ERDC/GSL TR-09-8

Geotechnical and Structures Laboratory

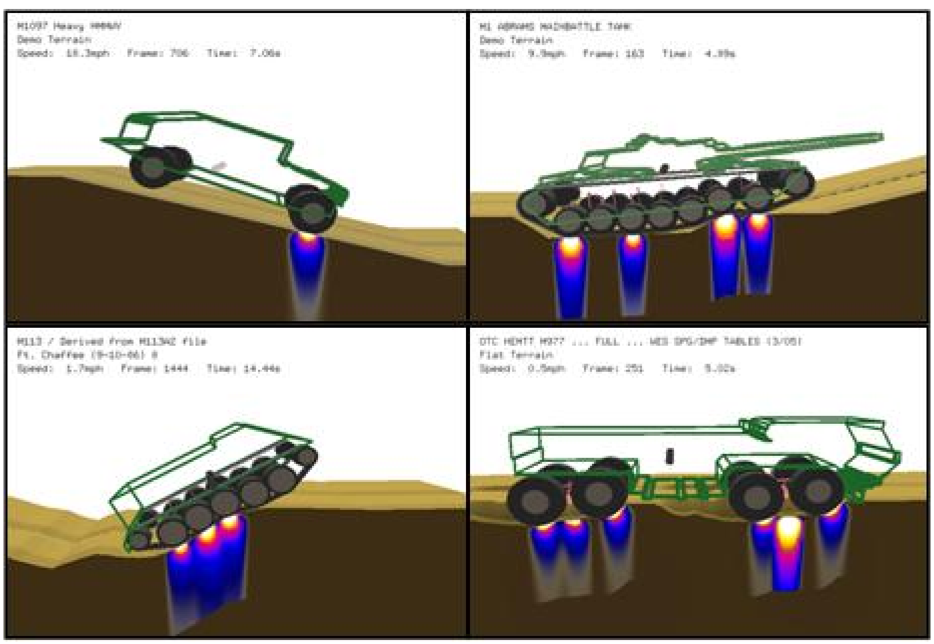
US Army Corps of Engineers®
Engineer Research and Development Center

Terrain Mechanics and Modeling Research Program

Enhanced Vehicle Dynamics Module

Daniel C. Creighton, George B. McKinley,
Randolph A. Jones, and Richard B. Ahlvin

May 2009



Terrain Mechanics and Modeling
Research Program

ERDC/GSL TR-09-8
May 2009

Enhanced Vehicle Dynamics Module

Daniel C. Creighton, George B. McKinley,
and Randolph A. Jones

Geotechnical and Structures Laboratory
U.S. Army Engineer Research and Development Center
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Richard B. Ahlvin

RBA Technical Services
3 Shadow Wood Drive
Vicksburg, MS 39180-9741

Final report

Prepared for Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000

Abstract: The vehicle dynamics module (VEHDYN 4.0) includes many enhancements to the previously documented and released VEHDYN II model and the internally developed, but not released, VEHDYN 3.0 model. The VEHDYN 3.0 model accepted the vehicle in a settled, or equilibrium, configuration instead of the "zero-force" configuration (the force acting on each suspension spring is assumed to be zero) required by VEHDYN II. VEHDYN 3.0 also included a special suspension and a band track model that were required to model the Caterpillar 30/30 engineering vehicle. VEHDYN 3.0 was further enhanced to add several other effects including: (1) a vehicle underside interference model that computes the normal and drag forces that occur when the underside of a vehicle's sprung mass comes in contact with the terrain profile, (2) a wheel damping model which computes an additional normal force component to represent the wheel's resistance to change, (3) a vehicle-terrain interface model to allow the terrain to deform, (4) accelerating and decelerating the vehicle using tractive force-speed relations and the soil-slope resistance, (5) rotational springs for enhanced suspension beam modeling, (6) a water-vehicle buoyancy and drag model to be used when a vehicle crosses a water-filled terrain obstacle, (7) numerous new run-type options including multi-run ride and shock performance sequences, slope climbing, obstacle crossing, etc., and (8) new postprocessors for animation. This report documents the VEHDYN 4.0 program, including the new enhancements, the input and output files and 12 example problems.

Contents

Figures and Tables

Figures

Tables

Preface

The study reported herein was conducted by personnel of the U.S. Army Engineer Research and Development Center (ERDC), Geotechnical and Structures Laboratory (GSL), Engineering Systems and Materials Division (ESMD), Mobility Systems Branch (MSB), Vicksburg, MS.

The work was conducted under the Military Engineering Business Area of the Terrain Mechanics and Modeling Research Program. The work was conducted between October 2002 and September 2006.

Overall development of the research was accomplished by George B. McKinley, Daniel C. Creighton, Randolph A. Jones, and Flossie B. Ponder, all of MSB, along with Richard B. Ahlvin, RBA Technical Services. Creighton, McKinley, Jones, and Ahlvin prepared this report.

The study was conducted under the general supervision of M. Wendell Gray, Chief, MSB; Dr. Larry N. Lynch, Chief, ESMD; Dr. William P. Grogan, Deputy Director, GSL; and Dr. David W. Pittman Director, GSL.

COL Gary E. Johnston was Commander and Executive Director of ERDC. Dr. James R. Houston was Director.

Unit Conversion Factors

Multiply	By	To Obtain
cubic inches	1.6387064 E-05	cubic meters
degrees (angle)	0.01745329	radians
	0.3048	meters
foot-pounds force	1.355818	joules
inches	0.0254	meters
miles per hour	0.44704	meters per second
pounds (force)	4.448222	newtons
pounds (force) per inch	175.1268	newtons per meter
pounds (mass)	0.45359237	kilograms
square feet	0.09290304	square meters
square inches	6.4516 E-04	square meters

1 Introduction

Background

The Vehicle Dynamics Module, referred to as VEHDYN, was developed by the Mobility Systems Division (MSD) of the Waterways Experiment Station (WES) in 1974 (Murphy and Ahlvin 1976) to provide ride and shock simulation capability for general use in support of the Army Mobility Model (AMM). The MSD is now the Mobility Systems Branch within the Engineering Systems and Materials Division, Geotechnical and Structures Laboratory (GSL), U.S. Army Engineer Research and Development Center (ERDC).

VEHDYN's primary purpose was to predict the gross motion of a vehicle chassis when negotiating rough terrain or discrete obstacles and to calculate the resulting absorbed power (a ride comfort criterion) or largest peak acceleration (shock criterion) from the vertical accelerations at a specified location in the vehicle. The VEHDYN predictions were used to calculate ride- and shock-limiting speeds (typically the speeds at which 6 W absorbed power or peak accelerations of 2.5 g's occur) as a function of terrain roughness or obstacle height. The intent of the VEHDYN model was not to assess the intricate effects of various vehicle components on ride dynamics, but rather to readily lump gross mass and suspension properties and reasonably predict ride-and shock-limiting speed relations.

In 1978, a North Atlantic Treaty Organization (NATO) working group composed of representatives from the United States, Canada, France, the Federal Republic of Germany, the Netherlands, and the United Kingdom adopted the AMM and its supporting VEHDYN module as standard references for evaluating the cross-country mobility performance of vehicles. The AMM was subsequently called the NATO Reference Mobility Model.

This adoption of VEHDYN as a standard reference cast it immediately into widespread usage, which brought about requests for immediate modifications to meet specific needs. After a variety of modifications by different users, the VEHDYN program evolved into one containing numerous inconsistencies, programming errors, redundant variables, and an unwieldy structure. In addition, as a result of these uncoordinated modifications, no suitable documentation was available for users.

Subsequently, it was decided to completely reprogram VEHDYN and develop a manual that would not only indicate how to use the program but also clearly explain how the program works. The first phase of this reprogramming effort culminated in the release of the VEHDYN II program and its corresponding user's guide (Creighton 1986).

One year later, another program, called PREVDYN2, was released along with its user's guide (Creighton 1987). PREVDYN2 is a preprocessor for VEHDYN II that allows a user to input vehicle data in a settled, or equilibrium, configuration rather than the "zero-force" configuration (zero force and no displacement in each spring) required by VEHDYN II. The settled configuration is much easier to measure from a vehicle. Since the release of PREVDYN2, very little work was done on the vehicle dynamics module until spring 1990.

The Belvoir Research, Development and Engineering Center (BRDEC), Fort Belvoir, VA, asked the WES to assist Caterpillar, Inc., in characterizing the dynamics and performance of a rubber-tracked vehicle called the 30/30 Engineer Support Tractor. This effort was actually one stage of a larger program to evaluate the effectiveness of Caterpillar's Mobil-Trac System, an undercarriage technology that was to provide on- and off-road mobility by combining features of tires and steel track. VEHDYN II was too simple a model to provide such an evaluation, for two primary reasons:

- The front-end suspension of the 30/30 was too complicated for VEHDYN II to model without developing a new suspension type.
- The track tension model in VEHDYN II was inadequate to handle the modeling of the rubber track.

A decision was made to modify VEHDYN II so that this characterization could be performed, which generated the VEHDYN 3.0 model. VEHDYN 3.0 was later enhanced to add several other features, including

- A vehicle underside interference model that computes the normal and drag forces that occur when the underside of a vehicle's sprung mass comes in contact with the terrain profile.
- A wheel damping model that computes an additional normal force component to represent the wheel's resistance to change.
- A vehicle-terrain interface (VTI) model to allow the terrain to deform.

- Accelerating and decelerating (ACC/DEC) the vehicle using tractive force-speed relations and soil-slope resistance.
- Rotational springs for enhanced suspension beam modeling.
- A water-vehicle buoyancy and drag model to be used when a vehicle crosses a water-filled terrain obstacle.
- Numerous new run-type options, including multi-run ride and shock performance sequences, slope climbing, obstacle crossing, etc.
- New postprocessors for animation.

The current version of the vehicle dynamics model is called VEHDYN 4.0. This report documents this program, including the new enhancements; specifies the input and output files; presents 12 sample problems, and serves as a user's guide for VEHDYN 4.0.

Scope

As with the VEHDYN II user's guide, this report attempts to teach the user of VEHDYN 4.0 how to set up a particular problem of interest (i.e., develop the required input files), run the problem, and interpret the results. Chapter 2 discusses the representation of the vehicle in VEHDYN 4.0 with particular emphases on the input configuration, improvements to existing suspensions, the various track model options, translational and rotational element model assumptions, the normal wheel damping model, the underside impact detection (UID) model, and the buoyancy resistance model, in addition to an example procedure for adding new suspensions. Chapter 3 presents an overview of the run capability of VEHDYN 4.0, including listing the numerous available run types, the required input and output files, and discussion of the postprocessing support programs. Chapter 4 presents input guides for each of the seven input files to assist the prospective user of VEHDYN 4.0. Chapter 5 demonstrates the use of VEHDYN 4.0 with sample problems. Chapter 6 discusses suggestions and expectations for future enhancements to the ERDC vehicle dynamics module. Appendix A is a detailed model discussion of the VEHDYN 4.0 normal wheel damping model. Appendix B similarly details the underside impact detection model in VEHDYN 4.0. Appendix C examines the vehicle-terrain interaction logic used to model deformable terrain in VEHDYN 4.0. Appendix D discusses the new band track model with options for uniform or local tension. Appendix E provides the variable-speed run model in which vehicle driving forces are applied to accelerate and decelerate a vehicle. Finally, Appendix F presents the logic used to model buoyancy and water drag forces acting on a fording vehicle.

2 Vehicle Representation

Background

In the process of reprogramming VEHDYN to produce VEHDYN II, several changes were made in the way the vehicle was modeled. The model was made fully two-dimensional, bogie and walking beam suspensions were enhanced, and spring and damper representations were changed to allow hysteretic effects. The input configuration was defined as a “zero-force” configuration in which the force acting on each suspension spring is assumed to be zero. Admittedly, this configuration is not one that, in general, represents a measurable state for a test vehicle. Rather, it is a mathematical convenience, as it gives the modeler a known starting or reference point (i.e., zero force and zero displacement in each spring). Recognizing the difficulty in establishing the data for this configuration for the average user, the WES developed the preprocessor PREVDYN2 that allows input of data in a settled, or equilibrium, configuration, a representation allowing data to easily be field-measured from a vehicle sitting on level ground.

The solution process in VEHDYN II involved two procedures, settlement of the vehicle under the force of gravity to equilibrium followed by the transient phase of either obstacle negotiation or ride course travel. The settlement phase uses an iterative stiffness method to obtain the initial spring displacements. This procedure has, from time to time, failed to converge for certain vehicle models. The VEHDYN 4.0 model no longer has this phase of the solution process, as the user inputs the settled configuration parameters directly and VEHDYN 4.0 works from that starting point.

Another weak part of earlier VEHDYN versions has been the track tension model. The primary features of the VEHDYN II track tension model involved interconnecting vertical linear springs between neighboring wheels inside the track envelope, as well as fore and aft “feelers” (sections of track that could provide resistance when they intersected the terrain profile). One obvious weakness with this model is that tracked vehicles would not demonstrate a “bridging” effect for road wheels in which the profile could deflect a section of track between road wheels, thereby causing an upward force on the wheels fore and aft of the profile protrusion. VEHDYN 4.0 has several track tension models that all include some type of "bridging" effect. The new models include a modification to the original interconnecting spring model that incorporates a terrain smoothing function to estimate the effect of bridging and two band track models where an actual track loop is calculated each time step with its resulting tension directly applied to each road wheel and each sprocket or idler (SPRIDLER).

VEHDYN 4.0 includes some other enhancements, including the addition of full hysteretic rotational springs to both the bogie and walking beam models, an example of a more complicated suspension (the front end of the Caterpillar 30/30 Engineer Support Tractor), and a postprocessing animation facility for personal computers. This chapter describes the changes to the manner in which a user models a vehicle, including some of the new available options. Other available options are shown in Chapter 4, where a line-by-line input guide is provided for each input file.

Direct input of settled configuration

Creating vehicle input for VEHDYN II was actually a two-step process. First, the user would develop a data set for a vehicle in equilibrium (settled) configuration. Those data were then input to the preprocessor PREVDYN2 to produce a vehicle data set in the "zero-force" configuration. This "zero-force" data is the format required as input to VEHDYN II.

The first change the user encounters with VEHDYN 4.0 is that there is no longer a preprocessor. The user develops vehicle data similar to the equilibrium configuration input for PREVDYN2 for direct input to VEHDYN 4.0. The iterative stiffness methodology is no longer used to settle the vehicle under its own weight from the "zero-force" position to equilibrium. Rather, VEHDYN 4.0 assumes that the input vehicle data represent the vehicle already in its equilibrium configuration.

As a simple example, look at the vehicle with two independent suspensions on each side pictured in Figure 1. The half-vehicle[1] has a sprung weight W_s, fore and aft wheel assembly weights W_1 and W_2, respectively, and forces F_1

---

[1] VEHDYN programs have always solved a "half-vehicle" problem. The vehicle's entire weight and moment of inertia are input, but the program takes one-half of each one and solves the problem using wheel assemblies, beam parameters, and spring/damper curves for one side of the vehicle only.

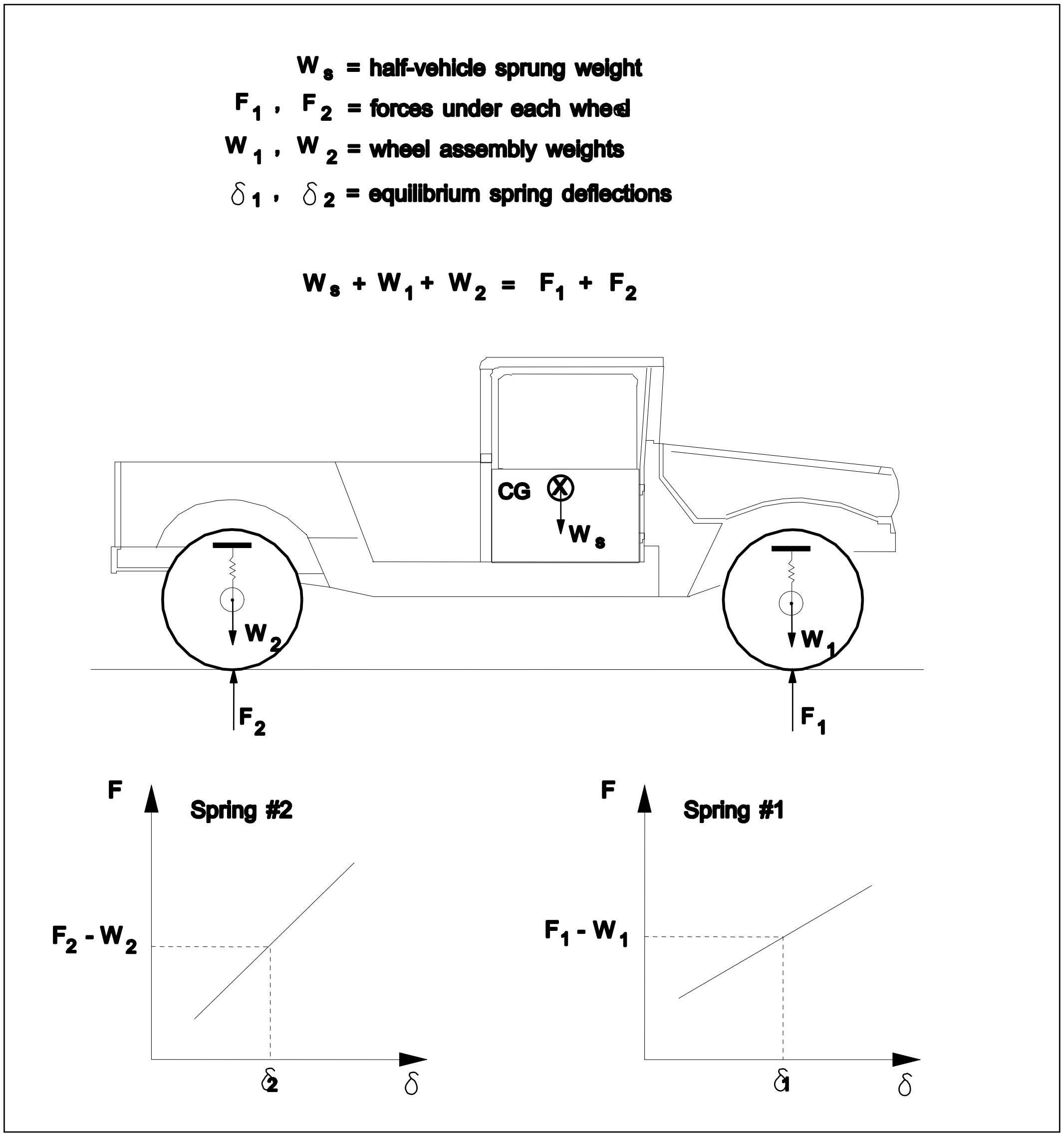


Figure 1. Static vehicle depiction for two-independent-suspension generic vehicle.

and F_2 under the fore and aft wheels, respectively. Thus, Equation 1 is a statement of equilibrium.

$$W_S + W_1 + W_2 = F_1 + F_2 \tag{1}$$

In both VEHDYN and VEHDYN II, an iterative stiffness solution methodology is employed to determine how much compression exists in each suspension spring at equilibrium. This information is required to initialize suspension spring displacements prior to the start of the transient phase of the problem.

Knowing the force under each wheel, however, allows direct computation of all of the suspension spring compressive displacements, regardless of the geometrical complexity of any of the suspensions. As seen in Figure 1 for a simple case, the displacement for each spring is selected directly from the associated spring force-deflection loading curve for a force equal to the load on top of the suspension. This load is the force under the wheel minus the wheel assembly weight. These equilibrium displacements in each suspension spring are the reference displacements needed to begin the next phase of the solution, that of moving the vehicle down the course.

Two questions arise at this point:

- Why change from the iterative stiffness method (ISM) used to settle the vehicle in earlier versions of VEHDYN?
- What potential problems are inherent with inputting forces under each wheel and computing spring displacements directly?

Following is a simple summary of how the ISM functions in VEHDYN II obtain vehicle equilibrium. Assuming there are *n* degrees of freedom (DOF) in our subject vehicle, approximate relationships can be developed (for relatively simple suspensions) of the form

$$\begin{aligned} K_{11}Z_1 + K_{12}Z_2 + \ldots + K_{1n}Z_n &= F_1 \\ K_{21}Z_1 + K_{22}Z_2 + \ldots + K_{2n}Z_n &= F_1 \\ &\vdots \\ K_{n1}Z_1 + K_{n2}Z_2 + \ldots + K_{nn}Z_n &= F_n \end{aligned} \tag{2}$$

where the matrix [K] is the n-by-n equilibrium stiffness matrix for this vehicle, the array [Z] is the set of displacements for the n DOF, and the array [F] is the set of forces associated with each of the n DOF. This method can only be applied to systems where we encounter neither any cross terms between two or more displacements nor any nonlinear functions of the displacements in any of the n equations above. These restrictions limit this methodology to relatively simple suspensions and small angular displacements.

The process begins by guessing a set of displacements [Z]. The set of stiffnesses [K] are, in general, nonlinear functions of [Z]. Given the forces [F], the simultaneous set of n equations is solved using a Gaussian elimination technique, obtaining a new set of [Z]. This procedure iterates until the array [Z] converges.

The problem with this process is how to "hardwire" into the model a set of initial [Z] guesses that leads to a converged solution for any vehicle. Specifically, a problem can arise with a spring force-displacement relationship that is not smoothly varying in the region of the curve at or near a guess of the spring displacement (which is a direct function of one or more of the displacements [Z]). This type of problem sometimes results in a solution procedure that will not converge. The user then has to determine which spring curve(s) is the offender and modify it so that the process will converge.

Another problem with the ISM concerns the restriction mentioned previously regarding no cross terms and no nonlinear terms in each of the n equations. This restriction proved to be a real problem in the spring of 1990 when the WES was tasked to model a vehicle (Caterpillar's 30/30 Engineer Support Tractor) whose front end contained a much more complicated suspension than could be modeled with any of the suspension elements already in VEHDYN II. A discussion of the development of the 30/30 front end is found later in this report. It can be shown that if the forces under each wheel of a vehicle are input, very complicated suspensions can be added to VEHDYN 4.0 because the restrictions concerning linearity only surface when trying to set up the equations required for the ISM. These restrictions are not applicable with regard to the solution equations used in the transient phase of the problem.

There are, however, potential problems with substituting the input of wheel forces for performing an equilibrium calculation using the ISM. A user might input any set of wheel forces he chooses even though there is only one true set of wheel forces for the vehicle data set provided. If the user inputs an incorrect set of wheel forces, then begins to simulate the movement of the vehicle down a flat course, the wheel forces will "readjust" to reach the true static position. In fact, it is recommended that, in developing a vehicle initially, the user simulate his vehicle running down a flat course, letting the vehicle "settle out," then take the wheel forces VEHDYN 4.0 computes and put them back into the input vehicle data file as the true wheel forces at equilibrium.

Springs and dampers

Another area that is different in VEHDYN 4.0 relates to the modeling of spring and damper elements. The changes to the translational elements are minor and involve only the transition function employed for hysteresis. The change to the rotational elements, however, is more extensive and gives the user much more flexibility in modeling resistance to beam rotation. The next two sections detail these changes.

Translational elements

Section 2.5 and Figures 2.7 and 2.8 of the VEHDYN II user's guide (Creighton 1986) explain the spring and damper hysteretic models employed by VEHDYN II. The concept of hysteresis implies separate loading and unloading relationships, which necessitates some type of transition curve to move between loading and unloading paths. VEHDYN II uses one exponential function for positive deflections (springs) or velocities (dampers) and a different exponential function for negative deflections or velocities. To specify these functions, the user must input four coefficients for each spring or damper data set. Many users find it difficult to determine these coefficients. To simplify the input for each spring/damper relation, linear transition functions have been implemented into VEHDYN 4.0. The user merely inputs a single value representing a straight-line slope that the routines in VEHDYN 4.0 use to move between loading and unloading curves.

Figure 2 shows generalized spring and damper relationships as modeled in VEHDYN 4.0. The upper graph shows a hypothetical spring relationship while the lower graph shows a hypothetical damper relationship. Each graph has separate loading and unloading curves. The numbers 1-2-3-4-5 on each graph indicate a possible path that could be taken during the course of a VEHDYN 4.0 calculation and show how the model would handle movement between loading and unloading curves.

