines CLEAR, FORCES, and MOVEB. An exit is made from the loop when the front wheel clears the departure slope.

1. Subroutine CLEAR

The relationship between the bottom frame of the vehicle and/or trailer and the obstacle profile can be illustrated by Figure II.F.1. Here the location of the obstacle profile breakpoints are given by (x'_{oi}, z'_{oi}) while that of the vehicle frame breakpoints are given by (x'_{vkn}, z'_{vkn}). The minimum and maximum clearance/interference between frame and surface will be found directly under a vehicle frame breakpoint or directly above an obstacle breakpoint. This is a consequence of approximating both the frame profile and the obstacle profile by straight line

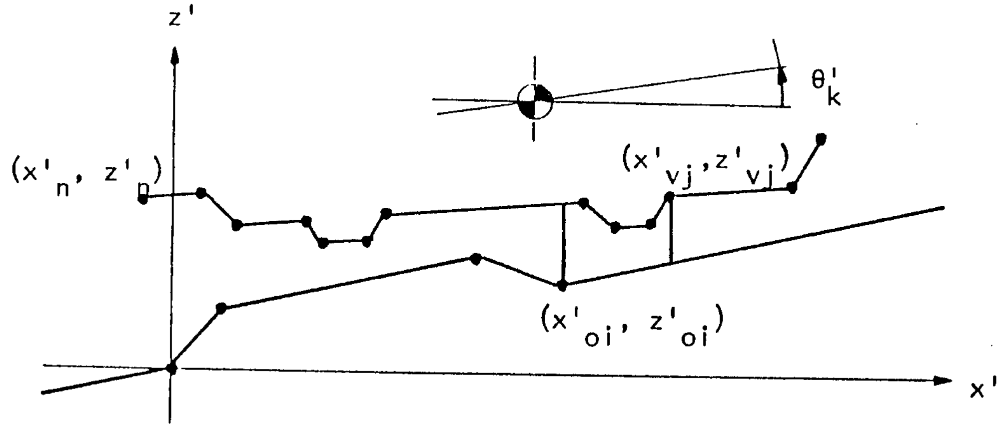


FIGURE II.F.1 -- Relation of Bottom Profile of Vehicle to Obstacle Profile

segments.

The subroutine first calculates the (x'_{vki}, z'_{vki}) for the current position and attitude by

$$x'_{vi} = x'_h + r_{cki} \cos(\theta'_k + \alpha_{cki})$$

$$z'_{vi} = z'_h + r_{cki} \sin(\theta'_k + \alpha_{cki})$$

where k = 1,2 is the vehicle unit number and i = 1,...,N designates the points on the frame profile of unit k. The routine then simply cycles through the obstacle breakpoints to determine if any part of the vehicle is above each point and calculates the clearance by linearly interpolating between the appropriate vehicle breakpoints. Similarly, for each frame profile breakpoint, the obstacle flank under the point is found and the clearance calculated. The minimum clearance/maximum interference is then found for the current position of the vehicle and an index is set pointing to that point which gave rise to the minimum clearance/maximum interference.

The determination of the overall minimum clearance or maximum interference for all positions of the vehicle across the obstacle is done with the code directly following the call to CLEAR in the main program.

2. Subroutine FORCES

This subroutine is used to estimate the tractive forces needed to overcome obstacles. This is done by evaluating the tangential tractive forces at the wheel/ground interface required to maintain the vehicle at the current position on the obstacle. Subroutine FORCES makes use of the equation solving subroutine EQSOL and subroutines NFORCE and CALFUN. The tractive force evaluation is performed for any combination of single wheel suspensions and bogie suspensions supported on both wheels or on one wheel.

To simplify and speed-up calculations eight assumptions were made:

1. Tires and suspensions are rigid.
2. Bogie beams can rotate about the pivot, but do not deflect.
3. Bogie beams take only normal forces, the tangential forces and torque are transmitted to the frame by parallel bars (A schematic version of such a bogie suspension is shown in Figure II.F.2).
4. The bogie pivot is in the middle of the line connecting the wheel centers.

5. Wheel radius is the same for all wheels on a bogie suspension
6. Each wheel can be powered, towed or braked as specified by the input data.
7. No provision is made to power some and brake other wheels at the same time.
8. Coefficients of power or brake forces can be specified by the ratios (POWERR, BRAKER) in the input data to allow for different soil conditions under each wheel.

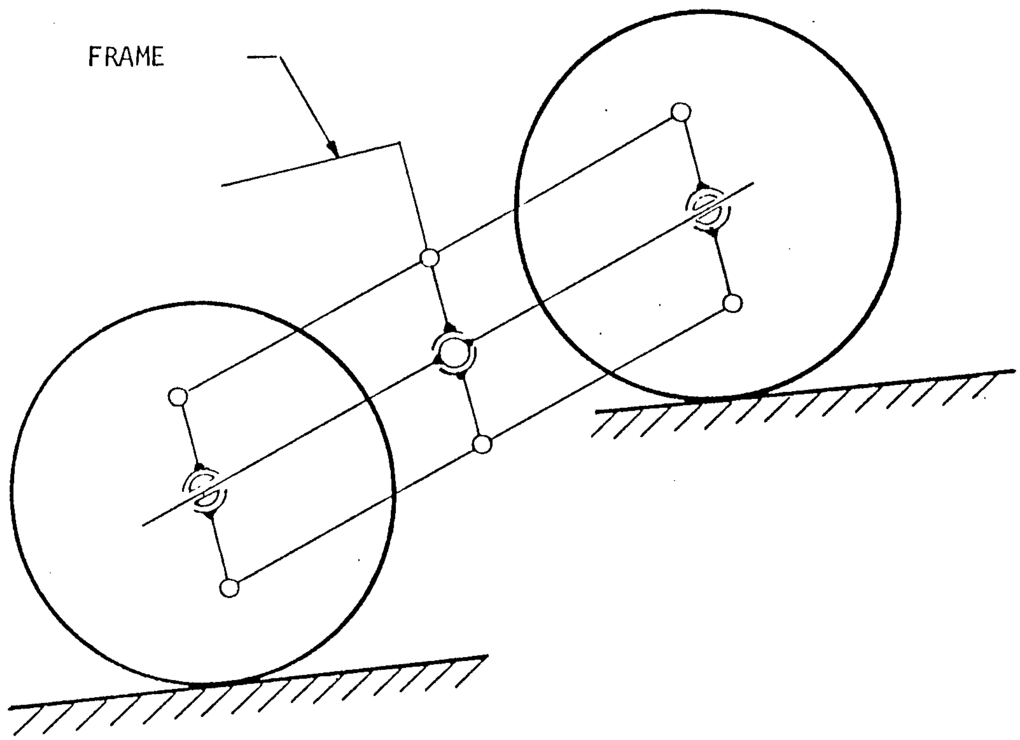


FIGURE II.F. 2 -- Schematic of Bogie Suspension

Based on the above, it is assumed that normal forces to the bogie beam are equal for both wheels of the same bogie support. The resulting system with any two suspension supports on the main unit and another on the trailer is statically determinant. The bogie assembly transmits force to the frame only at the bogie pivot point.

This routine uses the vehicle fixed-ground parallel coordinates x^F, z^F. Linear dimensions are measured from the hitch point parallel to the ground fixed coordinates x^F and z^F directions. The hitch point is the origin of the x^F, z^F coordinate systems, where the x^F axis is always horizontal and the z^F axis is vertical. Dimensions forward of the hitch are positive. Dimensions in the z^F-direction above the hitch are positive, below the hitch are negative. In the remainder of the description of Subroutine FORCES the superscript F will be omitted.

Based on previously made assumptions, the bogie can be treated as a single statically determined support point. In this case even the main unit with two bogie supports is statically determined. The sum of the forces (ground reactions, hitch forces and weight) must be zero in the x and z directions, and the moments produced by those forces about any given point also have to be equal to zero. For convenience the point about which the moments are summed is the hitch. The hitch is a common point for both units (main and trailer). For clarity, forces are always shifted to the wheel center and rotated to be parallel to the x-z coordinates. Forces at the hitch point are also resolved in the x and z direction (the hitch does not transmit a moment).

As input to this routine the main program and subroutine MOVEB supply the position of all wheels, bogie centers, bogie beam angles, bogie beam lengths, wheel radii, surface slope angles under the wheels, center of gravity locations and weights. Also entered are initial estimates for

XN(1)= overall coefficient of tractive force across all wheels,

XN(2)= normal force under the first wheel of the first suspension support,(F_{N11})

XN(3)= normal force under the first wheel of the second suspension support,(F_{21})

XN(4)= normal force under the first wheel of the third suspension support (if it exists),(F_{N31})

XN(5)= horizontal force on the hitch of the trailer (F_{HITCHx}) and

XN(6)= vertical force on the hitch of the trailer (F_{HITCHz}).

N.B.: The last three terms are included only in the case of a vehicle with a trailer.

Subroutine FORCES uses these values as initial values in an iteration, controlled by EQSOL, which will yield new values for XN(1) through XN(6) that result in the vehicle resting on the obstacle in a force and moment equilibrium state. These iterations depend on calculations performed by two subroutines, NFORCE and CALFUN, which essentially evaluate unbalanced forces and moments caused by non-equilibrium values of XN. The separation of the calculation into two subroutines is a matter of programming convenience. The description of the equations below does not distinguish in which subroutines the calculations are made.

a) Coefficient of Tractive Force

For wheel j of suspension support i:

$$C_{TFij} = XN(1)*POWERR_{ij}*IP_{ij} \qquad \text{for } XN(1) \geq 0$$

or

$$C_{TFij} = XN(1)*BRAKER_{ij}*IB_{ij} \qquad \text{for } XN(1) < 0$$

where

C_{TFij} = coefficient of tractive force

$POWERR_{ij}$ = Coefficients for distribution of tractive force among axles. The ratios of these coefficients in pairs define the force distributions.

$BRAKER_{ij}$ = Coefficients for distribution of braking force among axles. The ratios of these coefficients in pairs define the braking force distribution.

IP_{ij} = 1 , if wheel can be powered
= 0 , otherwise

IB_{ij} = 1 , if wheel can be braked
= 0 , otherwise.

Note: At any position on the obstacle, a combination of some wheels powered while others are braked is not modeled.

b) Force Relations for Single Wheel Support

Given normal force, tractive force, rolling force,wheel rolling radius and slope under wheel, the forces and the moment at the wheel center indicated in Fig.II.B.20 are calculated as follows:

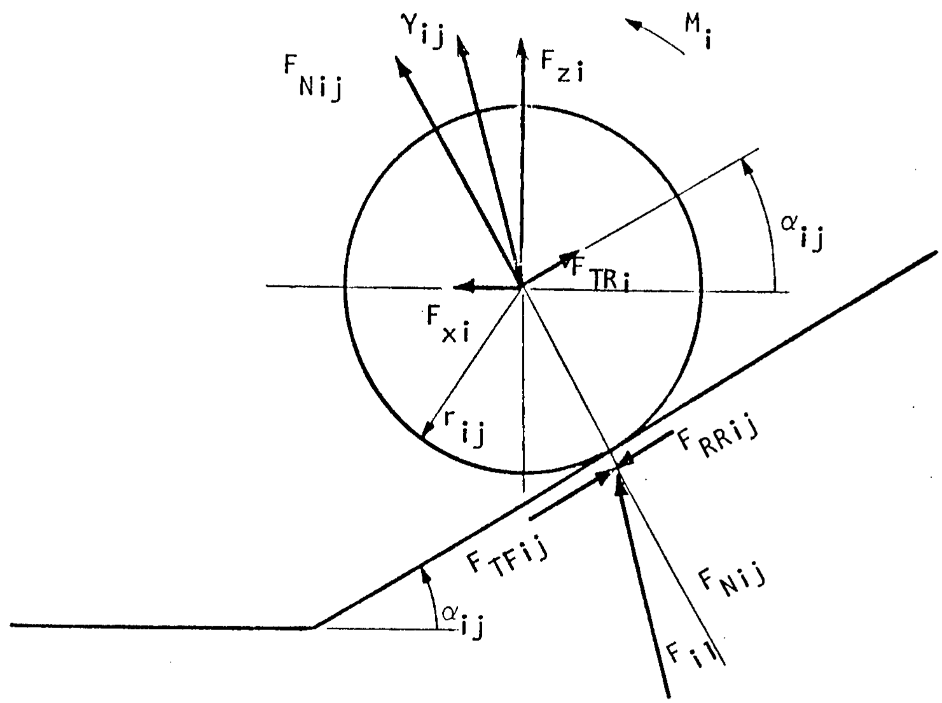


FIGURE II.F.3 -- Forces on a Single Wheel

$$F_{xi} = F_{Nij} * (C_{TRij} * \cos(\alpha_{ij}) - \sin(\alpha_{ij}))$$

$$F_{zi} = F_{Nij} * (\cos(\alpha_{ij}) + C_{TRij} * \sin(\alpha_{ij}))$$

$$M_i = C_{TFij} * F_{Nij} * r_{ij}$$

where j=1 and i designates the suspension support

C_{TRij} - Coefficient of rolling and tractive forces defined as: $C_{TRij} = C_{TFij} - C_{RRij}$

F_{TRi} - Sum of rolling resistance and tractive force $F_{TRi} = F_{Nij} * C_{TRij}$

C_{RRij} - Coefficient of rolling resistance

α_{ij} - Slope angle under wheel

F_{Nij} - Force under wheel normal to slope

F_{xi} - Force at wheel center in x-direction

F_{zi} - Force at wheel center in z-direction

M_i - Moment reaction reduced to wheel center. The moment reaction is due to the tractive force shift.The rolling force is shifted to the wheel center without a moment component.

r_{ij} - Wheel rolling radius

Note: For a single wheel, the above quantities are given for j=1. The corresponding quantities for j=2 are not used.

c) Force Relations for Bogie Support

As described below in section II.F.3, subroutine MOVEB, the vehicle may be located either with both wheels of a bogie assembly on the ground or with only one of the pair on the ground when the bogie angular motion limit is reached. The force relations are described separately for these two cases.

(1) Both wheels of the bogie support on the ground:

Assuming that the normal force, tractive force coefficient, rolling resistance coefficient and all needed geometry are known, the normal and the tangential forces acting on the bogie beam at wheel center are described as follows (see Fig.II.F.4):

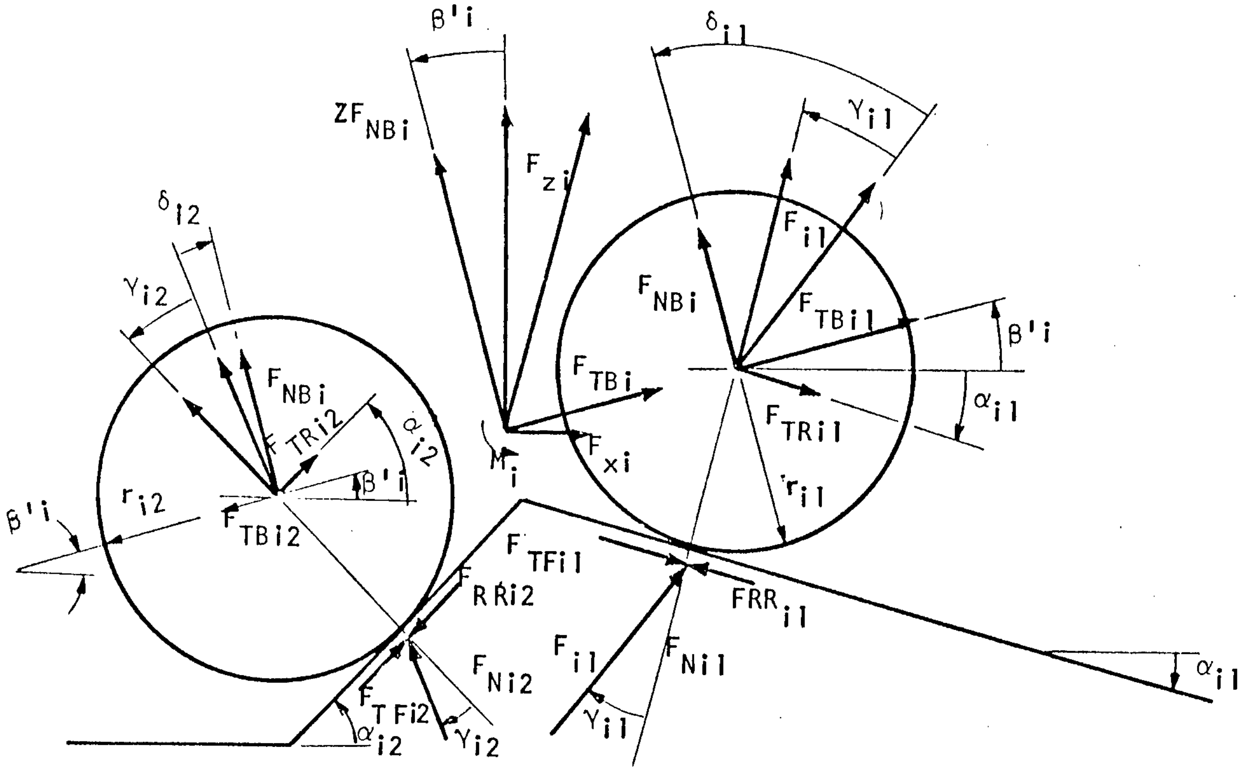


FIGURE II.F.4 -- Forces on Bogie Suspension When Both Wheels Contact the Surface

The angle (interface friction angle) that the resultant force vector under the wheel makes with the normal to the under-wheel-slope is:

$$\gamma_{ij} = \arctan(C_{TFij} - C_{RRij}).$$

The magnitude of the force vector at the center of the front wheel on the bogie is:

$$F_{i1} = F_{Ni1} / \cos(\gamma_{i1}).$$

The normal force to the bogie beam is:

$$F_{NBi} = F_{i1} * \cos(\delta_{i1})$$

where:

$$\delta_{ij} = \gamma_{ij} + \beta_i' - \alpha_{ij}$$

β_i' = angle of bogie beam with horizontal

α_{ij} = under-wheel-slope.

The tangential force on the bogie beam due to the first wheel is:

$$F_{TBi1} = F_{i1} * \sin(\delta_{i1}).$$

The equations for the normal force and the tangential force to the bogie beam due to the second wheel are calculated next, based on the previously made assumptions that the normal force to the bogie beam is equal for both wheels.

Force F_{i2} at the second wheel center is:

$$F_{i2} = F_{NBi} / \cos(\delta_{i2}).$$

The tangential force for the second wheel is:

$$F_{TBi2} = F_{i2} * \sin(\delta_{i2}).$$

The evaluated normal and tangential forces and moment on the bogie beam are shifted to the bogie pivot center and rotated to the vehicle fixed-ground parallel coordinates.

Forces at the pivot center are:

$$F_{TBi} = F_{TBi1} + F_{TBi2}$$

$$F_{xi} = -2F_{NBi} * \sin(\beta_i') + F_{TBi} * \cos(\beta_i')$$

$$F_{zi} = 2F_{NBi} * \cos(\beta_i') + F_{TBi} * \sin(\beta_i').$$

Moment at pivot center is:

$$M_i = C_{TFi1} * F_{Ni1} * r_{i1} + C_{TFi2} * F_{Ni2} * r_{i2}$$

where

r_{ij} = rolling radius of wheel j on suspension support i.

F_{xi}, F_{zi} = forces at bogie pivot center

M_i = moment reaction reduced to bogie pivot center

Note: The same rolling radius is used for all wheels on a suspension support

(2) Only one wheel of the bogie support on the ground:

Forces at the wheel center are evaluated as before for two wheel bogie support. The wheel in contact is designated by j. In the program this is indicated by the variables SFLAG and NW. The final force and moment equations reduced to the pivot center are:

$$F_{xi} = -F_{NBi} * \sin(\beta_i') + F_{TBij} * \cos(\beta_i')$$

$$F_{zi} = F_{NBi} * \cos(\beta_i') + F_{TBij} * \sin(\beta_i')$$

$$M_i = C_{TFij} * F_{Nij} * r_{ij} \pm FNBi * b_i/2$$

where:

+ if front wheel of bogie assembly is on the ground (j=1)

\- if rear wheel of bogie assembly is on the ground (j=2)

b_i = bogie arm length

Tractive force, rolling resistance force and reaction moments are calculated as follows:

$F_{Tij} = F_{Nij} * C_{TFij}$	Tractive force
$F_{Rij} = F_{Nij} * C_{RRij}$	Rolling resistance force
$M_{ij} = F_{Tij} * r_{ij}$	Reaction moment, due only to the tractive force

where:

F_{Nij} = Normal force under the wheel

The above quantities are used for information only, they are not needed by the rest of the program.

d) Force and Moment Summation for Entire Vehicle

Sum of the forces in x-direction for main unit

$$F_{Mx} = F_{x1} + F_{x2} + F_{MCGx} - F_{hx}$$

Sum of the forces in z-direction for main unit

$$F_{Mz} = F_{z1} + F_{z2} + F_{MCGz} - F_{hz}$$

Sum of the moments around hitch point for main unit

$$M_M = (M_1 + F_{x1} * z_1 + F_{z1} * x_1) + (M_2 + F_{x2} * z_2 + F_{z2} * x_2) - F_{MCGx} * z_{CGM} + F_{MCGz} * x_{CGM}$$

where:

(subscripts: M-for main unit, T- for trailer)

F_{MCGx}, F_{MCGz} = Forces at center of gravity in x-direction and z-direction respectively ($F_{MCGx} = 0$)

F_{hx}, F_{hz} = Force at trailer hitch point (negative sign for main unit, for single unit, both are equal to zero)

x_{CGM}, z_{CGM} = x and z location of center of gravity with reference to the hitch point (vehicle fixed-ground parallel coordinates)

The additional three equations for the main unit with a trailer are:

Sum of the forces in x-direction, for trailer only

$$F_{Tx} = F_{x3} + F_{TCGx} + F_{hx}$$

Sum of the forces in z-direction, for trailer only

$$F_{Tz} = F_{z3} + F_{TCGz} + F_{hz}$$

Sum of the moment around hitch point, for trailer only

$$M_T = M_i - F_{x3} * z_3 + F_{z3} * x_3 - F_{TCGx} * z_{CGT} + F_{TCGz} * x_{CGT}$$

where F_{TCGx}, F_{TCGz} are the forces at the center of gravity of the trailer in the x and z directions respectively.

These six unbalanced forces and moments F_{Mx}, F_{Mz}, M_M, F_{Tx}, F_{Tz} and M_T are all driven to zero by adjustments to XN(1), F_{N11}, F_{N21}, F_{N31}, F_{hx}, F_{hz} (the XN array) using the iterative procedure of subroutine EQSOL described in Powell (1970).

3. Subroutine MOVEB

This subroutine advances the vehicle to a new position on the obstacle profile and calculates the coordinates of the wheels, CG's, hitch, trailer, the vehicle pitch angle and the angle under the wheels, all at the new position and attitude.

MOVEB makes use of the equation solving routine EQSOL, also used by FORCES, to calculate the position of the prime mover (the vehicle) such that all the wheels are on their hub profiles (unless they are elevated above the hub profile by restrictions on the angular movement of the bogie arm with respect to the frame) in such a way that the new position of the CG is a distance of STEP away from the prior position. The value of STEP was calculated and set in subroutine OBGEOM. The independent variables of these equations are x'_{CG}, z'_{CG} and θ'_1 for single wheeled vehicle suspension elements and for those positions which yield all bogie arm positions at their limits. If the suspension elements are bogies and their equilibrium position is between their angular limits, then one or two additional independent variables are β_1 and/or β_2, the angle the bogie arm makes with respect to the vehicle x-axis.

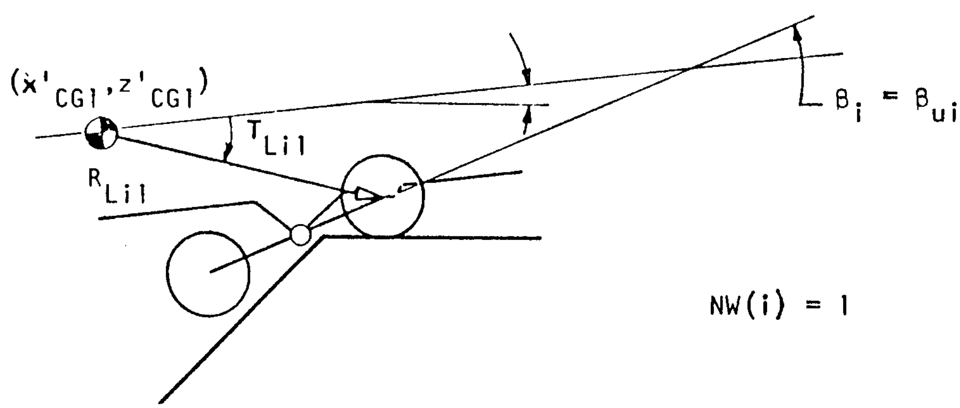
Initial estimates for these three, four, or five quantities are supplied to EQSOL; the equilibrium values of these variables are returned by EQSOL such that

$$[(x'_{CG1} + x'_{PCG1})^2 + (z'_{CG1} + z'_{PCG1})^2]^{1/2} = STEP$$

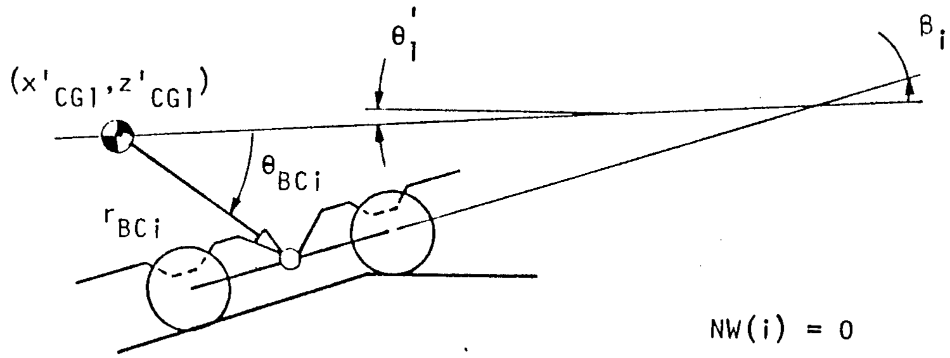
and the vertical distance of each wheel to its hub profile is zero, all within an overall tolerance of about one inch or less.