Point 1 is the starting point of this path and is assumed to be on the loading curve. The next point (2) has a spring displacement δ^S (or damper velocity δ^D) which is greater than the δ^S (or δ^D) for Point 1; thus, the path from Point 1 to Point 2 is a loading path. Because Point 1 is already on the loading curve, just march up the loading curve as indicated in the figure. The δ^S (or δ^D) for Point 3 is less than the δ^S (or δ^D) for Point 2; thus, unloading has begun. Because Point 2 is on the loading curve, unloading

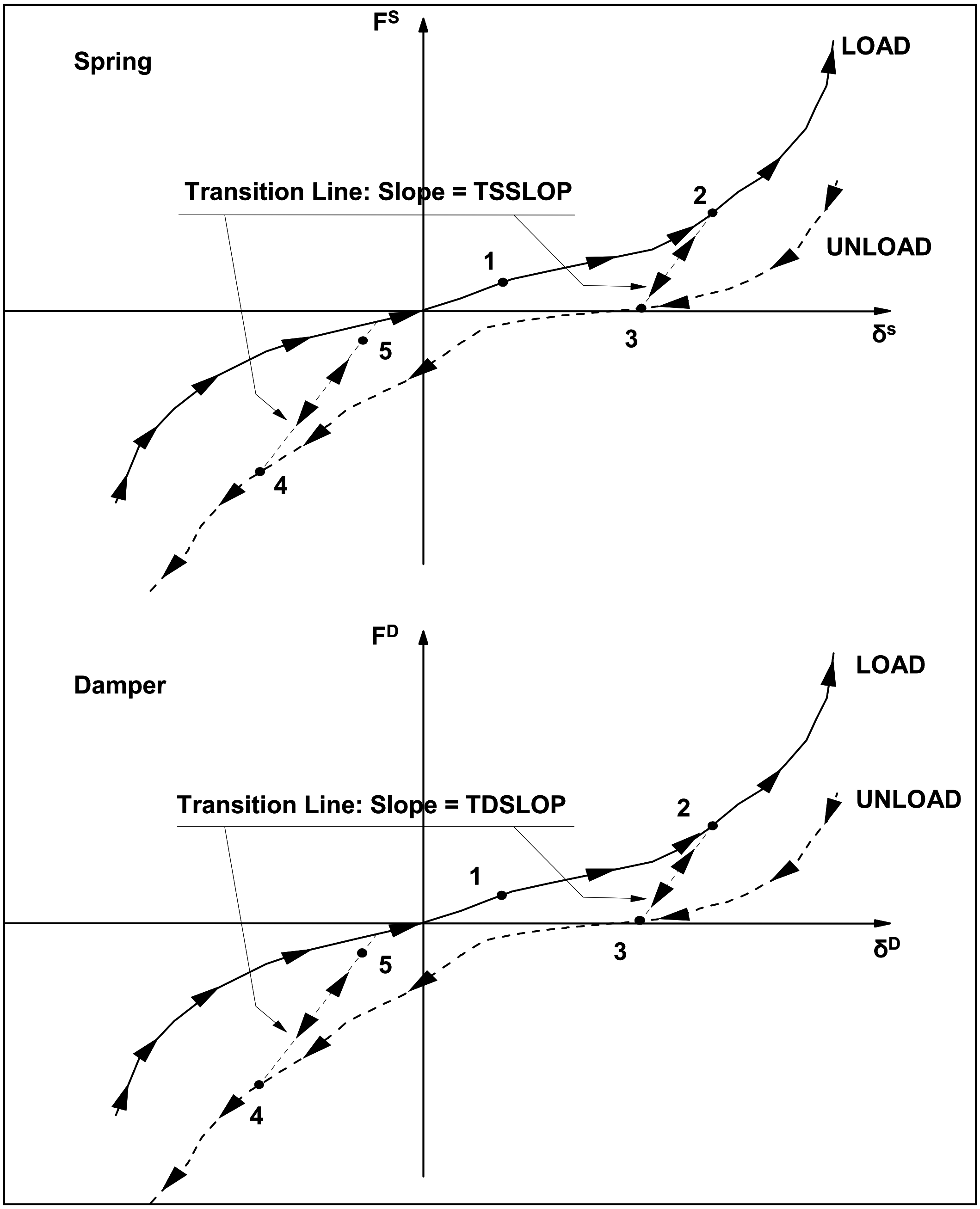


Figure 2. Nonlinear hysteretic spring and damper relations as modeled in VEHDYN 4.0.

must proceed down a transition line to Point 3. This transition line is a straight line in VEHDYN 4.0 of slope TSSLOP(i) (for the i^{th} spring curve) or TDSLOP(i) (for the i^{th} damper curve), which intersects Point 2, the point from which unloading begins.

Point 4 is a continuation of unloading from Point 3; just slide down the unloading curve from Point 3 to Point 4. Point 5 begins a reloading from

Point 4. This transition line has the same slope as earlier, but now intersects Point 4, which is the point where unloading switches to loading.

Besides being a simpler model than in VEHDYN II with the number of input parameters reduced from four to one, another advantage of this representation is the removal of the VEHDYN II restriction that both the loading and unloading curves needed to pass through the point (0,0). The only restrictions on the loading and unloading curves are that each curve must be a single-valued function and the unloading curve must lie entirely beneath the loading curve. Also, the slope for the transition line must be a positive value.

Rotational elements

The resistance to rotation for beams (walking beam and bogie suspensions) in VEHDYN II can be modeled using a frictional rotational damper and a rotational bump stop. The user-input frictional damper constant is a Coulomb-type resistance and is applied such that it always resists beam rotation. The rotational bump stop applies no resistance until the beam rotates a certain angle, clockwise or counterclockwise, from a static position.

The dash-double dot line in Figure 3 depicts the rotational bump stop model in VEHDYN II. The angle φ is the angle between the beam and the line joining the beam pivot point to the frame. The user inputs two quantities, the angle φ_{min} defining the minimum angle (counterclockwise beam rotation) prior to the onset of the stop's resistance and the slope of the linear stop portion of the resisting moment MR versus angle φ curve (i.e., $\varphi < \varphi_{min}$ or $\varphi > \varphi_{max}$). The stop is symmetric in the sense that φ_{min} and φ_{max} are equidistant from the static position (assumed to be $\varphi = \frac{\pi}{2}$); that is, the onset of the bump stop occurs at the same angle rotated from static, regardless of whether clockwise or counterclockwise rotation is performed. Equation 3 expresses the mathematical relationship between φ_{min} and φ_{max}.

$$\varphi_{max} = \pi - \varphi_{min} \qquad (3)$$

VEHDYN 4.0 still retains the rotational frictional damper from VEHDYN II, but a new, more general rotational spring model replaces the rotational bump stop described above. This new rotational spring model is

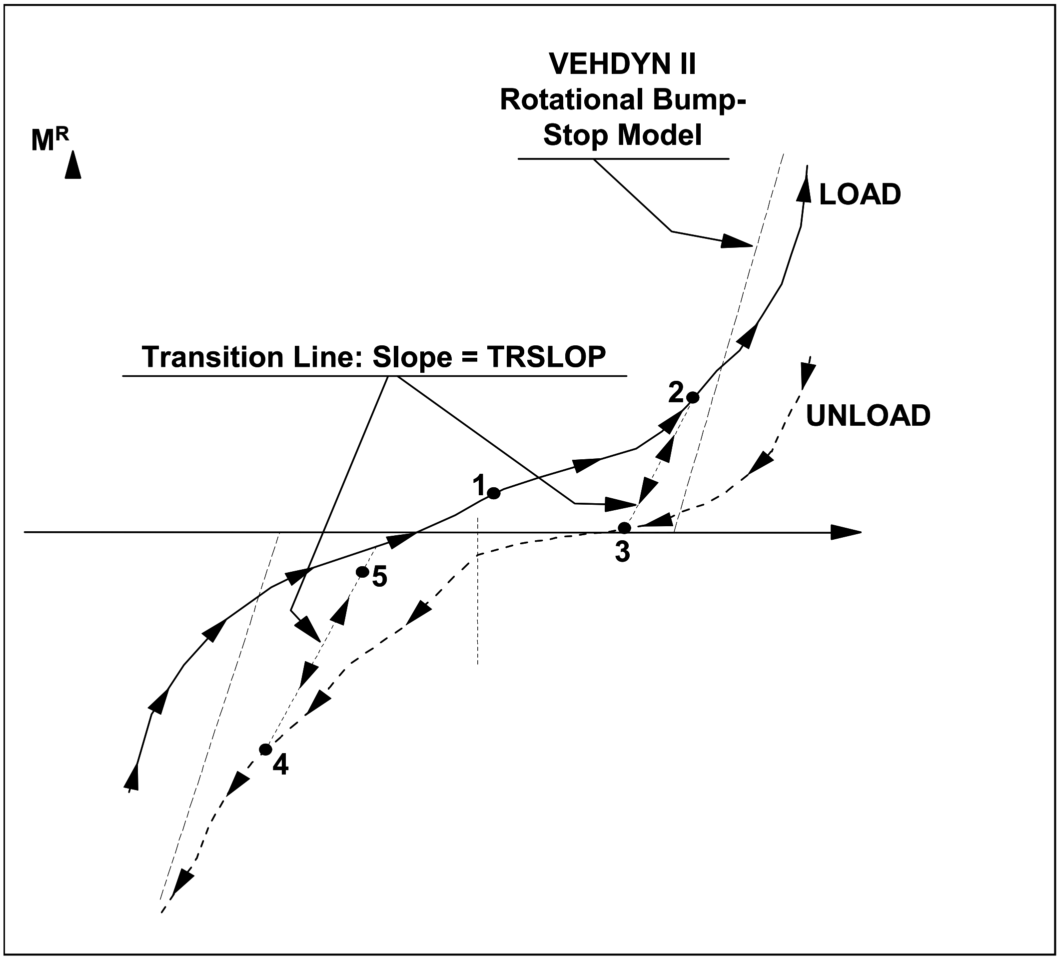


Figure 3. Nonlinear hysteretic rotational spring moment versus angular displacement model.

depicted in Figure 3 and is similar to the hysteretic spring and damper models described in the previous section. The user may input separate nonlinear loading (solid curve in Figure 3) and unloading (long-dash curve in Figure 3) relationships. If an unloading relationship is input, a transition line's slope TRSLOP is also required, which describes the path used to move between the loading and unloading curves the same way as illustrated for the translational elements.

By inputting the required loading-curve data to describe the dash-double dot line in Figure 3 representing the VEHDYN II rotational bump stop (i.e., four data points), this new model still allows the user to model this simple stop and also provides a means for modeling a much more complex hysteretic rotational spring.

Uniform track tension model

In VEHDYN II, the effects of a tracked vehicle's track are modeled via two separate mechanisms. First, as one tracked road wheel moves vertically relative to a neighboring tracked road wheel, the track tension exerts a vertical restoring force that tends to vertically realign the road wheels. This effect is modeled by a vertical interconnecting linear spring between each pair of tracked road wheels that exerts this restoring force on the road wheel axles (Creighton 1986, Figure 3.14). Second, the section of track between the forwardmost tracked road wheel and the forward SPRIDLER (i.e., the forward feeler) and the section of track between the aftmost tracked road wheel and the aft SPRIDLER (i.e., the rear feeler) are both modeled to account for interaction between that portion of the track and the terrain profile (Creighton 1986, Figures 3.12 and 3.13). Each feeler is assumed to have a linear spring (spring constant k_f) mounted perpendicular to the feeler and located at the maximum deflection point (δ_{max}) along the feeler. The normal force F_N (= $k_f \, \delta_{max}$) is resolved to a track-feeler tension and then applied directly to the corresponding road wheel and SPRIDLER.

For the predecessor to VEHDYN 4.0 (called VEHDYN 3.0, but never officially released), a new track tension model was developed and implemented. Rather than just modeling some of the effects of the track, a set of track points (which, when connected, form the entire track loop) was computed each time step, based on the instantaneous location of the wheels and SPRIDLERs relative to the nondeformable terrain profile. This "track" (called a band track) is assumed to have an instantaneous uniform tension T given by

$$T = max\left(0, T_0 + (L - L_0)\frac{dT}{dL}\right) \tag{4}$$

where:

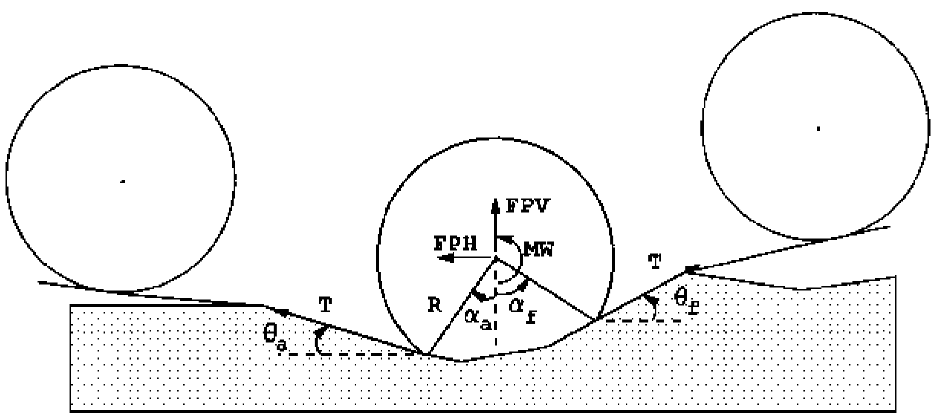
T_0 = track tension at static equilibrium (input)
L = instantaneous track length (computed each time step)
L_0 = track length at static equilibrium (computed from input vehicle configuration)
$\frac{dT}{dL}$ = change in tension per change in track length (input)

The "max" function in Equation 4 is necessary to maintain a nonnegative tension in the track. As the track length shortens, the tension will go to zero when the length reaches L_{min} given by

$$L_{min} = L_0 - \frac{T_0}{(dT/dL)} \tag{5}$$

For lengths $L \leq L_{min}$, the track tension is assumed to be zero.

Horizontal and vertical force components, as well as an applied moment, are computed each time step for each tracked road wheel and SPRIDLER, as shown in Figure 4, based on the instantaneous computed track points and track tension. These force and moment components are added to the components computed from the continuous spring model, which provides the interaction between wheels and terrain (Creighton 1986, Section 3.7.1).



$$FPV = T\ (\sin\theta_a + \sin\theta_f)$$

$$FPH = T\ (\cos\theta_a - \cos\theta_f)$$

$$MW = T\ R\ [\cos(\theta_f - \alpha_f) - \cos(\theta_a - \alpha_a)]$$

Figure 4. Track tension model in VEHDYN 4.0.

This track tension model is superior to the VEHDYN II model for two reasons. First, the computation of the track points comprising the instantaneous track takes into account the interference of the terrain profile between neighboring wheels (i.e., bridging effect). This effect was completely ignored in the VEHDYN II model except at the front and rear feelers. The VEHDYN 4.0 track model considers this bridging all along the track-profile interface.

Second, as shown in Figure 4, the force of track tension is applied (resolved) to wheel force and moment components exactly as a track actually pulls on each wheel, taking into account both horizontal and vertical effects. If θ_f (in Figure 4) is different from θ_a, which can occur even for interior tracked road wheels considering profile interference, there will be a net horizontal force due to the presence of the track not considered in the VEHDYN II track tension model.

In the implementation of this model in VEHDYN 4.0, a set of points is computed, which, when connected, forms an approximation of the instantaneous track loop. The procedure used in VEHDYN 4.0 (explained in detail in Appendix D) to accomplish this requires that each wheel inside the track loop, including any SPRIDLERs, be transformed into a set of discrete points that approximate the circular shape of the section of the wheel where the track is expected to touch under a wide range of conditions. For each wheel/SPRIDLER, the user must input three quantities that determine how to perform this transformation.

Figure 5 is an example of a wheel inside the track loop and how it might be transformed. ANGCTR defines a polar angle at the center of the portion of the wheel to be transformed. NWLPTS defines the total number of discrete points along the wheel's circumference. DELANG defines the size of the arc between discrete points on the circumference. The reason for generating points to represent only part of the wheel is that the CPU time to execute the new algorithm that computes the track loop is highly dependent on the total number of discrete wheel points, and it is only necessary to approximate the portion of each wheel where the track can reasonably be expected to touch. It has been found that a reasonable value for DELANG is in the range of 8 to 15 deg.

The obvious shortcoming of this new track model is the assumption of uniform tension throughout the entire length of track. This assumption is more in error with steel tracks than the new rubber tracks. Appendix D discusses some recent upgrades to the track model to incorporate a local tensioning scheme into the track model.

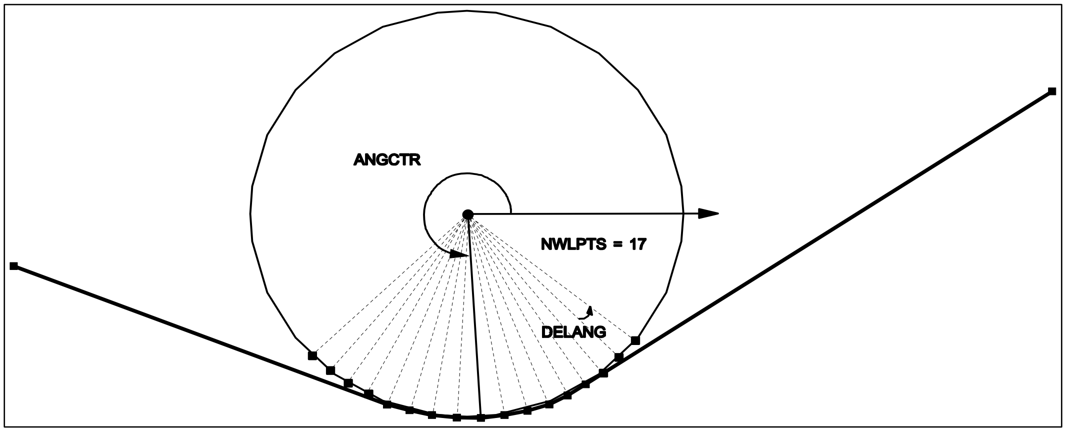


Figure 5. Generation of points representing tracked road wheels in VEHDYN 4.0.

Converting VEHDYN II track data to VEHDYN 4.0 track data

If the user has a VEHDYN II vehicle data set, one question that arises is how the VEHDYN II track tension input data can be used to obtain the VEHDYN 4.0 track tension input data. This section provides an approximate solution to give a reasonable starting set of data.

Consider the three wheels in Figure 6 that are all inside the track envelope.

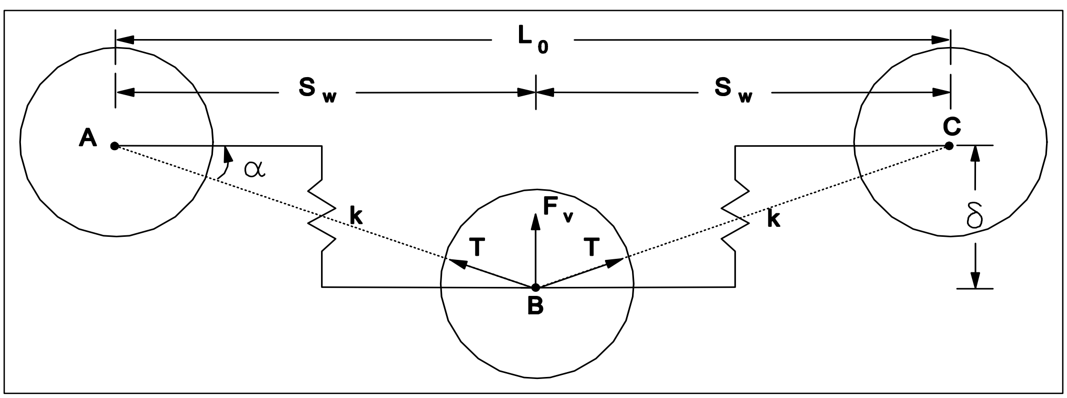


Figure 6. Track model representations in VEHDYN II and VEHDYN 4.0.

The distance labeled L_O represents the equilibrium track length that spans these three wheels and is twice the wheel spacing S_w. If the middle wheel is displaced vertically downward a distance δ, the track exerts an upward force attempting to restore the vertical alignment of the three wheels. In VEHDYN II, the model is represented by the two pictured interconnecting springs, each having a spring constant k, which together exert a vertical force F_v at point B given by

$$F_v = 2\,k\,\delta \tag{6}$$

Assuming that the distance ABC is a reasonable approximation for the stretched length of track L associated with these three wheels, the VEHDYN 4.0 tension T can be expressed as

$$T = T_0 + (L - L_0)\frac{dT}{dL} \tag{7}$$

As given by Equation 4. The length L is the given by

$$L = \frac{L_0}{\cos \alpha} \tag{8}$$

where the angle α can be expressed as

$$\alpha = \tan^{-1}\left(\frac{\delta}{S_w}\right) \tag{9}$$

The force F_v can be expressed in VEHDYN 4.0 terminology as

$$F_v = 2\,T \sin \alpha \tag{10}$$

To determine the initial tension T_o and the stretch parameter $\frac{dT}{dL}$ required for the VEHDYN 4.0 data set, all that is necessary is to eliminate F_v from Equations 6 and 10 and solve for two values of δ. Letting those two values be 1.0 and 2.0 in., the results are given as

$$T_0 = \frac{T_1 L_2 - T_2 L_1 + (T_2 - T_1)L_0}{L_2 - L_1} \tag{11}$$

and

$$\frac{dT}{dL} = \frac{T_2 - T_1}{L_2 - L_1} \tag{12}$$

The instantaneous track length and track tension for a displacement equal to 1 in., L_1 and T_1, respectively, are

$$L_1 = \frac{L_0}{\cos\left[\tan^{-1}\left(\frac{2}{L_0}\right)\right]} \tag{13}$$

$$T_1 = \frac{k}{\sin\left[\tan^{-1}\left(\frac{2}{L_0}\right)\right]} \tag{14}$$

while the instantaneous track length and track tension for a displacement equal to 2 in., L_2 and T_2, respectively, are

$$L_2 = \frac{L_0}{\cos\left[\tan^{-1}\left(\frac{4}{L_0}\right)\right]} \tag{15}$$

$$T_2 = \frac{k}{\sin\left[\tan^{-1}\left(\frac{4}{L_0}\right)\right]} \tag{16}$$

Example procedure for adding new suspensions --- 30/30 front end

Many of the upgrades to VEHDYN II to produce VEHDYN 4.0, particularly in the area of calculation initiation, were implemented to make the task easier for adding new, even complicated, suspensions to the VEHDYN model. This section is provided to show the steps required to develop a new suspension for inclusion in VEHDYN 4.0.

The vehicle to be considered is Caterpillar's 30/30 Engineer Support Tractor. Figure 7 shows a picture of the vehicle along with a view of the 30/30's suspension system. Most of the vehicle can be reasonably modeled using VEHDYN II with the exception of the front end, which includes the forwardmost three-road wheels and the spring-damping-beam elements attached to them. As a demonstration of the ability to upgrade VEHDYN 4.0 with more complicated suspension combinations, the front end of the 30/30, called the 30/30F suspension in this report and in VEHDYN 4.0, is discussed at this time.