With a bogie suspension element, three possible states of support exist:

(1) on the front wheel at its upper (toward the vehicle) limit



(2) on both wheels, or



(3) on the rear wheel at its upper limit.

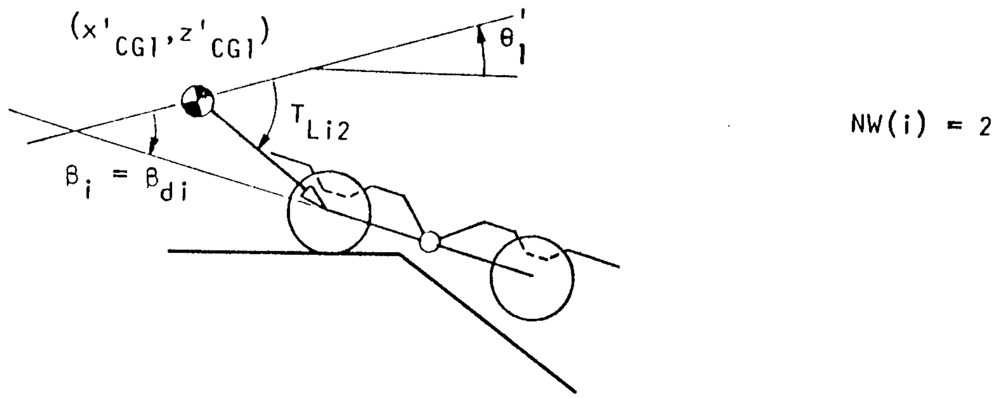


FIGURE II.F.5 -- Possible States of Support of Bogie Suspension Element

(4) In addition, for tracked vehicles, support by a spridler could be substituted for an entire suspension element.

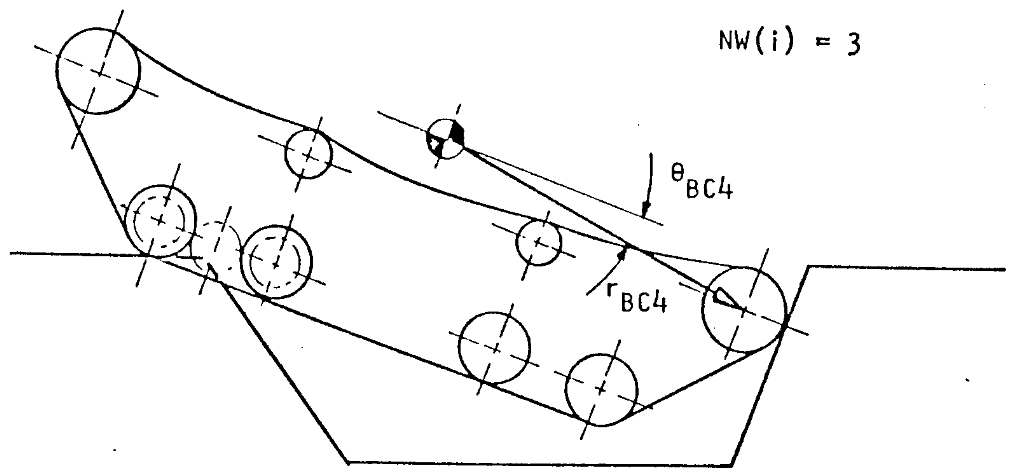


FIGURE II.F.6 -- Spridler Interference for Tracked Vehicles

If the rear spridler is supporting the vehicle, then NW(2) = 3. (In case (4), the "wheels" of the tracked vehicle that are used to model the track are much larger than pictured. The small wheels are shown for illustrative purposes only.)

Upon entry to MOVEB, the program assumes case (2) for all suspensions which are modeled with a bogie. (r_{BCi}, θ_{BCi} and β_i are passed to EQSOL to locate the supports.) This may result in up to five (NEQL = 5) independent variables and equations used to locate the vehicle. Upon return from EQSOL, the following values represent the location and attitude of the vehicle x'_{CG1}, z'_{CG1}, θ'_1 and β_1 and/or β_2. These returned values of

β_1 and/or β_2 are checked to be within their limits: $\beta_{di} \leq \beta_i \leq \beta_{ui}$, i = 1 and/or 2. If no violations to these inequalities occur, the position and attitude of the prime mover is considered final and the routine proceeds to calculate the position of the trailer, if there is one.

If, for example, $\beta_i \geq \beta_{ui}$ or $\beta_i \leq \beta_{di}$, a new entry is made to EQSOL, then the bogie of suspension i is replaced by a single wheel support with r_{BCi}, θ_{BCi}, β_i replaced by R_{Li1}, T_{Li1}, β_{ui} or R_{Li2}, T_{Li2}, β_{di} depending on which limit is exceeded. The number of independent location variables and equations is now reduced by one.

This procedure is repeated until no bogie angles exceed their limits or all bogies have been, temporarily, replaced by single wheel supports.

In case a tracked vehicle is being modeled, the location of both spridlers is now calculated. If either one is below their hub profile, EQSOL is called again with the front support replaced by one located at r_{BC4}, θ_{BC4} and/or the back support replaced by one at r_{BC5}, θ_{BC5}. Degrees of freedom may be reduced if, as shown in Figure II.F.6, the vehicle is being supported by a spridler rather than a bogie.

Once the vehicle location and attitude are returned from EQSOL all wheel and suspension support positions are calculated. This calculation, and the same ones performed during the equation solving done by EQSOL, are performed by a subroutine called ELEVAT. Given some set of x'_{CG1}, z'_{CG1}, θ'_1, β_1, β_2, flags indicating on what suspension elements the vehicle is being supported, and the length and direction of radius vectors from the CG to those vehicle support points, ELEVAT calculates x'_{wij}, z'_{wij}, x'_{BCi}, z'_{BCi} and ELEV(i), the vertical distance between wheel center i and its hub profile for all suspension elements on the prime mover.

When the above calculations and adjustments result in a position and attitude of the prime mover which does not violate any constraints and which has advanced the vehicle CG a distance of STEP across the obstacle, all the surface angles under the wheel in contact with the ground are calculated. This is done by a subroutine called WHEEL1. The hitch location is then calculated.

If a single wheel trailer is present, subroutine WHEEL2 is used to locate the trailer wheel on its hub profile maintaining the length of the radius vector, r_{BC3}, from the hitch to the trailer wheel center. The pitch angle of the trailer and the location of its CG are then calculated and a RETURN is made from MOVEB.

If a trailer is being modeled and it is fitted with a bogie suspension the trailer is first positioned on the obstacle with the front wheel at its upper most position ($\beta_3 = \beta_{u3}$) using subroutine WHEEL2 with R_{L31} and T_{L31}. If the second wheel is above its hub profile, it is concluded that this is the proper position for the trailer, its bogie center, pitch angle, and CG location are calculated and MOVEB exits.

If the second wheel is below its hub profile, the trailer is positioned on the obstacle with the rear wheel of the bogie at its upper most position ($\beta_3 = \beta_{d3}$) using subroutine WHEEL2 with R_{L32} and T_{L32}. If the first wheel is now above the hub profile, it is concluded that this is the proper position for the trailer, its bogie center, pitch angle, and CG position are calculated, and MOVEB exits.

If the first wheel is below its hub profile, it is concluded that the proper position of the trailer is such that both wheels of the bogie are in contact with the ground. A search for β_3 in the interval [β_{d3}, β_{u3}] is conducted until both wheels centers are on their hub profile to within 1/10 of an inch. It is concluded that this is the proper attitude of the bogie whereupon the location of the bogie center is calculated and thus the pitch angle and CG location of the trailer are determined. MOVEB then exits.

III INPUTS AND OUTPUTS

A. Vehicle Data

The data required to describe a vehicle for the Obstacle Module, OBS78B, is listed below together with the file formats required.

Most of the descriptions are self-explanatory. One should note that the equilibrium load and center of gravity location (lines 12,13) should be those of the empty vehicle. The weight and location of the payload are entered separately (line 14,15). The payload weight may be zero.

The data used to describe a tracked vehicle requires special attention. In OBS78B, the track is replaced by eight wheels, two bogie pairs on each side, as discussed in section II.A.1. In order to obtain the kind of path of motion expected at the CG, these wheels are quite large. In fact, the effective radius is the distance between the two support points if the vehicle has a girderized track and half this distance if the track is flexible. These wheels are placed on two bogie suspensions whose horizontal locations, bogie arm width and limits of angular motion are those specified in the input data file (lines 8-11). We have found that if the suspensions are too far apart the resulting enormous wheels can contact the obstacle far fore and/or aft of the vehicle resulting in false clearance information. In particular, the contact of the sprocket or idler (spridler) is not modeled in this case. If the suspensions are too close,the vehicle motion is not properly modeled. For the M60A1, placing these suspension supports over the second and next to last road wheels with the bogie arm width equal to the road wheel spacing seems to give reasonable results. To model the relative freedom of vertical motion of the first and last road wheels, the limits of angular motion are different in the clockwise and counter clockwise directions. For the M60A1, we allow the outer wheels about four times the motion toward the body of the vehicle allowed for the inner wheels.

The input file description forms Table III.A.1. The variable names are those in the program. The coordinate system for the input data is shown schematically in Fig III.A.1. An explanation of all the coordinate systems used in the Obstacle Module may be found in Section II.B, above. Sample vehicle input data files for wheeled and tracked vehicles are contained in Appendix B.

TABLE III.A.1

Vehicle Input File Format-OBS78B

Line No.	Variable Name	FORMAT	Description
1	TITLE1 TITLE2 TITLE3	A5 A5 A5	This line contains alphanumeric vehicle identification. The first 15 characters are printed in the program output.
2	NUNITS	I2	Number of units
	NSUSP	I2	Total number of suspension supports for entire vehicle
	NVEH1	I2	Vehicle type: 0-tracked 1 or greater- wheeled
	NFL	I2	Track type: 0- rigid 1- flexible
3	REFHT1	F7.2	Height of hitch above the ground when empty vehicle is at rest (in.)
	HTCHFZ	F7.2	Vertical force on hitch of trailer at rest (tongue weight) (lb.)
4	SFLAG(I) I=1,NSUSP	10I2	Suspension type at support I: 0-independent single wheel 1-bogie
5	IP(I,J) J=1,2 I=1,NSUSP	10I2	Power indicator for wheel J of support I: 0-unpowered 1-powered
6	IB(I,J) J=1,2 I=1,NSUSP	10I2	Brake indicator for wheel J of support I: 0-unbraked 1-braked
7	EFFRAD(I) I=1,NSUSP	10F7.2	Effective (loaded) radius of wheels at support I, i.e. the distance from the wheel centers to the contact point (including track thickness for a tracked vehicle)
8	ELL(I) I=1,NSUSP	10F7.2	Horizontal coordinate of suspension support point I with respect to hitch (in.)
9	BWIDTH(I) I=1,NSUSP	10F7.2	Bogie swing arm width at support I (0. If no bogie) (in.)
10	BALMU(I) I=1,NSUSP	10F7.2	Limit of angular movement in counter clockwise direction of bogie arm at support I (deg.)

TABLE III.A.1 (Continued)

Line No.	Variable Name	FORMAT	Description
11	BALMD(I) I=1,NSUSP	10F7.2	Limit of angular movement in clockwise direction of bogie arm at support I (This angle is negative if the front wheel is below the rear wheel at the extreme position) (deg.)
12	EQUILF(I) I=1,NSUSP	10F7.2	Equilibrium load on support I when vehicle is empty and at rest (If support I is a bogie, this is the sum of the loads on the two wheels of the bogie pair) (lb.)
13	CGZ1	F7.2	Vertical position from ground of center of gravity of unloaded first unit (in.)
	CGZ2	F7.2	Vertical position from ground of center of gravity of unloaded second unit (in.)
14	DEE1	F7.2	Horizontal coordinate of the first unit payload CG with respect to hitch (in.)
	ZEE1	F7.2	Vertical distance to the CG of the payload of the first unit from the ground at rest (in.)
	DEE2	F7.2	Horizontal coordinate of the trailer payload CG with respect to hitch (in.)
	ZEE2	F7.2	Vertical distance to the CG of payload of the second unit from the ground at rest (in.)
15	DELTW1	F7.2	Weight of the payload of the first unit (lb.)
	DELTW2	F7.2	Weight of the payload of the second unit (lb.)
16	NPTSC1	I2	Number of breakpoints used to describe the bottom profile of the first unit
	NPTSC2	I2	Number of breakpoints used to describe the bottom profile of the second unit
17	XCLC1(I), YCLC1(I) I=1,NPTSC1	10F7.2	Pairs of X and Z coordinates of breakpoints of the bottom profile of the first unit at equilibrium with no payload. Five pairs are entered per line, as many lines as needed (in.)

TABLE III.A.1 (Continued)

Line No.	Variable Name	FORMAT	Description

NOTE: IF A ONE UNIT VEHICLE IS BEING DESCRIBED, THE FOLLOWING LINE (18) IS SKIPPED.

Line No.	Variable Name	FORMAT	Description
18	XCLC2(I), YCLC2(I) I=1,NPTSC2	10F7.2	Pairs of X and Z coordinates of the breakpoints of the bottom profile of the second unit at equilibrium with no payload,five pairs per line with as many lines as needed (in.)

NOTE: THE FOLLOWING LINES (19 and 20) ARE INCLUDED ONLY FOR TRACKED VEHICLES.

Line No.	Variable Name	FORMAT	Description
19	SFLAG(I), IP(I),IB(I) I=4,5	6I2	Suspension type, power and brake indicator (see lines 4,5,6) for front and rear spridler (I=4,5 respectively)
20	ELL(4)	F7.2	Horizontal coordinate of center of front spridler with respect to hitch (in.)
	ZS(4)	F7.2	Vertical distance from ground to center of front spridler (in.)
	EFFRAD(4)	F7.2	Effective radius (distance from wheel center to contact point including track thickness of front spridler (in)
	ELL(5)	F7.2	Horizontal coordinate of center of rear spridler with respect to hitch (in.)
	ZS(5)	F7.2	Vertical distance from ground to center of rear spridler (in.)
	EFFRAD(5)	F7.2	Effective radius of rear spridler (in.)

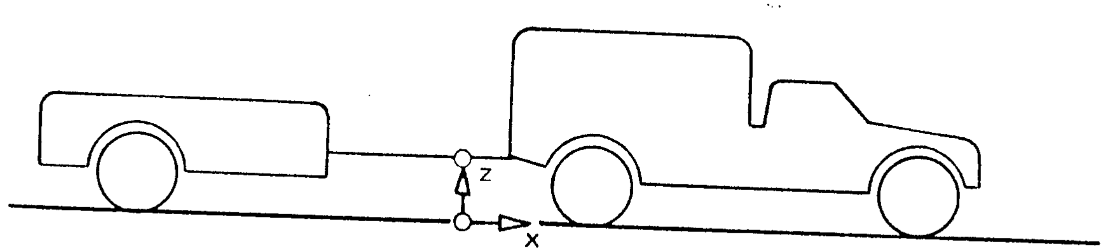


FIGURE III.A.1 -- Vehicle Input Data - Coordinate System

B. Terrain Data

Although OBS78B is currently to be used as a preprocessor, the program is designed to allow extension to in line use in the Areal Module or possible expansion to linear feature size obstacles. For these reasons, the topographic slope is included as a terrain input, although for present purposes, it should be entered as zero. In addition, data which describes the terrain vehicle interface is included as described in section III.C below.

At the present time, the obstacle modeled is a symmetric trapezoid and hence is defined by three numbers, the obstacle approach angle, height and width (see figure II.A.2). The user has the option of entering a single obstacle or a sequence of obstacles. The first line of the terrain file identifies the option selected. It is planned to extend the number of options. The value of the option identifier has been chosen to be consistent with those in data files existing at WES and TARADCOM. A sample terrain input file is contained in the Appendices.

TABLE III.B.1

Terrain File Format-OBS78B

Line No.	Variable Name	FORMAT	Description
1	LSIG	I2	Signal of data entry mode
2	GRADE	F7.2	Topographic slope (%)

NOTE: The only values currently allowed are LSIG=2 and LSIG=3. If LSIG=2,a single obstacle is expected while LSIG=3 indicates that the data contains a sequence of obstacles. If LSIG=2, the following line is skipped.

Line No.	Variable Name	FORMAT	Description
3	NANG	I2	Number of obstacle angles
	NOHGT	I2	Number of obstacle heights
	NWDTH	I2	Number of obstacle widths These three values are written in the output file for use by the Areal module. OBS78B does not need them.
4	OBH	F10.2	Obstacle height (in.)
	OBAA	F10.2	Obstacle approach angle (deg.)
	OBW	F10.2	Obstacle width (in.)

NOTE: If LSIG=3, the file should contain a line in the above format for each obstacle to be traversed. In this case, the last line of the file should contain all 9's. (The program terminates if OBH $\geq$ 99999.99)

C. Scenario/Control Data

For the nonce, variables to describe terrain/vehicle interaction and those containing control information for the computer system are read from unit LUN4 (i.e. the program contains FORTRAN "READ(LUN4,f) X" statements, with f the FORMAT label and X the variables). When the program is run interactively, the variables are entered from the terminal.

The first entry is DETAIL (FORMAT-I2), the output detail level indicator. At present the following output levels are implemented.

- 0 Only the minimum clearance, maximum force and average force for each obstacle are reported.
- 1 An additional output file is opened for detailed output. At detail level 1 or greater, the vehicle and terrain input data are echoed to this detailed output file.
- 4 In addition to the level 1 data, the clearance history is reported (i.e. the minimum clearance or maximum interference at each step in the traverse and its location on the vehicle or obstacle).
- 8 In addition to the level 4 data, intermediate calculations at the end of each major subsection (e.g. clearance computation, force balance, movement) are reported from the main program.
- 9 In addition to the above, the final computations in the movement and clearance subroutines are reported.
- 10 At this level intermediate results are reported from the subroutines as well as at the transition points selected for lower levels. This is the level normally required to debug the program. A complete report of each step is available. Care must be used as traversal of a single obstacle can produce more than 100 pages

of output at this level.

11 All level 10 output is also written at level 11 as well as a report on every call to the iterative non-linear equation solver. About 60% more output is produced than at level 10.

The final two lines are the vehicle/terrain interaction data. First is a line containing the limiting coefficient of friction for each assembly (FORMAT 3F7.2). In this edition of the Obstacle Module, this data is not used. The last line contains the rolling resistance coefficient for each assembly (FORMAT 3F7.2).

As this section is designed for interactive users, each of the READ statements is preceded by a prompt.

D. Output

The output of OBS78B consists of three files, one of which is optional. These contain control/execution information, the basic model output and detailed model output respectively. Each is described below.

1. Control/Execution Report

Several lines of output are generated for the guidance of the interactive users. These lines appear at the terminal or in a log file in the case of a batch run. The first few prompt the user to provide the scenario/control information described in the previous section. Next the first identification line of the vehicle data file is output. As each obstacle in the terrain file is completed, this is reported so that the interactive user knows how far the program has progressed. In addition, warning and error messages may be written. In particular, in certain cases an informational message is given about the error from the EQSOL subroutine although this error is relatively small and the results are satisfactory.

2. Basic Output

The final results of OBS78B are the minimum clearance (or maximum interference) between the vehicle and the obstacle during the override, the maximum propulsive force required during the override and the average propulsive force to override the obstacle. For ease in using this data as part of the vehicle data file for NRMM (see Volume I, Section III.B) the first six lines of the output file will contain the number of height values, angle values and width values from the terrain input file (section III.B), when appropriate with identifiers. Then a header is printed followed by the output and the corresponding terrain input in the format required for the vehicle data file for NRMM.

3. Detailed Output

As described before, the user of the Obstacle Module may choose to obtain an output file containing some of the results of the computations performed in modeling the override of the obstacle. The intent is to allow:

1. Verification that the input data is properly formatted and correctly read (level 1)
2. Examination of the clearance history to identify any points on the vehicle which appear to be problems (level 4)
3. Examination of the flow of computation to understand the geometry and force results and relate them to reality (level 8)
4. Generation of sufficient data to permit program verification and debugging (levels 10 and 11).

Care must be taken in selection of the output level for this program and that for the Operational Modules, NRMM, since the higher levels cause very large amounts of data to be written. We would expect levels 8 through 11 to be selected only for a single obstacle, not for runs with a multi-obstacle terrain file. An output level providing a force history is planned and several levels are unassigned to provide for expansion. Most of the output records written to the detailed output file contain an identification. These identifiers are listed in Table III.D.1 together with the subroutine from which the record is written and the output levels at which the record would appear. In the table, these identifiers are grouped by the originating subroutine and further arranged in order of placement in the program (which corresponds reasonably well to the order of appearance in the output).

Since the detailed output is intended primarily for the experienced analyst/programmer to use in uncovering anomalies, it would normally be used with a copy of the program and it is felt that the headers used as pointers to the appropriate place should suffice as labeling. The clearance data which is produced in level 4 output, however, is, hopefully, of potential use to vehicle designers and design evaluators.

This output (labeled MAINC) at each step is a line of five numbers, viz. the variables ILOC, CLRNC, CLRMIN, IDX and IDC. The first, ILOC, is the index of the step. The second is the minimum clearance or maximum interference (in inches) at that step. CLRMIN is the minimum clearance or maximum interference found at all steps from the initial position to the current position. The last two numbers, IDX and IDC are indices which contain, encrypted, the location (on vehicle or obstacle) at which CLRNC and CLRMIN respectively are obtained. As explained in section II.F.1, at each step of the obstacle traversal, clearances are checked at the obstacle breakpoints, the vehicle clearance array breakpoints and the vehicle hitch. The minimum is the reported clearance, CLRNC. If this occurs at the Nth obstacle breakpoint, the value reported in IDX is N. If the minimum occurs at the Nth breakpoint of the first unit's clearance array, the value of IDX is 10,000N. For a minimum at the Nth breakpoint of the second unit's clearance array, the value of IDX is 100N. If, finally, the minimum is found at the hitch point (which is checked separately), the value of IDX is 1,111.

TABLE III.D.1

Detailed Output Headers - OBS78B

Header	Originating Subprogram	Level	Comments
Descriptive Text	OBS78B	1 or greater	Echo of vehicle input
TERR1	OBS78B	1 or greater	Terrain input echo
NEW OBSTACLE	OBS78B	1 or greater	Terrain input echo
MBACKOFF	OBS78B	10,11	
MINIT1	OBS78B	8-11	
MINIT2	OBS78B	8-11	
MAINC	OBS78B	4,8-11	Clearance history
MAIN1	OBS78B	10,11	
MAIN2	OBS78B	10,11	
MAIN3	OBS78B	8-11	
MAIN4	OBS78B	8-11	
MAIN5	OBS78B	8-11	
MAIN7	OBS78B	1 or greater	
OBGI	OBGEOM	10,11	
----	OBGEOM	10,11	
----	OBGEOM	9-11	
K,I	OBGEOM	10,11	
----	OBGEOM	9-11	
STEP SIZE	OBGEOM	1 or greater	
CLEAR0	CLEAR	10,11	
CLEAR1	CLEAR	10,11	
CLEAR2	CLEAR	10,11	
CLEAR3	CLEAR	10,11	
04	CLEAR	10,11	
V1	CLEAR	10,11	
V2	CLEAR	10,11	
V3	CLEAR	10,11	
H1	CLEAR	10,11	
H2	CLEAR	10,11	
H3	CLEAR	10,11	
T1	CLEAR	10,11	
T2	CLEAR	10,11	
T3	CLEAR	10,11	
MIN	CLEAR	9-11	
SSQ	FORCES	10,11	
XN	FORCES	10,11	
XPH	FORCES	10,11	
X	FORCES	10,11	
Z	FORCES	10,11	
CGX(I),CGZ(I)	FORCES	10,11	
ALPHA	FORCES	10,11	
CGFX(I)	FORCES	10,11	
CGFZ(I)	FORCES	10,11	

TABLE III.D.1 (Continued)

Header	Originating Subprogram	Level	Comments
FHX,FHZ	FORCES	10,11	
SFLAG	FORCES	10,11	
NW	FORCES	10,11	
RR	FORCES	10,11	
BETAP	FORCES	10,11	
BWITH	FORCES	10,11	
BN	FORCES	10,11	
BT	FORCES	10,11	
CRR	FORCES	10,11	
CTF	FORCES	10,11	
FN	FORCES	10,11	
RF	FORCES	10,11	
TF	FORCES	10,11	
FX	FORCES	10,11	
FZ	FORCES	10,11	
PX	FORCES	10,11	
PZ	FORCES	10,11	
PM	FORCES	10,11	
MOVE2	MOVEB	10,11	
MOVE3	MOVEB	10,11	
MOVES4	MOVEB	10,11	
MOVES5	MOVEB	10,11	
MOVE11	MOVEB	10,11	
MOVE12	MOVEB	10,11	
MOVE21	MOVEB	10,11	
MOVE22	MOVEB	10,11	
MOVEA3	MOVEB	10,11	
MOVEA4	MOVEB	10,11	
MOVEA5	MOVEB	10,11	
MOVEA5A	MOVEB	10,11	
MOVEA5B	MOVEB	10,11	
MOVEA6	MOVEB	10,11	
ELEVAT1	ELEVAT	10,11	
ELEVAT2	ELEVAT	10,11	
ELEVAT3	ELEVAT	10,11	
ELEVAT4	ELEVAT	10,11	
WHEELS0	WHEEL2	11	
WHEELS1	WHEEL2	11	
WHEELS2	WHEEL2	11	
WHEEL3/1	WHEEL3	11	
WHEEL3/2	WHEEL3	11	
WHEEL3/3	WHEEL3	11	
%EQSOL:	EQSOL	11	

IV REFERENCES

American Association of State Highway Officials (AASHO), A Policy on Geometric Design of Rural Highways, 1965, The American Association of State Highway Officials, Washington, D. C., pp 152 to 159.

Jurkat, M. P., Nuttall, C. J., and Haley, P. W., "The AMC '74 Mobility Model", Technical Report No. 11921 (LL-149), May 1975, U. S. Army Tank-Automotive Command, Warren, Mich.

Lessem, A. S., "Dynamics of Wheeled Vehicles: A Mathematical Model for the Traversal of Rigid Obstacles by a Pneumatic Tire", Technical Report M-68-1, U. S. Army Waterways Experiment Station, CE, Vicksburg, Miss., May 1968.

Nuttall, C. J., Jr., and Dugoff, H. J., "A Hierarchial Structure of Models for the Analysis of Land Mobility Systems", presented at Twelfth Annual U. S. Army Operations Research Symposium, Durham, N. C., Oct 1973.

Nuttall, C. J., Jr., Rula, A. A., and Dugoff, H. J., "Computer Model for Comprehensive Evaluation of Cross-Country Vehicle Mobility", Earthmoving Industry Conference, SAE Paper No. 740426, presented at Society of Automotive Engineers, Apr 1974.

Powell, M. J. D., "A FORTRAN Subroutine for Solving Nonlinear Algebraic Equations", Numerical Methods for Nonlinear Algebraic Equations, (Editor: P. Rabinowitz), Gordon & Breach, 1970.

Pradko, F., Lee, R. A., and Kaluza, V., "Theory of Human Vibration Response", Paper 66-WA-BHF-15, presented at ASME Meeting, Nov 1966.

Rula, A. A., and Nuttall, C. J., Jr., "An Analysis of Ground Mobility Models (ANAMOB)", Technical Report M-71-4, Jul 1971, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss.

Rula, A. A. and Nuttall, C. J., Jr., "Terrain Modeling to Support Mobility Evaluation", Proceedings of Fifth International Conference, The International Society for Terrain-Vehicle Systems, Inc., Detroit, Mich., Vol 4, Jun 1975.

Smith, G., "Commercial Vehicles Performance and Fuel Economy", Society of Automotive Engineers, Special Publication 355, 1970.

Taborek, J. J., Mechanics of Vehicles, Penton Publications, Cleveland, Ohio, 1957.