The suspension attached to the middle pair of wheels (4 and 5 when counting from the front of the vehicle) can be modeled as a walking beam, and

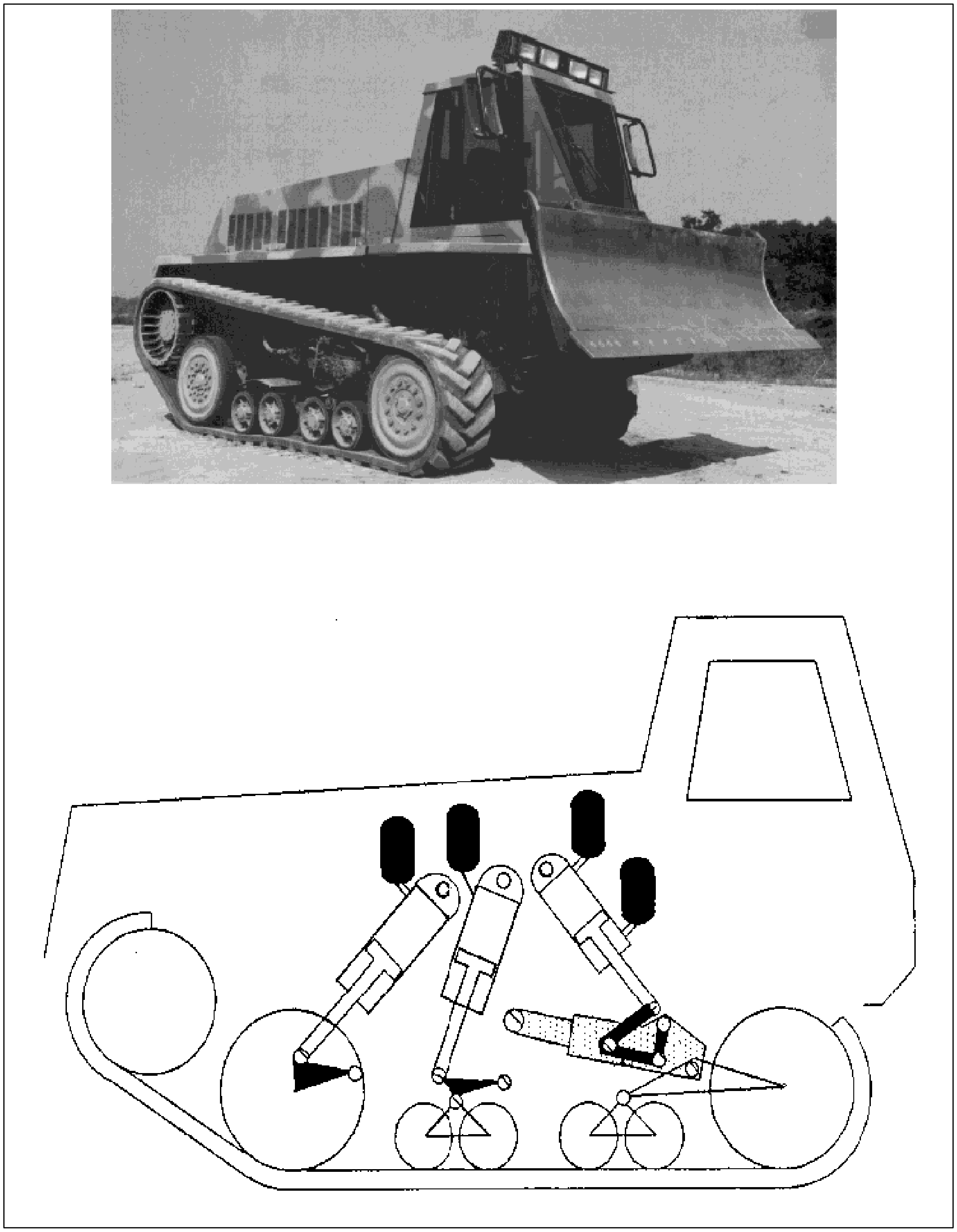


Figure 7. Caterpillar 30/30 engineer support tractor.

the suspension attached to wheel 6 can be modeled as independent. However, the forward section attached to wheels 1, 2, and 3 cannot be modeled with any of the existing suspension elements in VEHDYN II. Figure 8 shows a more detailed definition for the front end that was agreed upon by Caterpillar and the WES as being sufficient to accurately model the ride and shock characteristics for this vehicle.

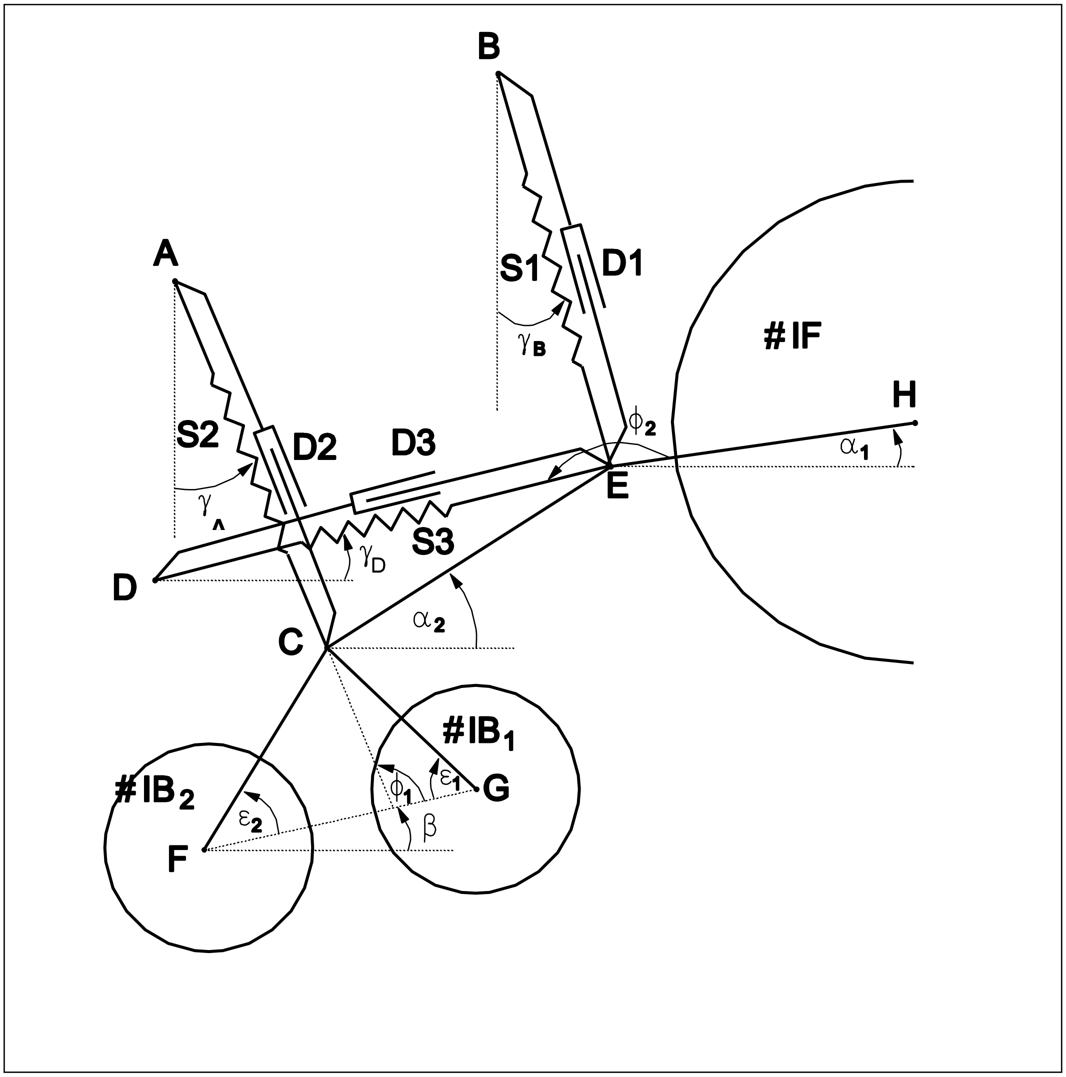


Figure 8. 30/30F suspension definition as modeled in VEHDYN 4.0.

Points A, B, and D in Figure 8 represent fixed points on the sprung mass, and each of the attached springs/dampers (A-C, B-E, and D-E) are free to pivot about each point, respectively. Lines CEH and FCG each represent bent beams. Beam CEH can pivot about Point E, with the beam's rotational movement being resisted by a frictional damper and a rotational spring. Even though the beam can rotate, the beam's angle CEH remains constant. Beam FCG is similar in that it can pivot about Point C, being restricted by both a frictional damper and a rotational spring. Also, the angle FCG is constant.

Given the spring/damper/beam elements in Figure 8, the next step is to determine the number of DOF for this suspension. If the vertical and horizontal movement of Point E (pivot point of major beam CEH) is specified, as well as the rotational movements of major beam CEH and minor beam FCG, the suspension's movement is completely determined; thus, there are four DOF. It is therefore necessary to develop four equations of motion, one for each DOF, and it is also necessary to modify the two equations describing the sprung mass movement (vertical translation and pitch). Using the terminology in Figure 8, the four equations of motion for the 30/30F suspension can be written as

$$\begin{aligned} F_E^v &= m_{3030F}\ddot{\xi}_E \\ &= F_{IF}^v + F_{IB_1}^v + F_{IB_2}^v - (m_{3030F})\,g - F_1 \cos\gamma_B \\ &\quad - F_2 \cos\gamma_A + F_3 \sin\gamma_D \end{aligned} \tag{17}$$

$$\begin{aligned} F_E^h &= m_{3030F}\ \ddot{\eta}_E \\ &= F_{IF}^h + F_{IB_1}^h + F_{IB_2}^h - F_1 \sin\gamma_B - F_2 \sin\gamma_A - F_3 \cos\gamma_D \end{aligned} \tag{18}$$

$$\begin{aligned} M_{CEH}^E &= I_{CEH}\ \ddot{\alpha}_1 \\ &= M_{IF}^w + BL_1^E\left[\left(F_{IF}^v - m_{IF}\,g\right)\cos\alpha_1 + F_{IF}^h \sin\alpha_1\right] \\ &\quad + BL_2^E\left[F_2 \cos\left(\alpha_2 - \gamma_A\right) - \cos\alpha_2\left(F_{IB_1}^v + F_{IB_2}^v\right.\right. \\ &\quad \left.\left. - m_{IB_1}\,g - m_{IB_2}\,g\right) - \sin\alpha_2\left(F_{IB_1}^h + F_{IB_2}^h\right)\right] \\ &\quad + \left[D_E \sin\left(\dot{\theta} - \dot{\alpha}_1\right) + M_{RS}^E\right] \\ &\quad - \left[D_C \sin\left(\dot{\alpha}_2 - \dot{\beta}\right) + M_{RS}^C\right] \end{aligned} \tag{19}$$

$$\begin{aligned} M_{FCG}^C &= I_{FCG}\ \ddot{\beta} \\ &= M_{IB_1}^w + M_{IB_2}^w + \left[D_C \sin\left(\dot{\alpha}_2 - \dot{\beta}\right) + M_{RS}^C\right] \\ &\quad + BL_1^C\left[\left(F_{IB_1}^v - m_{IB_1}\,g\right)\cos\left(\varepsilon_1 - \beta\right)\right. \\ &\quad \left. - F_{IB_1}^h \sin\left(\varepsilon_1 - \beta\right)\right] - BL_2^C\left[\left(F_{IB_2}^v - m_{IB_2}\,g\right)\right. \\ &\quad \left. \cos\left(\varepsilon_2 - \beta\right) + F_{IB_2}^h \sin\left(\varepsilon_2 - \beta\right)\right] \end{aligned} \tag{20}$$

where ξ_E and η_E are the vertical (positive upward) and horizontal (positive aftward) displacements of the major beam CEH's pivot point E. The forces F_1, F_2, and F_3 are combined spring and damping forces for spring-damper combination 1 (spring S1 and damper D1), 2 (S2 and D2), and 3 (S3 and D3), respectively. The M_{RS}^C and M_{RS}^E are the resisting moments due to the rotational springs mounted at Points E and C, respectively. The frictional damping coefficients D_E and D_C represent the frictional resistance to rotation of major beam *CEH* and minor beam *FCG*, respectively.

The subscripts *IF*, IB_1, and IB_2 refer to the wheels centered at Points H, G, and F, respectively. The large wheel (*IF*) is the forwardmost wheel, while the smaller wheels (IB_1 and IB_2) are the wheels associated with the minor beam FCG.

The quantities m_{IF}, m_{IB_1}, and m_{IB_2} are wheel assembly masses associated with each road wheel; each beam is explicitly assumed massless, but may reflect distribution of mass in its associated moment of inertia. The mass m_{3030F} is the mass of the entire 30/30F suspension and equals the sum of the three wheel assembly masses.

The forces F_F, F_{B_1}, and F_{B_2} are the vertical forces acting on each wheel center due to wheel-terrain interaction (continuous spring model, Creighton 1986, Section 3.7.1.1), while the forces F_F, F_{B_1}, and F_{B_2} are the corresponding horizontal forces, and the moments M_F, M_{B_1}, and M_{B_2} are the corresponding wheel moments.

Finally, the quantities BL_{EH}, BL_{CE}, BL_{CG}, and BL_{FC} are unsigned beam lengths representing line segments EH, CE, CG, and FC, respectively. Forces acting on the 30/30F suspension also affect the motion of the sprung mass, and the following adjustments to the two sprung mass equations of motion (vertical center of gravity, translation, and pitch) are necessary:

$$\left(F_z\right)_{3030F} = \left(m_s\,\ddot{\xi}_{CG}\right)_{3030F} = F_1 \cos\gamma_B + F_2 \cos\gamma_A - F_3 \sin\gamma_D \tag{21}$$

$$\begin{aligned} \left(M_\theta\right)_{3030F} &= \left(I_s\,\ddot{\theta}\right)_{3030F} \\ &= F_1\left[L_B^P \cos\left(\theta-\gamma_B\right) + H_B^P \sin\left(\theta-\gamma_B\right)\right] + \\ &\quad F_2\left[L_A^P \cos\left(\theta-\gamma_A\right) + H_A^P \sin\left(\theta-\gamma_A\right)\right] + \\ &\quad F_3\left[L_D^P \sin\left(\theta-\gamma_D\right) - H_D^P \cos\left(\theta-\gamma_D\right)\right] - \\ &\quad D_E \,\mathrm{sign}\left(\dot{\theta}-\dot{\alpha}_1\right) + M_{RS}^E \end{aligned} \tag{22}$$

where:

L_A^P, L_B^P, L_D^P = signed longitudinal distances from the center of gravity (CG) to pivot points A, B, and D, respectively

H_A^P, H_B^P, H_D^P = signed transverse distances from the CG to pivot points A, B, and D, respectively.

Given these equations (Eqs. 17–22), Table 1 lists the routines in VEHDYN 4.0 that required modification and what, specifically, was accomplished in each routine.

Table 1. VEHDYN 4.0 routines to modify for new suspension.

Routine	What To Do
MAIN	(1) Add calls (initial and transient) to new F3030ACC routine
CRUPDT	(1) Compute current coordinates of key locations (e.g., suspension-to-frame connection points, beam pivot points, and wheel centers) (2) Compute current beam angles and their trigonometric functions (3) Compute current spring/damper displacements and orientations, damper velocities, and rotational spring angles
F3030ACC	(1) Compute spring/damper forces (2) Compute accelerations for each DOF for the 30/30F (3) Compute force/moment contributions of the 30/30F to the vertical motion of the sprung mass CG and the pitch motion of the sprung mass about its CG
PRNTOT	(1) Add print statements pertinent to equilibrium position of the 30/30F for output to printed output file LUPRNT
PRNTRN	(1) Add print statements for transient data for the 30/30F for output to file LUPRNT
VEHINP	(1) Adjust computation of number of wheels to account for the 30/30F (2) Add call to new routine that reads input equilibrium data for the 30/30F
IN3030	(1) Reads in section of vehicle input data from file LUVEH for the 30/30F (2) Calculates initial (equilibrium) values of several quantities such as (a) pointer to location of the 30/30F's equations of motion (b) beam lengths and angles (c) horizontal coordinates of key locations (d) spring stretch lengths and orientation angles (e) rotational spring angles and rotational moment due to that spring (f) forces acting on each linear spring and resulting displacement
PLOTOT	(1) Equilibrium and transient write statements to plot output file LUPLOT for the 30/30F describing location and movement data required for postprocessing
TNINIT	(1) Add initialization of global displacement. and velocity arrays for each DOF for the 30/30F
STRIKE	(1) Add strike-through checks for each linear and rotational spring in the 30/30F

3 Simulation Capability Overview

Simulation types

VEHDYN 4.0 is different from earlier versions in that the user has several options available to him concerning the type of simulation he can make. The types of model simulations are listed in the last 12 rows of the first column in Table 2.

Data input

As many as seven input files are required to run VEHDYN 4.0. Table 2 indicates the 12 types of simulations that can be made and which input files are required for each type. Some simulations, such as the maximum slope climbing determination on a nondeformable surface terrain, require only two input files (vehicle and control) to execute. Others, such as the single ride (or shock) run down a deformable surface terrain, require as many as six input files (profile, vehicle, control, two UID (normal and belly), and VTI data) to run. After the user decides what type of simulation to make, Table 2 is used to determine which input files must be assembled and which output files are expected as the model is executed. Following is a generic description of what kind of data is found in each input file.

Vehicle data file

The first input file to examine is the vehicle data file (logical unit 43). Every model run requires this file containing the geometric description of the vehicle to be run as well as weight, moment of inertia, and suspension response tables. The vehicle is assumed to be sitting in an equilibrium static position under gravitational weight of the vehicle; this is the same configuration as was required for the input vehicle data file for the VEHDYN II preprocessor called PREVDYN2.

Control data file

The second input file is the control data file (logical unit 45); like the vehicle data file, this file is required by every type of run.

Table 2. Required files for each VEHDYN 4.0 simulation type.

Logical Units-->	42	43	45	46	47	48	49	50	51	52	53	54	55	56	57
VD4 Variable-->	LUPRF	LUVEH	LUINPT	LUPRNT	LUPLOT	LUPOW	LUUWIN	LUUWIN2	LUDRGOT	LUDIAG	LUFDAG	LUPRFO	LUPRFO2	LUPRFIN	LUVMI
Description-->	Profile Data	Vehicle Data	Control Data	Formatted Output	Plot Output	Absorbed Power and Driver Seat Acc Table	Standard UID Data	Full UID Data	Special Diagnostic Output	Assorted Diagnostics	Filtered Diagnostics	Deformed Profile No. 1	Deformed Profile No. 2	Profile Names with Starting Speeds	Soil Data for Deformable Profile
Input/Output-->	I	I	I	O	O	O	I	I	O	O	O	O	O	I	I
Required File?--> ------------------------ Run Type V V V															
single constant-speed ride/shock nondeformable	X	X	X ACCFLG <1	X	X	X	IUIDWL >0		X	NFILTS >0	NFILTS >0				
single constant-speed ride/shock deformable	X	X	X ACCFLG <1	X	X	X	IUIDWL >0	IUIDWL =2	X	NFILTS >0	NFILTS >0	X	X		X
single acc/dec ride/shock nondeformable	X	X	X ACCFLG =1	X	X	X	IUIDWL >0		X	NFILTS >0	NFILTS >0				
single acc/dec ride/shock deformable	X	X	X ACCFLG =1	X	X	X	IUIDWL >0	IUIDWL =2	X	NFILTS >0	NFILTS >0	X	X		X
VCI1 Determination	X	X	X ACCFLG =1				IUIDWL >0	IUIDWL =2	X						X
Maximum Slope Climbing Determination on Nondeformable		X	X ACCFLG =1						X						
Maximum Slope Climbing Determination on Deformable		X	X ACCFLG =1						X						X
Gap Crossing	X (special)	X	X ACCFLG =1		X		IUIDWL >0	IUIDWL =2	X						X
Obsmod	X (special)	X	X		X		IUIDWL >0	IUIDWL =2	X						
Ride Performance	X	X	X		X		IUIDWL >0		X					X	
Shock Performance		X	X		X		IUIDWL >0		X						
Drawbar Pull Test		X	X	X	X	X	IUIDWL >0	IUIDWL =2	X			X	X		X

The control file contains assorted data needed to describe the details of the model run. Information such as vehicle and profile descriptors, vehicle speed, time step size, run length in time or distance, output file increments, various flags to allow turning on/off UID effects, wheel damping, and using tractive force-speed data to allow ACC/DEC are some of the data found in the control file.

Profile data file

The third input file is the profile data file (logical unit 42) and is required for 6 of the 12 types of simulations. The gap crossing and ObsMod modes require special profile data. The simulation types not requiring a profile data file include maximum slope climbing, drawbar pull modeling, and shock performance. These simulations use program-calculated profiles, thus eliminating the need for the user to input the terrain description. When required, the profile data file primarily contains a set of world-frame coordinate pairs (horizontal, vertical) describing the shape of the terrain from left to right that the vehicle will be traversing.

Vehicle-terrain interaction data file

The fourth input file of interest is the vehicle-terrain interaction data file (logical unit 57). Every run in which the terrain is modeled as deformable requires this file. These runs include the single-run ride/shock runs, both constant-speed and ACC/DEC over deformable terrain, VCI1 determination, maximum slope-climbing determination on a deformable surface, drawbar pull simulation, and gap crossing. When required, this file contains soil strength in Rating Cone Index (RCI) and soil type data along the course. The RCI is described in Appendix C. The file also contains start locations for any bodies of water that may be encountered (e.g., puddles, streams, ponds, lakes, etc.). VEHDYN 4.0 contains buoyancy and drag effects when a vehicle interacts with such bodies of water as described in Appendix F.

Underside impact detection data files

The next input file(s) are the UID data files. These contain the data used to model the portion of the underside of the vehicle that may interact with the terrain by making contact as the undulations of the terrain are traversed. The first UID input file contains data describing the section of the underside that longitudinally is along the wheels; these data model the underside in front of the first wheel, aft of the last wheel, and between the interior wheels. The second UID input file contains data describing the belly section of the underside between the left and right wheels. For problems in which the user wishes to model underside impact, the first UID file is required and the second UID file is optional. The UID model is described in Appendix B.

Ride performance profile list input data file

The last input file the user may need to build is the profile list input data file. The file is required only when making a ride performance series of runs utilizing several terrain profiles of differing surface roughness. This file contains the eight-character-or-less identifier for each desired profile and a starting speed for each profile.

Data output

Several types of output files may be generated based on the type of run. For the single ride/shock type runs, the main history data are found in the formatted output file (logical unit 46), the plot output file (logical unit 47), and the absorbed power table (logical unit 48).

The formatted output file is designed for viewing only. It is an ASCII formatted file that can be listed and read. The beginning of the file has most of the vehicle input data from the vehicle data file, organized so that the user can verify the correctness of the vehicle components. Included are the identifiers for both the vehicle and the profile; integration time step; length of run (distance and time); output increments for all three output files; spring, damper, and rotational spring tables; vehicle weights (total and sprung) and moments of inertia (sprung mass and suspension beams); wheel data; track data; and initial location of key locations for each suspension.

At each output increment, instantaneous data of the vehicle's current configuration are then printed out, including coordinates of key locations (e.g., wheel centers, beam pivot points, CG of the sprung mass, suspension-to-frame connect points) and displacements, velocities, and accelerations of the sprung mass and all of the translational spring and damping elements. Rotational displacements, velocities, and accelerations are also output for the sprung mass (pitching motion about the CG) and all beam elements. Also output are the force components and moments acting on each wheel and tracked-vehicle's SPRIDLER. For tracked vehicles, the current track length and track tension are also printed out.

The format of VEHDYN 4.0's print file is almost identical to the format of VEHDYN II's formatted time-history output file (Section 3.10.1 of Creighton 1986). The primary differences occur in the portions of printout dealing with the improvements to VEHDYN 4.0, such as the new track tension algorithm, the new 3030F suspension, the new linear transition line for spring and damper tables, and elimination of the iterative stiffness method for vehicle settlement. The new print file now shows the data relevant to these upgrades.

One change that is not explained by the new improvements is in the area of wheel force components. VEHDYN II prints out the vertical force component acting on each wheel from all external sources (e.g., wheel-terrain, track-terrain, and wheel-track interactions). VEHDYN 4.0 prints out the net vertical force component acting on each wheel or SPRIDLER center. VEHDYN 4.0 subtracts the wheel assembly weight from the external forces being applied to the wheel or SPRIDLER. Thus, adding up the vertical force components acting on all wheels and SPRIDLERs results in one-half the sprung weight, whereas VEHDYN II produced one-half the entire vehicle weight. (All VEHDYN programs solve a half-vehicle problem—one side of the vehicle only.)

The plot output file is designed to be used by a postprocessing program to graphically display results such as an animation of the run. This file was called the binary output file in VEHDYN II. The data in the plot postprocessing file are similar to the data in the print file. Unlike ASCII files, binary files are generally not transportable across computer platforms; thus, this file is also in ASCII rather than binary, as was the case in VEHDYN II. An ASCII file allows a user to run VEHDYN 4.0 on one computer platform (e.g., a CRAY or SUN), then postprocess the results on another platform (e.g., a PC or Silicon Graphics). The plot postprocessing file contains all of the data that the print file contains except the spring, damper, and rotational spring tables. The plot postprocessing data are output every DTPLOT seconds, whereas the data in the print file are output every DTPRNT seconds. The two increments need not be the same, but they both should be multiples of the integration time step DT.

The absorbed power table contains a history of the instantaneous and average absorbed power computed at the driver's seat as well as instantaneous and maximum unfiltered and filtered driver's seat accelerations. This file is identical to the absorbed power table in VEHDYN II (Creighton 1986, Section 3.10.3). Acceleration and absorbed power data are printed to this table for a single user-defined location called the driver seat, which may be suspended on a spring-damper combination or unsuspended (i.e., a point located on the sprung mass). Data are written to the absorbed power table every DTPWR seconds; DTPWR should also be a multiple of the integration time step, DT.

Other output files possible with VEHDYN 4.0 include up to three diagnostic files (logical units 51, 52, and 53) and two deformed profile data files (logical units 54 and 55.) Logical unit 51 contains the maximum traction coefficient (when number of traction filters is greater than zero), ride performance speeds (for ride performance run), shock performance speeds (for shock performance run), and ObsMod results. Logical unit 52 contains the instantaneous traction coefficient history when number of traction filters is greater than zero. Logical unit 53 contains the filtered maximum traction coefficient for each traction filter when number of traction filters is greater than zero. Logical unit 54 describes the ground under the wheels or tracks, while logical unit 55 describes the ground under the vehicle's belly.

Postprocessing support programs

Currently, two postprocessing programs support the animation of output from VEHDYN 4.0. These programs are called PREANIM4 and VD4ANIMATOR. PREANIM4 reads data from VEHDYN 4.0's plot postprocessing data file (logical unit 47), the deformed profile data file(s) (logical units 54 and 55), and a frame file that contains an x-y array pair of the vehicle shape. PREANIM4 writes a single output file, called the preanimator output file, which contains data to subsequently be read by the VD4ANIMATOR. To execute theVD4ANIMATOR, the user must also construct a text-based configuration file containing data the animator requires, including some animation scaling information and some three-dimensional visualization parameters not required by the vehicle dynamics program. The use of these two programs will be demonstrated in Chapter 5 as the results for several sample problems are shown.

4 Input Guides

Unlike earlier VEHDYN programs that required three types of data in three separate input files for problem setup, VEHDYN 4.0 requires a variable number of input files dependent on the current problem type. A total of eight input files are used by VEHDYN 4.0, although no particular run requires all of them. As discussed in the previous chapter, there are 12 possible run types, each requiring a unique set of input files. This chapter discusses the format of each of the eight possible input files, including an input record-by-record guide. Once the run type is determined, Table 2 in Chapter 3 can be used to decide which of the seven possible input files will be required.

Control parameters

The control parameters data file contains the information that describes how the run will proceed. Each of the 12 simulation types requires this file. It includes descriptors to select a particular vehicle and a particular terrain profile, vehicle velocity, integration time step, run length, output time increments for each of three primary output files, and frequency for the low-pass filter for the absorbed power calculation. Additionally, the control file contains flags for turning on/off UID wheel and drag support, maximum traction coefficients for UID and drive wheels, coefficient of restitution for UID-wheel normal forces, UID-wheel damping coefficient, number of cutoff frequency filters and their corresponding frequency values for the traction coefficient array, an acceleration flag to select either constant-speed or variable-speed run, and the objective speed for variable-speed runs.

Following is a record-by-record guide for the control data file. The format for each of the first two records is shown, in parentheses, to the right of the record. The remaining records may be input in free-field format. A definition of each of the record's variables follows each record. Each new record is identified in this guide by a line that begins with the “>” character. If a record is too long to fit on one line, a continuation line is identified by a subsequent line that begins with the “&” character. Neither the “>” nor the “&” is part of the actual data that the user will enter.

> KVNAME (A8)

KVNAME: Character variable (maximum of eight characters) that is the vehicle search name for a given vehicle data set.

> KPNAME (A8)

KPNAME: Character variable (maximum of eight characters) that is the profile search name for a given profile data set.

> VELMPH DT YSTRT TMAX DMAX DTPRNT DTPLOT

VELMPH: Constant horizontal velocity at which the vehicle will run throughout the entire calculation, mph.

DT: Integration time step, sec.

YSTRT: Parameter used to position the vehicle anywhere horizontally along the profile. YSTRT is added to each vehicle horizontal coordinate to "move" the vehicle backward or forward along the profile. A positive value shifts the vehicle forward, and a negative value shifts the vehicle backward. A value of zero (0) positions the vehicle such that the forwardmost part of either road wheel no. 1 or SPRIDLER no. 1, whichever is forwardmost, is positioned at horizontal location equal to zero (i.e., the beginning of the profile), in.

TMAX: Maximum time for the vehicle to run down the profile; should be set equal to zero (0) if DMAX is to be used to determine the length of the run, sec.

DMAX: Maximum distance for the vehicle to run down the profile; is used only if TMAX has been set equal to zero (0), ft.

DTPRNT: Output increment for the formatted output file; should be a multiple of DT, sec.

DTPLOT: Output increment for the postprocessing output file formerly called the binary output file; should be a multiple of DT, sec.

> DTPWR FCUTOF STAPWR

DTPWR: Output increment for the tabulated driver/absorbed power information output file, should be a multiple of DT, sec.

FCUTOF: Cutoff frequency for low-pass filter used to filter driver's seat accelerations used to compute absorbed power, Hz.

STAPWR: Horizontal location at which, when the forwardmost road wheel (no. 1) has reached, absorbed power computations begin if the appropriate option for IFSEAT has been input, ft.

> **IUIDWL IUDRAG**

IUIDWL: Flag for UID computation; =0 (no detection), =1 (detection), =2 (additional file for belly impact).

IUDRAG: Flag to turn on/off drag effects for UID wheels; =0 (no drag effects), =1 (full drag effects).

> **XMUDMAX XMUUMAX COREST CUDAMP**

XMUDMAX: Maximum traction coefficient for drive wheels.

XMUUMAX: Maximum traction coefficient for UID wheels.

COREST: Coefficient of restitution for UID wheels (i.e., fraction of wheel-terrain interaction normal force to be applied back onto the vehicle as a reaction).

CUDAMP: Wheel damping coefficient for UID wheels.

> **NFILTS ACCFLG OBJMPH**

If NFILTS = 0, skip the next record.

ACCFLG: Flag for acceleration computation; =-1 (Constant speed and the data required for acceleration are not included in the vehicle file), =0 (Constant speed even though the acceleration data are present in the vehicle file), =1 (Acceleration is modeled).

OBJMPH: Objective speed for vehicle, mph.

> **FREOFF(1) FREOFF(2) ... FREOFF(NFILTS)**

FREOFF(i): i^{th} cutoff frequency for filtering traction coefficient array, Hz.

Profile data (standard)

The format of the profile data file is the same as for previous VEHDYN programs. The data in this file define a two-dimensional rigid nondeformable surface over which the vehicle travels during the course of the VEHDYN 4.0 run. An example profile is depicted in Figure 9.

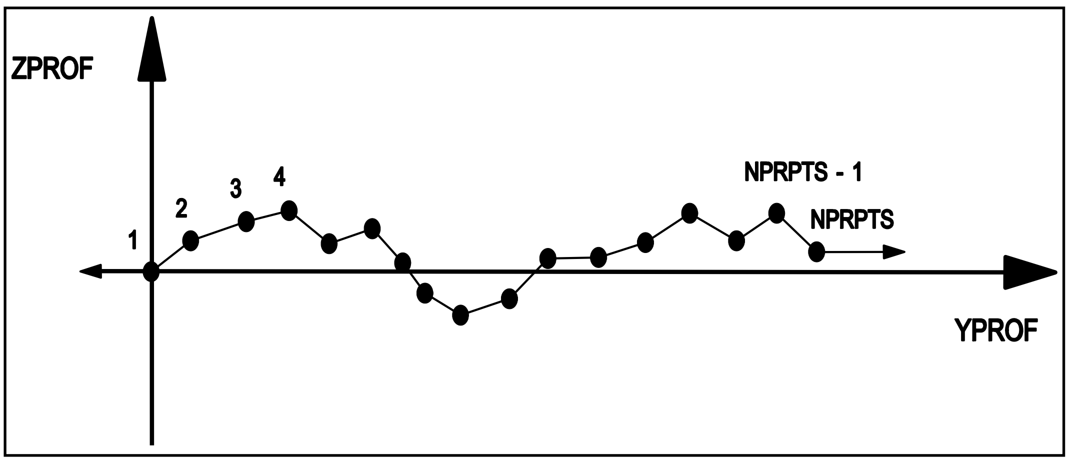


Figure 9. Terrain profile representation in VEHDYN 4.0.

The profile is determined by two arrays, YPROF and ZPROF, which define horizontal (Y) and vertical (Z) coordinates of successive points, respectively. The solid circles in Figure 9 are the points defined by (YPROF(i), ZPROF(i)), with i referring to the i^{th} point along the profile. The number of inputted points in each of the two arrays, YPROF and ZPROF, is NPRPTS.

The data can be input in two forms, equally or unequally spaced points. For equal spacing between adjacent points, the horizontal spacing value SPAC is input first as a positive quantity in inches. Then only the vertical coordinates of successive points are input (the unadjusted ZPROF array). The first input point is assigned a YPROF value equal to 0. The i^{th} point is assigned a YPROF value equal to (i-1) * SPAC . VEHDYN 4.0 adds a point at each end to define the profile for the entire Y range [$-\infty$, $+\infty$]. All of the data are adjusted in the vertical direction such that the initial input point has a vertical coordinate equal to 0. This determines a vertical offset by which all the other points are adjusted. A point is added at Y = $+\infty$ with the same adjusted ZPROF value as the last input point. A point is also added at Y = -∞ with a ZPROF value of 0. The resulting profile for this case is one in which the entire negative Y-axis is flat with a height of zero. The course actually begins at a YPROF-value of zero with the user's first input point, but now adjusted with a ZPROF value of zero.

For the "unequal spacing" format, the user inputs both Y and Z coordinates of each point defining the profile. The spacing variable SPAC must be set to 0. As with the "equal-spacing" case, the ZPROF value of the first input point is adjusted to 0; then the computed vertical offset is used to adjust all of the other points accordingly. The points at Y = +∞ and Y = -∞ are then added as described above. The resulting profile in this case is one in which the section of the Y-axis negative of the first input point's YPROF value is flat with a height of zero. The course actually begins at a YPROF value equal to the first input point's value of YPROF. Normally, this value would be zero, although it does not have to be. The negative Y-axis section of the profile is usually made flat, to provide a good starting location at which to place the vehicle. VEHDYN 4.0 then proceeds with the vehicle moving left to right, the direction of increasing YPROF values.

For the profile data file, it is possible to stack multiple data sets into a single data file. As can be seen in the input guide below, each data set would begin with an eight-character (or less) identifier. Like earlier versions, VEHDYN 4.0 has the ability to search through the profile data file to find the data set whose identifier matches the value of KPNAME from the control data file. If none of the data sets in the specified profile data file matches the input value of KPNAME, the VEHDYN 4.0 run aborts with an appropriate error message.

An input guide for the profile data file follows. The format for this guide is the same as for the control data file's input guide. A ">" character identifies each new record. If a record is too long to fit on one line, an "&" character identifies a continuation line.

> INNAME (A8)

INNAME: Character variable (maximum of eight characters) that identifies the forthcoming profile data set for search purposes. INNAME is compared to KPNAME from the control data file. If they are identical, the associated profile data are used to define the terrain profile for the current VEHDYN 4.0 run. VEHDYN 4.0 allows stacking of multiple profile data sets in the profile data file; comparison of INNAME to KPNAME is used to select the desired profile data.

> PRFID (A72)

PRFID: Character variable (maximum of 72 characters) used to identify the forthcoming profile data set in more detail than INNAME. PRFID is used only for output to each of the three output files.

> NPRPTS SPAC

NPRPTS: Number of profile data points stored in this data set.

SPAC: Horizontal spacing between consecutive points in this data set; set to 0 for unequal spacing option, in.

If SPAC = 0; skip the next record.

> ZPROF(1) ZPROF(2) . . . ZPROF(NPRPTS)

(Use as many lines of input as needed.)

ZPROF(i): Vertical coordinate of the i^{th} (evenly spaced) point, in.

If SPAC > 0; skip the next record.

> YPROF(1) ZPROF(1) YPROF(2) ZPROF(2) . . .
& . . . YPROF(NPRPTS) ZPROF(NPRPTS)

(Use as many lines of input as needed.)

YPROF(i), ZPROF(i): Horizontal and vertical coordinates, respectively, of the i^{th} profile point, in.

Profile data (special format for VGAP/ObsMod problems)

For VGAP (Gap Crossing) and ObsMod (Obstacle Crossing) problems, a special format profile data file is required that defines the sequence of gaps or obstacles to be generated by VEHDYN 4.0 and crossed during the execution of the program. Figure 10 defines the variables of interest for gaps and/or obstacles.

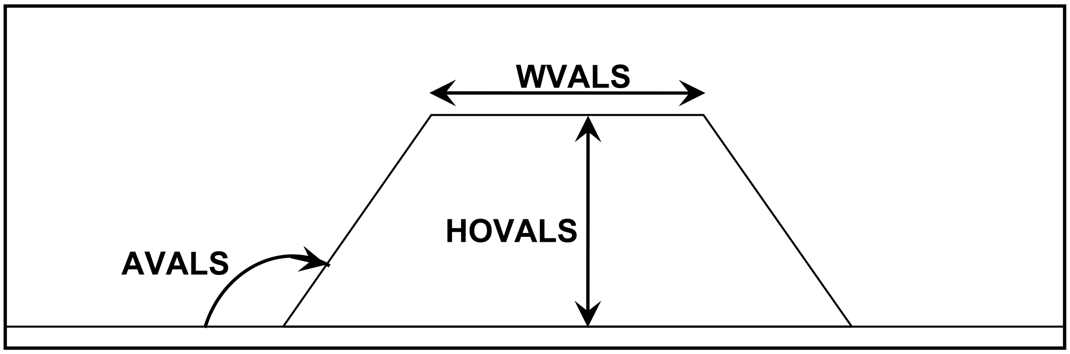


Figure 10. Symmetrical obstacle definition.

The format of the profile data for either the VGAP or ObsMod problem is as follows:

> **NOHGT**

NOHGT: Number of heights to be input.

> **HOVALS(1) HOVALS (2) … HOVALS (NOHGT)**

HOVALS(j): Obstacle height (or gap depth) of j^{th} obstacle (gap), in.

> **NWDTH**

NWDTH: Number of widths to be input.

> **WVALS(1) WVALS (2) … WVALS (NWDTH)**

WVALS(j): Obstacle (or gap) width of j^{th} obstacle (gap), in.

> **NANG**

NANG: Number of angles to be input.

> AVALS(1) AVALS (2) … AVALS (NANG)

AVALS(j): Approach angle for the j^{th} obstacle (or gap), radians.

For obstacles, angles should be between 0.5*pi and pi for bumps and between pi and 1.5*pi for ditches. For gaps, because generally only ditches are of interest, the angles should be between pi and 1.5*pi.

Vehicle data

The vehicle data file is the input file that has changed the most with this version of VEHDYN. In VEHDYN II, the vehicle's input configuration was a "zero-force," not an equilibrium, configuration. A preprocessor called PREVDYN2 was used to allow equilibrium input. The equilibrium, or settled, vehicle configuration is the format used for direct input into VEHDYN 4.0. In fact, the format is similar to the input file for PREVDYN2; the major differences involve the new capabilities of VEHDYN 4.0, such as rotational springs, the 30/30F suspension, and the transformation of tracked road wheels into discrete points.

The user can stack data sets in a vehicle data file just as with the profile data file. VEHDYN 4.0 searches through the vehicle data file to find the data set whose identifier INNAME matches the value of KVNAME from the control data file. If none of the data sets in the specified vehicle data file matches the input value of KVNAME, the VEHDYN 4.0 run aborts with an appropriate error message.

Following is a record-by-record input guide for the vehicle input file required by VEHDYN 4.0. Each record, except the first two, is to be input in free-field format. The format for each of the first two records is shown, in parentheses, to the right of the record. A definition of each of the record's variables follows each record. Each new record is identified in this guide by a line that begins with the ">" character. If a record is too long to fit on one line, a continuation line is identified by a subsequent line that begins with the "&" character. Neither the ">" nor the "&" is part of the actual data that the user will enter.

> INNAME (A8)

INNAME: Character variable (maximum of eight characters) that identifies the forthcoming vehicle data set for search purposes. INNAME is compared to KVNAME from the control data file; if they match, the upcoming vehicle data set is used to define the vehicle static configuration for the current VEHDYN 4.0 run.

> VID (A72)

VID: Character variable (maximum of 72 characters) used to identify the forthcoming vehicle data set in more detail than INNAME.

If control parameter ACCFLG<0, skip the next three records. Note: if ACCFLG=0, then include the tractive force-speed curve in the vehicle data file; however, the VEHDYN 4.0 run does not use it because it is not needed for a constant-speed run. If ACCFLG=1, the tractive force-speed curve is included in the vehicle data file and used by the VEHDYN 4.0 run in the variable-speed computations.

> NTRFSP

NTRFSP: Number of points in the tractive force-speed curve.

> SPD(1) TRF(1) SPD(2) TRF(2) . . .
& SPD(NTRFSP) TRF(NTRFSP)

SPD(i), TRF(i): Speed (mph) and tractive force (lb) for the i^{th} point on the tractive force-speed curve.

> NENG IDIESL CID NCYL RR QMAX

NENG: Number of engines.

IDIESL: Engine type (3=turbine, 2=two cycle diesel, 1=all others).

CID: Total cubic inch displacement of all engines.

NCYL: Total number of cylinders.

RR: Rolling radius or sprocket pitch radius, in.

QMAX: Maximum torque, ft-lb.

> ACD FAREA BFMX CD FDEPTH

ACD: Aerodynamic drag coefficient.

FAREA: Vehicle’s frontal area, sq ft.

BFMX: Maximum braking coefficient.

CD: Hydrodynamic drag coefficient.

FDEPTH: If positive, this is the maximum fording depth, in. Otherwise, the vehicle is a swimmer, and this number is the negated swimming speed, mph.

> **NUNITS NSTABL NDTABL NDMTBP NDMTBN NRSTAB**

NUNITS: Number of vehicle units; current program supports only one unit.

NSTABL: Number of unique spring force-deflection tables.

NDTABL: Number of unique damper force-velocity tables.

NDMTBP: Number of unique damper modification coefficient (DMC)-deflection tables for positive velocities.

NDMTBN: Number of unique DMC-deflection tables for negative velocities.

NRSTAB: Number of unique rotational spring moment-angular displacement tables.

If NSTABL = 0, skip the next five records.

If NSTABL > 0, repeat the next group of five records for each spring table. The index j refers to the j^{th} spring table; j = 1, 2, . . . NSTABL.

> **NSLOAD(j) NSUNLD(j) TSSLOP(j) STKNEG(j) STKPOS(j)**

NSLOAD(j): Number of data points in the loading portion of the j^{th} spring force-deflection relation.

NSUNLD(j): Number of data points in the unloading portion of the j^{th} spring force-deflection relation.

TSSLOP(j): Slope of the spring transition line for the j^{th} spring force-deflection relation, lb/in.

STKNEG(j): Displacement representing the rebound bump stop from the j^{th} spring force-deflection relation, in.

STKPOS(j): Displacement representing the jounce bump stop from the j^{th} spring force-deflection relation, in.

>   DELSLD(1,j) DELSLD(2,j) . . . DELSLD(NSLOAD(j),j)

(Use as many lines of free-field input as needed.)

DELSLD(i,j): The i^{th} deflection from the loading portion of the j^{th} spring force-deflection relation; an increasing function of i (Figure 3.6 in VEHDYN II User's Guide), in.

>   FORSLD(1,j) FORSLD(2,j) . . . FORSLD(NSLOAD(j),j)

(Use as many lines of free-field input as needed.)

FORSLD(i,j): The i^{th} force from the loading portion of the j^{th} spring force-deflection relation, lb.

If NSUNLD(j) = 0, skip the next two records.

>   DELSUN(1,j) DELSUN(2,j) . . . DELSUN(NSUNLD(j),j)

(Use as many lines of free-field input as needed.)

DELSUN(i,j): The i^{th} deflection from the unloading portion of the j^{th} spring force-deflection relation; an increasing function of i (Figure 3.6 in VEHDYN II User's Guide), in.

>   FORSUN(1,j) FORSUN(2,j) . . . FORSUN(NSLOAD(j),j)

(Use as many lines of free-field input as needed.)

FORSUN(i,j): The i^{th} force from the unloading portion of the j^{th} spring force-deflection relation, lb.

If NDTABL = 0, skip the next 11 records.

If NDTABL > 0, repeat the next group of five records for each damper table. The index j refers to the j^{th} damper table; j = 1, 2, . . . NDTABL.

> **NDLOAD(j) NDUNLD(j) TDSLOP(j)**

NDLOAD(j): Number of data points in the loading portion of the j^{th} damper force-velocity relation.

NDUNLD(j): Number of data points in the unloading portion of the j^{th} damper force-velocity relation.

TDSLOP(j): Slope of the damper transition line for the j^{th} damper force-velocity relation, lb-sec/in.

> **DDOTLD(1,j) DDOTLD(2,j) . . . DDOTLD(NDLOAD(j),j)**

(Use as many lines of free-field input as needed.)

DDOTLD(i,j): The i^{th} velocity from the loading portion of the j^{th} damper force-velocity relation; an increasing function of i (Figure 3.6 in VEHDYN II User's Guide), in./sec.

> **FORDLD(1,j) FORDLD(2,j) . . . FORDLD(NDLOAD(j),j)**

(Use as many lines of free-field input as needed.)

FORDLD(i,j): The i^{th} force from the loading portion of the j^{th} damper force-velocity relation, lb.

If NDUNLD(j) = 0, skip the next two records.

> **DDOTUN(1,j) DDOTUN(2,j) . . . DDOTUN(NDUNLD(j),j)**

(Use as many lines of free-field input as needed.)

DDOTUN(i,j): The i^{th} velocity from the unloading portion of the j^{th} damper force-velocity relation; an increasing function of i (Figure 3.6 in VEHDYN II User's Guide), in./sec.

> **FORDUN(1,j) FORDUN(2,j) . . . FORDUN(NDUNLD(j),j)**

(Use as many lines of free-field input as needed.)

FORDUN(i,j): The ith force from the unloading portion of the jth damperforce-velocity relation, lb.

If NDMTBP = 0, skip the next three records.

If NDMTBP > 0, repeat the next group of three records for each DMC-deflection table for positive velocities. The index j refers to the jth DMC-deflection table for positive velocities; j = 1, 2, . . . NDMTBP.

> **NDMODP(j)**

NDMODP(j): Number of data points in the jth DMC-deflection table for positive velocities.

> **DELMDP(1,j) DELMDP(2,j) . . . DELMDP(NDMODP(j),j)**

(Use as many lines of free-field input as needed.)

DELMDP(i,j): The ith deflection from the jth DMC-deflection table for positive velocities; an increasing function of i (Section 2.5.4 in VEHDYN II User's Guide), in.

> **CFMODP(1,j) CFMODP(2,j) . . . CFMODP(NDMODP(j),j)**

(Use as many lines of free-field input as needed.)

CFMODP(i,j): The ith DMC from the jth DMC-deflection table for positive velocities, dimensionless.

If NDMTBN = 0, skip the next three records.

If NDMTBN > 0, repeat the next group of three records for each DMC-deflection table for negative velocities. The index j refers to the jth DMC-deflection table for positive velocities; j = 1, 2, . . . NDMTBP.

> **NDMODN(j)**

NDMODN(j): Number of data points in the jth DMC-deflection table for negative velocities.

> **DELMDN(1,j) DELMDN(2,j) . . . DELMDN(NDMODN(j),j)**

(Use as many lines of free-field input as needed.)

DELMDN(i,j): The i^{th} deflection from the j^{th} DMC-deflection table for negative velocities; an increasing function of i (Section 2.5.4 in VEHDYN II User's Guide), in.

> **CFMODN(1,j) CFMODN(2,j) . . . CFMODN(NDMODN(j),j)**

(Use as many lines of free-field input as needed.)

CFMODN(i,j): The i^{th} DMC from the j^{th} DMC-deflection table for negative velocities, dimensionless.

If NRSTAB = 0, skip the next five records.

If NRSTAB > 0, repeat the next group of five records for each rotational spring table. The index j refers to the j^{th} rotational spring table; j = 1, 2, . . . NRSTAB.

> **NRSLOD(j) NRSULD(j) TRSLOP(j) RSSTKN(j) RSSTKP(j)**

NRSLOD(j): Number of data points in the loading portion of the j^{th} rotational spring moment-angular displacement relation.

NRSULD(j): Number of data points in the unloading portion of the j^{th} rotational spring moment-angular displacement relation.

TRSLOP(j): Slope of the rotational spring transition line for the j^{th} rotational spring moment-angular displacement relation, lb-in./rad.