U. S. ATAC, "The AMC-71 Mobility Model", Technical Report No. 11789 (LL 143), Vols. I and II, Jul 1973, U. S. Army Tank-Automotive Command, Warren, Mich.

```
C         PROGRAM OBS78B
C
C
C VEHICLE-OBSTACLE INTERFERENCE MODEL (CODING UNOPTIMIZED)
C
C DETERMINES INTERFERENCE/CLEARANCE BETWEEN 2-DIMENSIONAL
C VEHICLE PROFILE AND OBSTACLE PROFILE OF TRAPEZOIC SHAPE.
C DETERMINES TRACTION FORCE REQUIRED TO SURMOUNT. ACCOUNTS
C FOR ARTICULATION IN PITCH PLANE, BOGIES ALLOWED
C ON ALL SUSPENSIONS, BASIC ANALYSIS PROCEDURE: SOLUTION OF
C EQUATIONS OF STATIC EQUILIBRIUM FOR SEQUENTIAL PLACE-
C MENTS OF VEHICLE ON OBSTACLE TO YIELD TANGENTIAL FORCES
C AND POSITION OF VEHICLE CLEARANCE CONTOUR WITH RESPECT
C TO OBSTACLE.
C
C LOUT=DETAIL IS OUTPUT DETAIL LEVEL INDICATOR
C DETAIL  =   0  ONLY O78OUT FILE WILL BE WRITTEN
C DETAIL .GE. 1  O78DBG FILE WILL BE WRITTEN
C DETAIL  =   4  CLEARANCE HISTORY WRITTEN
C DETAIL  =   8  MAJOR SUBSECTION RESULTS
C DETAIL  =   9  SUBROUTINE TRACE
C DETAIL  =  10  ALL VARIABLES
C
C
      PROGRAM OBS78B (INPUT=150,OUTPUT=150,TAPE5=INPUT,TAPE6=OUTPUT,
     + TAPE1=150,TAPE20=150,TAPE21=150,TAPE22=150)
       COMMON ALPHA(5,2),
     + BALMC(3),BALMU(3),
     + BETA(3),BETAP(3),BN(3),BRAKER(5,2),BT(3,2),BWIDTH(3),
     + COSA(3,2),COSB(3),CCSG(3,2),CGFX(2),CGFZ(2),
     + CGX(2),CGZ(2),CGMY(2),CRR(3,2),CTF(3,2),
     + EFFRAD(5),ELL(5),
     + FHX,FHZ,FN(3,2),
     + HA(5,9),HB(5,9),HC(5,9),HD(5,9),HE(5,9),HF(5,9),
     + HFL(5,9),HX(5,10),HZ(5,10),
     + GAMMA(3,2),
     + IB(5,2),IP(5,2),IH(5,2),
     + LOUT,LUN6,
     + NSUSP,NUNITS,NW(5),NW2(5),
     + OA(9),OFL(9),OX(10),OZ(10),
     + PM(3),POWERR(5,2),PX(3),PXPCG(3),PZ(3),PZPCG(3),
     + RBC1,RBC2,RR(3,2),
     + SCALE(6),SFLAG(5),SINA(3,2),SINB(3),STEP,
     + THETB1,THETB2,
     + X(5),XPBC(5),XPW(5,2),
     + Z(5),ZPBC(5),ZPRCF(5,2),ZPW(5,2)
C
C
C
       DIMENSION
     + CAW1(15),CAW2(15),CRW1(15),CRW2(15),
     + EQUILF(5),EFTRAD(5),
     + FMU(3),
     + POW(3,2),
```

```
     +    RBC(5),RHTCH(2),RTCW(3),RWLIM(3,2),
     +    THETA(2),THETA0(5),THET0H(2),TWLIM(3,2),
     +    XCLC1(15),XCLC2(15),XN(6),XPCG(2),XPRF(20),
     +    YCLC1(15),YCLC2(15),YPRF(20),
     +    ZPCG(2),ZS(5)
C
C
          DOUBLE PRECISION VEHDAT
          INTEGER SFLAG,DETAIL
        REWIND 1
        REWIND 20
        REWIND 21
        REWIND 22
        CALL CONNEC( 5LINPUT )
        CALL CONNEC( 6LOUTPUT )
C INITIALIZATION OF I/O UNITS
C PROGRAM SUMMARY DATA
        LUN1=22
C TERRAIN OBSTACLE DATA
        LUN2=21
C VEHICLE DATA
        LUN3=20
C CONTROL INPUT FILE
        LUN4=5
C EXECUTION REPORT FILE
        LUN5=6
C DIAGNOSTICS
        LUN6=1
C
          PI=3.14159265
          PIM2=PI*2.
          PID2=PI/2.
          K1=0
          KAF=0.5
C
          WRITE(LUN5,10)
 10       FORMAT(20H PRINT OUTPUT LEVEL )
          READ(LUN4,11) DETAIL
 11       FORMAT(I2)
          WRITE(LUN5,15)
          READ(LUN4,4020) FMU(1),FMU(2),FMU(3)
          WRITE(LUN5,16)
          READ(LUN4,4020) RTCW(1),RTCW(2),RTCW(3)
 15       FORMAT(34H FRICTION COEFFICIENTS BY ASSEMBLY)
 16       FORMAT(43H ROLLING RESISTANCE COEFFCIENTS BY ASSEMBLY)
          LOUT=DETAIL
C READ IN VEHICLE DATA
C
          READ(LUN3,4000) TITLE1,TITLE2,TITLE3
          WRITE(LUN5,4000) TITLE1,TITLE2,TITLE3
 4000     FORMAT(3A5)
 4010     FORMAT(10I2)
 4020     FORMAT(10F7.2)
C
```

```
      READ(LUN3,4010)  NUNITS,NSUSP,NVEH1,NFL
      READ(LUN3,4020)  REFHT1,HTCHFZ
      READ(LUN3,4010)  (SFLAG(I),I=1,NSUSP)
      READ(LUN3,4010)  ((IF(I,J),J=1,2),I=1,NSUSP)
      READ(LUN3,4010)  ((IB(I,J),J=1,2),I=1,NSUSP)
      READ(LUN3,4020)  (EFFRAD(I),I=1,NSUSP)
      READ(LUN3,4020)  (ELL(I),I=1,NSUSP)
      READ(LUN3,4020)  (BWIDTH(I),I=1,NSUSP)
      READ(LUN3,4020)  (BALMU(I),I=1,NSUSP)
      READ(LUN3,4020)  (BALMD(I),I=1,NSUSP)
      READ(LUN3,4020)  (EQUILF(I),I=1,NSUSP)
      READ(LUN3,4020)  CGZ1,CGZ2
      CGZ1=CGZ1-REFHT1
      CGZ2=CGZ2-REFHT1
      READ(LUN3,4020)  CEE1,ZEE1,CEE2,ZEE2
      ZEE1=ZEE1-REFHT1
      ZEE2=ZEE2-REFHT1
      READ(LUN3,4020)  DELTW1,DELTW2
      READ(LUN3,4010)  NPTSC1,NPTSC2
      READ(LUN3,4020)  (XCLC1(I),YCLC1(I),I=1,NPTSC1)
      DO 80 I=1,NPTSC1
 80   YCLC1(I)=YCLC1(I)-REFHT1
      IF(NUNITS.EQ.1)GO TO 100
      READ(LUN3,4020)  (XCLC2(I),YCLC2(I),I=1,NPTSC2)
      DO 85 I=1,NPTSC2
 85   YCLC2(I)=YCLC2(I)-REFHT1
 100  CONTINUE
      IF(NVEH1.NE.0) GOTO 115
      READ(LUN3,4010)  (SFLAG(I),IF(I,1),IB(I,1),I=4,5)
      READ(LUN3,4020)  (ELL(I),ZS(I),EFFRAD(I),I=4,5)
      ZS(4)=ZS(4)-REFHT1
      ZS(5)=ZS(5)-REFHT1
 115  CONTINUE
C
C     OBS78 VEHICLE PREPROCESSOR
C
      IF(NUNITS.GE.2) GOTO 120
      HTCHFZ=0.
      EQUILF(3)=0.
      CGMY(2)=0.
      CGFX(2)=0.
      CGFZ(2)=0.
      CGX(2)=0.
      CGZ(2)=0.
 120  CGFZ1=-EQUILF(1)-EQUILF(2)
      CGX1=-(EQUILF(1)*ELL(1)+EQUILF(2)*ELL(2))/CGFZ1
      CGFZ2=-EQUILF(3)-HTCHFZ
      CGX2=0.
      IF(NSUSP .GE. 3) CGX2=-EQUILF(3)*ELL(3)/CGFZ2
      CGFZ(1)=CGFZ1-DELTW1
      CGX(1)=(CGFZ1*CGX1-DELTW1*CEE1)/CGFZ(1)
      CGZ(1)=(CGFZ1*CGZ1-DELTW1*ZEE1)/CGFZ(1)
      CGFX(1)=0.
      CGMY(1)=0.
```

```
      RHTCH(1)=SQRT(CGX(1)**2+CGZ(1)**2)
C
C FOLLOWING DISTANCES AND ANGLES WRT CG
C
      ACG=ATN2(CGZ(1),CGX(1))
      THETØH(1)=ACG+PI
C
C SET ANGLE OF VECTOR FROM CG TO HITCH BETWEEN -PI AND PI
C
      IF(THETØH(1).GE.PI) THETØH(1)=ACG-PI
      DO 122 I=1,2
      XB=ELL(I)-CGX(1)
      ZB=-REFHT1+EFFRAD(I)-CGZ(1)
      RBC(I)=SQRT(XB*XB+ZB*ZB)
      THETAØ(I)=ATN2(ZB,XB)
      RWLIM(I,1)=RBC(I)
      TWLIM(I,1)=THETAØ(I)
      RWLIM(I,2)=Ø.
      TWLIM(I,2)=Ø.
      IF(SFLAG(I).EQ.Ø) GOTO 122
      BALMU(I)=BALMU(I)*PI/180.
      BALMD(I)=BALMD(I)*PI/180.
      X1=XB+.5*BWIDTH(I)*COS(BALMU(I))
      Z1=ZB+.5*BWIDTH(I)*SIN(BALMU(I))
      X2=XB-.5*BWIDTH(I)*COS(BALMD(I))
      Z2=ZB-.5*BWIDTH(I)*SIN(BALMD(I))
      TWLIM(I,1)=ATN2(Z1,X1)
      TWLIM(I,2)=ATN2(Z2,X2)
      RWLIM(I,1)=SQRT(X1*X1+Z1*Z1)
      RWLIM(I,2)=SQRT(X2*X2+Z2*Z2)
 122  CONTINUE
      IF(NVEH1.NE. Ø) GOTO 124
      DO 123 I=4,5
      EFTRAD(I)=EFFRAD(I)
      XB=ELL(I)-CGX(1)
      ZB=ZS(I)-CGZ(1)
      RBC(I)=SQRT(XB*XB+ZB*ZB)
      THETAØ(I)=ATN2(ZB,XB)
 123  CONTINUE
 124  IF(NUNITS.EQ.1) GOTO 125
C
C ALL TRAILER DIST. AND ANGLES WRT HITCH
C
      CGFZ(2)=CGFZ2-DELTW2
      CGX(2)=(CGFZ2*CGX2-DELTW2*DEE2)/CGFZ(2)
      CGZ(2)=(CGFZ2*CGZ2-DELTW2*ZEE2)/CGFZ(2)
      CGFX(2)=Ø.
      CGMY(2)=Ø.
      RHTCH(2)=SQRT(CGX(2)**2+CGZ(2)**2)
      THETØH(2)=ATN2(CGZ(2),CGX(2))
      XHB=ELL(3)
      ZHB=-REFHT1+EFFRAD(3)
      RBC(3)=SQRT(XHB*XHB+ZHB*ZHB)
      THETAØ(3)=ATN2(ZHB,XHB)
```

```
      RWLIM(3,1)=RBC(3)
      TWLIM(3,1)=THETA0(3)
      RWLIM(3,2)=0.
      TWLIM(3,2)=0.
      IF(SFLAG(3).EQ.3) GOTO 125
      BALMU(3)=BALMU(3)*PI/180.
      BALMD(3)=BALMD(3)*PI/180.
      X1=XHB+.5*BWIDTH(3)*COS(BALMU(3))
      Z1=ZHB+.5*BWIDTH(3)*SIN(BALMU(3))
      RWLIM(3,1)=SQRT(X1*X1+Z1*Z1)
      TWLIM(3,1)=ATN2(Z1,X1)
      X2=XHB-.5*BWIDTH(3)*COS(BALMD(3))
      Z2=ZHB-.5*BWIDTH(3)*SIN(BALMD(3))
      RWLIM(3,2)=SQRT(X2*X2+Z2*Z2)
      TWLIM(3,2)=ATN2(Z2,X2)
125   CONTINUE
      DO 130 I=1,NSUSP
      EFTRAD(I)=EFFRAD(I)
      IF(NVEH1.EQ.0.AND.I.NE.3) EFTRAD(I)=.5*(ELL(1)-ELL(2))
      IF(NVEH1.EQ.0.AND.NFL.EQ.0.AND.I.NE.3)
     + EFTRAD(I)=ELL(1)-ELL(2)
      DO 130 J=1,2
      POWERR(I,J)=1.0
      BRAKFR(I,J)=1.0
      RR(I,J)=EFFRAD(I)
      CRR(I,J)=RTOW(I)
      POW(I,J)=FMU(I)
130   CONTINUE
      BPRFDL=0.
      IF(NVEH1.EQ.0) BPRFDL=EFTRAD(1)-EFFRAD(1)
      DO 135 I=1,NPTSC1
      YCLC1(I)=YCLC1(I)-BPRFDL
      IF(ABS(YCLC1(I))+ABS(XCLC1(I)).EQ.0.) GOTO 133
      CAW1(I)=ATN2(YCLC1(I),XCLC1(I))
      IF(ABS(CAW1(I)) .LE. .01) CAW1(I)=0.
      GOTO 135
133   CAW1(I)=0.
135   CRW1(I)=SQRT(XCLC1(I)**2+YCLC1(I)**2)
      IF(NUNITS.LE.1) GOTO 145
      DO 140 I=1,NPTSC2
      IF(ABS(YCLC2(I))+ABS(XCLC2(I)).EQ.0.) GOTO 138
      CAW2(I)=ATN2(YCLC2(I),XCLC2(I))
      IF(ABS(CAW2(I)) .LE. .01) CAW2(I)=0.
      GOTO 140
138   CAW2(I)=0.
140   CRW2(I)=SQRT(XCLC2(I)**2+YCLC2(I)**2)
C
C END OF VEHICLE PREPROCESSOR
C
C ECHO INPUT
C
145   IF(LOUT.EQ.0) GOTO 125
      WRITE(LUN6,5000) TITLE1,TITLE2,TITLE3,NVEH1,NFL
5000  FORMAT(1H1,37H THE FOLLOWING IS A LIST OF THE INPUT,
```

```
     +  11H VARIABLES /16H THE VEHICLE IS ,3A5/11H FIRST UNIT,
     +  28H TRACKED/WHEELED INDICATOR: ,I1,1H,,
     +  27H FLEXIBLE TRACK INDICATOR :,I2/)
        WRITE(LUN6,151) CGX1,CGZ1,CGFZ1,CGX2,CGZ2,CGFZ2,
     +  (CGX(I),CGZ(I),CGFX(I),CGFZ(I),RHTCH(I),THETØH(I),I=1,NUNITS)
151     FORMAT(6H OVPPF,6F12.3/6X,6F12.3/6X,6F12.3)
        WRITE(LUN6,5002) NUNITS,REFHT1,HTCHFZ
5002    FORMAT(11H THIS IS A ,I2,29H UNIT VEHICLE WITH THE HITCH ,
     +  F6.2,24H INCHES ABOVE THE GROUND/1X,14HHITCH LOAD IS ,F10.3)
        WRITE(LUN6,5004) NSUSP
5004    FORMAT(17H THE VEHICLE HAS ,I2,21H SUSPENSION SUPPORTS ,I2)
        WRITE(LUN6,5005)
5005    FORMAT(47H FOLLOWING IS A LIST OF SUSPENSION SUPPORT DATA,/)
        DO 160 I=1,NSUSP
        WRITE(LUN6,5006) SFLAG(I),EFFRAC(I),EFTRAD(I),ELL(I),
     +  EQUILF(I),BALMU(I),BALNC(I),BWIDTH(I),FMU(I),
     +  RTOW(I),RBC(I),THETAØ(I)
        WRITE(LUN6,5015) (IF(I,J),IE(I,J),RWLIM(I,J),TWLIM(I,J),
     +  RR(I,J),CRR(I,J),PCW(I,J),J=1,2)
5015    FORMAT(3X,2I2,2X,5F10.3/3X,2I2,2X,5F10.3)
5006    FORMAT(I3,12F10.3)
160     CONTINUE
        IF(NVEH1.NE. 0) GOTO 163
        WRITE(LUN6,5009) (SFLAG(I),IP(I,1),IB(I,1),ELL(I),
     +  ZS(I),EFFRAD(I),RBC(I),THETAØ(I),I=4,5)
5009    FORMAT(32H TRACKED VEHICLE BEING SIMULATED/2(3I3,5F10.3/))
163     CONTINUE
        WRITE(LUN6,5007) CGZ1,DEE1,ZEE1,DELTW1
5007    FORMAT(37HØFOR UNIT 1: VERT DIST HITCH TO CG = ,F7.3/
     +  13X,29HHORIZ DIST HITCH TO PAYLOAD= ,F7.3/
     +  13X,29H VERT DIST HITCH TO PAYLOAD= ,F7.3/
     +  13X,10H PAYLOAD= ,F7.3)
        WRITE(LUN6,5010) RAF
5010    FORMAT(35H THE REBOUND ATTENUATION FACTOR IS ,F5.2,/)
        WRITE(LUN6,5011) NPTSC1
5011    FORMAT(10H THERE ARE,I3,22H POINTS ON THE VEHICLE
     +  18H CLEARANCE CONTOUR,/)
        DO 165 I=1,NPTSC1
        WRITE(LUN6,5012) I,XCLC1(I),I,YCLC1(I),CAW1(I),CRW1(I)
5012    FORMAT(7H XCLC1(,I2,3H) =,F8.2,2X,6HYCLC1(,I2,3H) =,F8.2,
     +  2F10.3)
165     CONTINUE
        IF(NUNITS.EQ.1) GOTO 175
        WRITE(LUN6,5013) CGZ2,DEE2,ZEE2,DELTW2
5013    FORMAT(18HØFOR UNIT 2: CGZ= ,F7.3/
     +  13X,29HHORIZ DIST HITCH TO PAYLOAD= ,F7.3/
     +  13X,29H VERT DIST HITCH TO PAYLOAD= ,F7.3/
     +  13X,10H PAYLOAD= ,F7.3/1X,2F10.3)
        WRITE(LUN6,5014) NPTSC2
5014    FORMAT(10H THERE ARE,I3,23H POINTS ON THE 2ND UNIT
     +  18H CLEARANCE CONTOUR,/)
        DO 170 I=1,NPTSC2
        WRITE(LUN6,5016) I,XCLC2(I),I,YCLC2(I),CAW2(I),CRW2(I)
5016    FORMAT(7H XCLC2(,I2,3H) =,F8.2,2X,6HYCLC2(,I2,3H) =,
```

```
     +   F8.2,2F1Ø.3)
 17Ø     CONTINUE
C
C THIS PROGRAM DOES NOT HAVE CLASS INTERVAL OBSTACLES
C
C READ IN TERRAIN DATA
C
 175     CONTINUE
         NOBST=Ø
         READ(LUN2,4Ø1Ø) LSIG
         READ(LUN2,4Ø2Ø) GRADE
         SLOPE=ATAN(GRADE/1ØØ.)
         CSLOPE=COS(SLOPE)
         SSLOPE=SIN(SLOPE)
         IF(LCUT.GE. 1) WRITE(LUN6,5Ø18) LSIG,GRADE,SLOPE,
     +   CSLOPE,SSLOPE
 5Ø18    FORMAT(6HØTERR1,I2,4F1Ø.3)
         IF(LSIG.EQ.1)GO TO 2ØØ
         IF(LSIG.EQ.2)GO TO 185
         IF(LSIG.EQ.3)GO TO 18Ø
         WRITE(LUN1,5Ø17)
 5Ø17    FORMAT(19H TERRAIN FILE ERROR)
         CALL EXIT
 18Ø     READ(LUN2,4Ø4Ø) NANG,NOHGT,NWDTH
 4Ø4Ø    FORMAT(3(8X,I2))
C
C OBSTACLE LOOP
C
 185     READ(LUN2,4Ø5Ø) OBH,OBAA,OBW
 4Ø5Ø    FORMAT(3F1Ø.2)
         IF(OBH.GE.99999.99) CALL EXIT
         RAD=OBAA*PI/18Ø.
         IF(ABS(SLOPE)+ABS(18Ø.-OBAA)*PI/18Ø..LT.PID2) GOTO 195
         WRITE(LUN1,191) OBH,OBAA,OBW,GRADE
 191     FORMAT(5ØH OBSTACLE ANGLE-GRADE COMBINATION EXCEEDS VERTICAL,
     +   /4F1Ø.3)
         GOTO 185
 195     IF(18Ø.-OBAA .LT. Ø.) OBH=-ABS(OBH)
         IF(LCUT.GE. 1) WRITE(LUN6,4Ø3Ø) OBH,OBAA,OBW
 4Ø3Ø    FORMAT(13H1NEW OBSTACLE,4F1Ø.2)
         GO TO 21Ø
C
C READ OR CALCULATE OBSTACLE PROFILE BREAKPOINTS
C
 2ØØ     READ(LUN2,4Ø1Ø) NPTSPR
         NTOTAL=Ø
         IF(NPTSPR.EQ.99) CALL EXIT
         READ(LUN2,4Ø2Ø) (XPRF(I),YPRF(I),I=1,NPTSPR)
         WRITE(LUN1,4Ø35) LSIG
 4Ø35    FORMAT(42H WRONG DATA MODE FOR OBSTACLE DESCRIPTION ,I8)
         CALL EXIT
C
C CALCULATE OBSTACLE AND HUE PROFILE
C
```

```
 210     CALL OBGEOM (BWIDTH,BFTRAD,ELL,HA,HB,HC,HD,HE,HF,HFL,
     +   HX,HZ,LOUT,LUN6,NSUSP,NUNITS,NVEH1,OA,OBAA,OBH,OBW,OFL,
     +   OX,OZ,SFLAG,SLOPE,STEP)
C
C STARTING POINTS FOR EQ. SOLVER
C
         XN(1)=RTOW(1)
         XN(4)=0.
         N1=NSUSP+1
         DO 215 I=2,N1
         IM1=I-1
 215     XN(I)=EQUILF(IM1)*(DELTW1+DELTW2)/FLOAT(NSUSP)
         XN(5)=0.
         XN(6)=HTCHFZ
C
C INITIALIZE STORAGE
C
         NW(3)=0
         NW(4)=0
         NW(5)=0
         DO 216 I=1,5
 216     NW2(I)=0
         CLRMIN=1000.
         FOCMAX=0.
         FOC=0.
C
C CALCULATE INITIAL POSITION
C
C FIRST SUSPENSION
C
         C=-HE(1,1)/HFL(1,1)
         S=HD(1,1)/HFL(1,1)
         XPW(1,1)=HX(1,2)-.1*C
         ZPW(1,1)=HZ(1,2)-.1*S
         NW(1)=0
         IF(SFLAG(1).EQ.1) GOTO 220
C
C FIRST SUSPENSION BOGIE CENTER
C
 218     XPBC(1)=XPW(1,1)
         ZPBC(1)=ZPW(1,1)
         GOTO 230
C
C FIRST SUSPENSION BOGIE
C
 220     XPW(1,2)=XPW(1,1)-BWIDTH(1)*C
         ZPW(1,2)=ZPW(1,1)-BWIDTH(1)*S
         XTEMP=XPW(1,1)-XPW(1,2)
         ZTEMP=ZPW(1,1)-ZPW(1,2)
         BETA(1)=ATN2(ZTEMP,XTEMP)
         XPBC(1)=.5*(XPW(1,1)+XPW(1,2))
         ZPBC(1)=.5*(ZPW(1,1)+ZPW(1,2))
C
C LOCATE FIRST UNIT CG FROM FIRST SUSPENSION
```

```
C
 230    THETA(1)=ATN2(HD(1,1),-HE(1,1))
        IF(THETA(1).LE..01) THETA(1)=0.
        XPCG(1)=XPBC(1)-RBC(1)*COS(THETA0(1)+THETA(1))
        ZPCG(1)=ZPBC(1)-RBC(1)*SIN(THETA0(1)+THETA(1))
        XPBC(2)=XPCG(1)+RBC(2)*COS(THETA0(2)+THETA(1))
        ZPBC(2)=ZPCG(1)+RBC(2)*SIN(THETA0(2)+THETA(1))
C
C CHECK IF TRACKED
C
        IF(NVEH1.NE. 0) GOTO 235
C
C CHECK FRONT SPROCKET/IDLER INTERFERENCE
C
        XPS=XPCG(1)+RBC(4)*COS(THETA0(4)+THETA(1))
        ZPS=ZPCG(1)+RBC(4)*SIN(THETA0(4)+THETA(1))
        CALL WHEEL3 (E,HA,HD,HE,HF,HX,IH(4,1),4,LOUT,LUN6,
     +  XPS,ZPS,ZPROF(4,1))
        IF(E.GE.-.1) GOTO 235
C
C INTERFERENCE - BACKOFF FIRST WHEEL - ASSUME MOUND
C
        S1=S/C
        S2=(OZ(4)-OZ(2))/(OX(4)-OX(2))
        RISQ=(S1**2+1.)*(ZPS-HZ(4,2)+S2*(HX(4,2)-XPS))**2/(S1-S2)**2
        RI=SQRT(RISQ)
        XPW(1,1)=XPW(1,1)-RI*C
        ZPW(1,1)=ZPW(1,1)-RI*S
        IF(LOUT.GE.10) WRITE(LUN6,236) XPS,ZPS,E,IH(4,1),S1,S2,
     +  RISQ,RI,XPW(1,1),ZPW(1,1)
 236    FORMAT(9H MBACKOFF,3E10.3,I3,6F10.3)
        IF(SFLAG(1).EQ.1) GOTO 220
        GOTO 218
C
C SECOND SUSPENSION
C
 235    NW(2)=0
        IF(SFLAG(2).EQ.1) GOTO 240
C
C SECOND SUSPENSION SINGLE WHEEL
C
        XPW(2,1)=XPBC(2)
        ZPW(2,1)=ZPBC(2)
        GOTO 250
C
C SECOND SUSPENSION BOGIE
C
 240    XPW(2,1)=XPBC(2)+.5*BWIDTH(2)*COS(THETA(1))
        ZPW(2,1)=ZPBC(2)+.5*BWIDTH(2)*SIN(THETA(1))
        XPW(2,2)=XPBC(2)-.5*BWIDTH(2)*COS(THETA(1))
        ZPW(2,2)=ZPBC(2)-.5*BWIDTH(2)*SIN(THETA(1))
        XTEMP=XPW(2,1)-XPW(2,2)
        ZTEMP=ZPW(2,1)-ZPW(2,2)
        BETA(2)=ATN2(ZTEMP,XTEMP)
```

```
C
C LOCATE HITCH
C
 250      XPH=XPCG(1)+RHTCH(1)*COS(THETØH(1)+THETA(1))
          ZPH=ZPCG(1)+RHTCH(1)*SIN(THETØH(1)+THETA(1))
          IF(NUNITS.EQ.1) GOTO 280
C
C SECOND UNIT - LOCATE WHEEL/BOGIE CENTER
C
          THETA(2)=THETA(1)
          RSQ=RBC(3)**2
          CALL WHEEL2 (EFFRAD,HA,HD,HE,HF,HX,HZ,1,IH(3,1),
     +    3,LOUT,LUN6,OX,OZ,ALPHA(3,1),RBC(3),RSQ,XPH,
     +    XPBC(3),ZPH,ZPBC(3))
          NW(3)=0
          IF(SFLAG(3).EQ.1) GOTO 260
C
C THIRD SUSPENSION SINGLE WHEEL
C
          XPW(3,1)=XPBC(3)
          ZPW(3,1)=ZPBC(3)
          GOTO 270
C
C THIRD SUSPENSION BOGIE
C
 260      XPW(3,1)=XPBC(3)+.5*BWIDTH(3)*COS(THETA(2))
          ZPW(3,1)=ZPBC(3)+.5*BWIDTH(3)*SIN(THETA(2))
          XPW(3,2)=XPBC(3)-.5*BWIDTH(3)*COS(THETA(2))
          ZPW(3,2)=ZPBC(3)-.5*BWIDTH(3)*SIN(THETA(2))
          XTEMP=XPW(3,1)-XPW(3,2)
          ZTEMP=ZPW(3,1)-ZPW(3,2)
          BETA(3)=ATN2(ZTEMP,XTEMP)
 270      XPCG(2)=XPH+RHTCH(2)*COS(THETØH(2)+THETA(2))
          ZPCG(2)=ZPH+RHTCH(2)*SIN(THETØH(2)+THETA(2))
 280      DO 290 I=1,NSUSP
          ALPHA(I,1)=THETA(1)
          IF(SFLAG(I).EQ.0) GOTO 290
          ALPHA(I,2)=THETA(1)
 290      CONTINUE
          ILOC=0
          IF(LOUT.GE.8) WRITE(LUN6,291) XPH,ZPH,(XPCG(I),
     +    ZPCG(I),THETA(I),I=1,NUNITS)
 291      FORMAT(7H MINIT1,8F10.3)
          IF(LOUT.GE.8) WRITE(LUN6,296) (XPBC(I),ZPBC(I),NW(I),
     +    (XPW(I,J),ZPW(I,J),ALPHA(I,J),J=1,2),I=1,NSUSP)
 296      FORMAT(7H MINIT2,2F10.3,I3,6F10.3/2(7X,2F10.3,I3,6F10.3/))
C
C VEHICLE MOVEMENT LOOP
C
C CALCULATE CLEARANCE
C
 300      ILOC=ILOC+1
          CALL CLEAR (CAW1,CAW2,CRW1,CRW2,IDX,LOUT,
     +    LUN6,CLRNC,NPTSC1,NPTSC2,NUNITS,OX,OZ,THETA,XPH,ZPH)
```

```
      IF(CLRNC.GE.CLRMIN) GOTO 310
      IDXCLR=IDX
      LOCATC=ILOC
      CLRMIN=CLRNC
  310 IF((LOUT.EQ.4).OR.(LOUT.GE.8)) WRITE(LUN6,311) ILOC,CLRNC,
     + CLRMIN,IDX,IDXCLR
  311 FORMAT(6H MAINC,I5,2F10.3,2I10)
C
C CALCULATE FORCES UNDER WHEELS
C
      CGX(1)=XPCG(1)-XPH
      CGZ(1)=ZPCG(1)-ZPH
      IF(NUNITS.EQ.1) GOTO 320
      CGX(2)=XPCG(2)-XPH
      CGZ(2)=ZPCG(2)-ZPH
  320 IF(LOUT.GE.10) WRITE(LUN6,326) CGX(1),CGZ(1),
     + CGX(2),CGZ(2)
  326 FORMAT(6H MAIN1,4F10.3)
      IF(SFLAG(1).EQ.1) BETAP(1)=BETA(1)+THETA(1)
      IF(SFLAG(2).EQ.1) BETAP(2)=BETA(2)+THETA(1)
      IF(NSUSP.GE.3.AND.SFLAG(3).EQ.1) BETAP(3)=BETA(3)+THETA(2)
      DO 340 I=1,5
      X(I)=XPBC(I)-XPH
      Z(I)=ZPBC(I)-ZPH
      IF(LOUT.GE.10) WRITE(LUN6,336) X(I),Z(I)
  336 FORMAT(6H MAIN2,4F10.3)
  340 CONTINUE
      CALL FORCES (XN,MAXC,NTCTAL,SSC,XPH,ZPH)
C
C CAPTURE OUTPUT
C
      FSUM=0.
      DO 350 I=1,NSUSP
      DO 355 J=1,2
      FSUM=FSUM+FN(I,J)*CTF(I,J)
      IF(LOUT.GE.8) WRITE(LUN6,351) ILOC,FSUM,
     + FN(I,J),CTF(I,J)
  355 CONTINUE
  350 CONTINUE
  351 FORMAT(6H MAIN3,I3,7F12.3)
      IF(FSUM.LE.FUOMAX) GOTO 360
      LOCATF=ILOC
      FUOMAX=FSUM
  360 IF(FSUM.LT.0.) FSUM=RAF*FSUM
      FOO=FOO+FSUM
      IF(SSQ.GT.100.) GOTO 981
C
C ADVANCE VEHICLE
C
      CALL MOVEB (CSLOPE,NBCL,NVEH1,RBC,
     + REFHT1,RHTCH,RWLIM,SSLOPE,SSQM,THETA,THETA0,THET0H,TWLIM,
     + XPCG,XPH,ZPCG,ZPH)
      IF(SSQM.GT.100.) GOTO 983
      IF(LOUT.GE.8) WRITE(LUN6,366) XPH,ZPH,(XPCG(I),ZPCG(I),
```

```
     +     THETA(I),I=1,NUNITS)
 366       FORMAT(6H MAIN4,8F10.3)
           IF(LCUT.LT.8) GOTO 390
           DO 380 I=1,NSUSP
           WRITE(LUN6,371) I,SFLAG(I),NW(I),XPBC(I),ZPBC(I),BETA(I),
     +     (XPW(I,J),ZPW(I,J),ALPHA(I,J),IH(I,J),J=1,2)
 371       FORMAT(6H MAIN5,3I3,3F10.3,2(3F10.3,I3))
 380       CONTINUE
 390       IF(XPW(1,1).LE.HX(1,10)) GOTO 300
C
C END OF VEHICLE MOVEMENT LOOP
C
           FOO=FOO/FLOAT(ILOC)
           IF(LCUT.GT.0) WRITE(LUN6,811) LOCATC,CLRMIN,
     +     IDXCLR,LOCATF,FOCMAX,FOO
 811       FORMAT(6H MAIN7,I5,F10.3,I10/6X,I5,2F10.3)
C
C WRITE AMC'74 AREAL MODULE INPUT FILE
C
           IF(LSIG.EQ.1) GOTO 989
           IF(LSIG.EQ.2)GO TO 991
           IF(K1.EQ.1) GOTO 995
           WRITE(LUN1,9076) NOHGT,NANG,NWDTH
 9076      FORMAT(5HNOHGT,/,5X,I2,/,4HNANG,/,5X,I2,/,5HNWDTH,/,5X,I2)
 991       K1=1
           WRITE(LUN1,9071)
 9071      FORMAT(/1X,6HCLRMIN,4X,6HFOCMAX,4X,3HFOO,7X,6HHOVALS,
     +     4X,5HAVALS,5X,5HWVALS)
           WRITE(LUN1,9072)
 9072      FORMAT(1X,6HINCHES,4X,6HPOUNDS,4X,6HPOUNDS,4X,6HINCHES,
     +     4X,7HRADIANS,3X,6HINCHES)
 995       CONTINUE
           IF(LSIG.EQ.1)GO TO 984
           OBH=ABS(OBH)
 981       WRITE(LUN1,9073) CLRMIN,FOCMAX,FOO,OBH,RAD,OBW
 9073      FORMAT(1X,F6.2,1X,F9.1,1X,F9.1,4X,F6.2,4X,F6.2,3X,F7.2)
           IF(SSQ.GT.100.) WRITE(LUN1,982)
 982       FORMAT(1H+,60X,39H EQSOL CANNOT SOLVE FORCE & MOMENT EQS.)
           GO TO 983
 989       IF(K1.EQ.1)GO TO 984
           K1=1
           WRITE(LUN1,9077)
 9077      FORMAT(/1X,6HCLRMIN,4X,6HFOCMAX,4X,3HFOO)
           WRITE(LUN1,9078)
 9078      FORMAT(1X,6HINCHES,4X,6HPOUNDS,4X,6HPOUNDS)
 984       WRITE(LUN1,9079) CLRMIN,FOCMAX,FOO
 9079      FORMAT(1X,F6.2,1X,F9.1,1X,F9.1)
 983       NOBST=NOBST+1
           WRITE(LUN5,985) NOBST
 985       FORMAT(1X,19H END OF OBSTACLE # ,I3)
           IF(LSIG.EQ.1) GOTO 200
           IF(LSIG.EQ.2) CALL EXIT
           IF(LSIG.EQ.3)GO TO 185
C
```

```
C END OF OBSTACLE LOOP
C
      END
C
C
      SUBROUTINE OBGEOM (BWIDTH,EFFRAD,ELL,HA,HB,HC,HD,HE,HF,HFL,
     + HX,HZ,LOUT,LUN6,NSUSP,NUNITS,NVEH1,OA,OBAA,OBH,OBW,OFL,
     + OX,OZ,SFLAG,SLOPE,STEP)
C
      INTEGER SFLAG
      DIMENSION BWIDTH(3),EFFRAD(5),ELL(5),HA(5,9),HB(5,9),
     + HC(5,9),HD(5,9),HE(5,9),HF(5,9),HFL(5,9),HX(5,10),HZ(5,10),
     + OA(9),OFL(9),OX(10),OZ(10),SFLAG(3)
C
C     OBSTACLE AND HUB BREAK POINTS BEFORE MAIN SLOPE
C
      OANG=(180.-OBAA)*3.14159265/180.
      CANG2=COS(OANG/2.)
      SANG2=SIN(OANG/2.)
      TANG2=SANG2/CANG2
      CANG=COS(OANG)
      SANG=SIN(OANG)
      TANG=SANG/CANG
      WA=OBW+2.*OBH/TANG
      RUNL=ELL(1)-ELL(NSUSP)
      IF(SFLAG(1).EQ.1) RUNL=RUNL+BWIDTH(1)/2.
      IF(SFLAG(NSUSP).EQ.1) RUNL=RUNL+BWIDTH(NSUSP)/2.
      IF(LOUT.GE.10) WRITE(LUN6,121) CANG,OBH,OBW,
     + SANG,CANG,TANG,WA,SLOPE,RUNL
 121  FORMAT(5H OBG1,9F10.3)
      IF(OANG.LT.0.) GOTO 1300
C
C     MOUND
C     SET OBSTACLE POINTS
C
      OX(1)=-RUNL-EFFRAD(1)*TANG2-1.
      IF(NVEH1.EQ.0) OX(1)=OX(1)+ELL(1)-ELL(4)
      OZ(1)=0.
      OX(2)=0.
      OZ(2)=0.
      OX(3)=0.
      OZ(3)=0.
      OX(4)=OBH/TANG
      OZ(4)=OBH
      OX(5)=OX(4)
      OZ(5)=OBH
      OX(6)=WA-OX(4)
      OZ(6)=OBH
      OX(7)=OX(6)
      OZ(7)=OBH
      OX(8)=WA
      OZ(8)=0.
      OX(9)=WA
      OZ(9)=0.
```

```
      OX(10)=WA+RUNL+EFFRAD(NSUSP)*TANG2
      IF(NVEH1.EQ.0) OX(10)=OX(10)+ELL(NSUSP)-ELL(5)
      OZ(10)=0.
C
C     SET HUB PROFILE POINTS
C
      DO 1200 K=1,5
      IF(K.GT.NSUSP.AND.NVEH1.NE.0) GOTO 1200
      IF(K.EQ.3.AND.NUNITS.EQ.1) GOTO 1200
      RK=EFFRAD(K)
      HX(K,1)=OX(1)
      HZ(K,1)=RK
      HX(K,5)=OX(5)
      HZ(K,5)=OBH+RK
      HX(K,6)=OX(6)
      HZ(K,6)=OBH+RK
      HX(K,10)=OX(10)
      HZ(K,10)=RK
C
      HZ(K,4)=OZ(4)+RK*CANG
      IF(HZ(K,4).LT.RK) GOTO 1100
      HX(K,4)=OX(4)-RK*SANG
      HX(K,3)=OX(3)-RK*TANG2
      HZ(K,3)=RK
      HX(K,2)=HX(K,3)
      HZ(K,2)=RK
      HX(K,7)=OX(7)+RK*SANG
      HZ(K,7)=OZ(7)+RK*CANG
      HX(K,8)=OX(8)+RK*TANG2
      HZ(K,8)=RK
      HX(K,9)=HX(K,8)
      HZ(K,9)=RK
      GOTO 1200
 1100 HX(K,4)=OX(4)-SQRT(2.*RK*OBH-OBH*OBH)
      HZ(K,4)=RK
      HX(K,3)=HX(K,4)
      HZ(K,3)=HZ(K,4)
      HX(K,2)=HX(K,3)
      HZ(K,2)=HZ(K,3)
      HX(K,7)=OX(6)+SQRT(2.*RK*OBH-OBH*OBH)
      HZ(K,7)=RK
      HX(K,8)=HX(K,7)
      HZ(K,8)=RK
      HX(K,9)=HX(K,8)
      HZ(K,9)=RK
 1200 CONTINUE
      GOTO 1800
C
C     DITCH
C     SET OBSTACLE POINTS
C
 1300 OX(1)=-RUNL-1.
      OZ(1)=0.
      OX(2)=0.
```

```
      OZ(2)=0.
      OX(3)=0.
      OZ(3)=0.
      OX(4)=OBH/TANG
      OZ(4)=OBH
      OX(5)=OX(4)
      OZ(5)=OBH
      OX(6)=WA-OBH/TANG
      OZ(6)=OBH
      OX(7)=OX(6)
      OZ(7)=OBH
      OX(8)=WA
      OZ(8)=0.
      OX(9)=WA
      OZ(9)=0.
      OX(10)=WA+RUNL+1.
      OZ(10)=0.
C
C     SET HUB PROFILE
C
      DO 1700 K=1,5
      IF(K.GT.NSUSP.AND.NVEH1.NE.0) GOTO 1700
      IF(K.EQ.3.AND.NUNITS.EQ.1) GOTO 1700
      RK=EFFRAD(K)
      HX(K,1)=OX(1)
      HZ(K,1)=RK
      HX(K,2)=0.
      HZ(K,2)=RK
      HX(K,9)=WA
      HZ(K,9)=RK
      HX(K,10)=OX(10)
      HZ(K,10)=RK
      HX(K,3)=OX(3)-RK*SANG
      HX(K,8)=OX(8)+RK*SANG
      IF(HX(K,3).LT.HX(K,8)) GOTO 1400
C
C  CASE 1 - WHEEL TOUCHES OBSTACLE POINTS 3 AND 8
C
      HX(K,3)=.5*(OX(3)+OX(8))
      HX(K,4)=HX(K,3)
      HX(K,5)=HX(K,3)
      HX(K,6)=HX(K,3)
      HX(K,7)=HX(K,3)
      HX(K,8)=HX(K,3)
      HZ(K,3)=SQRT(RK*RK-(HX(K,3)-HX(K,2))**2)
      HZ(K,4)=HZ(K,3)
      HZ(K,5)=HZ(K,3)
      HZ(K,6)=HZ(K,3)
      HZ(K,7)=HZ(K,3)
      HZ(K,8)=HZ(K,3)
      GOTO 1700
 1400 HZ(K,3)=OZ(3)+RK*CANG
      IF(HZ(K,3).GT.OBH+RK) GOTO 1500
C
```

```
C CASE 2 - WHEEL TOUCHES POINT 3 AND BOTTOM
C
      HX(K,3)=HX(K,2)+SQRT(-2.*RK*OBH-OBH*OBH)
      HZ(K,3)=RK+OBH
      HX(K,4)=HX(K,3)
      HZ(K,4)=HZ(K,3)
      HX(K,5)=HX(K,3)
      HZ(K,5)=HZ(K,3)
      HX(K,8)=HX(K,9)-SQRT(-2.*RK*OBH-OBH*OBH)
      HZ(K,8)=HZ(K,3)
      HX(K,7)=HX(K,8)
      HZ(K,7)=HZ(K,8)
      HX(K,6)=HX(K,8)
      HZ(K,6)=HZ(K,8)
      GOTO 1700
 1500 HZ(K,8)=HZ(K,3)
      HX(K,4)=OX(4)-RK*TANG2
      HX(K,7)=OX(7)+RK*TANG2
      IF(HX(K,4).LT.HX(K,7)) GOTO 1600
C
C CASE 3 - WHEEL TOUCHES BOTH SLOPES BEFORE BOTTOM
C
      HX(K,4)=(OX(5)+OX(6))/2.
      HX(K,5)=HX(K,4)
      HX(K,6)=HX(K,4)
      HX(K,7)=HX(K,4)
      HZ(K,4)=.5*(HZ(K,3)+HZ(K,8)+(HX(K,8)-HX(K,3))*TANG)
      HZ(K,5)=HZ(K,4)
      HZ(K,6)=HZ(K,4)
      HZ(K,7)=HZ(K,4)
      GOTO 1700
C
C CASE 4 - WHEEL TOUCHES SLOPES AND BOTTOM
C
 1600 HX(K,5)=HX(K,4)
      HX(K,6)=HX(K,7)
      HZ(K,4)=RK+OBH
      HZ(K,5)=HZ(K,4)
      HZ(K,6)=HZ(K,4)
      HZ(K,7)=HZ(K,4)
 1700 CONTINUE
 1800 IF(LCUT.GE.10) WRITE(LUN6,1900) (OX(I),I=1,10),(OZ(I),I=1,10),
     + ((HX(K,I),I=1,10),(HZ(K,I),I=1,10),K=1,5)
 1900 FORMAT(/8(1X,10F10.3/))
C
C     TRANSFORM PROFILES FOR SLOPE
C
      DO 2000 I=1,10
      RP=SQRT(OX(I)**2+OZ(I)**2)
      PHI=ATN2(OZ(I),OX(I))
      OX(I)=RP*COS(PHI+SLOPE)
      OZ(I)=RP*SIN(PHI+SLOPE)
      DO 2000 K=1,5
      IF(K.GT.NSUSP.AND.NVBH1.NE.0) GOTO 2000
```

```
      IF(K.EQ.3.AND.NUNITS.EQ.1) GOTO 2000
      RP=SQRT(HX(K,I)**2+HZ(K,I)**2)
      PHI=ATN2(HZ(K,I),HX(K,I))
      IF(ABS(PHI).LE..01) PHI=0.
      HX(K,I)=RP*COS(PHI+SLOPE)
      HZ(K,I)=RP*SIN(PHI+SLOPE)
 2000 CONTINUE
      IF(LCUT.GE.9) WRITE(LUN6,1900) (OX(I),I=1,10),(OZ(I),I=1,10),
     + ((HX(K,I),I=1,10),(HZ(K,I),I=1,10),K=1,5)
      DO 2010 I=1,9
 2010 OFL(I)=SQRT((OX(I+1)-OX(I))**2+(OZ(I+1)-OZ(I))**2)
      DO 2150 K=1,5
      IF(K.GT.NSUSP.AND.NVEH1.NE.0) GOTO 2150
      IF(K.EQ.3.AND.NUNITS.EQ.1) GOTO 2150
      RK=EFFRAD(K)
      IF(OANG.LT.0.) GOTO 2100
C
C     MOUND
C
      DO 2060 I=1,9
      IF((I.EQ.4).OR.(I.EQ.6)) GOTO 2040
 2030 HFL(K,I)=SQRT((HX(K,I+1)-HX(K,I))**2 +
     + (HZ(K,I+1)-HZ(K,I))**2 )
      GOTO 2060
C
C     ELEMENT OF ARC
C
 2040 IF((HX(K,I+1).EQ.HX(K,I)).AND.(HZ(K,I+1).EQ.
     + HZ(K,I))) GOTO 2030
      SPROD=(HX(K,I+1)-OX(I+1))*(HX(K,I)-OX(I))+
     + (HZ(K,I+1)-OZ(I+1))*(HZ(K,I)-OZ(I))
      ANGLE=ACOS(SPROD/(RK*RK))
      HFL(K,I)=RK*ANGLE
 2060 CONTINUE
      GOTO 2150
C
C     DITCH
C
 2100 CONTINUE
      DO 2140 I=1,9
      IF((I.EQ.2).OR.(I.EQ.8)) GOTO 2130
 2110 HFL(K,I)=SQRT((HX(K,I+1)-HX(K,I))**2+(HZ(K,I+1)-HZ(K,I))**2)
      GOTO 2140
C
C     ELEMENT OF ARC
C
 2130 IF((HX(K,I+1).EQ.HX(K,I)).AND.(HZ(K,I+1).EQ.
     + HZ(K,I))) GOTO 2110
      SPROD=(HX(K,I+1)-OX(I+1))*(HX(K,I)-OX(I))+
     + (HZ(K,I+1)-OZ(I+1))*(HZ(K,I)-OZ(I))
      ANGLE=ACOS(SPROD/(RK*RK))
      HFL(K,I)=RK*ANGLE
      IF(LCUT.GE.10) WRITE(LUN6,2145) K,I,HX(K,I),HX(K,I+1),
     + OX(I),OX(I+1),HZ(K,I),HZ(K,I+1),OZ(I),OZ(I+1),RK,SPROD
```

```
 2140   CONTINUE
 2150   CONTINUE
 2145   FORMAT(5H K,I ,2X,2I3,6H HX    ,2(2X,F12.3),2X,6H OX    ,
     +  2(2X,F12.3),/,6H HZ    ,2(2X,F12.3),6H OZ    ,2(2X,F12.3),
     +  10H RK,SPROD ,2(2X,F12.3))
C
C       DEFINITION OF OBSTACLE ELEMENTS
C       OA - ANGLE BETWEEN ELEMENT AND HORIZONTAL
C
        OA(1)=SLOPE
        OA(2)=0.
        OA(3)=SLOPE+OANG
        OA(4)=0.
        OA(5)=SLOPE
        OA(6)=0.
        OA(7)=SLOPE-OANG
        OA(8)=0.
        OA(9)=SLOPE
C
C       DEFINITION OF HUE ELEMENTS BY QUADRATIC
C
        DO 2300 K=1,5
        IF(K.GT.NSUSP.AND.NVEH1.NE.0) GOTO 2300
        IF(K.EQ.3.AND.NUNITS.EQ.1) GOTO 2300
        RK=EFFRAD(K)
        DO 2280 I=1,9
        IF(HFL(K,I).EQ.0.) GOTO 2220
        IF(OFL(I).EQ.0.) GOTO 2250
C
C       ELEMENT IS LINE SEGMENT
C
        HA(K,I)=0.
        HB(K,I)=0.
        HC(K,I)=0.
        HD(K,I)=HZ(K,I+1) - HZ(K,I)
        HE(K,I)= - (HX(K,I+1) - HX(K,I))
        HF(K,I)= - (HD(K,I) * HX(K,I) + HE(K,I) * HZ(K,I))
        GOTO 2280
C
C       ELEMENT IS POINT
C
 2220   HA(K,I)=0.
        HB(K,I)=0.
        HC(K,I)=0.
        HD(K,I)=0.
        HE(K,I)=0.
        HF(K,I)=0.
        GOTO 2280
C
C       ELEMENT IS ARC
C
 2250   HA(K,I)=1.
        HB(K,I)=0.
        HC(K,I)=1.
```

```
      HD(K,I)= - 2.* OX(I)
      HE(K,I)= - 2.* OZ(I)
      HF(K,I)=OX(I) * OX(I) + OZ(I) * OZ(I) - RK * RK
 2280 CONTINUE
 2300 CONTINUE
      IF(LCUT.GE.9) WRITE(LUN6,2500) (OFL(I),I=1,9),(OA(I),I=1,9),
     + ((HFL(K,I),I=1,9),(HA(K,I),I=1,9),(HB(K,I),I=1,9),
     + (HC(K,I),I=1,9),(HD(K,I),I=1,9),(HE(K,I),I=1,9),
     + (HF(K,I),I=1,9),K=1,5)
 2500 FORMAT(9F10.3)
C
C     CALCULATION OF STEP SIZE
C
      STEP=1000.
      DO 2400 K=1,NSUSP
      DO 2400 I=1,9
      IF(HFL(K,I).EQ.0.) GOTO 2400
      IF(STEP.LE.HFL(K,I)) GOTO 2400
      STEP=HFL(K,I)
 2400 CONTINUE
      STEP=AMAX1(.49*STEP,1.)
      IF(LCUT.GE.1) WRITE(LUN6,2550) STEP
 2550 FORMAT(12H STEP SIZE= ,F10.3/)
      RETURN
      END
C
C
      SUBROUTINE CLEAR (CAW1,CAW2,CRW1,CRW2,IDX,
     + LOUT,LUN6,MINCLR,NPTSC1,NPTSC2,NUNITS,OX,OZ,THETA,
     + XH,ZH)
      DIMENSION CAW1(15),CAW2(15),CLO(20),CLV1(20),CLV2(20),
     + CRW1(15),CRW2(15),OX(10),OZ(10),THETA(2),
     + XPV1(20),XPV2(20),ZPV1(20),ZPV2(20)
      REAL MINCLR
C
C     LOCATE VEHICLE POINTS
C
      VPA1=THETA(1)
      VPA2=THETA(2)
      DO 110 I=1,NPTSC1
      XPV1(I)=XH+CRW1(I)*COS(VPA1+CAW1(I))
      ZPV1(I)=ZH+CRW1(I)*SIN(VPA1+CAW1(I))
  110 CONTINUE
      IF(LCUT.GE.10)WRITE(LUN6,111) (XPV1(I),I=1,NPTSC1)
      IF(LCUT.GE.10)WRITE(LUN6,111) (ZPV1(I),I=1,NPTSC1)
  111 FORMAT(7H CLEAR0,13F10.3)
      IF(NUNITS.LE.1) GOTO 130
      DO 120 I=1,NPTSC2
      XPV2(I)=XH+CRW2(I)*COS(VPA2+CAW2(I))
      ZPV2(I)=ZH+CRW2(I)*SIN(VPA2+CAW2(I))
  120 CONTINUE
      IF(LCUT.GE.10)WRITE(LUN6,111) (XPV2(I),I=1,NPTSC2)
      IF(LCUT.GE.10)WRITE(LUN6,111) (ZPV2(I),I=1,NPTSC2)
C
```

```
C         CALCULATE CLEARANCE ABOVE OBSTACLE POINTS
C
 130      DO 200 IO=1,10
          CLO(IO)=1000.
          X=OX(IO)
          Z=OZ(IO)
C
C         TEST IF VEHICLE IS ABOVE OBSTACLE POINT
C
          IF(XPV1(1).LT.X) GOTO 200
          IF(XH.LE.X) GOTO 180
          IF(NUNITS.LE.1) GOTO 200
          IF(XPV2(NPTSC2).GE.X) GOTO 200
C
C         TRAILER ABOVE POINT
C
          IF(XPV2(1).GE.X) GOTO 150
          VPZ=ZPV2(1)+(ZH-ZPV2(1))*(X-XPV2(1))/(XH-XPV2(1))
          CLO(IO)=VPZ-Z
          IF(LCUT.GE.10)WRITE(LUN6,141) IO,X,Z,VPZ,CLO(IO)
 141      FORMAT(7H CLEAR1,I3,4F10.3)
          GOTO 200
 150      DO 170 IV=2,NPTSC2
          IF(XPV2(IV).GE.X) GOTO 170
          VPZ=ZPV2(IV)+(ZPV2(IV-1)-ZPV2(IV))*(X-XPV2(IV))/
     +    (XPV2(IV-1)-XPV2(IV))
          CLO(IO)=VPZ-Z
          IF(LCUT.GE.10)WRITE(LUN6,161) IO,X,Z,VPZ,CLO(IO)
 161      FORMAT(7H CLEAR2,I3,4F10.3)
          GOTO 200
 170      CONTINUE
          WRITE(LUN1,176) IO,X,Z
 176      FORMAT(6H OERR1,I3,2F10.3)
          CALL EXIT
C
C         VEHICLE ABOVE POINT
C
 180      DO 190 IV=1,NPTSC1
          IF(XPV1(IV).GE.X) GOTO 190
          VPZ=ZPV1(IV)+(ZPV1(IV-1)-ZPV1(IV))*(X-XPV1(IV))/
     +    (XPV1(IV-1)-XPV1(IV))
          CLO(IO)=VPZ-Z
          IF(LCUT.GE.10)
     +    WRITE(LUN6,186) IO,X,Z,IV,VPZ,CLO(IO)
 186      FORMAT(7H CLEAR3,I3,2F10.3,I3,2F10.3)
          GOTO 200
 190      CONTINUE
          VPZ=ZH+(ZPV1(NPTSC1) -ZH)*(X-XH)/(XPV1(NPTSC1)-XH)
          CLO(IO)=VPZ-Z
          IF(LCUT.GE.10)WRITE(LUN6,196) IO,X,Z,VPZ,CLO(IO)
 196      FORMAT(3H O4,I3,4F10.3)
 200      CONTINUE
C
C         CALCULATE CLEARANCE BELOW VEHICLE POINTS
```

```
C
      DO 240 IV=1,NPTSC1
      CLV1(IV)=1500.
      X=XPV1(IV)
      Z=ZPV1(IV)
      IF(X.GE.OX(1)) GOTO 220
      OPZ=OZ(1)+(OZ(2)-OZ(1))*(X-OX(1))/(OX(2)-OX(1))
      CLV1(IV)=Z-OPZ
      IF(LOUT.GE.10)WRITE(LUN6,216) IV,X,Z,OPZ,CLV1(IV)
216   FORMAT(3H V1,I3,4F10.3)
      GOTO 240
220   DO 230 IO=2,10
      IF(X.GE.OX(IO)) GOTO 230
      OPZ=OZ(IO-1)+(OZ(IO)-OZ(IO-1))*(X-OX(IO-1))/(OX(IO)-OX(IO-1))
      CLV1(IV)=Z-OPZ
      IF(LOUT.GE.10)
     + WRITE(LUN6,226) IV,X,Z,IO,OPZ,CLV1(IV)
226   FORMAT(3H V2,I3,2F10.3,I3,2F10.3)
      GOTO 240
230   CONTINUE
      OPZ=OZ(9)+(OZ(10)-OZ(9))*(X-OX(9))/(OX(10)-OX(9))
      CLV1(IV)=Z-OPZ
      IF(LOUT.GE.10)WRITE(LUN6,236) IV,X,Z,OPZ,CLV1(IV)
236   FORMAT(3H V3,I3,4F10.3)
240   CONTINUE
C
C     CALCULATE CLEARANCE BELOW HITCH
C
      CLH=2000.
      IF(XH.GE.OX(1)) GOTO 260
      OPZ=OZ(1)+(OZ(2)-OZ(1))*(XH-OX(1))/(OX(2)-OX(1))
      CLH=ZH-OPZ
      IF(LOUT.GE.10)WRITE(LUN6,256) XH,ZH,OPZ,CLH
256   FORMAT(3H H1,4F10.3)
      GOTO 280
260   DO 270 IO=2,10
      IF(XH.GE.OX(IO)) GOTO 270
      OPZ=OZ(IO-1)+(OZ(IO)-OZ(IO-1))*(XH-OX(IO-1))/(OX(IO)-OX(IO-1))
      CLH=ZH-OPZ
      IF(LOUT.GE.10)WRITE(LUN6,266) XH,ZH,IO,OPZ,CLH
266   FORMAT(3H H2,2F10.3,I3,2F10.3)
      GOTO 280
270   CONTINUE
      OPZ=OZ(9)+(OZ(10)-OZ(9))*(XH-OX(9))/(OX(10)-OX(9))
      CLH=ZH-OPZ
      IF(LOUT.GE.10)WRITE(LUN6,276) XH,ZH,OPZ,CLH
276   FORMAT(3H H3,4F10.3)
C
C     CALCULATE CLEARANCE BELOW TRAILER POINTS
C
280   IF(NUNITS.LE.1) GOTO 325
      DO 320 IV=1,NPTSC2
      CLV2(IV)=2500.
      X=XPV2(IV)
```

```
      Z=ZPV2(IV)
      IF(X.GE.OX(1)) GOTO 300
      OPZ=OZ(1)+(OZ(2)-OZ(1))*(X-OX(1))/(OX(2)-OX(1))
      CLV2(IV)=Z-OPZ
      IF(LCUT.GE.10)WRITE(LUN6,291) IV,X,Z,OPZ,CLV2(IV)
 291  FORMAT(3H T1,I3,4F10.3)
      GOTO 320
 300  DO 310 IO=2,10
      IF(X.GE.OX(IO)) GOTO 310
      OPZ=OZ(IO-1)+(OZ(IO)-OZ(IO-1))*(X-OX(IO-1))/(OX(IO)-OX(IO-1))
      CLV2(IV)=Z-OPZ
      IF(LCUT.GE.10)
     + WRITE(LUN6,306) IV,X,Z,IO,OPZ,CLV2(IV)
 306  FORMAT(3H T2,I3,2F10.3,I3,2F10.3)
      GOTO 320
 310  CONTINUE
      OPZ=OZ(9)+(OZ(10)-OZ(9))*(X-OX(9))/(OX(10)-OX(9))
      CLV2(IV)=Z-OPZ
      IF(LCUT.GE.10)WRITE(LUN6,316) IV,X,Z,OPZ,CLV2(IV)
 316  FORMAT(3H T3,I3,4F10.3)
 320  CONTINUE
C
C     MINIMUM CLEARANCE
C
 325  MINCLR=CLO(1)
      IDX=1
      DO 330 IO=2,10
      IF(CLO(IO).GE.MINCLR) GOTO 330
      MINCLR=CLO(IO)
      IDX=IO
 330  CONTINUE
      DO 340 IV=1,NPTSC1
      IF(CLV1(IV).GE.MINCLR) GOTO 340
      MINCLR=CLV1(IV)
      IDX=10000*IV
 340  CONTINUE
      IF(CLH.GE.MINCLR) GOTO 350
      MINCLR=CLH
      IDX=1111
 350  IF(NUNITS.LE.1) GOTO 370
      DO 360 IV=1,NPTSC2
      IF(CLV2(IV).GE.MINCLR) GOTO 360
      MINCLR=CLV2(IV)
      IDX=100*IV
 360  CONTINUE
 370  IF(LCUT.GE.9) WRITE(LUN6,371) MINCLR
 371  FORMAT(4H MIN,F10.3,I10)
      RETURN
      END
C
C
      SUBROUTINE FORCES (XN,MAXC,NTOTAL,SSC,XPH,ZPH)
      DIMENSION AJINV(6,6),W(110),XN(6),F(6)
      DIMENSION ALPHD(3,2),BETAD(3),FX(3,2),FZ(3,2),RF(3,2),TF(3,2)
```

```
C
C
      COMMON ALPHA(5,2),
     + BALMD(3),BALMU(3),
     + BETA(3),BETAP(3),BN(3),BRAKER(5,2),BT(3,2),BWIDTH(3),
     + COSA(3,2),COSB(3),CCSG(3,2),CGFX(2),CGFZ(2),
     + CGX(2),CGZ(2),CGMY(2),CRR(3,2),CTF(3,2),
     + EFFRAD(5),ELL(5),
     + FHX,FHZ,FN(3,2),
     + HA(5,9),HB(5,9),HC(5,9),HD(5,9),HE(5,9),HF(5,9),
     + HFL(5,9),HX(5,10),HZ(5,10),
     + GAMMA(3,2),
     + IB(5,2),IP(5,2),IH(5,2),
     + LOUT,LUN6,
     + NSUSP,NUNITS,NW(5),NW2(5),
     + OA(9),OFL(9),OX(10),OZ(10),
     + PM(3),POWERR(5,2),PX(3),PXPCG(3),PZ(3),PZPCG(3),
     + RBC1,RBC2,RR(3,2),
     + SCALE(6),SFLAG(5),SINA(3,2),SINB(3),STEP,
     + THETB1,THETB2,
     + X(5),XPBC(5),XPW(5,2),
     + Z(5),ZPBC(5),ZPRCF(5,2),ZPW(5,2)
C
C
      INTEGER SFLAG
      EXTERNAL CALFUN
      DSTEP=.0001
      DMAX=100.
      ACC=1.
      MAXFUN=500
      RADIAN=57.29577951
      DO 100 I=1,NSUSP
      SINB(I)=SIN(BETAP(I))
      COSB(I)=COS(BETAP(I))
      DO 100 J=1,2
      SINA(I,J)=SIN(ALPHA(I,J))
      COSA(I,J)=COS(ALPHA(I,J))
      IF(NW2(J).NE.0.AND.NW(J).EQ.0) XN(1)=.01
  100 CONTINUE
      IF(NUNITS .EQ. 1) NEQ=3
      IF(NUNITS .EQ. 2) NEQ=6
      N=0
      SALPHA=0.
      DO 150 I=1,NSUSP
      IF(NW(I).EQ.2)GOTO 130
      N=N+1
      SALPHA=SALPHA+SINA(I,1)-CRR(I,1)
      IF(SFLAG(I).EQ.0.OR.NW(I).EQ.1) GOTO 150
  130 N=N+1
      SALPHA=SALPHA+SINA(I,2)-CRR(I,2)
  150 CONTINUE
      IF(N.EQ.0) GOTO 180
      SCALE(1)=1.
      XN(1)=SALPHA/FLOAT(N)
```

```
      GOTO 190
180   WRITE(LUN6,181)
181   FORMAT(31H FORCES: ERROR IN NO. OF WHEELS)
      CALL EXIT
190   CONTINUE
      DO 200 L=2,NEQ
      IF(-.01.LT.XN(L).AND.XN(L).LT..01) XN(L)=.01
      IF(XN(L).EQ.0.) SCALE(L)=1.
      IF(XN(L).NE.0.) SCALE(L)=10.**IFIX(ALOG10(ABS(XN(L))))
      XN(L)=XN(L)/SCALE(L)
200   CONTINUE
      IPRINT=LOUT-10
      CALL ECSOL (NEQ,XN,F,AJINV,DSTEP,DMAX,ACC,MAXFUN,
     + W,MAXC,LUN6,IPRINT,CALFUN)
      NTOTAL=NTOTAL+MAXC
      DO 300 L=1,NEQ
300   XN(L)=XN(L)*SCALE(L)
      SSQ=0.
      DO 400 K=1,NEQ
400   SSQ=SSQ+F(K)*F(K)
      IF(SSQ.GT.100.) WRITE(LUN5,600) XN,F,SSQ
      IF(LOUT.LT. 10) RETURN
      DO 500 I=1,NSUSP
      BETAC(I)=BETAP(I)*RADIAN
      DO 500 J=1,2
      TF(I,J)=FN(I,J)*CTF(I,J)
      RF(I,J)=-FN(I,J)*CRR(I,J)
      TFRF=TF(I,J)+RF(I,J)
      FX(I,J)=-FN(I,J)*SINA(I,J)+TFRF*COSA(I,J)
      FZ(I,J)= FN(I,J)*COSA(I,J)+TFRF*SINA(I,J)
      ALPHC(I,J)=ALPHA(I,J)*RADIAN
500   CONTINUE
600   FORMAT(16H SSQ OVER LIMIT ,/,5H XN= ,6(2X,F12.3),/5H F=   ,
     + 6(2X,F12.3),/,6H SSQ= ,2X,F12.3)
      WRITE(LUN6,900) SSQ,MAXC,NTOTAL
      IF(SSQ.GT.100.) WRITE(LUN6,910) XN,F
      WRITE(LUN6,920) XPH,ZPH
      WRITE(LUN6,930) (X(I),I=1,NSUSP)
      WRITE(LUN6,940) (Z(I),I=1,NSUSP)
      WRITE(LUN6,950) ((CGX(I),CGZ(I)),I=1,2)
      WRITE(LUN6,960) ((ALPHC(I,J),J=1,2),I=1,NSUSP)
      WRITE(LUN6,970) ((CGFX(I),CGFZ(I)),I=1,2)
      WRITE(LUN6,980) FHX,FHZ
      WRITE(LUN6,990) (SFLAG(I),I=1,NSUSP)
      WRITE(LUN6,1000) (NW(I),I=1,NSUSP)
      WRITE(LUN6,1010) ((RR(I,J),J=1,2),I=1,NSUSP)
      DO 700 I=1,NSUSP
      IF(SFLAG(I).EQ.1) GOTO 800
700   CONTINUE
      GOTO 850
800   WRITE(LUN6,1020) (BETAC(I),I=1,NSUSP)
      WRITE(LUN6,1025) (BWIDTH(I),I=1,NSUSP)
      WRITE(LUN6,1030) (BN(I),I=1,NSUSP)
      WRITE(LUN6,1040) ((ET(I,J),J=1,2),I=1,NSUSP)
```

```
 850     WRITE(LUN6,1050) ((CRR(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1060) ((CTF(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1070) ((FN(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1080) ((RF(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1090) ((TF(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1100) ((FX(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1110) ((FZ(I,J),J=1,2),I=1,NSUSP)
         WRITE(LUN6,1120) (PX(I),I=1,NSUSP)
         WRITE(LUN6,1130) (PZ(I),I=1,NSUSP)
         WRITE(LUN6,1140) (PM(I),I=1,NSUSP)
 900     FORMAT(6H SSQ  ,F12.3,4X,7H CALFUN,2X,I4,4X,8H TCALFUN,2X,I8)
 910     FORMAT(6H XN   ,6(2X,F12.3)/6H F     ,
      +  6(2X,F12.3))
 920     FORMAT(6H XPH  ,2X,F12.3,8X,6H ZPH  ,2X,F12.3)
 930     FORMAT(6H X    ,10(2X,F10.2))
 940     FORMAT(6H Z    ,10(2X,F10.2))
 950     FORMAT(14H CGX(I),CGZ(I),8(2X,F10.2))
 960     FORMAT(6H ALPHA,10(2X,F10.2))
 970     FORMAT(17H CGFX(I),CGFZ(I) ,10(2X,F10.1))
 980     FORMAT(33H FHX,FHZ FORCES AT TRAILER HITCH ,2(2X,F10.2))
 990     FORMAT(6H SFLAG,10(2X,I10))
 1000    FORMAT(6H NW   ,10(2X,I10))
 1010    FORMAT(6H RR   ,10(2X,F10.2))
 1020    FORMAT(6H BETAP,10(2X,F10.2))
 1025    FORMAT(7H BWIDTH,10(2X,F10.2))
 1030    FORMAT(6H BN   ,10(2X,F10.2))
 1040    FORMAT(6H BT   ,10(2X,F10.3))
 1050    FORMAT(6H CRR  ,10(2X,F10.2))
 1060    FORMAT(6H CTF  ,10(2X,F10.2))
 1070    FORMAT(6H FN   ,10(2X,F10.2))
 1080    FORMAT(6H RF   ,10(2X,F10.2))
 1090    FORMAT(6H TF   ,10(2X,F10.2))
 1100    FORMAT(6H FX   ,10(2X,F10.2))
 1110    FORMAT(6H FZ   ,10(2X,F10.2))
 1120    FORMAT(6H PX   ,10(2X,F10.2))
 1130    FORMAT(6H PZ   ,10(2X,F10.2))
 1140    FORMAT(6H PM   ,10(2X,F10.1))
         RETURN
         END
C
C
C
         SUBROUTINE NFORCE (XX,XXT,XZM,XZMT,ZZ,ZZT)
C
C
         COMMON ALPHA(5,2),
      +  BALMD(3),BALMU(3),
      +  BETA(3),BETAP(3),BN(3),BRAKER(5,2),BT(3,2),BWIDTH(3),
      +  COSA(3,2),COSB(3),COSG(3,2),CGFX(2),CGFZ(2),
      +  CGX(2),CGZ(2),CGMY(2),CRR(3,2),CTF(3,2),
      +  EFFRAD(5),ELL(5),