RSSTKN(j): Angular displacement representing the rebound bump stop from the j^{th} rotational spring moment-angular displacement relation, rad.

RSSTKP(j): Angular displacement representing the jounce bump stop from the j^{th} rotational spring moment-angular displacement relation, rad.

> **PHILOD(1,j) PHILOD(2,j) . . . PHILOD(NRSLOD(j),j)**

(Use as many lines of free-field input as needed.)

PHILOD(i,j): The i^{th} angular displacement from the loading portion of the j^{th} rotational spring moment-angular displacement relation; an increasing function of i, rad.

> **MOMLOD(1,j) MOMLOD(2,j) . . .MOMLOD(NRSLOD(j),j)**

(Use as many lines of free-field input as needed.)

MOMLOD(i,j): The i^{th} moment from the loading portion of the j^{th} rotational spring moment-angular displacement relation, lb-in.

If NRSULD(j) = 0, skip the next two records.

> **PHIULD(1,j) PHIULD(2,j) . . . PHIULD(NRSULD(j),j)**

(Use as many lines of free-field input as needed.)

PHIULD(i,j): The i^{th} angular displacement from the unloading portion of the j^{th} rotational spring moment-angular displacement relation; an increasing function of i, rad.

> **MOMULD(1,j) MOMULD(2,j) . . .MOMULD(NRSULD(j),j)**

(Use as many lines of free-field input as needed.)

MOMULD(i,j): The i^{th} moment from the unloading portion of the j^{th} rotational spring moment-angular displacement relation, lb-in.

All of the variables listed in the remainder of this section have an index denoting a specific unit of a multiunit vehicle. This index has been omitted in this guide, however, because VEHDYN 4.0 supports only single-unit vehicles at this time.

> **NTRKS NB NW NI NU I3030 IFSEAT**

NTRKS: Number of track sections.

NB: Number of bogie suspensions.

NW: Number of walking beam suspensions.

NI: Number of independent suspensions.

NU: Number of unsprung suspensions.

I3030: Number of 30/30 front-end type (3030F) suspensions (0 or 1).

IFSEAT: Integer to select the type of driver's seat computations.
= 1 compute absorbed power without seat dynamics
= 2 compute absorbed power with seat dynamics
= 3 don't compute absorbed power at the driver's seat

> **LDBASE HDBASE**

LDBASE: Longitudinal distance from the CG to the base of the driver's seat; positive if the seat is forward of the CG, negative if it is rearward, in.

HDBASE: Normal distance from the CG to the base of the driver's seat; positive if the seat base is above the CG, negative if it is below, in.

> **DRVWGT HDSEAT ISDRV IDDRV**

DRVWGT: Combined weight of the driver and the driver's seat, lb.

HDSEAT: Normal distance from the base of the driver's seat to the driver's seat; positive if the driver's seat is above its base, negative if it is below, in.

ISDRV: Number of the spring force-deflection table to be used in computing the driver's seat dynamics.

IDDRV: Number of the damper force-velocity table to be used in computing the driver's seat dynamics.

> **GVW VMMI ZCG H1 H2 L1 L2 THETA**

GVW: Gross weight of the entire vehicle; if the vehicle is tracked, include only that portion of the track that is to be counted in the vehicle's sprung weight, lb.

VMMI: Pitch mass moment of inertia about the CG of the vehicle's sprung mass, lb-sec^2-in.

ZCG: Settled height of the CG above the ground surface, in.

H1: Normal distance from the bottom of the vehicle's sprung mass (i.e., average location of suspension-to-hull connection points) to the CG; a positive number, in.

H2: Normal distance from the CG to the top of the vehicle's sprung mass; a positive number, in.

L1: Longitudinal distance from the rear end of the vehicle's sprung mass to the CG; a positive number, in.

L2: Longitudinal distance from the CG to the front end of the vehicle's sprung mass; a positive number, in.

THETA: Angle of the settled sprung mass with respect to and measured counterclockwise from the horizontal, rad.

Repeat the next record for each road wheel on the vehicle. The index j refers to the wheel number; j = 1, 2, . . . NWLS. Wheels are numbered fore to aft.

> **R(j) W(j) LWHL(j) ZWHL(j) DEFL(j) ZFORCE(j)**
& FUNDWL(j) IDRIVE(j) CDWHLS(j) TSECTW(j) TSECTH(j)
& TSFLG(j) NUMWHLONAXLE(j) ANGCTR(j) DELANG(j)
& NWLPTS(j)

R(j): Undeflected radius of the jth wheel; include the track thickness for a tracked road wheel, in.

W(j): Weight of the jth wheel assembly; input value for one side of the vehicle only, lb.

LWHL(j): Longitudinal distance from the CG to the center of the jth wheel; positive if the wheel center is forward of the CG, negative if it is rearward, in.

ZWHL(j): Settled height of the center of the j^{th} wheel above the ground, in.

DEFL(j), ZFORCE(j): Deflection (in.) and corresponding force (lb), respectively, of the representative point from the force-deflection relation for the j^{th} wheel.

FUNDWL(j): Measured settled load under the j^{th} wheel, lb.

IDRIVE(j): Integer that indicates whether the j^{th} wheel is a drive wheel (=1) or a towed wheel (=0). All road wheels inside a track envelope should be designated as drive wheels.

CDWHLS(j): Damping coefficient for j^{th} wheels; see Appendix A.

TSECTW(j): Tire section width for j^{th} wheel, in.

TSECTH(j): Tire section height for j^{th} wheel, in.

TSFLG(j): Tire stiffness flag for j^{th} wheel, in.
= 1 for "flexible" (0.000 <= RR < 0.020 at 25% deflection)
= 2 for "medium" (0.020 <= RR < 0.035 at 25% deflection)
= 3 for "stiff" (RR >= 0.035 at 25% deflection)
where RR = normalized rolling resistance.

NUMWHLONAXLE(j): Number of wheels on one side of vehicle per axle.

The next three variables relate to the process of transforming the wheels/SPRIDLERs into a set of discrete points used in the track model. Realistic values are required only for a tracked vehicle. If NWLPTS(j)<0, the default values are in effect.

ANGCTR(j): Central polar angle of j^{th} wheel's arc, deg; default value=0.

DELANG(j): Polar angle between each discrete wheel point on the j^{th} wheel, deg; default value=12.

NWLPTS(j): Number of total discrete points on the j^{th} wheel; default value=30.

If NTRKS = 0, skip the next three record groups.

\> **NSPX NWB(1) NWE(1) ITRKTYPE**

NSPX: Number of SPRIDLERs; possible values = 0, 1, 2.

NWB(1): Number of the forwardmost road wheel within the 1st track envelope.

NWE(1): Number of the rearmost road wheel within the 1st track envelope.

ITRKTYPE: Type of track model

= 0	wheeled vehicle
= 1	band track model with uniform tension
= 2	band track model with localized tension
= 3	interconnecting spring model with profile smoothing and band track loop calculation
= 4	interconnecting spring model with profile smoothing but with no band track loop computation or output

Repeat the next record for each additional track. The index j refers to the track number; j = 2, . . . NTRKS. Note that, for multiple tracks, NSPX must be 0 and ITRKTYPE must be either 1 or 2.

\> **NWB(j) NWE(j)**

NWB(j): Number of the forwardmost road wheel within the j^{th} track envelope.

NWE(j): Number of the rearmost road wheel within the j^{th} track envelope.

\> **B NTRFLG GROUSH L ASHOE NTRPAD ZMAX**

B: Track width, in.

NTRFLG: Track type flag; =0 (flexible track), =1 (girder-type track).

GROUSH: Grouser height, in.

L: Length of track on ground, in.

ASHOE: Area of a track shoe, sq in.

NTRPAD: Track pad flag; =0 (no track pads), =1 (track pads).

ZMAX: Maximum possible sinkage, in.

If ITRKTYPE<3, include the next record(s).

Repeat the next record for each track. The index j refers to the track number; j = 1, . . . NTRKS. Note that, for multiple tracks, NSPX must be 0 and ITRKTYPE must be either 1 or 2.

\> **TTENS0(j) TNSDTL(j)**

TTENS0(j): Value of track tension at settlement, lb.

TNSDTL(j): Change in track tension per change in track length, lb/in.

If ITRKTYPE>2, include the next record.

\> **FSPRNG TRACKK(NWB) TRACKK(NWB+1) . . .**
& TRACKK(NWE-1)

FSPRNG: Feeler spring constant, lb/in.; see page 39, VEHDYN II User's Guide (Creighton 1986).

TRACKK(j): j^{th} interconnecting spring constant for spring joining wheels numbered j and j+1 , lb/in.; see page 39, VEHDYN II User's Guide (Creighton 1986).

If NSPX = 0, skip the next record.

If NSPX > 0, repeat the next record for each SPRIDLER to be considered. The index j refers to the j^{th} SPRIDLER, numbered fore to aft; j = 1, NSPX.

\> **RAD(j) LSPX(j) ZSPX(j) SXDEFL(j) SXFORC(j)**
& CDSPX(j) ANGCTR(j+NWLS) DELANG(j+NWLS)
& NWLPTS(j+NWLS)

RAD(j): Undeflected radius of the j^{th} SPRIDLER measured from the SPRIDLER center to the outside of the track, in.

LSPX(j): Longitudinal distance from the CG to the center of the j^{th} SPRIDLER; positive if the SPRIDLER center is forward of the CG, negative if it is rearward, in.

ZSPX(j): Settled height of the center of the j^{th} SPRIDLER above the ground surface, in.

SXDEFL(j), SXFORC(j): Deflection (in.) and corresponding force (lb), respectively, of the representative point from the force-deflection relation for the j^{th} SPRIDLER.

CDSPX(j): Damping coefficient for the j^{th} SPRIDLER.

The next three variables relate to the process of transforming the wheel into a set of discrete points used in the track model. If NWLPTS(j+NWLS)<0, the default values are in effect.

ANGCTR(j+NWLS): Central polar angle of j^{th} SPRIDLER's arc, deg; default value=0.

DELANG(j+NWLS): Polar angle between each discrete SPRIDLER point on the j^{th} SPRIDLER, deg; default value=12.

NWLPTS(j+NWLS): Number of total discrete points on the j^{th} SPRIDLER; default value=30.

If I3030 = 0, skip the next five records.

> L3030(1) Z03030(1) L3030(2) Z03030(2) L3030(3)
& Z03030(3) L3030(4) Z03030(4) L3030(5) Z03030(5)

L3030(i): Longitudinal distance from the CG to the i^{th} point of the 3030F suspension (Figure 9, A=1, B=2, C=3, D=4, E=5).

Z03030(i): Vertical distance from the ground to the i^{th} point of the 3030F suspension.

> **I3STB(1) I3DTB(1) I3MTP(1) I3MTN(1)**

I3STB(1): Number of spring table for spring connecting points B and E of the 3030F.

I3DTB(1): Number of damper force-velocity table for damper connecting points B and E of the 3030F.

I3MTP(1): Number of damper DMC-deflection table for positive velocities for damper connecting points B and E of the 3030F.

I3MTN(1): Number of damper DMC-deflection table for negative velocities for damper connecting points B and E of the 3030F.

> **I3STB(2) I3DTB(2) I3MTP(2) I3MTN(2) I3RSTB(1)**

I3STB(2): Number of spring table for spring connecting points A and C of the 3030F.

I3DTB(2): Number of damper force-velocity table for damper connecting points A and C of the 3030F.

I3MTP(2): Number of damper DMC-deflection table for positive velocities for damper connecting points A and C of the 3030F.

I3MTN(2): Number of damper DMC-deflection table for negative velocities for damper connecting points A and C of the 3030F.

I3RSTB(1): Number of rotational spring table for small beam's rotational spring located at C and keyed off angle ($\varphi_1 = \gamma_A + \frac{\pi}{2} - \beta$).

> **I3STB(3) I3DTB(3) I3MTP(3) I3MTN(3) I3RSTB(2)**

I3STB(3): Number of spring table for spring connecting points D and E of the 3030F.

I3DTB(3): Number of damper force-velocity table for damper connecting points D and E of the 3030F.

I3MTP(3): Number of damper DMC-deflection table for positive velocities for damper connecting points D and E of the 3030F.

I3MTN(3): Number of damper DMC-deflection table for negative velocities for damper connecting points D and E of the 3030F.

I3RSTB(2): Number of rotational spring table for big beam's rotational spring located at E and keyed off angle DEH ($\varphi_2 = \gamma_D + \pi - \alpha_1$).

> B3INRT B3RDMP S3INRT S3RDMP

B3INRT: Big beam CEH's mass moment of inertia, lb-sec^2-in.

B3RDMP: Big beam's frictional damping coefficient for resisting rotation, lb-in.

S3INRT: Small beam FCG's mass moment of inertia, lb-sec^2-in.

S3RDMP: Small beam's frictional damping coefficient for resisting rotation, lb-in.

If NU = 0, skip the next record.

> IU(1) IU(2) . . . IU(NU)

IU(i): Number of the wheel attached to the ith unsprung suspension.

If NI = 0, skip the next record.

If NI > 0, repeat the next record for each independent suspension. The index i refers to the ith independent suspension; i = 1, 2, . . . NI.

> II(i) IINSTB(i) IINDTB(i) INDMTP(i) INDMTN(i)

II(i): Number of the wheel attached to the ith independent suspension.

IINSTB(i): Number of the spring force-deflection table to be used with the spring from the ith independent suspension.

IINDTB(i): Number of the damper force-velocity table to be used with the damper from the i^{th} independent suspension.

INDMTP(i): Number of the DMC-deflection table for positive velocities to be used with the damper from the i^{th} independent suspension.

INDMTN(i): Number of the DMC-deflection table for negative velocities to be used with the damper from the i^{th} independent suspension.

If NW = 0, skip the next three records.

If NW > 0, repeat the next group of three records for each walking beam. The index k refers to the k^{th} walking beam; k = 1, 2, . . . NW.

\> **IWBSTB(k) IWBDTB(k) IWDMTP(k) IWDMTN(k) & IWRSTB(k)**

IWBSTB(k): Number of the spring force-deflection table to be used with the spring from the k^{th} walking beam.

IWBDTB(k): Number of the damper force-velocity table to be used with the central damper from the k^{th} walking beam.

IWDMTP(k): Number of the DMC-deflection table for positive velocities to be used with the central damper from the k^{th} walking beam.

IWDMTN(k): Number of the DMC-deflection table for negative velocities to be used with the central damper from the k^{th} walking beam.

IWRSTB(k): Number of the rotational spring moment-angular displacement table to be used with the beam rotational spring from the k^{th} walking beam.

\> **LW(k) WBINRT(k) WBRDMP(k)**

LW(k): Longitudinal distance from the CG to the pivot point of the k^{th} walking beam; positive if the pivot point is forward of the CG, negative if it is rearward, in.

WBINRT(k): Mass moment of inertia of the k^{th} walking beam's beam, lb-sec^2-in.

WBRDMP(k): Beam's frictional damping coefficient for resisting rotation for the k^{th} walking beam, lb-in.

Include the next record for the fore (i=1), then the aft (i=2) sections of the k^{th} walking beam.

> **IW(i,k) IWODTB(i,k) IWOMTP(i,k) IWOMTN(i,k)**
& LWDF(i,k) LWDB(i,k)

IW(i,k): Number of the i^{th} wheel contained in the k^{th} walking beam.

IWODTB(i,k): Number of the damper force-velocity table to be used with the i^{th} outboard damper from the k^{th} walking beam.

IWOMTP(i,k): Number of the DMC-deflection table for positive velocities to be used with the i^{th} outboard damper from the k^{th} walking beam.

IWOMTN(i,k): Number of the DMC-deflection table for negative velocities to be used with the i^{th} outboard damper from the k^{th} walking beam.

LWDF(i,k): Longitudinal distance from the CG to the connection point of the i^{th} outboard damper from the k^{th} walking beam to the vehicle frame; positive if the connection point is forward of the CG, negative if it is rearward, in.

LWDB(i,k): Longitudinal distance from the CG to the connection point of the i^{th} outboard damper from the k^{th} walking beam to the beam; positive if the connection point is forward of the CG, negative if it is rearward, in.

If NB = 0, skip the next two records.

If NB > 0, repeat the next group of two records for each bogie. The index j refers to the j^{th} bogie; j = 1, 2, . . . NB.

> **LB(j) ZBGPIV(j) BGANGL(j) BGINRT(j) BGRDMP(j)**
& IBRSTB(j)

LB(j): Longitudinal distance from the CG to the pivot point of the j^{th} bogie; positive if the pivot point is forward of the CG, negative if it is rearward, in.

ZBGPIV(j): Settled height of the pivot point from the j^{th} bogie above the ground surface, in.

BGANGL(j): Settled orientation angle of the j^{th} bogie's beam with respect to the horizontal, rad.

BGINRT(j): Mass moment of inertia of the j^{th} bogie's beam, lb-sec^2-in.

BGRDMP(j): Beam's frictional damping coefficient for resisting rotation for the j^{th} bogie, lb-in.

IBRSTB(j): Number of the rotational spring moment-angular displacement table to be used with the beam rotational spring from the j^{th} bogie.

Include the next record for the fore (i=1), then the aft (i=2) sections of the j^{th} bogie.

> **IB(i,j) IBSTB(i,j) IBDTB(i,j) IBDMTP(i,j) IBDMTN(i,j) & DLB(i,j)**

IB(i,j): Number of the i^{th} wheel of the j^{th} bogie.

IBSTB(i,j): Number of the spring force-deflection table to be used with the i^{th} spring of the j^{th} bogie.

IBDTB(i,j): Number of the damper force-velocity table to be used with the i^{th} damper of the j^{th} bogie.

IBDMTP(i,j): Number of the DMC-deflection table for positive velocities to be used with the i^{th} damper of the j^{th} bogie.

IBDMTN(i,j): Number of the DMC-deflection table for negative velocities to be used with the i^{th} damper of the j^{th} bogie.

DLB(i,j): Longitudinal distance from the CG to the connection point of the i^{th} damper of the j^{th} bogie to the vehicle frame; positive if the connection point is forward of the CG, negative if it is rearward, in.

Vehicle-terrain interaction data

The vehicle-terrain interaction data file contains information regarding the terrain profile for runs over deformable terrain. Also contained in this data file are start locations for any bodies of water that may be encountered during a run. Following is a record-by-record guide for the vehicle-terrain interaction data file in the same format as the previous input files. Each new record is identified by a “>” sign, and continuation lines are identified with the “&” sign.

> **SURFAC**

SURFAC: Number of distinct surface sections or materials; must be an integer greater than 0.

Repeat the next record for each surface section; the index j refers to the j^{th} surface section; j=1, 2, ..., SURFAC.

> **SSTART(j) SSTEND(j) TRCI(j) STYPE(j)**

SSTART(j): Horizontal start location of the j^{th} surface, in.

SSTEND(j): Horizontal ending location of the j^{th} surface, in.

TRCI(j): Surface/soil strength of the j^{th} surface, RCI.

STYPE(j): USCS soil classification type (character *4).

> **NUMPUD**

NUMPUD: Number of distinct bodies of water.

Repeat the next record for each body of water when NUMPUD>0; the index j refers to the j^{th} body of water; j=1, 2, ...NUMPUD.

> **XWATER(j) H2OVEL(j)**

XWATER(j): Horizontal location for the beginning of the j^{th} body of water, in.

H2OVEL(j): Water (cross current) velocity, mph.

Standard underside impact detection data file

To model interactions between the underside of the vehicle and the ground, the user must include one or two UID data files. The first one, called the standard UID data file, models the section of the vehicle along where the wheels are located, describing the underside between the wheels, ahead of the first wheel, and behind the last wheel. The second UID file, called the full UID data file, models the belly of the vehicle between the right and left wheels. If the user wishes, he may leave this data file out and VEHDYN 4.0 will use the data in the standard UID file to model the belly as well. As with the descriptions of the previous data files, this section uses “>” to designate a new record and “&” to designate a record continuation.

> **UDEL UFORCE**

UDEL, UFORCE: Single data point from a UID wheel’s force-deflection response; UDEL deflection (in.), UFORCE force (lb).

> **ZHITCH LWHL2HCH**

ZHITCH: Height of the vehicle’s rear hitch above the ground at equilibrium, in.

LWHL2HCH: Longitudinal distance between the rear hitch and the first wheel’s axle, in.

> **DUIDMX**

DUIDMX: Maximum allowable diameter of UID wheels; actual diameter will be computed and will be no bigger than this value, in.

> **NUNDPTS**

NUNDPTS: Number of underside definition points.

> **YUNDHCH(1) ZUND(1) YUNDHCH(2) ZUND(2) . . .**
> **& YUNDHCH(NUNDPTS) ZUND(NUNDPTS)**

YUNDHCH(j): Horizontal coordinate of the j^{th} underside point measured relative to the rear hitch; begin entering data from the vehicle front end and work toward the rear of the vehicle. This horizontal array should therefore be a descending array, in.

ZUND(j): Vertical height of the j^{th} underside point measured relative to the ground surface, in.

Full underside impact detection (UID) data file

The full UID data file defines the belly section between the right and left wheels. If not included, the data from the standard UID file are used to define the belly. If included, the format of the full UID data file is exactly the same as the format of the standard UID data file merely defining an underside contour for the belly which is different from the underside contour for the vehicle section along the wheels.

Ride performance profile list input data file

The ride performance profile list input data file is required when the user desires to make a series of runs with a single vehicle over several terrain ride profiles and obtain the ride performance (such as the 6-W speed) for that vehicle for each of the entered profiles. This file merely contains a list of profiles and a starting speed for each profile for the initial run to be made for that profile. The format is as follows:

Repeat the next two records for each different course to be run. All of the profiles listed in this file should be among a stack of profile data sets in a single data file in the proper format for profile data, as indicated earlier in this chapter.

> **KPNAME (A8)**

KPNAME: Character variable (maximum of eight characters) that is the profile descriptive name for a given profile data set.

> VSTART

VSTART: Starting speed for the vehicle over the course designated by KPNAME; a sequence of constant-speed runs will be made to zero in on the target speed (e.g., 6-W speed) with the first such run executing at VSTART, mph.

5 Sample Problems

This chapter now presents several sample problems to demonstrate the usage of VEHDYN 4.0 and its postprocessing support programs. Table 3 is a list of the problems discussed in this chapter. An attempt has been made to provide a broad range of problem types to give the prospective user a chance to realize the broad capability of VEHDYN 4.0. The discussion of each sample problem will proceed by providing a short description of the problem, followed by a screen capture of the interactive running of the problem, followed by an examination of both the input and output files. It is beyond the scope of this report to present a lengthy discussion of the results of each run. Rather, this chapter merely demonstrates the use of VEHDYN 4.0.

Table 3. List of sample problems.

Sample Problem Number	Problem Description
1	LAV 8×8 riding over 500-ft nondeformable LET7AVG course at constant 10-mph speed
2	M1 tank riding over 500-ft deformable LET7AVG course at constant 10-mph speed
3	M113 APC ACC/DEC over 500-ft nondeformable LET7AVG course – speed varying from 0.1 mph to 10 mph
4	M977 HEMTT ACC/DEC over 500-ft deformable LET7AVG course – speed varying from 0.1 mph to 10 mph
5	VCI1 determination for M2 BFV over deformable SMSC soil type surface
6	Maximum slope climb of M998 HMMWV over nondeformable surface
7	Maximum slope climb of LMTV 4×4 truck over 300-RCI SM-type deformable soil
8	Gap crossing (VGAP) of LAV 8x8 over deformable gap obstacle
9	Obstacle crossing (ObsMod) of M1 tank over nondeformable series of obstacles
10	Ride performance of CAT30/30 tractor over series of nondeformable ride courses
11	Shock performance of M998 HMMWV over series of nondeformable half-round obstacles
12	Drawbar pull determination for an M2 BFV on a deformable 200-RCI SM soil

Problem 1: LAV 8×8 wheeled vehicle over nondeformable 500-ft LET7AVG ride course at 10-mph constant speed

The first problem to be demonstrated involves the movement of a light armored vehicle (LAV), an 8×8 wheeled vehicle, at a constant speed of

10 mph down a nondeformable 500-ft LET7AVG ride course. Using Table 2 as a guide, the required input files for this type of run (called "single constant-speed ride/shock nondeformable" in the table) are the profile data file, vehicle data file, control data file, and standard UID data file. The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown as Figure 11.