      +  FHX,FHZ,FN(3,2),
      +  HA(5,9),HB(5,9),HC(5,9),HD(5,9),HE(5,9),HF(5,9),
      +  HFL(5,9),HX(5,10),HZ(5,10),
```

```
     +   GAMMA(3,2),
     +   IB(5,2),IP(5,2),IR(5,2),
     +   LOUT,LUN6,
     +   NSUSP,NUNITS,NW(5),NW2(5),
     +   OA(9),OFL(9),OX(10),OZ(10),
     +   PM(3),POWERR(5,2),PX(3),PXPCG(3),PZ(3),PZPCG(3),
     +   RBC1,RBC2,RR(3,2),
     +   SCALE(6),SFLAG(5),SINA(3,2),SINB(3),STEP,
     +   THETB1,THETB2,
     +   X(5),XPBC(5),XPW(5,2),
     +   Z(5),ZPBC(5),ZPRCF(5,2),ZPW(5,2)
C
C
      INTEGER SFLAG
      DIMENSION ANGLE(3,2),COSANG(3,2),FORCE(3,2),SINANG(3,2)
      XX=-FHX+CGFX(1)
      ZZ=-FHZ+CGFZ(1)
      XZM=CGFZ(1)*CGX(1)-CGFX(1)*CGZ(1)+CGMY(1)
      DO 50 I=1,NSUSP
C     SET TO ZERO
      BN(I)=0.
      BT(I,1)=0.
      BT(I,2)=0.
      FORCE(I,1)=0.
      FORCE(I,2)=0.
C     IF SINGLE WHEEL ASSEMBLY GOTO 10
      IF(SFLAG(I).EQ.0.OR.(SFLAG(I).EQ.1.AND.NW(I).EQ.3)) GOTO 10
C     IF BOGIE ASSEMBLY IS SUPPORTED ON BOTH WHEELS GOTO 20
      IF((SFLAG(I).EQ. 1).AND.(NW(I).EQ. 0)) GOTO 20
C     IF BOGIE ASSEMBLY IS SUPPORTED ON ONE WHEEL ONLY GOTO 30
      IF(SFLAG(I).EQ.1.AND.(NW(I).EQ.1.OR.NW(I).EQ.2)) GOTO 30
      WRITE(LUN5,5) I,SFLAG(I),NW(I)
 5    FORMAT(42H ERROR IN WHEEL SUPPORT SPEC. I,SFLAG,NW= ,
     +   3(2X,I3))
C     SINGLE WHEEL ASSEMBLY
 10   J=1
      CTF(I,2)=0.
      CTR=CTF(I,J)-CRR(I,J)
      IF(FN(I,J).LE.0.) CTR=0.
      PX(I)=FN(I,J)*(COSA(I,J)*CTR - SINA(I,J))
      PZ(I)=FN(I,J)*(COSA(I,J) + SINA(I,J)*CTR)
      PM(I)=FN(I,J)*RR(I,J)*CTF(I,J)
      GOTO 40
C     BOGIE ASSEMBLY SUPPORTED ON BOTH WHEELS
 20   DO 25 J=1,2
C     ANGLE OF THE VECTOR ATTACHED AT WHEEL CENTER
      ANGLE(I,J)=GAMMA(I,J)+BETAP(I)-ALPHA(I,J)
      SINANG(I,J)=SIN(ANGLE(I,J))
      COSANG(I,J)=COS(ANGLE(I,J))
 25   CONTINUE
      J=1
      IF(NW2(I).EQ.2) FN(I,1)=.5*FN(I,1)
      FORCE(I,J)=FN(I,J)/COSG(I,J)
      IF(FN(I,J).LE.0.) FORCE(I,J)=FN(I,J)
```

```
C         NORMAL FORCE ON BOGIE BEAM(EQ. FOR BOTH WHEELS)
          BN(I)=FORCE(I,J)*COSANG(I,J)
C         TANGENTIAL FORCE ON BOGIE BEAM
          BT(I,J)=FORCE(I,J)*SINANG(I,J)
C         NORMAL FORCE TO THE GROUND UNDER WHEEL J=2
          J=2
          FORCE(I,J)=BN(I)/COSANG(I,J)
          FN(I,J)=FORCE(I,J)*COSG(I,J)
C         TANGENTIAL FORCE UNDER WHEEL J=2
          BT(I,J)=FORCE(I,J)*SINANG(I,J)
C         FORCES ACTING ON PIVOT
          BN2=BN(I)*2.
C         TOTAL TANGENTIAL FORCE
          BTT=BT(I,1)+BT(I,2)
C         COMPONENTS OF THE PIVOT FORCE
          PX(I)=-BN2*SINB(I)+BTT*COSB(I)
          PZ(I)=BN2*COSB(I)+BTT*SINB(I)
C         MOMENT AT PIVOT
          PM(I)=FN(I,1)*RR(I,1)*CTF(I,1)+FN(I,2)*RR(I,2)*CTF(I,2)
          GOTO 40
C         BOGIE ASSEMBLY SUPPORTED ON ONE WHEEL ONLY ( ON OBST.)
 30       J=NW(I)
          BW=.5*BWIDTH(I)
          IF(J.EQ.1) K=2
          IF(J.EQ.2) K=1
          FN(I,J)=FN(I,1)
          FN(I,K)=0.
          CTF(I,K)=0.
          IF(J.EQ.2) BW=-BW
          ANGLE(I,J)=GAMMA(I,J)+BETAP(I)-ALPHA(I,J)
          SINANG(I,J)=SIN(ANGLE(I,J))
          COSANG(I,J)=COS(ANGLE(I,J))
          FORCE(I,J)=FN(I,J)/COSG(I,J)
          IF(FN(I,J).LE.0.) FORCE(I,J)=FN(I,J)
C         NORMAL FORCE ON BOGIE BEAM(EQ. FOR BOTH WHEELS)
          BN(I)=FORCE(I,J)*COSANG(I,J)
C         TANGENTIAL FORCE ON BOGIE BEAM
          BT(I,J)=FORCE(I,J)*SINANG(I,J)
          PX(I)=-BN(I)*SINB(I)+BT(I,J)*COSB(I)
          PZ(I)=BN(I)*COSB(I)+BT(I,J)*SINB(I)
          PM(I)=FN(I,J)*RR(I,J)*CTF(I,J)+BN(I)*BW
 40       CONTINUE
 50       CONTINUE
C         SIGN CONVENTION FOR LENGTH OF THE MOMENTS ARMS
C         + FROM HITCH TO THE RIGHT SIDE, + IN UP DIRECTION
C         + FOR MOMENTS CCW.
          DO 100 I=1,2
          XX=XX+PX(I)
          ZZ=ZZ+PZ(I)
          XZM=XZM+PX(I)*Z(I)+PZ(I)*X(I)+PM(I)
 100      CONTINUE
          IF(NSUSP .EQ. 2) GOTO 200
C         FORCE SUMMATION FOR TRAILER
          XXT=PX(3)+FHX+CGFX(2)
```

```
      ZZT=PZ(3)+FHZ+CGFZ(2)
      XZMT=-PX(3)*Z(3)+PZ(3)*X(3)+CGFZ(2)*CGX(2)+PM(3)-CGFX(2)*CGZ(2)
     + +CGMY(2)
      RETURN
 200  XXT=0.
      ZZT=0.
      XZMT=0.
      RETURN
      END
C
C
C
      SUBROUTINE CALFUN(N,XN,F)
      INTEGER SFLAG
C
C
      COMMON ALPHA(5,2),
     + BALMD(3),BALMU(3),
     + BETA(3),BETAP(3),BN(3),BRAKER(5,2),BT(3,2),BWIDTH(3),
     + COSA(3,2),COSB(3),COSG(3,2),CGFX(2),CGFZ(2),
     + CGX(2),CGZ(2),CGMY(2),CRR(3,2),CTF(3,2),
     + EFFRAD(5),ELL(5),
     + FHX,FHZ,FN(3,2),
     + HA(5,9),HB(5,9),HC(5,9),HD(5,9),HE(5,9),HF(5,9),
     + HFL(5,9),HX(5,10),HZ(5,10),
     + GAMMA(3,2),
     + IB(5,2),IP(5,2),IH(5,2),
     + LOUT,LUN6,
     + NSUSP,NUNITS,NW(5),NW2(5),
     + OA(9),OFL(9),OX(10),OZ(10),
     + PM(3),POWERR(5,2),PX(3),PXPCG(3),PZ(3),PZPCG(3),
     + RBC1,RBC2,RR(3,2),
     + SCALE(6),SFLAG(5),SINA(3,2),SINB(3),STEP,
     + THETB1,THETB2,
     + X(5),XPBC(5),XPW(5,2),
     + Z(5),ZPBC(5),ZPRLF(5,2),ZPW(5,2)
C
C
      DIMENSION XN(6),F(6)
      CTFR=XN(1)*SCALE(1)
      FN(1,1)=XN(2)*SCALE(2)
      FN(2,1)=XN(3)*SCALE(3)
      FN(3,1)=XN(4)*SCALE(4)
      FHX=XN(5)*SCALE(5)
      FHZ=XN(6)*SCALE(6)
      DO 100 I=1,3
      FN(I,2)=0.
      DO 100 J=1,2
      IF(CTFR.GE.0.) CTF(I,J)=CTFR*POWERR(I,J)*FLOAT(IP(I,J))
      IF(CTFR.LT.0.) CTF(I,J)=CTFR*BRAKER(I,J)*FLOAT(IB(I,J))
      GAMMA(I,J)=ATAN(CTF(I,J)-CRR(I,J))
      COSG(I,J)=COS(GAMMA(I,J))
 100  CONTINUE
      CALL NFORCE (XX,XXT,XZM,XZMT,ZZ,ZZT)
```

```
      F(1)=XX
      F(2)=ZZ
      F(3)=XZM
      F(4)=XXT
      F(5)=ZZT
      F(6)=XZMT
      RETURN
      END
      SUBROUTINE MOVEB (CSLOPE,NEQL,
     + NVEH1,RBC,REFHT1,RHTCH,RWLIM,SSLOPE,SSQM,THETA,THETA0,THET0H,
     + TWLIM,XPCG,XPH,ZPCG,ZPH)
C
C
      COMMON ALPHA(5,2),
     + BALMC(3),BALMU(3),
     + BETA(3),BETAP(3),BN(3),BRAKER(5,2),BT(3,2),BWIDTH(3),
     + COSA(3,2),COSB(3),COSG(3,2),CGFX(2),CGFZ(2),
     + CGX(2),CGZ(2),CGMY(2),CHR(3,2),CTF(3,2),
     + EFFRAD(5),ELL(5),
     + FHX,FHZ,FN(3,2),
     + HA(5,9),HB(5,9),HC(5,9),HD(5,9),HE(5,9),HF(5,9),
     + HFL(5,9),HX(5,10),HZ(5,10),
     + GAMMA(3,2),
     + IB(5,2),IP(5,2),IH(5,2),
     + LOUT,LUN6,
     + NSUSF,NUNITS,NW(5),NW2(5),
     + OA(9),OFL(9),OX(10),OZ(10),
     + PM(3),POWERR(5,2),PX(3),PXPCG(3),PZ(3),PZPCG(3),
     + RBC1,RBC2,RR(3,2),
     + SCALE(6),SFLAG(5),SINA(3,2),SINE(3),STEP,
     + THETB1,THETB2,
     + X(5),XPBC(5),XPW(5,2),
     + Z(5),ZPBC(5),ZPRCF(5,2),ZPW(5,2)
C
C
      INTEGER SFLAG
C
      DIMENSION AJINV(6,6),ELEV(5),
     + RBC(5),RHTCH(2),RWLIM(3,2),THETA(2),THETA0(5),
     + THET0H(2),TWLIM(3,2),W(110),XL(5),XPCG(2),ZPCG(2)
      EXTERNAL ELEVAT
      DO 10 I=1,5
   10 NW2(I)=NW(I)
      DSTEP=.0001
      DMAX=100.
      ACC=.1*STEP
      MAXFUN=500
      PXPCG(1)=XPCG(1)
      PZPCG(1)=ZPCG(1)
      PTHETA=THETA(1)
      NEQL=3
      NAGAIN=0
      NW(1)=0
      NW(2)=0
```

```
      THETB1=THETA0(1)
      THETB2=THETA0(2)
      RBC1=RBC(1)
      RBC2=RBC(2)
      IF(SFLAG(1).EQ.0) GOTO 20
      NEQL=4
      XL(4)=BETA(1)
 20   IF(SFLAG(2).EQ.0) GOTO 30
      NEQL=NEQL+1
      XL(NEQL)=BETA(2)
 30   XL(1)=PXPCG(1)+STEP*CSLOPE
      XL(2)=PZPCG(1)+STEP*SSLOPE
      XL(3)=PTHETA
      IF(LOUT.GE.10) WRITE(LUN6,46) NEQL,
     + THETB1,RBC1,THETB2,RBC2,(XL(L),L=1,NEQL)
 46   FORMAT(6H MOVE1,I4,14F8.3)
      LOUT=LOUT+1
      CALL ELEVAT (NEQL,XL,ELEV)
      LOUT=LOUT-1
      IPRINT=LOUT-10
      CALL EQSOL (NEQL,XL,ELEV,AJINV,CSTEP,
     + DMAX,ACC,MAXFUN,W,MAXC,LUN6,IPRINT,ELEVAT)
      LOUT=LOUT+1
      CALL ELEVAT (NEQL,XL,ELEV)
      LOUT=LOUT-1
      SSQM=0.
      DO 50 L=1,NEQL
 50   SSQM=SSQM+ELEV(L)**2
      XPCG(1)=XL(1)
      ZPCG(1)=XL(2)
      THETA(1)=XL(3)
      IF(LOUT.GE.10) WRITE(LUN6,61) XPCG(1),ZPCG(1),THETA(1),
     + XPBC(1),ZPBC(1),XPW(1,1),ZPW(1,1),IH(1,1),XPBC(2),ZPBC(2),
     + XPW(2,1),ZPW(2,1),IH(2,1)
 61   FORMAT(6H MOVE2,7F10.3,I3,4F10.3,I3)
      IF(SSQM.GT.10.) WRITE(LUN5,66) SSQM,MAXC
 66   FORMAT(23H SSQM OVER LIMIT: SSQM=,E15.7,
     + 8H,   MAXC=,I6)
      IF(NEQL.EQ.3) GOTO 300
C
C ONE SUSPENSION ON UNIT 1 IS A BOGIE
C
      IF(SFLAG(1).EQ.1.AND.NW(1).EQ.0) GOTO 70
      BETA(2)=XL(4)
      GOTO 80
 70   BETA(1)=XL(4)
      IF(LOUT.GE.10) WRITE(LUN6,71) BETA(1),XPW(1,2),ZPW(1,2),
     + IH(1,2)
 71   FORMAT(6H MOVE3,3F10.3,I3)
      IF(SFLAG(2).EQ.0.OR.NW(2).NE.0) GOTO 85
      BETA(2)=XL(5)
 80   IF(LOUT.GE.10) WRITE(LUN6,81) BETA(2),XPW(2,2),ZPW(2,2),
     + IH(2,2)
 81   FORMAT(6H MOVE4,3F10.3,I3)
```

```
C
C CHECK FIRST SUSPENSION BOGIE OUT OF LIMIT
C IF SINGLE AXLE OR BOGIE ON BOTH WHEELS LEAVE
C THETB1 AND RBC1
C
 85      IF(SFLAG(1).EQ.0.OR.NW(1).NE.0) GOTO 190
         IF(BETA(1).GE.BALMU(1)) NW(1)=1
         IF(BETA(1).LE.BALMD(1)) NW(1)=2
         IF(SFLAG(1).EQ.0.OR.(SFLAG(1).EQ.1.AND.
     +   NW(1).EQ.0)) GOTO 190
         IF(SFLAG(1).EQ.1.AND.NW(1).EQ.1) GOTO 150
C
C FIRST SUSPENSION BOGIE ON REAR WHEEL ONLY
C
C
         THETB1=TWLIM(1,2)
         RBC1=RWLIM(1,2)
         BETA(1)=BALMD(1)
         GOTO 170
C
C FIRST SUSPENSION BOGIE ON FRONT WHEEL ONLY
C
 150     THETB1=TWLIM(1,1)
         RBC1=RWLIM(1,1)
         BETA(1)=BALMU(1)
 170     IF(NEQL.EQ.5) XL(4)=XL(5)
         NEQL=NEQL-1
         NAGAIN=1
C
C CHECK SECOND SUSPENSION BOGIE OUT OF LIMIT
C IF SINGLE AXLE OR BOGIE ON BOTH WHEELS LEAVE
C THETB2 AND RBC2
C
 190     IF(SFLAG(2).EQ.0.OR.NW(2).NE.0) GOTO 280
         IF(BETA(2).GE.BALMU(2)) NW(2)=1
         IF(BETA(2).LE.BALMD(2)) NW(2)=2
         IF(SFLAG(2).EQ.0.OR.(SFLAG(2).EQ.1.AND.
     +   NW(2).EQ.0)) GOTO 280
         IF(SFLAG(2).EQ.1.AND.NW(2).EQ.1) GOTO 250
C
C SECOND SUSPENSION BOGIE ON REAR WHEEL ONLY
C
         THETB2=TWLIM(2,2)
         RBC2=RWLIM(2,2)
         BETA(2)=BALMD(2)
         GOTO 270
C
C SECOND SUSPENSION BOGIE ON FRONT WHEEL ONLY
C
 250     THETB2=TWLIM(2,1)
         RBC2=RWLIM(2,1)
         BETA(2)=BALMU(2)
 270     NEQL=NEQL-1
         NAGAIN=1
```

```
C
 280     IF(NAGAIN.EQ.0) GOTO 300
         NAGAIN=0
         GOTO 30
C
C
C UNIT 1 POSITIONED ON WHEELS - CHECK FOR
C SPROCKET/IDLER INTERFERENCE IF TRACKED
C
 300     IF(NVEH1.NE.0) GOTO 600
C
C TRACKED VEHICLE
C
C ****** IDLER AND SPROCKET SUPPORT CHECK HERE ******
C
         XSF=XPCG(1)+RBC(4)*COS(THETA0(4)+THETA(1))
         ZSF=ZPCG(1)+RBC(4)*SIN(THETA0(4)+THETA(1))
         CALL WHEEL3 (E,HA,HC,HE,HF,HX,IH(4,1),4,LOUT,LUN6,
       * XSF,ZSF,ZPROF(4,1))
         IF(LOUT.GE.10) WRITE(LUN6,311) XSF,ZSF,ZPROF(4,1),IH(4,1),E
 311     FORMAT(7H MOVES4,3F10.3,I5,F10.3)
         IF(E.GE.-.1) GOTO 400
C
C FRONT SPROCKET/IDLER INTERFERENCE
C
         THETB1=THETA0(4)
         RBC1=RBC(4)
         IF(SFLAG(1).EQ.0.OR.NW(1).NE.0) GOTO 320
         IF(NEQL.EQ.5) XL(4)=XL(5)
         NEQL=NEQL-1
 320     NAGAIN=1
         NW(1)=3
C
 400     XSR=XPCG(1)+RBC(5)*COS(THETA0(5)+THETA(1))
         ZSR=ZPCG(1)+RBC(5)*SIN(THETA0(5)+THETA(1))
         CALL WHEEL3 (E,HA,HC,HE,HF,HX,IH(5,1),5,LOUT,LUN6,
       * XSF,ZSF,ZPROF(5,1))
         IF(LOUT.GE.10) WRITE(LUN6,411) XSR,ZSR,ZPROF(5,1),IH(5,1),E
 411     FORMAT(7H MOVES5,3F10.3,I5,F10.3)
         IF(E.GE.-.1) GOTO 500
C
C REAR SPROCKET/IDLER INTERFERENCE
C
         THETB2=THETA0(5)
         RBC2=RBC(5)
         IF(SFLAG(2).EQ.0.OR.NW(2).NE.0) GOTO 420
         NEQL=NEQL-1
 420     NAGAIN=1
         NW(2)=3
C
 500     IF(NAGAIN.EQ.0) GOTO 600
         NAGAIN=0
         GOTO 30
C
```

```
C ANGLE UNDER WHEELS
C
 600      IF(NW(1).EQ.2) GOTO 610
          CALL WHEEL1 (ALPHA(1,1),HA,HD,HE,IH(1,1),1,OX,OZ,
     +    XPW(1,1),ZPW(1,1))
          IF(LCUT.GE.10) WRITE(LUN6,606) XPW(1,1),ZPW(1,1),
     +    IH(1,1),ALPHA(1,1)
 606      FORMAT(7H MOVE11,2F10.3,I4,F10.3)
 610      IF(NW(1).EQ.1.OR.SFLAG(1).EQ.0) GOTO 620
          CALL WHEEL1 (ALPHA(1,2),HA,HD,HE,IH(1,2),1,OX,OZ,
     +    XPW(1,2),ZPW(1,2))
          IF(LCUT.GE.10) WRITE(LUN6,616) XPW(1,2),ZPW(1,2),
     +    IH(1,2),ALPHA(1,2)
 616      FORMAT(7H MOVE12,2F10.3,I4,F10.3)
 620      IF(NW(2).EQ.2) GOTO 630
          CALL WHEEL1 (ALPHA(2,1),HA,HD,HE,IH(2,1),2,OX,OZ,
     +    XPW(2,1),ZPW(2,1))
          IF(LCUT.GE.10) WRITE(LUN6,626) XPW(2,1),ZPW(2,1),
     +    IH(2,1),ALPHA(2,1)
 626      FORMAT(7H MOVE21,2F10.3,I4,F10.3)
 630      IF(NW(2).EQ.1.OR.SFLAG(2).EQ.0) GOTO 640
          CALL WHEEL1 (ALPHA(2,2),HA,HD,HE,IH(2,2),2,OX,OZ,
     +    XPW(2,2),ZPW(2,2))
          IF(LCUT.GE.10) WRITE(LUN6,636) XPW(2,2),ZPW(2,2),
     +    IH(2,2),ALPHA(2,2)
 636      FORMAT(7H MOVE22,2F10.3,I4,F10.3)
 640      CONTINUE
C
C LOCATE HITCH
C
          XPH=XPCG(1)+RHTCH(1)*COS(THETØH(1)+THETA(1))
          ZPH=ZPCG(1)+RHTCH(1)*SIN(THETØH(1)+THETA(1))
          IF(NUNITS.EQ.1) RETURN
C
C SECOND UNIT
C
          IF(SFLAG(3).EQ.1) GOTO 670
C
C SINGLE AXLE TRAILER
C
          RSQ=RWLIM(3,1)**2
          CALL WHEEL2 (EFFRAD,HA,HD,HE,HF,HX,HZ,IH(2,1),IH(3,1),
     +    3,LOUT,LUN6,OX,OZ,ALPHA(3,1),RWLIM(3,1),RSQ,XPH,
     +    XPW(3,1),ZPH,ZPW(3,1))
          XPBC(3)=XPW(3,1)
          ZPBC(3)=ZPW(3,1)
          A=ATN2(ZPBC(3)-ZPH,XPBC(3)-XPH)
          THETA(2)=A-TWLIM(3,1)
          XPCG(2)=XPH+RHTCH(2)*COS(THETØH(2)+THETA(2))
          ZPCG(2)=ZPH+RHTCH(2)*SIN(THETØH(2)+THETA(2))
          IF(LCUT.GE.10) WRITE(LUN6,656) XPH,ZPH,XPW(3,1),ZPW(3,1),
     +    ALPHA(3,1),XPBC(3),ZPBC(3),A,THETA(2),XPCG(2),ZPCG(2)
 656      FORMAT(7H MOVEA3,11F10.3)
          RETURN
```

```
C
C BOGIE AXLE TRAILER - TEST IF CN FRONT WHEEL ONLY
C
 670      RSQ=RWLIM(3,1)**2
          CALL WHEEL2 (EFFRAD,HA,HD,HE,HF,HX,HZ,IH(2,1),IH(3,1),
     +    3,LOUT,LUN6,CX,CZ,ALPHA(3,1),RWLIM(3,1),RSQ,XPH,
     +    XPW(3,1),ZPH,ZPW(3,1))
          A=ATN2(ZPW(3,1)-ZPH,XPW(3,1)-XPH)
          T=A-TWLIM(3,1)
          XPW(3,2)=XPW(3,1)-BWIDTH(3)*COS(BALMU(3)+T)
          ZPW(3,2)=ZPW(3,1)-BWIDTH(3)*SIN(BALMU(3)+T)
          CALL WHEEL3 (ELE,HA,HD,HE,HF,HX,IH(3,2),3,LOUT,LUN6,
     +    XPW(3,2),ZPW(3,2),ZFRCF(3,2))
          IF(ELE.LE.0.) GOTO 690
C
C TRAILER BOGIE CN FRONT WHEEL CNLY
C
          NW(3)=1
          BETA(3)=BALMU(3)
          XPBC(3)=XPW(3,1)-.5*BWIDTH(3)*COS(BALMU(3)+T)
          ZPBC(3)=ZPW(3,1)-.5*BWIDTH(3)*SIN(BALMU(3)+T)
          THETA(2)=T
          XPCG(2)=XPH+RHTCH(2)*COS(THET0H(2)+T)
          ZPCG(2)=ZPH+RHTCH(2)*SIN(THET0H(2)+T)
          IF(LOUT.GE.10) WRITE(LUN6,686) XPH,ZPH,XPW(3,1),ZPW(3,1),
     +    ALPHA(3,1),XPBC(3),ZPBC(3),A,T,XPCG(2),ZPCG(2),NW(3)
 686      FORMAT(7H MOVEA4,11F10.3,2I3)
          RETURN
C
C TRAILER BOGIE NOT CN FRONT WHEEL CNLY - TEST IF ON REAR WHEEL ONLY
C
 690      RSQ=RWLIM(3,2)**2
          CALL WHEEL2 (EFFRAD,HA,HC,HE,HF,HX,HZ,IH(2,1),IH(3,2),
     +    3,LOUT,LUN6,CX,CZ,ALPHA(3,2),RWLIM(3,2),RSQ,XPH,
     +    XPW(3,2),ZPH,ZPW(3,2))
          A=ATN2(ZPW(3,2)-ZPH,XPW(3,2)-XPH)
          T=A-TWLIM(3,2)
          XPW(3,1)=XPW(3,2)+BWIDTH(3)*COS(BALMD(3)+T)
          ZPW(3,1)=ZPW(3,2)+BWIDTH(3)*SIN(BALMD(3)+T)
          CALL WHEEL3 (ELE,HA,HD,HE,HF,HX,IH(3,1),3,LOUT,LUN6,
     +    XPW(3,1),ZPW(3,1),ZFRCF(3,1))
          IF(ELE.LE.0.) GOTC 720
C
C TRAILER BOGIE CN REAR WHEEL ONLY
C
          NW(3)=2
          BETA(3)=BALMD(3)
          XPBC(3)=XPW(3,2)+.5*BWIDTH(3)*COS(BALMD(3)+T)
          ZPBC(3)=ZPW(3,2)+.5*BWIDTH(3)*SIN(BALMD(3)+T)
          THETA(2)=T
          XPCG(2)=XPH+RHTCH(2)*COS(THET0H(2)+T)
          ZPCG(2)=ZPH+RHTCH(2)*SIN(THET0H(2)+T)
          IF(LOUT.GE.10) WRITE(LUN6,716) XPH,ZPH,XPW(3,2),ZPW(3,2),
     +    ALPHA(3,2),XPBC(3),ZPBC(3),A,T,XPCG(2),ZPCG(2),NW(3)
```

```
 716    FORMAT(7H MOVEA5,11F10.3,2I3)
        RETURN
C
C TRAILER BOGIE ON BOTH WHEELS - SEARCH ON BOGIE ANGLE
C UNTIL BOTH WHEELS ARE ON HUB PROFILE TO WITHIN TOLERANCE
C
 720    IF(ABS(ELE).LE..1) GOTO 800
        BO2=.5*BWIDTH(3)
        BETA(3)=BALMD(3)
        IF(LOUT.GE.11) WRITE(LUN6,721) ELE,BO2,BETA(3)
 721    FORMAT(8H MOVEA5A,3F10.3)
 725    DELTB=ATN2(-ELE,BO2)
        BETA(3)=BETA(3)+DELTB
        X2=ELL(3)-BO2*COS(BETA(3))
        Z2=-REFHT1+EFFRAD(3)-BO2*SIN(BETA(3))
        RH2SQ=X2*X2+Z2*Z2
        RH2=SQRT(RH2SQ)
        THET2=ATN2(Z2,X2)
        IF(THET2.GT.0.) THET2=THET2-6.2831853
        CALL WHEEL2 (EFFRAD,HA,HD,HE,HF,HX,HZ,IH(2,1),IH(3,2),
     +  3,LOUT,LUN6,OX,OZ,ALPHA(3,2),RH2,RH2SQ,XPH,
     +  XPW(3,2),ZPH,ZPW(3,2))
        A=ATN2(ZPW(3,2)-ZPH,XPW(3,2)-XPH)
        IF(A.GT.0.) A=A-6.2831853
        THETA(2)=A-THET2
        XPW(3,1)=XPW(3,2)+BWIDTH(3)*COS(THETA(2)+BETA(3))
        ZPW(3,1)=ZPW(3,2)+BWIDTH(3)*SIN(THETA(2)+BETA(3))
        CALL WHEEL3 (ELE,HA,HD,HE,HF,HX,IH(3,1),3,LOUT,LUN6,
     +  XPW(3,1),ZPW(3,1),ZFACF(3,1))
        IF(LOUT.GE.11) WRITE(LUN6,751) DELTB,BETA(3),X2,Z2,RH2SQ,
     +  RH2,THET2,XPW(3,2),ZPW(3,2),A,THETA(2),XPW(3,1),ZPW(3,1),ELE
 751    FORMAT(8H MOVEA5B,7F10.3/8X,7F10.3)
        IF(ABS(ELE).GT..1) GOTO 725
C
C BOTH WHEELS ON HUB PROFILE TO WITHIN .1 INCH
C
 800    CALL WHEEL1 (ALPHA(3,1),HA,HD,HE,IH(3,1),3,OX,OZ,
     +  XPW(3,1),ZPW(3,1))
        NW(3)=0
        XPBC(3)=.5*(XPW(3,1)+XPW(3,2))
        ZPBC(3)=.5*(ZPW(3,1)+ZPW(3,2))
        XPCG(2)=XPH+RHTCH(2)*COS(THET0H(2)+THETA(2))
        ZPCG(2)=ZPH+RHTCH(2)*SIN(THET0H(2)+THETA(2))
        XTEMP=XPW(3,1)-XPW(3,2)
        ZTEMP=ZPW(3,1)-ZPW(3,2)
        BETA(3)=ATN2(ZTEMP,XTEMP)
        IF(LOUT.GE.10) WRITE(LUN6,811) XPCG(2),ZPCG(2),THETA(2),
     +  XPBC(3),ZPBC(3),(XPW(3,J),ZPW(3,J),ALPHA(3,J),
     +  J=1,2),XPH,ZPH,NW(3)
 811    FORMAT(7H MOVEA6,5F10.3/2(3F10.3),2F10.3,I3)
        RETURN
        END
C
C
```

```
C
          SUBROUTINE ELEVAT(NEQL,XL,ELEV)
C
C
          COMMON ALPHA(5,2),
     +    BALMC(3),BALMU(3),
     +    BETA(3),BETAP(3),BN(3),BRAKER(5,2),BT(3,2),BWIDTH(3),
     +    COSA(3,2),COSB(3),COSG(3,2),CGFX(2),CGFZ(2),
     +    CGX(2),CGZ(2),CGMY(2),CRR(3,2),CTF(3,2),
     +    EFFRAD(5),ELL(5),
     +    FHX,FHZ,FN(3,2),
     +    HA(5,9),HB(5,9),HC(5,9),HD(5,9),HE(5,9),HF(5,9),
     +    HFL(5,9),HX(5,10),HZ(5,10),
     +    GAMMA(3,2),
     +    IB(5,2),IP(5,2),IH(5,2),
     +    LOUT,LUN6,
     +    NSUSP,NUNITS,NW(5),NW2(5),
     +    OA(9),OFL(9),OX(10),OZ(10),
     +    PM(3),POWERR(5,2),PX(3),PXPCG(3),PZ(3),PZPCG(3),
     +    RBC1,RBC2,RR(3,2),
     +    SCALE(6),SFLAG(5),SINA(3,2),SINB(3),STEP,
     +    THETB1,THETB2,
     +    X(5),XPBC(5),XPW(5,2),
     +    Z(5),ZPBC(5),ZPRCF(5,2),ZPW(5,2)
C
C
          INTEGER SFLAG
C
          DIMENSION XL(5),ELEV(5),XLL(5)
C
C XL(1)= X-POSITION OF CG OF UNIT 1
C XL(2)= Z-POSITION OF CG OF UNIT 1
C XL(3)= PITCH ANGLE OF UNIT 1 WRT GROUND COORDINATES
C
C XL(4)= PITCH ANGLE OF FORWARD MOST BOGIE
C        ASSEMBLY ON UNIT 1 WRT VEHICLE COORDINATES
C XL(5)= PITCH ANGLE OF SECOND BOGIE
C        ASSEMBLY ON UNIT 1 WRT VEHICLE COORDINATES
C
C ELEV(1)= DISTANCE OF CG FROM LAST EQUILIBRIUM
C          POSITION MINUS STEP
C ELEV(2)= ELEVATION OF FIRST WHEEL WRT
C          ITS HUB PROFILE
C ELEV(3)= ELEVATION OF SECOND WHEEL WRT
C          ITS HUB PROFILE
C ELEV(4)= ELEVATION OF THIRD WHEEL (WHEN PRESENT) WRT
C          ITS HUB PROFILE
C ELEV(5)= ELEVATION OF FOURTH WHEEL (WHEN PRESENT) WRT
C          ITS HUB PROFILE
C
          DO 10 L=1,NEQL
 10       XLL(L)=XL(L)
          XSQ=STEP*STEP-(XLL(2)-PZPCG(1))**2
          ELEV(1)=XLL(1)-PXPCG(1)-SQRT(ABS(XSQ))
```

```
      THET=XLL(3)
      C=COS(THETB1+THET)
      XPBC(1)=XLL(1)+RBC1*C
      S=SIN(THETB1+THET)
      ZPBC(1)=XLL(2)+RBC1*S
      C=COS(THETB2+THET)
      XPBC(2)=XLL(1)+RBC2*C
      S=SIN(THETB2+THET)
      ZPBC(2)=XLL(2)+RBC2*S
      IF(LOUT.GE.11) WRITE(LUN6,21) C,S,XPBC(1),
     + ZPBC(1),XPBC(2),ZPBC(2),(XLL(I),I=1,NEQL)
 21   FORMAT(8H ELEVAT1,11E10.3)
      IF(SFLAG(1).EQ.1.AND.NW(1).EQ.0) GOTO 30
C
C FIRST ASSEMBLY IS ON SINGLE WHEEL
C
      IF(SFLAG(1).EQ.1.AND.NW(1).NE.3) GOTO 23
      CALL WHEEL3 (ELEV(2),HA,HD,HE,HF,HX,IH(1,1),1,LOUT,LUN6,
     + XPBC(1),ZPBC(1),ZPROF(1,1))
      XPW(1,1)=XPBC(1)
      ZPW(1,1)=ZPBC(1)
      GOTO 50
 23   IF(NW(1).EQ.2) GOTO 27
      XPW(1,1)=XPBC(1)
      ZPW(1,1)=ZPBC(1)
      CALL WHEEL3 (ELEV(2),HA,HD,HE,HF,HX,IH(1,1),1,LOUT,
     + LUN6,XPW(1,1),ZPW(1,1),ZPROF(1,1))
      BETA(1)=BALMU(1)
      XPBC(1)=XPW(1,1)-.5*BWIDTH(1)*COS(BALMU(1)+THET)