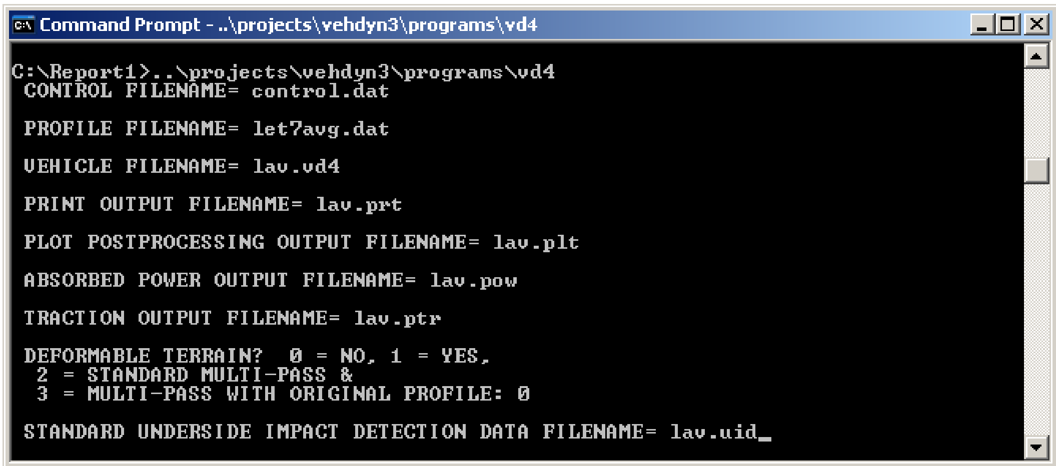


Figure 11. Screen capture for problem 1.

The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 12.

```
LAV
LET7AVG
10.0,0.001,0,60.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,0,0
```

Figure 12. Control data file.

The first few lines of the profile data file, named let7avg.dat, are shown in Figure 13.

```
LET7AVG
LETOURNEAU COURSE 7 AVG 31MAY83
  501       12.
    0.     -0.06   -0.18   -0.90   -1.86   -3.12   -5.28   -6.06   -6.84   -7.50
  -7.44    -7.44   -7.32   -7.14   -7.08   -7.02   -6.90   -5.22   -3.90   -2.88
  -1.50    -0.42    0.54    1.38    2.22    2.88    3.06    3.24    2.22    0.48
  -2.10    -4.56   -5.22   -5.58   -5.88   -5.82   -5.76   -6.60   -6.48   -6.06
  -6.18    -5.76   -5.58   -5.64   -5.64   -5.28   -3.48    0.60    2.82    3.00
   3.18     3.24    3.30    3.48    3.90    3.84    3.00    1.68    1.14    0.60
```

Figure 13. Top of profile data file.

This file continues until all 501 data points are given, defining the entire 500-ft LeTourneau 7 ride course. The vehicle data file for the 8×8 LAV is shown in Figure 14.

```
LAV
LIGHT ARMOR VEHICLE WHEELED 8X8
47
0,39293,1,36764,2,31574,2.53,28700,3,26363,3.36,24600,4,21408
4.39,19541,5,14359,6,13150,7,11960,8,9903,9,9704,10,9502
11,8798,12,6735,13,6645,14,6561,15,6433,16,6085,16.83,5789
18,4419,19,4372,20,4336,21,4303,22,4261,23,4201,24,4067,25,3895
26,3754,27,3210,28,3191,29,3167,30,3141,31,3093,32,3015,33,2915
34,2831,35,2752,36,2634,37,2569,38,2490,39,2425,40,2364,41,2304
42,1703,42.63,1225
1 2 318 6 18.24 586
0.75 50.0 0.49 0.8 9999.9
1,2,1,0,0,0
4,0,0.,-1.5,11.8
-2.5,-1.5,11.8,12.8
-21125,1125,8850,28850
4,0,0.,-.5,12.8
-1.5,-.5,12.8,13.8
-20275,-275,7040,27040
2,0,0.
-10,10
-1180,1180
0,0,0,4,0,0,1
61.25,0.
170.,0.,0,0
39412,357500,53.8,29.8,42.2,113.75,129.25,0.
19.25,345,72.25,17.8,1.45,3445,4577.9,1,0,12.2,12.24,1,1,0,0,-1
19.25,345,26.95,17.8,1.45,3445,4868.6,1,0,12.2,12.24,1,1,0,0,-1
19.25,345,-23.65,17.8,1.45,3445,4991.5,1,0,12.2,12.24,1,1,0,0,-1
19.25,345,-64.55,17.8,1.45,3445,5268.0,1,0,12.2,12.24,1,1,0,0,-1
1,1,1,0,0
2,1,1,0,0
3,2,1,0,0
4,2,1,0,0
```

Figure 14. LAV vehicle data file.

After these three key input files, a fourth and final input file is required which describes the underside geometry of the vehicle. This UID data file is shown in Figure 15 for the LAV.

Finally, the VEHDYN 4.0 interaction asks for the user to input an integer describing whether there is to be deformable terrain and/or multiple pass calculations and the names for four output files: the formatted print output file, the plot postprocessing output file, the absorbed power output file, and the traction output data file. These files do not need to be created before the run; the program will create each one as needed. Care must be taken not to use the same name as an existing file in the working directory, as the program will overwrite an existing file.

After the program completes its run, the output files can be examined and plot postprocessing can be accomplished if the user desires. The first

```
.1,750.
26.5,186.0
3.0
8
251.,50.
187.,15.4
180.,20.
55.,20.
50.,15.4
45.,20.
23.5,20.
0.,66.5
39.0
```

Figure 15. LAV UID file.

thing the user notices about this run is that the traction output file contains no data. This file contains data only for (1) a drawbar pull run, (2) filtered maximum traction coefficients when nfilts>0, (3) ride/shock performance data for ride/shock performance runs, (4) ObsMod profiles generated for an ObsMod run, and (5) general error diagnostic data when a run has an algorithmic computational problem during execution. None of these conditions exists in this problem; hence, the file is empty.

The other three output files are generally quite lengthy and are given below in partial format in order to give the user an idea of what should be expected as a result of executing this first sample problem. The start and end of the absorbed power output file are shown in Figure 16.

Figures 17, 18 and 19 show portions of the formatted print output file, including the output for the first and last print increment.

Figure 20 shows a listing of the start of the plot postprocessing output file.

This last file is primarily to be used as an input file to a postprocessing program such as ERDC's animation program VD4Animator. An example snapshot of the kind of output that can be displayed with VD4Animator from this first sample problem is shown as Figure 21.

Throughout this chapter, snapshots from the animation program will be exhibited to partially demonstrate the animator's capability.

```
ABSORBED POWER STARTS AT  0.00 FEET,  FILTER CUTOFF FREQUENCY= 30.0 HZ.
VEHICLE...LIGHT ARMOR VEHICLE WHEELED 8X8
 PROFILE...LETOURNEAU COURSE 7 AVG 31MAY83

Unit    Longitudinal Distance     Normal Distance     Settled Normal Distance     Dynamics
 No.          CG to Base             CG to Base            Base to Seat           Included?
                 (in)                   (in)                   (in)
  1             61.25                   0.00                   0.00                  NO

    T (SEC)     Y (FT)      UNIT  **ABSORBED POWER (W)**  ***UNFILTERED ACCELERATION (G'S)***  ***FILTERED ACCELERATION (G'S)***
                                   INSTANT.    AVERAGE      INSTANT.  RMS     MIN     MAX        INSTANT.    RMS    MAX ABS

     0.000      0.000       1        0.00        0.00         0.00    0.00    0.00    0.00         0.00      0.00    0.00
     0.500      7.385       1        0.55        2.83        -0.13    0.21   -0.92    0.13        -0.09      0.21    0.47
     1.000     14.557       1        0.14        1.63         0.13    0.17   -0.92    0.16         0.13      0.16    0.47
     1.500     22.159       1        0.34        1.29         0.14    0.17   -0.92    0.34         0.14      0.17    0.47
     2.000     29.691       1        0.02        1.08        -0.27    0.17   -0.92    0.34        -0.26      0.16    0.47
     2.500     36.523       1        0.79        1.31         0.25    0.20   -0.92    0.34         0.23      0.20    0.47
     3.000     43.949       1        0.10        1.14         0.13    0.20   -0.92    0.34         0.15      0.20    0.47
     3.500     51.618       1        3.53        1.64        -0.00    0.21   -0.92    0.50        -0.04      0.21    0.50
.
.
.
    30.500    447.520       1        0.02        2.10        -0.44    0.24   -0.92    0.74        -0.43      0.24    0.74
    31.000    454.534       1        2.57        2.14         0.16    0.24   -0.92    0.74         0.20      0.24    0.74
    31.500    462.214       1        0.03        2.12        -0.22    0.24   -0.92    0.74        -0.27      0.24    0.74
    32.000    469.414       1        2.43        2.14         0.23    0.24   -0.92    0.74         0.29      0.24    0.74
    32.500    477.020       1        0.53        2.15        -0.28    0.24   -0.92    0.74        -0.25      0.24    0.74
    33.000    483.785       1        5.09        2.30         0.41    0.25   -0.92    0.74         0.43      0.25    0.74
    33.500    491.507       1        0.06        2.28        -0.13    0.25   -0.92    0.74        -0.10      0.25    0.74
    34.000    498.903       1        2.09        2.25         0.10    0.24   -0.92    0.74         0.13      0.24    0.74
    34.075    500.001       1        1.82        2.25        -0.05    0.24   -0.92    0.74        -0.04      0.24    0.74

FOO=          764.8 lbs
FOOMAX=         6085.3 lbs
```

Figure 16. Top and bottom of absorbed power output file.

```
VEHICLE   LAV           PROFILE   LET7AVG         VEHICLE SPEED= 10.0 MPH ( 176.0 IPS)
INTEGRATION STEP SIZE DT=0.001000 SEC            VEHICLE WILL RUN FOR   34.09 SECONDS (   500.0 FEET)
OUTPUT INCREMENTS: DTPRNT= 0.500000 SEC, DTPLOT= 0.010000 SEC, DTPWR= 0.500000 SEC
ABSORBED POWER STARTS AT  0.00 FEET,  FILTER CUTOFF FREQUENCY= 30.0 HZ.

VEHICLE ID.   LIGHT ARMOR VEHICLE WHEELED 8X8

PROFILE ID.   LETOURNEAU COURSE 7 AVG 31MAY83

 <NUMBER OF POINTS IN TRACTIVE FORCE-SPEED CURVE>  47
 PT#     TF (POUNDS)    SPEED (MPH)
   1      39293.000         0.000
   2      36764.000         1.000
   3      31574.000         2.000
   4      28700.000         2.530
   5      26363.000         3.000
   6      24600.000         3.360
   7      21408.000         4.000
   8      19541.000         4.390
   9      14359.000         5.000
  10      13150.000         6.000
  11      11960.000         7.000
  12       9903.000         8.000
  13       9704.000         9.000
  14       9502.000        10.000
  15       8798.000        11.000
  16       6735.000        12.000
  17       6645.000        13.000
  18       6561.000        14.000
  19       6433.000        15.000
  20       6085.000        16.000
  21       5789.000        16.830
  22       4419.000        18.000
  23       4372.000        19.000
  24       4336.000        20.000
  25       4303.000        21.000
  26       4261.000        22.000
  27       4201.000        23.000
  28       4067.000        24.000
  29       3895.000        25.000
  30       3754.000        26.000
  31       3210.000        27.000
  32       3191.000        28.000
  33       3167.000        29.000
  34       3141.000        30.000
  35       3093.000        31.000
  36       3015.000        32.000
  37       2915.000        33.000
  38       2831.000        34.000
  39       2752.000        35.000
  40       2634.000        36.000
  41       2569.000        37.000
  42       2490.000        38.000
  43       2425.000        39.000
  44       2364.000        40.000
  45       2304.000        41.000
  46       1703.000        42.000
  47       1225.000        42.630

 <AERODYNAMIC COEFFICIENT OF DRAG>  0.75
 <FRONTAL AREA (SQUARE FEET)>  50.
 <BRAKING COEFFICIENT>  0.49
```

Figure 17. Top of print output file.

Problem 2: M1 Abrams tank over deformable 500-ft LET7AVG ride course at 10-mph constant speed

The second sample problem differs from the first problem in that it is a tracked vehicle (M1 Abrams tank) traveling over a deformable version (soil strength of 40 RCI and SM soil classification) of the 500-ft LET7AVG ride course at the same 10-mph constant speed. For the “single constant-speed ride/shock deformable” run, use Table 2 to determine which additional input files are needed. An extra underside detection file to model the vehicle belly could be added (but will be the same data as is in the standard UID file for this sample problem), as well as the VMI (vehicle-media interaction) file containing the soil strength data. The VEHDYN 4.0 program is again run from a command line in a DOS window, as shown in the screen capture (Figure 22).

```
THERE ARE   2 SPRING TABLES,  1 DAMPER TABLES, AND  0 DAMPER-MODIFICATION TABLES ( 0 FOR POSITIVE VELOCITIES, 0 FOR NEGATIVE)
THERE ARE   0 ROTATIONAL SPRING TABLES

<<<SPRING TABLES>>>     (TABLE TYPE="SPG")

 << #1 >> TSSLOPE= 0.00E+00 LB/IN, BUMP STOPS AT  -1.50 AND  11.80 INCHES DISP.
<LOADING>
 DISP(IN)        -2.50      -1.50      11.80      12.80
 FORCE(LB)     -21125.      1125.      8850.     28850.

 << #2 >> TSSLOPE= 0.00E+00 LB/IN, BUMP STOPS AT  -0.50 AND  12.80 INCHES DISP.
<LOADING>
 DISP(IN)        -1.50      -0.50      12.80      13.80
 FORCE(LB)     -20275.      -275.      7040.     27040.

<<<DAMPER TABLES>>>     (TABLE TYPE="DMP")

 << #1 >> TDSLOPE= 0.00E+00 LB-SEC/IN
<LOADING>
 VEL(IPS)       -10.00      10.00
 FORCE(LB)      -1180.      1180.

 <<< VEHICLE UNIT #1 >>>

<<UNIT TYPE>>WHEEL <<GROSS VEHICLE WEIGHT>> 39412. LBS <<UNIT MOMENT OF INERTIA (SPRUNG MASS)>>    357500. LB-SEC^2-IN
                   <<SPRUNG VEHICLE WEIGHT>> 36652. LBS

 ************UF            Y(UF),Z(UF)=   37.75 ,    96.00              <<FRAME CONSTANTS>>
 *           *     *                                                CG-TO-BOTTOM, H1=   29.800
 *    *CG    *    Z*       Y(CG),Z(CG)=  -91.50 ,    53.80              CG-TO-TOP , H2=   42.200
 *           *     ****                                               CG-TO-REAR, L1=  113.750
LR************        Y      Y(LR),Z(LR)= -205.25 ,    24.00             CG-TO-FRONT, L2=  129.250

<WHEEL NUMBER>                       #1        #2        #3        #4
<DRIVE WHEEL?>                        Y         Y         Y         Y
<WHEEL RADIUS (IN)>                 19.25     19.25     19.25     19.25
<WHEEL ASSEMBLY WEIGHT (LBS)>      345.00    345.00    345.00    345.00
<WHEEL CENTER HOR. COORD. (IN)>    -19.25    -64.55   -115.15   -156.05
<WHEEL CENTER VERT. COORD. (IN)>    17.50     17.43     17.39     17.33
<WHEEL SPRING DEFLECTION (IN)>       1.75      1.82      1.86      1.92
<WHEEL SPRING FORCE (LBS)>        4577.90   4868.60   4991.50   5268.00
<WHEEL SPRING CONST.(LB/IN/RAD)>  4573.47   4573.47   4573.47   4573.47

  <<SUSPENSIONS>>
4  INDEPENDENT


 <INDEPENDENT>
   SUSPENSION  WHEEL      HOR DIST FROM   EQUILIBRIUM VERT DIST      TABLE NUMBERS       WHEEL TRAVEL (IN)
     NUMBER    NUMBER    CG-TO-W.C.(IN)    FROM CG-TO-W.C. (IN)    SPG/ DMP/DM+V/DM-V     REBOUND / JOUNCE
       #1        #1           72.25              -36.30            #1 / #1 / #0 / #0        -0.07 / 13.37
       #2        #2           26.95              -36.37            #1 / #1 / #0 / #0        -0.07 / 13.37
       #3        #3          -23.65              -36.41            #2 / #1 / #0 / #0        -0.00 / 13.30
       #4        #4          -64.55              -36.47            #2 / #1 / #0 / #0        -0.00 / 13.30
```

Figure 18. Second part of print output file

```
 <<TIME=  0.000 SEC>>                                          <<ELAPSED CPU TIME=   0.02 SEC,  CPU TIME SINCE LAST PRINT=   0.00 SEC>>
  UNIT    CG COORDINATES (IN)    CG VERT    CG VERT    CG VERT    PITCH DISP   PITCH VEL   PITCH ACC
   NO.        Y         Z       DISP (IN)  VEL (IPS)  ACC (G'S)     (RAD)     (RAD/SEC)  (RAD/SEC^2)       (RAD)       (RAD/SEC)  (RAD/SEC^2)
   #1    -91.50     53.80         0.00     176.00       0.00        0.000       0.000       0.000          0.000        0.000        0.000

 INDEP. WHEEL  WHEEL CENTER      CONN-TO-FRAME     WHEEL CENTER VERT MOTION     SPRING/DAMPER COMPRESSIVE MOTION
 SUSPEN  NO.     COORD. (IN)      COORD. (IN)      DISP      VEL       ACC            DISP           VELOCITY
   NO.             Y       Z        Y        Z      (IN)     (IPS)     (G'S)           (IN)            (IPS)
  #1     #1   -19.25  17.50    -19.25   24.00    0.00      0.00     13.27          -1.57            0.00
  #2     #2   -64.55  17.43    -64.55   24.00    0.00      0.00     14.11          -1.57            0.00
  #3     #3  -115.15  17.39   -115.15   24.00    0.00      0.00     14.47          -0.50            0.00
  #4     #4  -156.05  17.33   -156.05   24.00    0.00      0.00     15.27          -0.50            0.00

    WHEEL FORCES/MOMENT              #1         #2         #3         #4

  FPH (HORIZONTAL FORCE, LB)         0.0       -0.0       -0.0        0.0
  FPV (VERTICAL FORCE, LB)        4232.9     4523.6     4646.5     4923.0
  MW  (APPLIED MOMENT, LB-IN)       -0.0       -0.0       -0.0       -0.0

REBOUND STRIKE-THROUGH OCCURRED FOR INDEP. SUSPENSION # 1 (UNIT #1) FROM TIME=   0.001 UNTIL   0.006 SEC.

REBOUND STRIKE-THROUGH OCCURRED FOR INDEP. SUSPENSION # 2 (UNIT #1) FROM TIME=   0.001 UNTIL   0.006 SEC.



 <<TIME= 34.075 SEC>>                                          <<ELAPSED CPU TIME=  39.16 SEC,  CPU TIME SINCE LAST PRINT=   0.09 SEC>>
  UNIT    CG COORDINATES (IN)    CG VERT    CG VERT    CG VERT    PITCH DISP   PITCH VEL   PITCH ACC
   NO.        Y         Z       DISP (IN)  VEL (IPS)  ACC (G'S)     (RAD)     (RAD/SEC)  (RAD/SEC^2)       (RAD)       (RAD/SEC)  (RAD/SEC^2)
   #1   5905.70     50.11      5997.29     176.00       0.00       -3.684      10.756      -0.069          0.100       -0.010        0.126

 INDEP. WHEEL  WHEEL CENTER      CONN-TO-FRAME     WHEEL CENTER VERT MOTION     SPRING/DAMPER COMPRESSIVE MOTION
 SUSPEN  NO.     COORD. (IN)      COORD. (IN)      DISP      VEL       ACC            DISP           VELOCITY
   NO.             Y       Z        Y        Z      (IN)     (IPS)     (G'S)           (IN)            (IPS)
  #1     #1  5980.77  25.80   5980.57   27.71    8.31      8.96      0.08           3.02            -1.10
  #2     #2  5935.55  22.72   5935.50   23.16    5.30      9.47     -0.29           4.56            -0.96
  #3     #3  5884.98  19.89   5885.16   18.09    2.49      5.22     -0.06           7.91            -5.76
  #4     #4  5844.00  18.55   5844.46   13.99    1.22      4.95     -0.19          10.75            -6.42

    WHEEL FORCES/MOMENT              #1         #2         #3         #4

  FPH (HORIZONTAL FORCE, LB)       114.2       18.7      -36.4      -96.5
  FPV (VERTICAL FORCE, LB)        3633.6     4411.0     3628.7     5065.1
  MW  (APPLIED MOMENT, LB-IN)     2425.4     2869.9     2419.8     3204.0

THERE WERE    12 REBOUND AND     4 JOUNCE SPRING STRIKE-THROUGHS DURING THIS EVENT.

TERRAIN-NEGOTIATION PHASE COMPLETED...CPU ELAPSED TIME (SEC)=     39.1406  (THIS PHASE ONLY)

TOTAL ELAPSED CPU TIME (SEC)=     39.1562
```

Figure 19. First and last time increment in print output file.

```
LAV      LIGHT ARMOR VEHICLE WHEELED 8X8
LET7AVG LETOURNEAU COURSE 7 AVG 31MAY83
  0.10000E+02  0.10000E-02  0.34091E+02  0.50000E+03  0.10000E-01
    1     0
  2229
    -342.000        0.000    -339.000        0.000    -336.000        0.000
    -333.000        0.000    -330.000        0.000    -327.000        0.000
    -324.000        0.000    -321.000        0.000    -318.000        0.000
    -315.000        0.000    -312.000        0.000    -309.000        0.000
    -306.000        0.000    -303.000        0.000    -300.000        0.000
    -297.000        0.000    -294.000        0.000    -291.000        0.000
    -288.000        0.000    -285.000        0.000    -282.000        0.000
    -279.000        0.000    -276.000        0.000    -273.000        0.000
    -270.000        0.000    -267.000        0.000    -264.000        0.000
    -261.000        0.000    -258.000        0.000    -255.000        0.000
    -252.000        0.000    -249.000        0.000    -246.000        0.000
    -243.000        0.000    -240.000        0.000    -237.000        0.000
    -234.000        0.000    -231.000        0.000    -228.000        0.000
    -225.000        0.000    -222.000        0.000    -219.000        0.000
    -216.000        0.000    -213.000        0.000    -210.000        0.000
    -207.000        0.000    -204.000        0.000    -201.000        0.000
    -198.000        0.000    -195.000        0.000    -192.000        0.000
    -189.000        0.000    -186.000        0.000    -183.000        0.000
    -180.000        0.000    -177.000        0.000    -174.000        0.000
    -171.000        0.000    -168.000        0.000    -165.000        0.000
    -162.000        0.000    -159.000        0.000    -156.000        0.000
    -153.000        0.000    -150.000        0.000    -147.000        0.000
    -144.000        0.000    -141.000        0.000    -138.000        0.000
    -135.000        0.000    -132.000        0.000    -129.000        0.000
    -126.000        0.000    -123.000        0.000    -120.000        0.000
    -117.000        0.000    -114.000        0.000    -111.000        0.000
    -108.000        0.000    -105.000        0.000    -102.000        0.000
     -99.000        0.000     -96.000        0.000     -93.000        0.000
     -90.000        0.000     -87.000        0.000     -84.000        0.000
     -81.000        0.000     -78.000        0.000     -75.000        0.000
     -72.000        0.000     -69.000        0.000     -66.000        0.000
     -63.000        0.000     -60.000        0.000     -57.000        0.000
     -54.000        0.000     -51.000        0.000     -48.000        0.000
     -45.000        0.000     -42.000        0.000     -39.000        0.000
     -36.000        0.000     -33.000        0.000     -30.000        0.000
     -27.000        0.000     -24.000        0.000     -21.000        0.000
     -18.000        0.000     -15.000        0.000     -12.000        0.000
      -9.000        0.000      -6.000        0.000      -3.000        0.000
       0.000        0.000       3.000       -0.015       6.000       -0.030
       9.000       -0.045      12.000       -0.060      15.000       -0.090
      18.000       -0.120      21.000       -0.150      24.000       -0.180
      27.000       -0.360      30.000       -0.540      33.000       -0.720
      36.000       -0.900      39.000       -1.140      42.000       -1.380
      45.000       -1.620      48.000       -1.860      51.000       -2.175
      54.000       -2.490      57.000       -2.805      60.000       -3.120
      63.000       -3.660      66.000       -4.200      69.000       -4.740
      72.000       -5.280      75.000       -5.475      78.000       -5.670
      81.000       -5.865      84.000       -6.060      87.000       -6.255
      90.000       -6.450      93.000       -6.645      96.000       -6.840
      99.000       -7.005     102.000       -7.170     105.000       -7.335
     108.000       -7.500     111.000       -7.485     114.000       -7.470
     117.000       -7.455     120.000       -7.440     123.000       -7.440
```

Figure 20. Top of the plot postprocessing output file.