      ZPBC(1)=ZPW(1,1)-.5*BWIDTH(1)*SIN(BALMU(1)+THET)
      GOTO 50
 27   XPW(1,2)=XPBC(1)
      ZPW(1,2)=ZPBC(1)
      CALL WHEEL3 (ELEV(2),HA,HD,HE,HF,HX,IH(1,2),1,LOUT,
     + LUN6,XPW(1,2),ZPW(1,2),ZPROF(1,2))
      BETA(1)=BALMD(1)
      XPBC(1)=XPW(1,2)+.5*BWIDTH(1)*COS(BALMD(1)+THET)
      ZPBC(1)=ZPW(1,2)+.5*BWIDTH(1)*SIN(BALMD(1)+THET)
      GOTO 50
C
C FIRST ASSEMBLY IS BOGIE
C
 30   RW1=.5*BWIDTH(1)
      C=COS(XLL(4)+THET)
      XPW(1,1)=XPBC(1)+RW1*C
      S=SIN(XLL(4)+THET)
      ZPW(1,1)=ZPBC(1)+RW1*S
      CALL WHEEL3 (ELEV(2),HA,HD,HE,HF,HX,IH(1,1),1,LOUT,LUN6,
     + XPW(1,1),ZPW(1,1),ZPROF(1,1))
      XPW(1,2)=XPBC(1)-RW1*C
      ZPW(1,2)=ZPBC(1)-RW1*S
      CALL WHEEL3 (ELEV(3),HA,HD,HE,HF,HX,IH(1,2),1,LOUT,LUN6,
     + XPW(1,2),ZPW(1,2),ZPROF(1,2))
      IF(LOUT.GE.11) WRITE(LUN6,41) C,S,(XPW(1,J),
```

```
     +  ZPW(1,J),ZPROF(1,J),IH(1,J),J=1,2)
 41     FORMAT(8H ELEVAT2,2F10.3/2(3F10.3,I3))
 50     IF(SFLAG(2).EQ.1.AND.NW(2).EQ.0) GOTO 70
C
C SECOND ASSEMBLY IS ON SINGLE WHEEL
C
        IF(SFLAG(2).EQ.1.AND.NW(2).NE.3) GOTO 53
        CALL WHEEL3 (ELEV(NEQL),HA,HD,HE,HF,HX,IH(2,1),2,LOUT,LUN6,
     +  XPBC(2),ZPBC(2),ZPROF(2,1))
        XPW(2,1)=XPBC(2)
        ZPW(2,1)=ZPBC(2)
        GOTO 60
 53     IF(NW(2).EQ.2) GOTO 57
        XPW(2,1)=XPBC(2)
        ZPW(2,1)=ZPBC(2)
        CALL WHEEL3 (ELEV(NEQL),HA,HD,HE,HF,HX,IH(2,1),2,LOUT,
     +  LUN6,XPW(2,1),ZPW(2,1),ZPROF(2,1))
        BETA(2)=BALMU(2)
        XPBC(2)=XPW(2,1)-.5*BWIDTH(2)*COS(BALMU(2)+THET)
        ZPBC(2)=ZPW(2,1)-.5*BWIDTH(2)*SIN(BALMU(2)+THET)
        GOTO 60
 57     XPW(2,2)=XPBC(2)
        ZPW(2,2)=ZPBC(2)
        CALL WHEEL3 (ELEV(NEQL),HA,HD,HE,HF,HX,IH(2,2),2,LOUT,
     +  LUN6,XPW(2,2),ZPW(2,2),ZPROF(2,2))
        BETA(2)=BALMD(2)
        XPBC(2)=XPW(2,2)+.5*BWIDTH(2)*COS(BALMD(2)+THET)
        ZPBC(2)=ZPW(2,2)+.5*BWIDTH(2)*SIN(BALMD(2)+THET)
 60     IF(LOUT.GE.11) WRITE(LUN6,61) (ELEV(I),I=1,NEQL)
 61     FORMAT(8H ELEVAT3,5F10.3)
        RETURN
C
C SECOND ASSEMBLY BOGIE
C
 70     NM1=NEQL-1
        RW2=.5*BWIDTH(2)
        C=COS(XLL(NEQL)+THET)
        XPW(2,1)=XPBC(2)+RW2*C
        S=SIN(XLL(NEQL)+THET)
        ZPW(2,1)=ZPBC(2)+RW2*S
        NEQLM1=NEQL-1
        CALL WHEEL3 (ELEV(NEQLM1),HA,HD,HE,HF,HX,IH(2,1),2,
     +  LOUT,LUN6,XPW(2,1),ZPW(2,1),ZPROF(2,1))
        XPW(2,2)=XPBC(2)-RW2*C
        ZPW(2,2)=ZPBC(2)-RW2*S
        CALL WHEEL3 (ELEV(NEQL),HA,HD,HE,HF,HX,IH(2,2),2,LOUT,LUN6,
     +  XPW(2,2),ZPW(2,2),ZPROF(2,2))
        IF(LOUT.GE.11) WRITE(LUN6,61) (ELEV(I),I=1,NEQL)
        IF(LOUT.GE.11) WRITE(LUN6,81) C,S,(XPW(2,J),
     +  ZPW(2,J),ZPROF(2,J),IH(2,J),J=1,2)
 81     FORMAT(8H ELEVAT4,2F10.3/2(3F10.3,I3))
        RETURN
        END
C
```

```
C
C
      SUBROUTINE WHEEL1 (ANGLE,HA,HD,HE,IHUB,K,OX,OZ,XW,ZW)
C
C
      DIMENSION HA(5,9),HD(5,9),HE(5,9),OX(10),OZ(10)
C
C SUBROUTINE TO FIND ANGLE UNDER WHEEL AT XW,ZW,
C OF SUSPENSION K ON HUB PROFILE ELEMENT IHUB
C
      IF(HA(K,IHUB).EQ.1.) GOTO 100
C
C HUB PROFILE ELEMENT A LINE
C
      ANGLE=ATN2(HD(K,IHUB),-HE(K,IHUB))
      IF(ABS(ANGLE).LE..01) ANGLE=0.
      RETURN
C
C HUB PROFILE ELEMENT AN ARC
C
 100  A=ATN2(ZW-OZ(IHUB),XW-OX(IHUB))
      IF(ABS(A).LE..01) A=0.
      ANGLE=A-1.5707963
      RETURN
      END
C
C
C
      SUBROUTINE WHEEL2 (EFFRAD,HA,HD,HE,HF,HX,
     + HZ,IHUB,IH2,K,LCUT,LUN6,OX,OZ,PSLP2,R12,R12SQ,XP1,XP2,ZP1,ZP2)
      DIMENSION EFFRAD(5),HA(5,9),HD(5,9),HE(5,9),HF(5,9),HX
     + (5,10),HZ(5,10),OX(10),OZ(10)
C
C SUBROUTINE TO LOCATE SECOND WHEEL GIVEN ONE
C WHEEL AT XP1,ZP1
C
      DO 100 I=1,IHUB
      DSQ=(HX(K,I)-XP1)**2+(HZ(K,I)-ZP1)**2
      IF(LCUT.EQ.11) WRITE(LUN6,96) I,DSQ,R12SQ,HX(K,I),HZ(K,I)
 96   FORMAT(8H WHEELS0,I2,4F10.3)
      IF(DSQ .LE. R12SQ) GOTO 110
 100  CONTINUE
C
C     SECOND AXLE ON HUB PROFILE ELEMENT IHUB
C
      IH2=IHUB
      GOTO 115
 110  IH2=I-1
      IF(IH2.LT.1) IH2=1
 115  D=SQRT(DSQ)
      IF(HA(K,IH2) .EQ. 1.) GOTO 160
C
C     ELEMENT (K,IH2) IS A LINE
C
```

```
      S=-HC(K,IH2)/HE(K,IH2)
      T=-HF(K,IH2)/HE(K,IH2)
      A=S**2+1.
      B=S*(T-ZP1)-XP1
      C=(T-ZP1)**2+XP1**2-R12SQ
      BOA=B/A
      COA=C/A
      IF(-BOA .GE. 0.) X1=-BOA+SQRT(BOA*BOA-COA)
      IF(-BOA .LT. 0.) X1=-BOA-SQRT(BOA*BOA-COA)
      X2=COA/X1
      Z1=S*X1+T
      Z2=S*X2+T
      IF(X1 .GT. XP1) XP2=X2
      IF(X2 .GT. XP1) XP2=X1
      IF(X1 .GT. XP1 .OR. X2 .GT. XP1) GOTO 150
      IH2P1=IH2+1
      IF(X1.LT.HX(K,IH2).OR.X1.GT.HX(K,IH2P1)) XP2=X2
      IF(X2.LT.HX(K,IH2).OR.X2.GT.HX(K,IH2P1)) XP2=X1
      IF(X1.LT.HX(K,IH2).OR.X2.LT.HX(K,IH2)) GOTO 150
      IF(X1.GT.HX(K,IH2P1).OR.X2.GT.HX(K,IH2P1)) GOTO 150
      IF(Z1 .GE. Z2) XP2=X1
      IF(Z2 .GT. Z1) XP2=X2
 150  ZP2=S*XP2+T
      PSLP2=ATN2(HD(K,IH2),-HE(K,IH2))
      IF(ABS(PSLP2) .LE. .01) PSLP2=0.
      IF(LOUT.EQ.11) WRITE(LUNO,156) IH2,D,S,T,A,B,C,BOA,COA,
     + X1,X2,Z1,Z2,XP2,ZP2,PSLP2
 156  FORMAT(8H0WHEELS1,I3,7F10.3/8F10.3)
      RETURN
C
C     ELEMENT (K,IH2) IS AN ARC
C
 160  CHORD=SQRT((HX(K,IH2+1)-HX(K,IH2))**2+(HZ(K,IH2+1)
     + -HZ(K,IH2))**2)
      A=2.*ASIN(.5*CHORD/EFFRAD(K))
      B=ATN2(HZ(K,IH2)-CZ(IH2),HX(K,IH2)-OX(IH2))
      IF(ABS(B) .LE. .01) B=0.
      IF(B .LE. -1.570796327) B=B+6.283185307
      AHGH=B
      ALOW=B-A
      DO 180 I=1,6
      AMID=.5*(AHGH+ALOW)
      HXM=CX(IH2)+EFFRAD(K)*COS(AMID)
      HZM=CZ(IH2)+EFFRAD(K)*SIN(AMID)
      RM2=(HXM-XP1)**2+(HZM-ZP1)**2
      IF(RM2 .LE. R12SQ) GOTO 170
      AHGH=AMID
      GOTO 180
 170  IF(RM2 .EQ. R12SQ) GOTO 190
      ALOW=AMID
 180  CONTINUE
 190  XP2=HXM
      ZP2=HZM
      RKANG=ATN2(ZP2-OZ(IH2),XP2-OX(IH2))
```

```
      IF(ABS(RKANG) .LE. .01) RKANG=0.
      PSLP2=RKANG-1.5707963267
 195  CONTINUE
      IF(LCUT.EQ.11) WRITE(LUN6,196) IH2,D,CHORD,A,B,
     + XP2,ZP2,PSLP2
 196  FORMAT(8H0WHEELS2,I3,7F10.3)
      RETURN
      END
C
C
C
      SUBROUTINE WHEEL3 (ELEV,HA,HD,HE,HF,HX,IH,K,LOUT,
     + LUN6,XP,ZP,ZPROF)
      DIMENSION HA(5,9),HD(5,9),HE(5,9),HF(5,9),HX(5,10)
C
C SUBROUTINE TO FIND ELEVATION OF WHEEL CENTER
C AT XP,ZP,WRT HUE PROFILE
C
      DO 20 I=1,10
      IF(HX(K,I).GT.XP) GOTO 30
 20   CONTINUE
      IH=9
      GOTO 40
 30   IH=I-1
      IF(IH.LT.1) IH=1
C
C FIND POINT ON PROFILE
C
 40   IF(HA(K,IH).EQ.1.) GOTO 60
C
C PROFILE ELEMENT A LINE
C
      S=-HD(K,IH)/HE(K,IH)
      T=-HF(K,IH)/HE(K,IH)
      ZPROF=S*XP+T
      IF(LCUT.GE.11) WRITE(LUN6,56) IH,S,T,ZPROF
 56   FORMAT(9H WHEEL3/1,I3,3F10.3)
      GOTO 80
C
C PROFILE ELEMENT AN ARC
C
 60   B=.5*HE(K,IH)
      C=XP*XP+HD(K,IH)*XP+HF(K,IH)
      D=B*B-C
      IF(-B.GE.0.) Z1=-B+SQRT(D)
      IF(-B.LT.0.) Z1=-B-SQRT(D)
      Z2=C/Z1
      IF(Z1.GE.Z2) ZPROF=Z1
      IF(Z1.LT.Z2) ZPROF=Z2
      IF(LCUT.GE.11) WRITE(LUN6,71) IH,B,C,D,Z1,
     + Z2,ZPROF
 71   FORMAT(9H WHEEL3/2,I3,6F10.3)
C
C ELEVATION
```

```
C
 80     ELEV=ZP-ZPROF
        IF(LCUT.GE.11) WRITE(LUN6,86) XF,ZP,K,IH,
     +  ELEV,ZPROF
 86     FORMAT(9H WHEEL3/3,2F10.3,2I3,2F10.3)
        RETURN
        END
C
C       ..................................................
C
        SUBROUTINE MINV(A,N,D,L,M)
        DIMENSION A(1),L(1),M(1)
C
C       MATRIX INVERSION WITH PIVOTING
C       ..................................................
C
C           SEARCH FOR LARGEST ELEMENT
C
        D=1.0
        NK=-N
        DO 80 K=1,N
        NK=NK+N
        L(K)=K
        M(K)=K
        KK=NK+K
        BIGA=A(KK)
        DO 20 J=K,N
        IZ=N*(J-1)
        DO 20 I=K,N
        IJ=IZ+I
 10     IF (ABS(BIGA)-ABS(A(IJ))) 15,20,20
 15     BIGA=A(IJ)
        L(K)=I
        M(K)=J
 20     CONTINUE
C
C           INTERCHANGE ROWS
C
        J=L(K)
        IF(J-K) 35,35,25
 25     KI=K-N
        DO 30 I=1,N
        KI=KI+N
        HOLD=-A(KI)
        JI=KI-K+J
        A(KI)=A(JI)
 30     A(JI)=HOLD
C
C           INTERCHANGE COLUMNS
C
 35     I=M(K)
        IF(I-K) 45,45,33
 33     JP=N*(I-1)
        DO 40 J=1,N
```

```
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
 40   A(JI)=HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT
C        IS CONTAINED IN BIGA)
C
 45   IF(BIGA) 48,46,48
 46   D=0.0
      RETURN
 48   DO 55 I=1,N
      IF(I-K) 50,55,50
 50   IK=NK+I
      A(IK)=A(IK)/(-BIGA)
 55   CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
 60   IF(J-K) 62,65,62
 62   KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
 65   CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
 70   A(KJ)=A(KJ)/BIGA
 75   CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPRICAL
C
      A(KK)=1.0/BIGA
 80   CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
 100  K=(K-1)
```

```
      IF(K) 150,150,105
 105  I=L(K)
      IF (I-K) 120,120,108
 108  JC=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JC+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
 110  A(JI)=HOLD
 120  J=M(K)
      IF(J-K) 100,100,125
 125  KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
 130  A(JI)= HOLD
      GO TO 100
 150  RETURN
      END
C
C
      FUNCTION ATN2(X,Y)
      ATN2=0.
      IF(X.NE.0..OR.Y.NE.0.) ATN2=ATAN2(X,Y)
      RETURN
      END
C
C
C ****************************************************************************
C
C                         SUBROUTINE EQSOL
C
C ****************************************************************************
C     SUBROUTINE EQSOL - FROM M.J.D. POWELL -A FORTRAN SUBROUTINE
C              FOR SOLVING NONLINEAR ALGEBRAIC EQUATIONS
C             IN NUMERICAL METHODS FOR NONLINEAR ALGEBRAIC EQUATIONS
C             ED: PHILIP RABINOWITZ, PUB: GORDON & BREACH, 1970
      SUBROUTINE EQSOL (N,X,F,AJINV,DSTEP,DMAX,ACC,MAXFUN,
     1 W,MAXC,LUN6,IPRINT,CALFUN)
      DIMENSION X(N),F(N),AJINV(N,N),W(110),L(10),M(10)
      EXTERNAL CALFUN
C    SET VARIOUS PARAMETERS
      MAXC=0
C     'MAXC' COUNTS THE NUMBER OF CALLS OF CALFUN
      NT=N+4
      NTEST=NT
C     'NT' AND 'NTEST' CAUSE AN ERROR RETURN IF F(X) DOES
C     NOT DECREASE
      DTEST=FLOAT(N+N)-0.5
C     'DTEST' IS USED TO MAINTAIN LINEAR INDEPENDENCE
```

```
      NX=N*N
      NF=NX+N
      NW=NF+N
      MW=NW+N
      NDC=MW+N
      ND=NDC+N
C     THESE PARAMETERS SEPARATE THE WORKING SPACE
C     ARRAY W
      FMIN=0.
C     USUALLY 'FMIN' IS THE LEAST CALCULATED VALUE OF F(X),
C     AND THE BEST X IS IN W(NX+1) TO W(NX+N)
      DD=0.
C     USUALLY DD IS THE SQUARE OF THE CURRENT STEP LENGTH
      DSS=DSTEP*DSTEP
      DM=DMAX*DMAX
      DMM=4.*DM
      IS=5
C     'IS' CONTROLS A 'GO TO' STATEMENT FOLLOWING A CALL OF
C     CALFUN
      TINC=1.
C     'TINC' IS USED IN THE CRITERION TO INCREASE THE STEP
C     LENGTH
C     START A NEW PAGE FOR PRINTING
      IF(IPRINT)1,1,85
85    WRITE(LUN6,86)
86    FORMAT(1H1)
C     CALL THE SUBROUTINE CALFUN
1     MAXC=MAXC+1
      CALL CALFUN (N,X,F)
C     TEST FOR CONVERGENCE
      FSQ=0.
      DO 2 I=1,N
      FSQ=FSQ+F(I)*F(I)
2     CONTINUE
      IF (FSQ-ACC)3,3,4
C     PROVIDE PRINTING OF FINAL SOLUTION IF REQUESTED
3     CONTINUE
      IF (IPRINT)5,5,6
6     WRITE(LUN6,7)MAXC
7     FORMAT (///8H% EQSOL=/
     1 5X,39HTHE FINAL SOLUTION CALCULATED BY EQSOL
     2 8HREQUIRED,I5,23H CALLS OF CALFUN, AND IS)
      WRITE(LUN6,8)  (I,X(I),F(I),I=1,N)
8     FORMAT (//4X,1HI,7X,4HX(I),12X, 4HF(I)//(I5,2E17.8))
      WRITE(LUN6,9) FSQ
9     FORMAT (/5X,21HTHE SUM OF SQUARES IS,E17.8)
5     RETURN
C     TEST FOR ERROR RETURN BECAUSE F(X) DOES NOT DECREASE
4     GO TO (10,11,11,10,11),IS
10    IF(FSQ-FMIN)15,20,20
20    IF(DD-DSS)12,12,11
12    NTEST=NTEST-1
      IF(NTEST)13,14,11
14    WRITE(LUN6,16)NT
```

```
 16      FORMAT(///8H %EQSOL :/5X,31HERROR RETURN FROM EQSOL BECAUSE,I5,
     1   47HCALLS OF CALFUN FAILED TO IMPROVE THE RESIDUALS)
 17      DO 18 I=1,N
         NXI=NX+I
         NFI=NF+I
         X(I)=W(NXI)
         F(I)=W(NFI)
 18      CONTINUE
         FSQ= FMIN
         GO TO 3
C        ERROR RETURN BECAUSE A NEW JACOBIAN IS UNSUCCESSFUL
 13      WRITE(LUN6,19)
 19      FORMAT(///8H% EQSOL :/
     1   5X,36HERROR RETURN FROM EQSOL BECAUSE F(X),
     2   39HFAILED TO DECREASE USING A NEW JACOBIAN)
         GO TO 17
 15      NTEST=NT
C        TEST WHETHER THERE HAVE BEEN MAXFUN CALLS OF
C        CALFUN
 11      IF(MAXFUN-MAXC)21,21,22
 21      WRITE(LUN6,23)MAXC
 23      FORMAT(///8H% EQSOL :/
     1   5X,31HERROR RETURN FROM EQSOL BECAUSE
     2   16HTHERE HAVE BEEN ,I5,15HCALLS OF CALFUN)
         IF(FSQ-FMIN)3,17,17
C        PROVIDE PRINTING IF REQUESTED
 22      IF (IPRINT)24,24,25
 25      WRITE(LUN6,26) MAXC
 26      FORMAT(///8H% EQSOL:/
     1   5X,6HAT THE,I5,25HTH CALL OF CALFUN WE HAVE)
         WRITE(LUN6,8)(I,X(I),F(I),I=1,N)
         WRITE(LUN6,9)FSQ
 24      GO TO(27,28,29,87,32),IS
C        STORE THE RESULT OF THE INITIAL CALL OF CALFUN
 30      FMIN=FSQ
         DO 31 I=1,N
         NXI=NX+I
         NFI=NF+I
         W(NXI)=X(I)
         W(NFI)=F(I)
 31      CONTINUE
C        CALCULATE A NEW JACOBIAN APPROXIMATION
 32      IC=0
         IS=3
 33      IC=IC+1
         X(IC)=X(IC)+DSTEP
         GO TO 1
 29      K=IC
         DO 34 I=1,N
         NFI=NF+I
         W(K)=(F(I)-W(NFI))/DSTEP
         K=K+N
 34      CONTINUE
         NXIC=NX+IC
```

```
      X(IC)=W(NXIC)
      IF(IC-N)33,35,35
C     CALCULATE THE INVERSE OF THE JACOBIAN AND SET THE
C     DIRECTION MATRIX
 35   K=0
      DO 36 I=1,N
      DO 37 J=1,N
      K=K+1
      NDK=ND+K
      AJINV(I,J)=W(K)
      W(NDK)=0.
 37   CONTINUE
      NDCI=NDC+I
      NDCKI=NDCI+K
      W(NDCKI)=1.
      W(NDCI)=1.+FLOAT(N-I)
 36   CONTINUE
      CALL MINV(AJINV,N,DA,L,M)
C     START ITERATION BY PREDICTING THE DESCENT AND
C     NEWTON MINIMA
 38   DS=0.
      DN=0.
      SP=0.
      DO 39 I=1,N
      X(I)=0.
      F(I)=0.
      K=I
      DO 40 J=1,N
      NFJ=NF+J
      X(I)=X(I)-W(K)*W(NFJ)
      F(I)=F(I)-AJINV(I,J)*W(NFJ)
      K=K+N
 40   CONTINUE
      DS=DS+X(I)*X(I)
      DN=DN+F(I)*F(I)
      SP=SP+X(I)*F(I)
 39   CONTINUE
C     TEST WHETHER A NEARBY STATIONARY POINT IS
C     PREDICTED
      IF(FMIN*FMIN-DMM*DS)41,41,42
C     IF SO THEN RETURN OR REVISE JACOBIAN
 42   GO TO(43,43,44),IS
 44   WRITE(LUN6,45)
 45   FORMAT(///8H% EQSOL :/
     1 5X,33HERROR RETURN FROM EQSOL BECAUSE A,
     2 44HNEARBY STATIONARY POINT OF F(X) IS PREDICTED)
      GO TO 17
 43   NTEST=0
      DO 46 I=1,N
      NXI=NX+I
      X(I)=W(NXI)
 46   CONTINUE
      GO TO 32
C     TEST WHETHER TO APPLY THE FULL NEWTON CORRECTION
```

```
 41      IS=2
         IF(DN-DD)47,47,48
 47      DD=AMAX1(DN,DSS)
         DS=0.25*DN
         TINC=1.
         IF(DN-DSS)49,58,58
 49      IS=4
         GO TO 80
C        CALCULATE THE LENGTH OF THE STEEPEST DESCENT STEP
 48      K=0
         DMULT=0.
         DO 51 I=1,N
         DW=0.
         DO 52 J=1,N
         K=K+1
         DW=DW+W(K)*X(J)
 52      CONTINUE
         DMULT=DMULT+DW*DW
 51      CONTINUE
         DMULT=DS/DMULT
         DS=DS*DMULT*DMULT
C        TEST WHETHER TO USE THE STEEPEST DESCENT DIRECTION
         IF(DS-DD)53,54,54
C        TEST WHETHER THE INITIAL VALUE OF DD HAS BEEN SET
 54      IF(DD)55,55,56
 55      DD=AMAX1(DSS,AMIN1(DM,DS))
         DS=DS/(DMULT*DMULT)
         GO TO 41
C        SET THE MULTIPLIER OF THE STEEPEST DESCENT DIRECTION
 56      ANMULT=0.
         DMULT=DMULT*SQRT(DD/DS)
         GO TO 98
C        INTERPOLATE BETWEEN THE STEEPEST DESCENT AND THE
C        NEWTON DIRECTIONS
 53      SP=SP*DMULT
         ANMULT=(DD-DS)/((SP-DS)+SQRT((SP-DD)**2+(DN-DD)
     1   *(DD-DS)))
         DMULT=DMULT*(1.-ANMULT)
C        CALCULATE THE CHANGE IN XAND ITS ANGLE WITH THE
C        FIRST DIRECTION
 98      DN=0.
         SP=0.
         DO 57 I=1,N
         F(I)=DMULT*X(I)+ANMULT*F(I)
         DN=DN+F(I)*F(I)
         NDI=ND+I
         SP=SP+F(I)*W(NDI)
 57      CONTINUE
         DS=0.25*DN
C        TEST WHETHER AN EXTRA STEP IS NEEDED FOR
C        INDEPENDENCE
         IF(W(NDC+1)-DTEST)58,58,59
 59      IF(SP*SP-DS)60,58,58
C        TAKE THE EXTRA STEP AND UPDATE THE DIRECTION MATRIX
```

```
 50     IS=2
 60     DO 61 I=1,N
        NXI=NX+I
        NDI=ND+I
        NDCI=NDC+I
        X(I)=W(NXI)+DSTEP*W(NDI)
        W(NDCI)=W(NDCI+1)+1.
 61     CONTINUE
        W(ND)=1.
        DO 62 I=1,N
        K=ND+I
        SP=W(K)
        DO 63 J=2,N
        KN=K+N
        W(K)=W(KN)
        K=KN
 63     CONTINUE
        W(K)=SP
 62     CONTINUE
        GO TO 1
C       EXPRESS THE NEW DIRECTION IN TERMS OF THOSE OF THE
C       DIRECTION MATRIX,AND UPDATE THE COUNTS IN W(NDC+1)
C       ETC.
 58     SP=0.
        K=ND
        DO 64 I=1,N
        X(I)=DW
        DW=0.
        DO 65 J=1,N
        K=K+1
        DW=DW+F(J)*W(K)
 65     CONTINUE
        GOTO (68,66),IS
 66     NDCI=NDC+I
        W(NDCI)=W(NDCI)+1.
        SP=SP+DW*DW
        IF (SP-DS)64,64,67
 67     IS=1
        KK=I
        X(1)=DW
        GO TO 69
 68     X(I)=DW
 69     NDCI=NDC+I
        W(NDCI)=W(NDCI+1)+1.
 64     CONTINUE
        W(ND)=1.
C       REORDER THE DIRECTIONS SO THAT KK IS FIRST
        IF (KK-1)70,70,71
 71     KS=NDC+KK*N
        DO 72 I=1,N
        K=KS+I
        SP=W(K)
        DO 73 J=2,KK
        KN=K-N
```

```
      W(K)=W(KN)
      K=KN
 73   CONTINUE
      W(K)=SP
 72   CONTINUE
C     GENERATE THE NEW ORTHOGONAL DIRECTION MATRIX
 70   DO 74 I=1,N
      NWI=NW+I
      W(NWI)=0.
 74   CONTINUE
      SP=X(1)*X(1)
      K=ND
      DO 75 I=2,N
      DS=SQRT(SP*(SP+X(I)*X(I)))
      DW=SP/DS
      DS=X(I)/DS
      SP=SP+X(I)*X(I)
      DO 76 J=1,N
      K=K+1
      NWJ=NW+J
      KN=K+N
      W(NWJ)=W(NWJ)+X(I-1)*W(K)
      W(K)=DW*W(KN)-DS*W(NWJ)
 76   CONTINUE
 75   CONTINUE
      SP=1./SQRT(DN)
      DO 77 I=1,N
      K=K+1
      W(K)=SP*F(I)
 77   CONTINUE
C     CALCULATE THE NEXT VECTOR X,AND PREDICT THE RIGHT
C     HAND SIDES
 80   FNP=0.
      K=0
      DO 78 I=1,N
      NXI=NX+I
      NFI=NF+I
      NWI=NW+I
      X(I)=W(NXI)+F(I)
      W(NWI)=W(NFI)
      DO 79 J=1,N
      K=K+1
      W(NWI)=W(NWI)+W(K)*F(J)
 79   CONTINUE
      FNP=FNP+W(NWI)**2
 78   CONTINUE
C     CALL CALFUN USING THE NEW VECTOR OF VARIABLES
      GO TO 1
C     UPDATE THE STEP SIZE
 27   DMULT=0.9*FMIN+0.1*FNP-FSQ
      IF (DMULT)82,81,81
 82   DD=AMAX1(DSS,0.25*DD)
      TINC=1.
      IF (FSQ-FMIN)83,28,28
```

```
C         TRY THE TEST TO DECIDE WHETHER TO INCREASE THE STEP
C         LENGTH
 81       SP=0.
          SS=0.
          DO 84 I=1,N
          NWI=NW+I
          SP=SP+ABS(F(I)*(F(I)-W(NWI)))
          SS=SS+(F(I)-W(NWI))**2
 84       CONTINUE
          PJ=1.+DMULT /(SP+SQRT(SP*SP+DMULT*SS))
          SP=AMIN1(4.,TINC,PJ)
          TINC=PJ/SP
          DD=AMIN1(DM,SP*DD)
          GO TO 83
C         IF F(X) IMPROVES STORE THE NEW VALUE OF X
 87       IF(FSQ-FMIN)83,50,50
 83       FMIN=FSQ
          DO 88 I=1,N
          SP=X(I)
          NXI=NX+I
          NFI=NF+I
          NWI=NW+I
          X(I)=W(NXI)
          W(NXI)=SP
          SP=F(I)
          F(I)=W(NFI)
          W(NFI)=SP
          W(NWI)=-W(NWI)
 88       CONTINUE
          IF(IS-1)28,28,50
C         CALCULATE THE CHANGES IN F AND IN X
 28       DO 89 I=1,N
          NXI=NX+I
          NFI=NF+I
          X(I)=X(I)-W(NXI)
          F(I)=F(I)-W(NFI)
 89       CONTINUE
C         UPDATE THE APPROXIMATIONS TO J AND TO AJINV
          K=0
          DO 90 I=1,N
          MWI=MW+I
          NWI=NW+I
          W(MWI)=X(I)
          W(NWI)=F(I)
          DO 91 J=1,N
          W(MWI)=W(MWI)-AJINV(I,J)*F(J)
          K=K+1
          W(NWI)= W(NWI)-W(K)*X(J)
 91       CONTINUE
 90       CONTINUE
          SP=0.
          SS=0.
          DO 92 I=1,N
          DS=0.
```

```
      DO 93 J=1,N
      DS=DS+AJINV(J,I)*X(J)
 93   CONTINUE
      SP=SP+DS*F(I)
      SS=SS+X(I)*X(I)
      F(I)=DS
 92   CONTINUE
      DMULT=1.
      IF (ABS(SP)-0.1*SS)94,95,95
 94   DMULT=0.8
 95   PJ=DMULT/SS
      PA=DMULT/(DMULT*SP+(1.-DMULT)*SS)
      K=0
      DO 96 I=1,N
      NWI=NW+I
      MWI=MW+I
      SP=PJ*W(NWI)
      SS=PA*W(MWI)
      DO 97 J=1,N
      K=K+1
      W(K)=W(K)+SP*X(J)
      AJINV(I,J)=AJINV(I,J)+SS*F(J)
 97   CONTINUE
 96   CONTINUE
      GO TO 38
      END
```

APPENCIX E

VEHICLE INFUT FILES FUR PROGRAM OBS78B

```
M60LIM2
1 2 0 1                                            NUNITS,NSUSP,NVEH,NFL
 40.      0.                                       HITCH HEIGHT AND LOAD
1 1                                                BOGIE INDICATORS
1 1 1 1                                            POWER INDICATORS
1 1 1 1                                            BRAKE INDICATORS
 17.5    17.5                                      ROLLING RADIUS
186.0    86.0                                      HITCH TO SUPPORT CENTER
 33.3    33.3                                      BOGIE WIDTH
 30.      7.                                       BOGIE LIMIT-UP
 -7.    -30.                                       BOGIE LIMIT-DOWN
61246. 47754.                                      AXLE LOAD-EMPTY
 53.62    0.                                       VEH. CG ABOVE GROUND
144.2    53.62      0.        0.                   LOAD CG WRT GROUND
  0.      0.                                       LOAD
2 0                                                VEH BOTTOM POINTS  NPTSC1,NPTSC2
273.5    45.       0.       40.                              XCLC1(I),YCLC1(I),I=1,NPTSC1
0 1 1 0 1 1                                        SFLAG(I),IP(I,1),IB(I,1),I=4,5
253.31   40.      17.62    23.6     41.25   14.62        ELL(I),ZS(I),EFFRAD(I),
                                                                  I=4,5
```

```
M151A2 - 4X4
 1 2 1 0                                                  NUNITS,NSUSP,NVEHL,NFL
    18.      0.                                           HITCH HEIGHT AND LOAD
 0 0 0                                                    BOGIE INDICATORS
 1 0 1 0 0 0                                              POWER INDICATORS
 1 0 1 0 0 0                                              BRAKE INDICATORS
    14.     14.      0.                                   ROLLING RADIUS
   113.     28.      0.                                   HITCH TO SUPPORT POINT
     0.      0.      0.                                   BOGIE WIDTH
     0.      0.      0.                                   BOGIE LIMIT-UP
     0.      0.      0.                                   BOGIE LIMIT-DOWN
  1340.   1060.      0.                                   AXLE LOAD-EMPTY
    25.     18.                                           VEH. CG ABOVE GROUND
    56.     30.      0.      0.                           LOAD CG WRT GROUND
   500.      0.                                           LOAD
 9 0                                                      VEH BOTTOM POINTS
   132.     17.    123.     10.     88.  13.15   86.   12.   85.  13.15
    47.     14.     26.     10.     13.   18.     0.   18.
```

APPENDIX C

SAMPLE TERRAIN INPUT FILE FOR PROGRAM OBS78B

SAMPLE TERRAIN INPUT FILE FOR PROGRAM OBS78B

```
03
    0.
       08          03            03
     3.15       112.00          5.88
    15.75       112.00          5.88
    33.46       112.00          5.88
     3.15       142.00          5.88
    15.75       142.00          5.88
    33.46       142.00          5.88
     3.15       154.00          5.88
    15.75       154.00          5.88
    33.46       154.00          5.88
     3.15       164.00          5.88
    15.75       164.00          5.88
    33.46       164.00          5.88
     3.15       196.00          5.88
    15.75       196.00          5.88
    33.46       196.00          5.88
     3.15       206.00          5.88
    15.75       206.00          5.88
    33.46       206.00          5.88
     3.15       218.00          5.88
    15.75       218.00          5.88
    33.46       218.00          5.88
     3.15       248.00          5.88
    15.75       248.00          5.88
    33.46       248.00          5.88
     3.15       112.00         29.88
    15.75       112.00         29.88
    33.46       112.00         29.88
     3.15       142.00         29.88
    15.75       142.00         29.88
    33.46       142.00         29.88
     3.15       154.00         29.88
    15.75       154.00         29.88
    33.46       154.00         29.88
     3.15       164.00         29.88
    15.75       164.00         29.88
    33.46       164.00         29.88
     3.15       196.00         29.88
    15.75       196.00         29.88
    33.46       196.00         29.88
     3.15       206.00         29.88
    15.75       206.00         29.88
    33.46       206.00         29.88
     3.15       218.00         29.88
    15.75       218.00         29.88
    33.46       218.00         29.88
     3.15       248.00         29.88
    15.75       248.00         29.88
    33.46       248.00         29.88
     3.15       112.00        141.60
    15.75       112.00        141.60
    33.46       112.00        141.00
```

3.15	142.00	141.60
15.75	142.00	141.60
33.46	142.00	141.60
3.15	154.00	141.60
15.75	154.00	141.60
33.46	154.00	141.60
3.15	164.00	141.60
15.75	164.00	141.60
33.46	164.00	141.60
3.15	196.00	141.60
15.75	196.00	141.60
33.46	196.00	141.60
3.15	206.00	141.60
15.75	206.00	141.60
33.46	206.00	141.60
3.15	218.00	141.60
15.75	218.00	141.60
33.46	218.00	141.60
3.15	248.00	141.60
15.75	248.00	141.60
33.46	248.00	141.60

9999999.99999999.99999999.99

APPENDIX C

SAMPLE OUTPUT FROM PROGRAM OBS78B

NUHGT
3
NANG
8
NWDTH
3

CLRMIN INCHES	FCOMAX POUNDS	FCC POUNDS	HOVALS INCHES	AVALS RADIANS	WVALS INCHES
37.Ø3	8948.5	372.1	3.15	1.95	5.88
24.42	27Ø76.2	1842.Ø	15.75	1.95	5.88
6.57	89773.8	5211.1	33.46	1.95	5.88
37.Ø3	8948.5	394.3	3.15	2.48	5.88
24.38	24473.2	16Ø4.8	15.75	2.48	5.88
6.72	5Ø134.8	38ØØ.Ø	33.46	2.48	5.88
37.Ø3	8948.5	399.Ø	3.15	2.69	5.88
24.56	18569.2	139Ø.5	15.75	2.69	5.88
11.43	32415.7	3Ø16.3	33.46	2.69	5.88
36.9Ø	8456.Ø	386.8	3.15	2.86	5.88
24.3Ø	17646.6	1259.3	15.75	2.86	5.88
2Ø.43	3Ø244.5	2787.9	33.46	2.86	5.88
38.22	8281.7	7Ø7.Ø	3.15	3.42	5.88
21.27	18699.8	2246.3	15.75	3.42	5.88
2.87	3Ø244.5	2696.Ø	33.46	3.42	5.88
39.64	4124.4	224.7	3.15	3.6Ø	5.88
31.Ø1	13744.6	1544.Ø	15.75	3.6Ø	5.88
-1.3Ø	3Ø816.3	2642.5	33.46	3.6Ø	5.88
4Ø.ØØ	3757.7	174.5	3.15	3.8Ø	5.88
36.Ø3	13166.8	982.9	15.75	3.8Ø	5.88
2Ø.Ø1	31678.1	262Ø.5	33.46	3.8Ø	5.88
4Ø.ØØ	1612.7	3Ø.6	3.15	4.33	5.88
39.54	4149.3	145.9	15.75	4.33	5.88
37.79	5566.1	-125.5	33.46	4.33	5.88
37.13	9272.2	484.4	3.15	1.95	29.88
24.26	12489.2	-316.4	15.75	1.95	29.88
6.57	79647.8	4974.4	33.46	1.95	29.88
37.13	9272.2	5ØØ.Ø	3.15	2.48	29.88
24.22	2Ø272.6	862.5	15.75	2.48	29.88
6.62	51346.5	4342.5	33.46	2.48	29.88
37.13	9272.2	516.7	3.15	2.69	29.88
24.36	2Ø378.Ø	1717.Ø	15.75	2.69	29.88
11.72	34Ø87.7	3769.5	33.46	2.69	29.88
36.99	8456.Ø	527.7	3.15	2.86	29.88
24.57	15926.4	1465.5	15.75	2.86	29.88
2Ø.55	3Ø244.5	3131.9	33.46	2.86	29.88
37.17	8448.1	629.9	3.15	3.42	29.88
14.79	18895.7	1864.3	15.75	3.42	29.88
2.92	3Ø244.5	334Ø.6	33.46	3.42	29.88
36.88	72Ø8.2	-219.2	3.15	3.6Ø	29.88
22.Ø8	31861.Ø	2261.9	15.75	3.6Ø	29.88
-11.56	34784.1	3152.8	33.46	3.6Ø	29.88
36.71	9361.9	1ØØ1.2	3.15	3.8Ø	29.88
27.21	2Ø261.7	1637.8	15.75	3.8Ø	29.88
3.49	48386.8	4522.6	33.46	3.8Ø	29.88
38.68	5564.9	196.1	3.15	4.33	29.88

37.04	7279.0	-102.6	15.75	4.33	29.88
35.01	12253.2	759.8	33.46	4.33	29.88
37.17	9272.2	231.1	3.15	1.95	141.60
24.77	20814.9	1040.4	15.75	1.95	141.60
6.59	79704.9	4401.1	33.46	1.95	141.00
37.17	9272.2	236.3	3.15	2.48	141.60
24.44	35968.2	1861.0	15.75	2.48	141.60
6.62	52815.6	3648.1	33.46	2.48	141.60
37.17	9272.2	241.8	3.15	2.69	141.60
24.40	27603.5	1707.9	15.75	2.69	141.60
11.59	34088.9	3306.2	33.46	2.69	141.60
36.93	8456.0	429.9	3.15	2.86	141.60
24.46	18740.7	1827.2	15.75	2.86	141.60
20.55	30044.5	3062.1	33.46	2.86	141.60
34.03	8295.3	471.2	3.15	3.42	141.60
22.76	19012.2	2295.4	15.75	3.42	141.60
20.46	30044.5	3493.0	33.46	3.42	141.60
34.12	9326.8	741.4	3.15	3.60	141.60
16.75	32341.8	2497.8	15.75	3.60	141.60
9.38	34368.4	4266.5	33.46	3.60	141.60
33.89	9787.3	452.9	3.15	3.80	141.60
12.40	38383.1	2027.9	15.75	3.80	141.60
-1.83	48928.4	3741.5	33.46	3.80	141.60
33.91	8474.2	608.2	3.15	4.33	141.60
10.90	18269.4	955.9	15.75	4.33	141.60
-23.03	79892.1	5167.6	33.46	4.33	141.60

SAMPLE OUTPUT FROM PROGRAM CBS78B - VEHICLE:M151 JEEP

NOHGT
3
NANG
8
NWDTH
3

CLRMIN INCHES	FCOMAX POUNDS	FOO POUNDS	HOVALS INCHES	AVALS RADIANS	WVALS INCHES
6.85	941.6	31.2	3.15	1.95	5.88
-3.75	2179.6	127.1	15.75	1.95	5.88
-21.21	2208.5	237.5	33.46	1.95	5.88
6.85	1015.5	35.6	3.15	2.48	5.88
-3.54	1061.2	116.7	15.75	2.48	5.88
-13.36	960.9	160.6	33.46	2.48	5.88
6.85	696.1	25.5	3.15	2.69	5.88
-2.31	696.7	124.9	15.75	2.69	5.88
-3.95	646.3	98.2	33.46	2.69	5.88
7.45	411.2	34.3	3.15	2.86	5.88
2.93	424.0	69.7	15.75	2.86	5.88
2.61	799.3	98.3	33.46	2.86	5.88
7.19	417.7	40.9	3.15	3.42	5.88
5.50	444.5	88.7	15.75	3.42	5.88
3.10	799.3	103.9	33.46	3.42	5.88
7.42	704.7	35.5	3.15	3.60	5.88
1.20	757.6	135.1	15.75	3.60	5.88
-4.83	839.1	135.3	33.46	3.60	5.88
8.20	662.5	16.3	3.15	3.80	5.88
.08	1170.4	180.3	15.75	3.80	5.88
-9.54	1301.5	240.0	33.46	3.80	5.88
9.65	344.3	4.8	3.15	4.33	5.88
5.79	1150.8	43.5	15.75	4.33	5.88
-.23	2378.0	146.0	33.46	4.33	5.88
6.85	592.1	-2.8	3.15	1.95	29.88
-3.75	2163.4	99.1	15.75	1.95	29.88
-21.46	2029.6	150.9	33.46	1.95	29.88
6.85	1015.5	29.3	3.15	2.48	29.88
-3.75	1052.4	98.4	15.75	2.48	29.88
-4.92	1110.3	109.8	33.46	2.48	29.88
6.85	698.1	24.7	3.15	2.69	29.88
.59	658.0	69.2	15.75	2.69	29.88
.52	837.9	116.9	33.46	2.69	29.88
7.45	411.2	28.8	3.15	2.86	29.88
4.86	443.4	50.1	15.75	2.86	29.88
4.75	799.3	105.0	33.46	2.86	29.88
7.29	417.6	31.1	3.15	3.42	29.88
5.40	444.5	57.0	15.75	3.42	29.88
4.92	799.3	100.6	33.46	3.42	29.88
6.63	708.6	39.9	3.15	3.60	29.88
.78	761.3	119.2	15.75	3.60	29.88
-2.82	842.2	137.0	33.46	3.60	29.88
6.70	991.4	34.9	3.15	3.80	29.88
-2.46	1178.4	145.1	15.75	3.80	29.88
-10.26	1318.0	195.9	33.46	3.80	29.88
6.68	575.1	4.9	3.15	4.33	29.88

-3.01	2401.8	157.0	15.75	4.33	29.88
-23.83	2551.4	228.7	33.46	4.33	29.88
6.85	541.3	-6.0	3.15	1.95	141.60
-.52	2428.4	87.4	15.75	1.95	141.60
-11.40	2556.1	128.8	33.46	1.95	141.00
6.85	1093.9	18.1	3.15	2.48	141.60
2.04	1170.6	68.6	15.75	2.48	141.60
-.73	1304.9	145.9	33.46	2.48	141.60
6.85	707.5	16.9	3.15	2.69	141.60
4.40	758.7	75.1	15.75	2.69	141.60
3.83	837.9	132.5	33.46	2.69	141.60
7.45	416.8	17.0	3.15	2.86	141.60
6.75	443.4	65.4	15.75	2.86	141.60
6.88	799.3	123.0	33.46	2.86	141.60
7.67	417.2	19.1	3.15	3.42	141.60
7.28	388.0	65.9	15.75	3.42	141.60
6.85	799.3	106.6	33.46	3.42	141.60
6.84	707.1	20.1	3.15	3.60	141.60
4.25	760.1	78.2	15.75	3.60	141.60
3.88	839.7	135.9	33.46	3.60	141.60
7.08	1094.0	18.6	3.15	3.80	141.60
2.04	1168.7	83.3	15.75	3.80	141.60
-.60	1312.2	164.2	33.46	3.80	141.60
6.80	1131.4	30.3	3.15	4.33	141.60
-.03	2397.2	80.3	15.75	4.33	141.60
-15.46	2549.8	147.3	33.46	4.33	141.60

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 12503	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NATO REFERENCE MOBILITY MODEL, EDITION I USERS GUIDE, VOLUME II		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER SIT-DL-79-9-2058
7. AUTHOR(s) M. Peter Jurkat, Peter M. Brady, Jr., Peter W. Haley		8. CONTRACT OR GRANT NUMBER(s) DAAK30-77-C-0027
9. PERFORMING ORGANIZATION NAME AND ADDRESS Stevens Institute of Technology, Davidson Lab Castle Point Station, Hoboken, NJ 07030		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Tank-Automotive Rsch & Dev Command DRDTA-ZSA Warren, MI 48090		12. REPORT DATE OCT 1979
		13. NUMBER OF PAGES 165
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Includes: Obstacle Module; App A: Program Listing; App B: Vehicle Input Files; AppC: Terrain Input Files; App D: Sample Output of Program

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Mobility	Vehicle Performance
Mobility Modeling	Terrain
Computerized Simulation	Obstacle Crossing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Instructions in the organization and use of the computer programs which implement the Initial NATO Reference Mobility Model (INRMM) are presented. Volume II is devoted to the INRMM Obstacle-Crossing Module. A brief description of the mathematical equations and computing algorithms which predict the speed of a vehicle over a variety of terrain, the input data required, and the outputs generated is included. Some aid to the interpretation of various output variables is given.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)