As previously noted, all of the input files must be created prior to running VEHDYN 4.0. Using the guidelines in Chapter 4, the control file, named control.dat, is shown in Figure 23.

The profile data file, named let7avg.dat, is the same as listed for Sample Problem 1. The next input file is the vehicle data file named m1.vd4. It is shown in Figures 24 and 25.

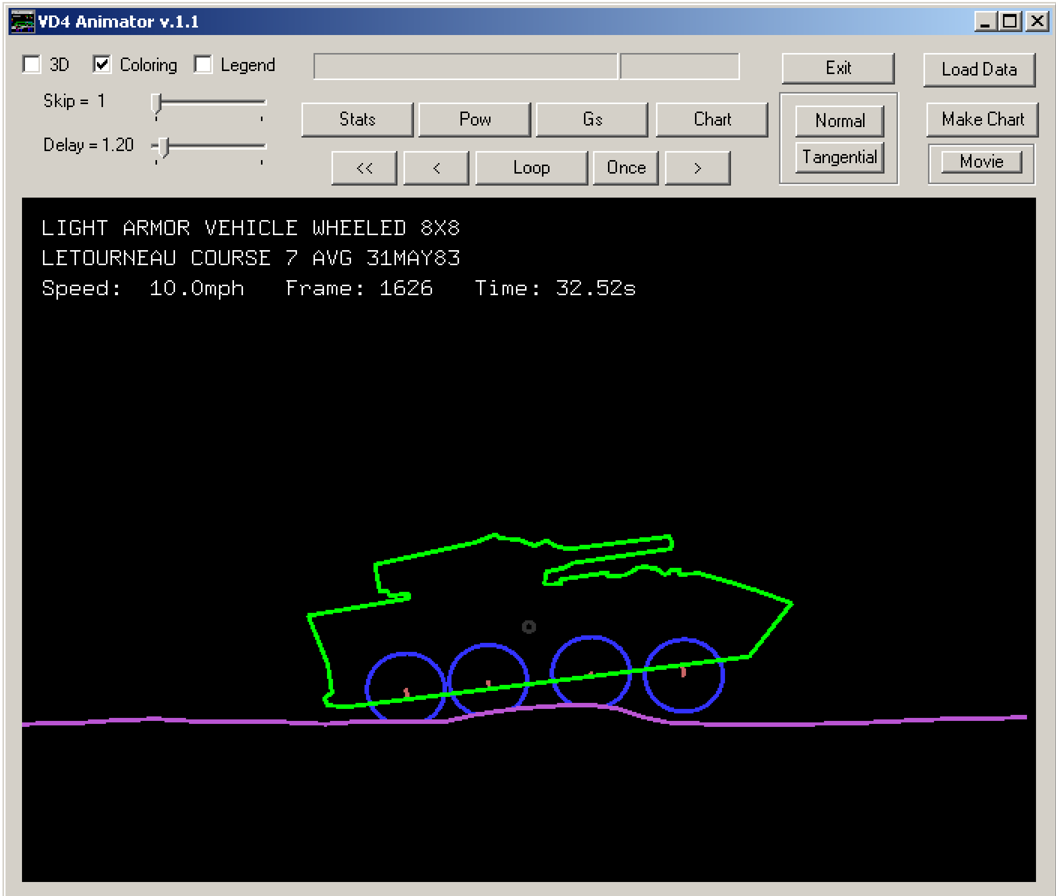


Figure 21. Snapshot of LAV animation.

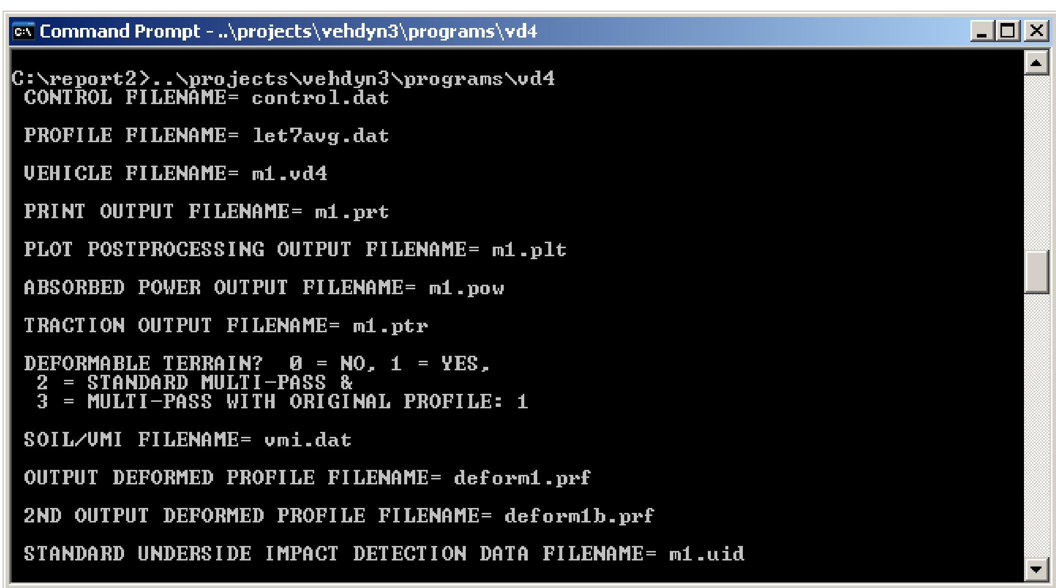


Figure 22. Screen capture for problem 2.

```
M1
LET7AVG
10.0,0.001,0,0.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,0,0
```

Figure 23. Control data file.

```
M1
M1 ABRAMS MAINBATTLE TANK
42
0.0,103500,1.2,103500,2.5,96500,3.7,84500,5.0,67500,6.2,51500,7.5,43718
8.7,38772,9.9,35849,11.2,32927,12.4,31353,13.7,29105,14.5,25772
15.0,24781,16.0,22772,17.0,20734,18.0,20149,19.0,19396,20.0,18636
21.0,17869,22.0,17094,23.0,16313,24.0,15528,25.0,14738,26.0,13941
27.0,13139,28.0,13131,29.0,12774,30.0,12414,31.0,12050,32.0,11685
33.0,11316,34.0,10945,35.0,10573,36.0,10198,37.0,9821,38.0,9441
39.0,9059,40.0,8675,41.0,8291,41.5,6825,41.6,6300
1 3 1500 8 13.13 3825
1.2 78 0.81 1.2 48.0
     1     4     4     0     0     0
    10     0  0.000E+00  -5.000E-01  1.971E+01
 -5.000E-01  0.000E+00  8.200E-01  2.690E+00  4.820E+00  7.190E+00  1.380E+01
  1.658E+01  1.832E+01  1.971E+01
 -2.500E+04  0.000E+00  2.621E+03  6.729E+03  9.972E+03  1.279E+04  1.955E+04
  2.259E+04  2.469E+04  5.000E+04
    16     0  0.000E+00  -1.000E+00  2.147E+01
 -1.000E+00  0.000E+00  1.290E+00  2.670E+00  4.140E+00  5.690E+00  7.300E+00
  8.970E+00  1.067E+01  1.240E+01  1.414E+01  1.588E+01  1.761E+01  1.931E+01
  2.098E+01  2.147E+01
 -2.500E+04  0.000E+00  2.632E+03  4.923E+03  6.985E+03  8.900E+03  1.073E+04
  1.253E+04  1.433E+04  1.619E+04  1.815E+04  2.024E+04  2.252E+04  2.505E+04
  2.790E+04  4.000E+04
    16     0  0.000E+00  -1.000E+00  2.193E+01
 -1.000E+00  0.000E+00  1.490E+00  3.050E+00  4.670E+00  6.340E+00  8.050E+00
  9.790E+00  1.118E+01  1.292E+01  1.465E+01  1.636E+01  1.802E+01  1.963E+01
  2.118E+01  2.193E+01
 -2.500E+04  0.000E+00  2.305E+03  4.415E+03  6.398E+03  8.313E+03  1.021E+04
  1.213E+04  1.371E+04  1.579E+04  1.802E+04  2.046E+04  2.318E+04  2.625E+04
  2.980E+04  4.000E+04
    18     0  0.000E+00  -1.000E+00  2.309E+01
 -1.000E+00  0.000E+00  1.130E+00  2.380E+00  3.720E+00  5.160E+00  6.680E+00
  8.270E+00  9.910E+00  1.160E+01  1.332E+01  1.506E+01  1.576E+01  1.750E+01
  1.923E+01  2.093E+01  2.260E+01  2.309E+01
 -2.500E+04  0.000E+00  2.956E+03  5.426E+03  7.574E+03  9.511E+03  1.132E+04
  1.305E+04  1.476E+04  1.649E+04  1.828E+04  2.018E+04  2.097E+04  2.308E+04
  2.539E+04  2.797E+04  3.090E+04  5.000E+04
```

Figure 24. Top of M1 vehicle data file.

```
    7     0  0.000E+00
 -1.618E+03 -8.090E+01 -1.618E+01  0.000E+00  1.618E+01  1.618E+02  1.618E+03
 -2.500E+04 -3.399E+03 -2.472E+03  0.000E+00  7.725E+03  1.236E+04  5.562E+04
    7     0  0.000E+00
 -1.813E+03 -9.063E+01 -1.813E+01  0.000E+00  1.813E+01  1.813E+02  1.813E+03
 -2.207E+04 -3.034E+03 -2.207E+03  0.000E+00  6.896E+03  1.103E+04  4.965E+04
    5     0  0.000E+00
 -1.000E+03 -1.000E+00  0.000E+00  1.000E+00  1.000E+03
 -2.000E+02 -2.000E+02  0.000E+00  2.000E+02  2.000E+02
    7     0  0.000E+00
 -1.827E+03 -9.135E+01 -1.827E+01  0.000E+00  1.827E+01  1.827E+02  1.827E+03
 -2.189E+04 -3.015E+03 -2.189E+03  0.000E+00  6.842E+03  1.095E+04  4.926E+04
    1     0     0     7     0     0     1
79.18,-24.0
0.,0.,0,0
127451.,1960000.,62.80,21.71,89.03,146.41,165.3,-.013
17.,372.,86.01,16.49,1.,25000.,11110.5,1,0,0,0,0,1,300.,12.,17
17.,339.,50.09,16.43,1.,25000.,10561.8,1,0,0,0,0,1,270.,12.,13
17.,339.,21.29,16.50,1.,25000.,10216.6,1,0,0,0,0,1,270.,12.,13
17.,339.,-7.41,16.50,1.,25000.,9319.1,1,0,0,0,0,1,270.,12.,13
17.,339.,-36.01,16.53,1.,25000.,8933.4,1,0,0,0,0,1,270.,12.,13
17.,339.,-64.71,16.51,1.,25000.,8488.7,1,0,0,0,0,1,270.,12.,13
17.,339.,-94.21,16.60,1.,25000.,8221.0,1,0,0,0,0,1,240.,12.,17
2,1,7,1
25.0,0,1.86,183.1,187.5,1,17.0
15000.,500.
17.,126.09,30.7,1.,25000.,0,0.,12.,25
17.,-123.91,44.7,1.,25000.,0,160.,12.,25
    1     1     1     0     0
    2     2     2     0     0
    3     2     3     0     0
    4     3     3     0     0
    5     3     3     0     0
    6     3     3     0     0
    7     4     4     0     0
```

Figure 25. Bottom of M1 vehicle data file.

The next input file that must be entered for this run is the UID data file (named m1.uid) containing the underside geometry for the M1 tank; it is shown in Figure 26.

Because this problem contains a deformable terrain course, the vehicle-media interaction data file named vmi.dat is also required and is shown as Figure 27.

As shown in the DOS window run of VEHDYN 4.0, the names of several output files are also input during the question-answer sequence of running

```
.1,750
40.,246.73
3.0
5
315.,44.
303.,31.
18.,34.
13.,40.
0.,40.
43.5
```

Figure 26. UID data file for M1.

```
1
-5000 50000  40  'SM'
0
```

Figure 27. Vehicle-media interaction file.

this problem. These files are the formatted output file (m1.prt), the plot postprocessing output file (m1.plt), the absorbed power output file (m1.pow), the traction output file (m1.ptr), and two deformed profile data files (deform1.prf and deform1b.prf) containing data describing the profile deformation as the vehicle moves down the course both under the outside portion of the vehicle (i.e., where the wheels/track goes) as well as under the interior belly of the vehicle. Each of these files will be created by VEHDYN 4.0 as the problem executes. As mentioned for Problem 1, the user must be careful to use a unique name, as any existing file in the directory where the run is proceeding will be overwritten during program execution.

After this problem completes its run, the output files can be examined and plot postprocessing can commence. As with Problem 1, the traction file m1.ptr again contains no data. For this particular run, the file would have contained data only if an execution error generated a diagnostic message.

The other five output files are generally lengthy and, in the interest of brevity, only a partial listing of the deformable profiles will be given below. The m1.pow (absorbed power output file), m1.prt (formatted print output file) and m1.plt (plot postprocessing output file) will follow the same basic format as those for the LAV in Example 1.

The start of the first deformed profile data file (the portion where the track runs) is shown in Figure 28.

```
    0.000000  (Current Time in Seconds)
LET7AVG

 1 246
     -450.000         0.000          0.0         0.0
     -447.000         0.000          0.0         0.0
     -444.000         0.000          0.0         0.0
     -441.000         0.000          0.0         0.0
     -438.000         0.000          0.0         0.0
     -435.000         0.000          0.0         0.0
     -432.000         0.000          0.0         0.0
     -429.000         0.000          0.0         0.0
     -426.000         0.000          0.0         0.0
     -423.000         0.000          0.0         0.0
     -420.000         0.000          0.0         0.0
     -417.000         0.000          0.0         0.0
     -414.000         0.000          0.0         0.0
     -411.000         0.000          0.0         0.0
     -408.000         0.000          0.0         0.0
     -405.000         0.000          0.0         0.0
     -402.000         0.000          0.0         0.0
     -399.000         0.000          0.0         0.0
     -396.000         0.000          0.0         0.0
     -393.000         0.000          0.0         0.0
```

Figure 28. Top of first deformed profile data file.

Shown next, in Figure 29, is the second deformed profile data file (the portion where the belly runs).

```
    0.000000  (Current Time in Seconds)
LET7AVG

 1 246
     -450.000         0.000          0.0         0.0
     -447.000         0.000          0.0         0.0
     -444.000         0.000          0.0         0.0
     -441.000         0.000          0.0         0.0
     -438.000         0.000          0.0         0.0
     -435.000         0.000          0.0         0.0
     -432.000         0.000          0.0         0.0
     -429.000         0.000          0.0         0.0
     -426.000         0.000          0.0         0.0
     -423.000         0.000          0.0         0.0
     -420.000         0.000          0.0         0.0
     -417.000         0.000          0.0         0.0
     -414.000         0.000          0.0         0.0
     -411.000         0.000          0.0         0.0
     -408.000         0.000          0.0         0.0
     -405.000         0.000          0.0         0.0
     -402.000         0.000          0.0         0.0
     -399.000         0.000          0.0         0.0
     -396.000         0.000          0.0         0.0
     -393.000         0.000          0.0         0.0
```

Figure 29. Top of second deformed profile data file.

Although the two deformed profile output files look the same initially, they differ considerably as the run proceeds and soil deformation occurs. The snapshot shown in Figure 30 from a point just before the end of the run provides a visual aid for this simulation; note the purple line

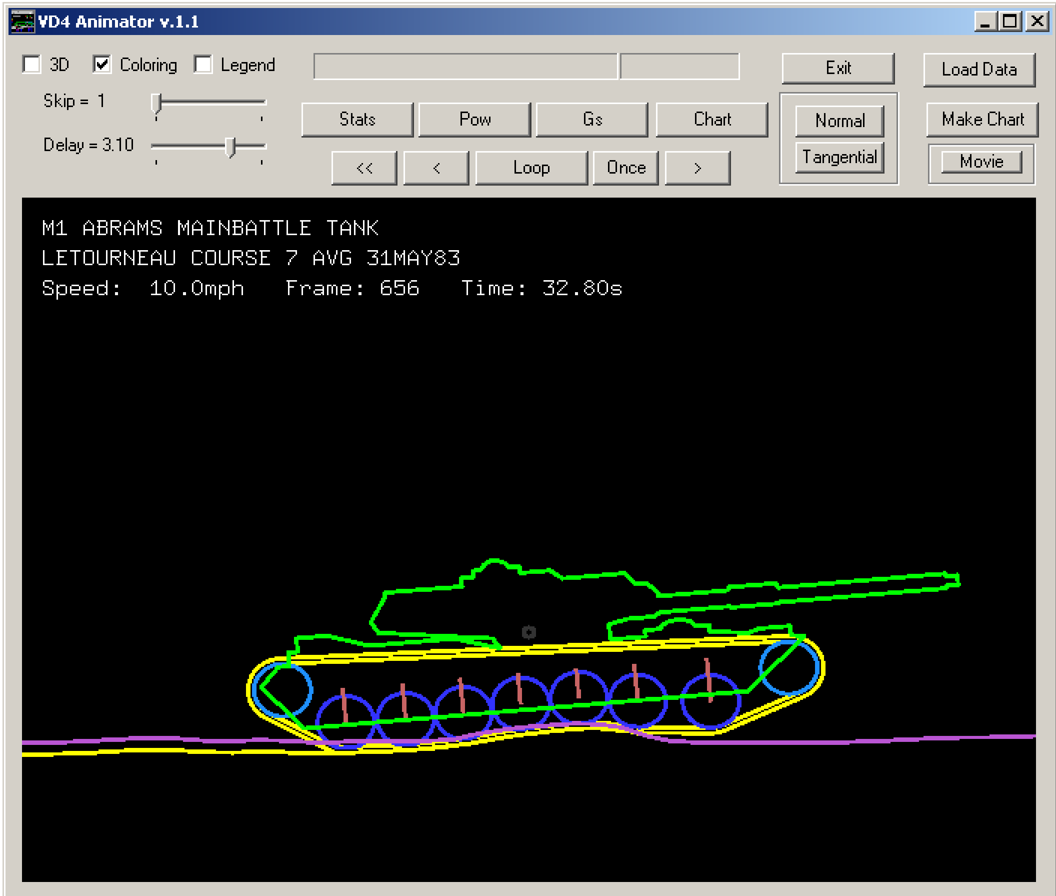


Figure 30. Snapshot of M1 animation.

(undeformed profile) and the yellow line (deformed profile), giving the user a visual examination of the amount of soil deformation taking place in this problem.

Problem 3: M113 armored personnel carrier ACC/DEC over 500-ft nondeformable LET7AVG course - speed varying from 0.1 to 10 mph

The third problem to be demonstrated involves the movement of an M113 armored personnel carrier (APC), a tracked vehicle, beginning at 0.1 mph and accelerating up to 10 mph moving down the nondeformable 500-ft LET7AVG ride course. Using Table 2 as a guide, the required input files for this type of run (called “single ACC/DEC ride/shock nondeformable” in the table) are the profile data file, vehicle data file, control data file, and standard UID data file. The VEHDYN 4.0 program is run from a command line in a DOS window. A screen capture of what this would look like for this problem is shown as Figure 31.

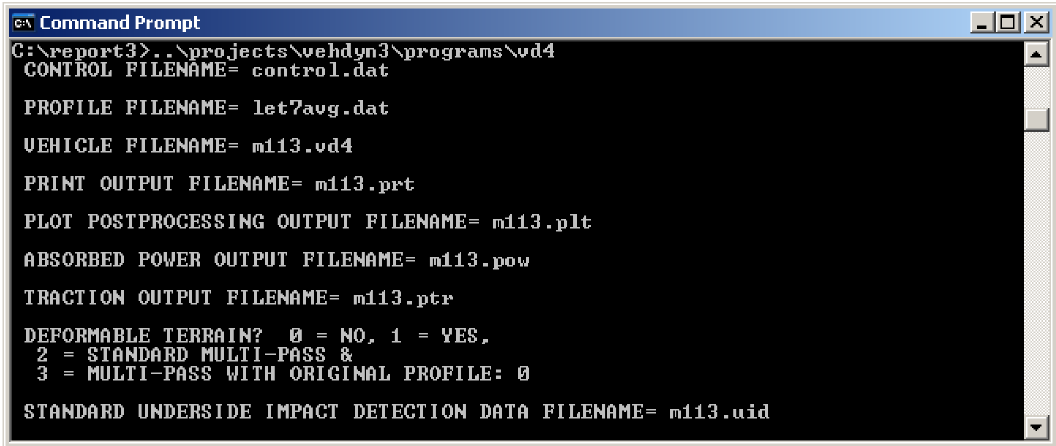


Figure 31. Screen capture for problem 3.

The input files must all be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is as shown in Figure 32.

```
M113
LET7AVG
0.1,0.001,0,60.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,1,10.0
```

Figure 32. Control data file.

Note the last line of the control file in which the ACCFLG=1 (for a nonconstant-speed run) and the OBJMPH=10.0 (setting our objective speed to 10 mph). In line 3, the initial speed is set to 0.1 mph; thus, the vehicle begins at 0.1 mph, accelerates to 10 mph by computing the possible accelerating force from the tractive force-speed data in the vehicle file, then attempts to maintain the objective speed for the remainder of the run.

The profile data file is the same let7avg.dat that was used previously in problems 1 and 2. The vehicle data file for the M113 APC is shown in Figure 33.

After these three key input files, a fourth and final input file is required that describes the underside geometry of the vehicle; this UID data file is shown in Figure 34 for the M113 APC.

```
M113
M113 / Derived from M113A2 file
80
0.0,21450.0,0.5,20048.0,1.0,18517.4,1.5,16861.4,2.0,15129.3,2.5,13593.3
3.0,12222.0,3.5,11056.4,4.0,10062.8,4.5,9133.33,5.0,8325.0,5.5,7600.0C
6.5,7185.71,7.0,7010.53,7.5,6846.05,3.0,6636.36,8.5,6405.59,9.0,6158.E8
9.5,5914.29,10.0,5662.50,10.5,5350.0,11.0,4534.21,11.5,3582.86
13.5,3553.97,14.0,3511.67,14.5,3470.J,15.0,3428.33,15.5,3363.67
16.0,3297.0,16.5,3230.33,17.0,3164.07,17.5,3097.85,18.0,3031.62
18.5,2965.0,19.0,2898.33,19.5,2831.67,20.0,2750.5,20.5,2667.72
21.0,2584.93,21.5,2311.0,23.5,1817.81,24.0,1800.68,24.5,1783.56
25.0,1766.44,25.5,1749.32,26.0,1732.19,26.5,1717.53,27.0,1704.64
27.5,1691.75,28.0,1678.87,28.5,1665.98,29.0,1653.09,29.5,1633.68
30.0,1612.2,30.5,1590.72,31.0,1569.24,31.5,1547.77,32.0,1526.29
32.5,1508.85,33.0,1491.67,33.5,1474.48,34.0,1457.3,34.5,1440.12
35.0,1422.42,35.5,1400.94,36.0,1379.47,36.5,1357.99,37.0,1336.51
37.5,1315.03,38.0,1292.29,38.5,1266.61,39.0,1240.92,39.5,1215.24
40.0,1189.55,40.5,1163.87,41.0,1110.62,41.5,1025.0,42.0,939.383
42.5,746.052,42.8,575.0
1 2 318 6 9.55 435
1.1  70.0  0.8  0.7 9999.9
1,1,2,0,0,0
15,14,2500.,0.,8.5
-.5,0.,.04,.24,.55,1.,1.64,2.39,3.17,4.02,4.52,5.5,7.,8.5,9.
-45000.,0.,100.,300.,600.,1000.,1500.,2000.,2500.,3000.,3300.
3750.,4360.,4810.,50000.
-.5,0.,.13,.34,.71,1.23,2.,2.81,3.68,4.58,5.5,7.,8.5,9.
-50000.,0.,100.,300.,600.,1000.,1500.,2000.,2500.,2900.
3300.,3900.,4350.,45000.
```

```
4,0,0.
-10.,-7.42,7.42,10.
-2350.,-2350.,2350.,2350.
4,0,0.
-20.,-1.,1.,20.
-700.,-700.,700.,700.
1,0,0,5,0,0,1
44.25,0.70
170.,0.,0,0
23400.,119468.,35.8,5.,20.,87.,92.75,0.
14.,250.,53.13,13.18,1.,3500.,2268.,1,0,12.3,1.25,3,1,0,0,-1
14.,250.,26.94,13.17,1.,3500.,2306.,1,0,12.3,1.25,3,1,0,0,-1
14.,250.,0.75,13.16,1.,3500.,2339.,1,0,12.3,1.25,3,1,0,0,-1
14.,250.,-25.25,13.15,1.,3500.,2378.,1,0,12.3,1.25,3,1,0,0,-1
14.,250.,-51.25,13.14,1.,3500.,2409.,1,0,12.3,1.25,3,1,0,0,-1
2,1,5,2
15.0,0,1.6,109.0,90.0,1,16.0
2000.,100.
9.8,82.25,22.12,1.,25000.,0,0,0,-1
9.8,-79.75,22.08,1.,25000.,0,0,0,-1
1,1,1,0,0
2,1,2,0,0
3,1,2,0,0
4,1,2,0,0
5,1,1,0,0
```

Figure 33. M113 vehicle data file.

```
.1,750
20.,130.88
3.0
4
171.,30.
80.,40.
0.,20.
-17.5,22.
35.0
```

Figure 34. M113 UID data file.

Finally, the VEHDYN 4.0 interaction asks for the user to input an integer describing whether there is to be deformable terrain and/or multiple pass calculations and the names for four output files: the formatted print output file, the plot postprocessing output file, the absorbed power output file, and the traction output data file. These files will be created by the program as needed. Remember that care must be taken not to use the same name as an existing file in the working directory, as the program will overwrite an existing file.

After program completion, the output files can be examined and plot postprocessing can be accomplished if the user desires. Again, the traction output file contains no data, as expected with a problem running normally with no diagnostic errors. The other three output files (m113.pow, m113.prt, and m113.plt) are generally quite lengthy and follow the same general format as shown for Example 1.

The m113.plt file is primarily to be used as an input file to a postprocessing program, such as ERDC's animation program VD4Animator. An example snapshot of the kind of output that can be displayed with VD4Animator from this sample problem is shown in Figure 35.

Problem 4: M977 HEMTT ACC/DEC over 500-ft deformable LET7AVG course – speed varying from 0.1 to 10 mph

This fourth problem to be demonstrated involves the movement of an M977 HEMTT 8×8 wheeled vehicle, beginning at 0.1 mph and accelerating up to 10 mph moving down the deformable 500-ft LET7AVG ride course. Using Table 2 as a guide, the required input files for this type of run, called

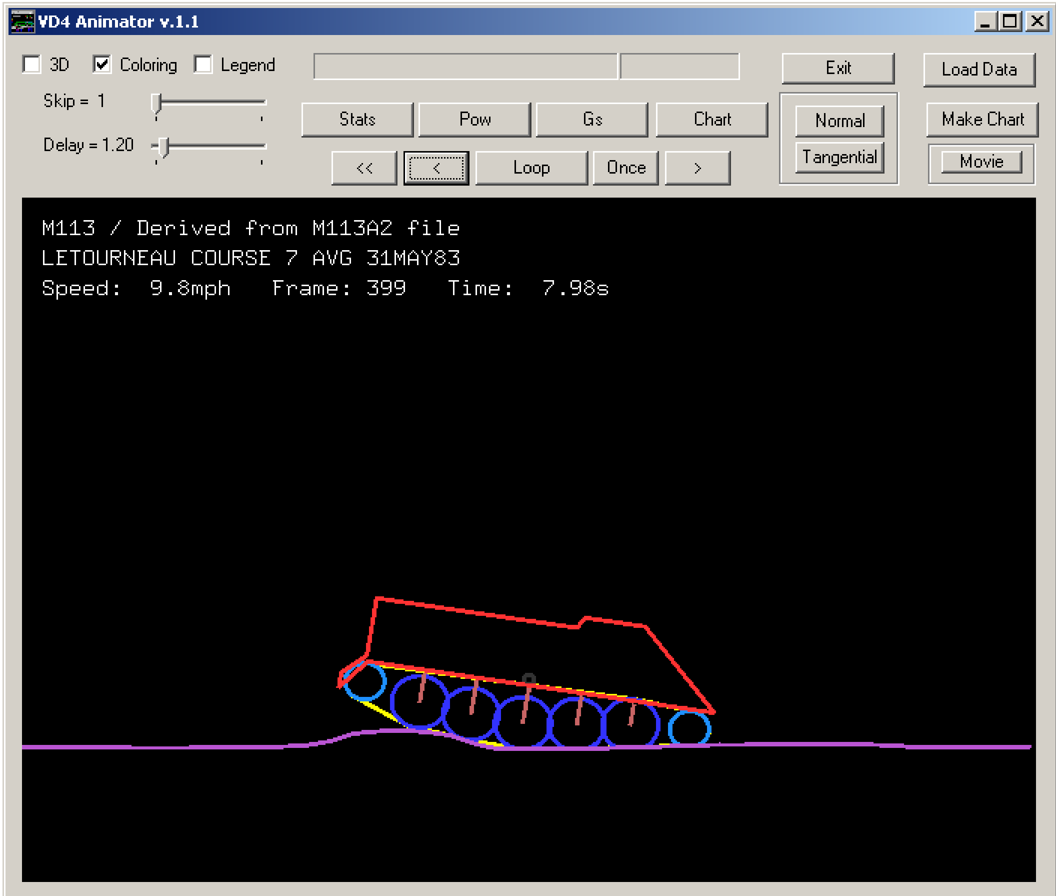


Figure 35. Snapshot of M113 animation.

"single ACC/DEC ride/shock deformable" in the table, are the profile data file, vehicle data file, control data file, and standard (or full) UID data file. The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 36.

The input files must all be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 37.

The profile data file is the same one named let7avg.dat used previously.

Next is the vehicle data file for the M977 HEMTT, shown in Figure 38.

Because we answered that the profile was going to be deformable terrain, the next input file that is required is the vehicle-media interaction file

```
Command Prompt - ..\projects\vehdyn3\programs\vd4

C:\report4>..\projects\vehdyn3\programs\vd4
 CONTROL FILENAME= control.dat

 PROFILE FILENAME= let7avg.dat

 VEHICLE FILENAME= m977.vd4

 PRINT OUTPUT FILENAME= m977.prt

 PLOT POSTPROCESSING OUTPUT FILENAME= m977.plt

 ABSORBED POWER OUTPUT FILENAME= m977.pow

 TRACTION OUTPUT FILENAME= m977.ptr

 DEFORMABLE TERRAIN?  0 = NO, 1 = YES,
  2 = STANDARD MULTI-PASS &
  3 = MULTI-PASS WITH ORIGINAL PROFILE: 1

 SOIL/VMI FILENAME= vmi.dat

 OUTPUT DEFORMED PROFILE FILENAME= deform1.prf

 2ND OUTPUT DEFORMED PROFILE FILENAME= deform1b.prf

 STANDARD UNDERSIDE IMPACT DETECTION DATA FILENAME= m977.uid
```

Figure 36. Screen capture for problem 4.

```
M977
LET7AVG
0.1,0.001,0,60.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,1,10.0
```

Figure 37. Control data file.

describing the deformable terrain's properties; for a single-layer 40-RCI SM surface material, this file is shown in Figure 39.

After these four input files, a fifth input file is required that describes the underside geometry of the vehicle; this UID data file is shown in Figure 40 for the M977 HEMTT.

Finally, the VEHDYN 4.0 interaction requests the user to input the names for six output files; these are the formatted print output file (m977.prt), the plot postprocessing output file (m977.plt), the absorbed power output file (m977.pow), the traction output data file (m977.ptr), and two deformed profiles (wheel rut [deform1.prf] and belly rut [deform1b.prf]). These files will be created by VEHDYN 4.0 as needed.

```
M977
OTC HEMTT M977 ... FULL ... WES SPG/DMP TABLES (3/05)
28
0,52206,1.0,47144,2.0,41262,3.0,35500,4.0,26894,4.9,23654,5.9,18322
6.9,16860,7.9,15386,8.9,13892,9.9,12890,10.9,12395,12.8,9882,14.8,8875
16.8,6753,19.8,5954,22.7,5556,24.7,5100,27.7,4566,29.6,3536,34.6,3328
39.5,3042,44.4,2425,49.4,2351,52.3,2260,54.3,2159,59.3,2108,62.2,2057
1 2 736 8 24.72 1230
1.0  68.0  0.8  0.7 48.0
1,2,2,0,0,1
4,0,0,0,4.35
-1.,0.,4.35,5.35
-25000.,0.,21902.,46902.
4,0,0,0,4.35
-1.,0.,4.35,5.35
-25000.,0.,18222.,43222.
4,0,0
-100.,-45.45,45.45,100.
-1500.,-1500.,1500.,1500.
4,0,0
-100.,-38.46,38.46,100.
-1500.,-1500.,1500.,1500.
4,0,0,1.4486,1.6930
.4486,1.4486,1.6930,2.6930
-3.65E6,0.,0.,3.65E6
0,0,2,0,0,0,1
150.,28.
0,0,0,0
60400.,1.717E6,62.,22.,45.,150.,187.,0.
26.5,1406,143.,24.23,2.11,6325,7045.,1,0,16.0,13.8,2,1,0,0,-1
26.5,1406,83.,24.22,2.11,6325,7129.,1,0,16.0,13.8,2,1,0,0,-1
26.5,1406,-67.,23.96,2.28,6825,8018.,1,0,16.0,13.8,2,1,0,0,-1
26.5,1406,-127.,23.96,2.28,6825,8008.,1,0,16.0,13.8,2,1,0,0,-1
1,0,0,0,1
113.,6168.,3000.
1,1,0,0,157,150
2,1,0,0,72,76
2,0,0,0,1
-97,6168.,3000.
3,2,0,0,-56,-60
4,2,0,0,-139,-134
```

Figure 38. M977 vehicle data file.

```
1
-5000 50000  40  'SM'
0
```

Figure 39. Vehicle-media interaction file.

```
0.1,750.0
35.43,325.5
3.0
15
400.5,70.3,400.5,58.8,382.64,37.2,367.34,28.5
351.05,28.5,324.5,24.4,294.6,13.3,265.4,24.5
234.2,25.4,157.8,23.4,115.3,23.6,84.9,13.0
54.4,23.9,10.3,27.3,0,36.5
48.0
```

Figure 40. M977 UID data file.

After the program completes its run, the output files can be examined and plot postprocessing can be accomplished if the user desires. Again, the traction output file contains no data, as there are no diagnostic reporting issues with this run. The other five output files are generally quite lengthy and follow the same basic formats as shown in Examples 1 and 2.

The m977.plt file is primarily to be used as an input file to a postprocessing program such as ERDC's animation program VD4Animator. An example snapshot of the kind of output that can be displayed with VD4Animator from this sample problem is shown in Figure 41, which depicts the extreme rutting taking place in the fairly soft 40-RCI terrain.

Figure 42 is a snapshot from VD4Animator's 3D capability giving a more realistic display of the rutting as well as the pressure bulbs that the soil is experiencing under each wheel.

Problem 5: VCI1 determination for M2 Bradley fighting vehicle over deformable SMSC soil type surface

The fifth problem to be demonstrated involves the determination of VCI1 for a tracked M2 Bradley Fighting Vehicle (BFV) on a silty, clayey sand (SMSC) surface. The VCI1 is the value of soil RCI where this vehicle can just make a single pass without getting stuck. Any soil softer than the VCI1 would produce excessive sinkage, and the vehicle would get stuck. In VEHDYN 4.0, an initial value for VCI1 is guessed and a determination made whether the vehicle over this terrain is a GO or NOGO. If a NOGO, the guessed value is then increased successively until a GO is reached. Similarly, if the initial guess produces a GO, the guessed value is reduced as necessary until a NOGO is reached. Through this

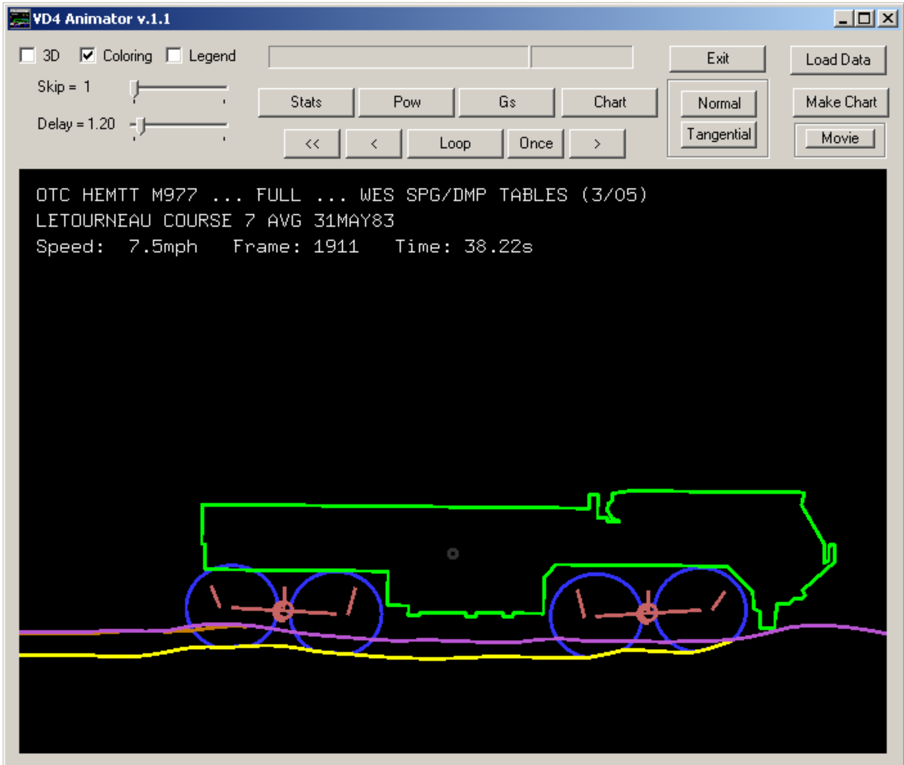


Figure 41. Snapshot of M977 animation.

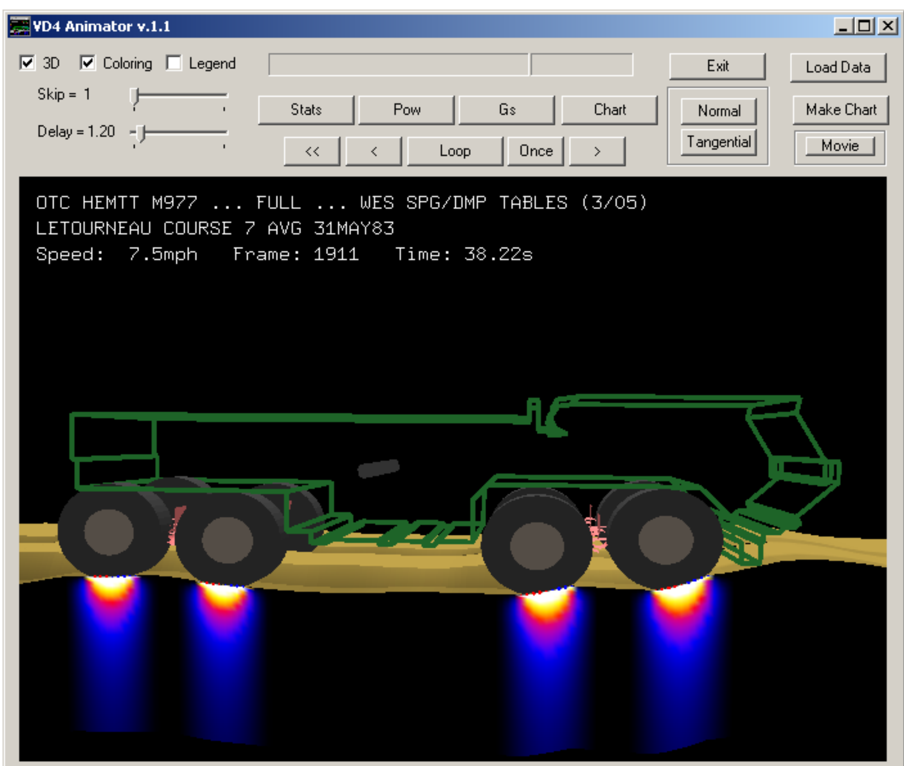


Figure 42. Snapshot of M977 animation.

successive guessing process, the VCI1 is zeroed-in on. Using Table 2 as a guide, the required input files for this type of run, called “VCI1 Determination” in the table, are the vehicle data file (m2.vd4), control data file (control.dat), and standard UID data file (m2.uid). Additionally, a special name for profile data filename (VCI1-Special) is required to initiate the VCI1 determination calculation sequence. The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 43.

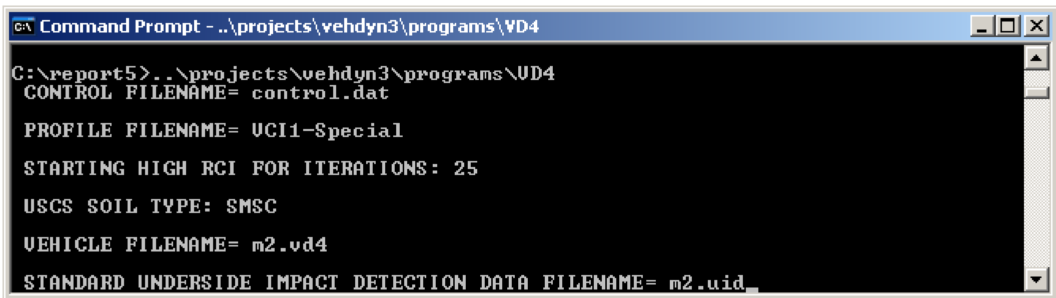


Figure 43. Screen capture for problem 5.

The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 44.

```
M2
LET7AVG
0.1,0.001,0,60.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,1,10.0
```

Figure 44. Control data file.

Note that the second line lists the profile name as LET7AVG; for this type of problem, the name of the profile in this control data file is irrelevant, and anything can be put on this line. The actual course for a VCI1 Determination is a flat course and is automatically generated by VEHDYN 4.0.

The vehicle data file for the M2 BFV is given in Figure 45.

In addition to the control file and the vehicle file, the only other input file required for this type of problem contains the description of the underside geometry of the vehicle; this UID data file is shown in Figure 46 for the M2 BFV.

```
M2
M2 Bradley Fighting Vehicle --- 1985
31
0.00,57222,0.43,57222,0.90,55242,1.41,50424,1.96,44286,2.56,38676
3.19,34188,3.87,30294,4.65,25806,5.53,22308,6.92,19536,7.61,18480
8.30,17424,8.99,16434,10.38,14388,11.07,13530,12.45,11946,13.84,10560
15.22,9438,16.94,8250,19.37,7590,20.75,7194,22.14,6864,23.52,6468
24.91,6138,26.29,5808,27.67,5478,30.44,4818,33.21,4290,35.76,3828
38.00,3696
1 1 900 8 10.5 1225
0.75  80  0.5 0.7 9999.9
1,2,2,0,0,0
12,0,0.,0.,22.14
-1,0,1,2,4,6,12,15,17,19.14,22.14,23.14
-25000,0,1500,2500,4100,5500,9100,11200,12900,15150,30150,40153
11,0,0.,0.,22.14
-1,0,2,4,6,12,15,17,19.14,22.14,23.14
-25000,0,1600,3100,4300,7400,9200,10600,12530,27530,37528
7,0,0.
-100,-68,-9.6,0,11,96,100
-5072,-2630,-2040,0,3940,5110,5346
4,0,0.
-100,-5,5,100
-125,-125,125,125
1,0,0,6,0,0,1
46.,-1
0.,0.,0,0
66000,565000,53.4,35.4,54.6,110,120,0.
14.75,195,81,14.35,1.,15000,6181,1,0,0,0,0,1,0,0,-1
14.75,195,53,14.36,1.,15000,5082,1,0,0,0,0,1,0,0,-1
14.75,195,22,14.44,1.,15000,5016,1,0,0,0,0,1,0,0,-1
14.75,195,-16,14.25,1.,15000,5940,1,0,0,0,0,1,0,0,-1
14.75,195,-45,14.32,1.,15000,5907,1,0,0,0,0,1,0,0,-1
14.75,195,-73,14.32,1.,15000,5874,1,0,0,0,0,1,0,0,-1
2,1,6,2
21,0,1.6,157,136,1,17.5
12628.,205.
13.12,112.5,30.,1.,10000.,0,0,0,-1
12.75,-98.6,27.5,1.,10000.,0,0,0,-1
1,1,1,0,0
2,1,1,0,0
3,1,1,0,0
4,2,2,0,0
5,2,2,0,0
6,2,1,0,0
```

Figure 45. M2 vehicle data file.

```
.1,750.
28.,177.5
3.0
6
230,35,230,27,0,28
-19,25,-19,38,-28,38
37.5
```

Figure 46. M2 UID data file.

Finally, VEHDYN 4.0 generates a single output file named VCI1.out, which contains the final results of the VCI1 sequence of runs. The only datum of interest is the final value of VCI1; the resulting output file is shown in Figure 47.

```
M2 Bradley Fighting Vehicle --- 1985
  VCI1 FOR THIS VEHICLE: 20
```

Figure 47. VCI1 output data file.

Problem 6: Maximum slope climb for M998 HMMWV over nondeformable surface

The sixth problem to be demonstrated involves the determination of the maximum slope that an M998 HMMWV 4×4 wheeled vehicle can climb on a nondeformable surface. VEHDYN 4.0 generates a series of runs to zero-in on the desired slope, which is the maximum percent slope this vehicle can climb and on which it can still accelerate without coming to a stop. The program first runs a 0-percent slope, then a 100-percent slope. Assuming the 100-percent result is a NOGO, an intermediate slope (half the value between the NOGO slope and the last GO result) is run. If the new slope is a GO, another intermediate slope (half the value of this current GO and the last NOGO) is run. This type of sequence is continued until the vehicle can no longer climb the slope and this NOGO slope is within a set tolerance to the last GO slope.

Using Table 2 as a guide, the required input files for this type of run, called “Maximum Slope Climbing Determination on Nondeformable” in the table, are vehicle data file (m998.vd4), control data file (control.dat), and standard UID data file (m998.uid). The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 48.

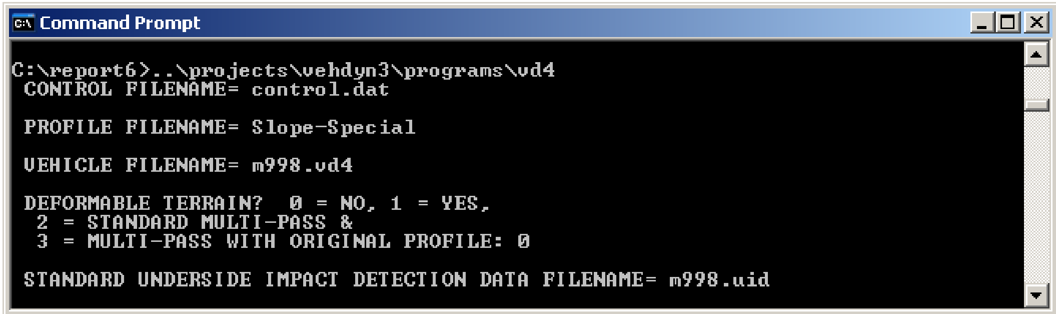


Figure 48. Screen capture for problem 6.

No profile data file is required. In answer to the VEHDYN 4.0 request for a profile filename, the user inputs Slope-Special, and the program will generate the necessary profile data. The other input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 49.

```
M998
LET7AVG
0.1,0.001,0,60.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,1,10.0
```

Figure 49. Control data file.

As with the VCI1 problem, the second line in the control file where the user would put the name of the terrain course is not used by the program; the user may put anything on that line as the course is autogenerated by VEHDYN 4.0. The vehicle data file for the M998 HMMWV is given in Figure 50.

The final input file that is required describes the underside geometry of the vehicle; this UID data file is shown in Figure 51 for the M998 HMMWV.

After the program completes its run, the only output file is a file named Slope.out. This file is shown in Figure 52.

```
M998
M998 HMMWV (2/28/92)
81
0.,5880.,1.,5880.,2.,5880.,3.,5880.,4.,5880.,5.,5638.,6.,5122.03
7.,4744.07,8.,4690.76,9.,4602.20,10.0,4444.97,11.,4269.86,12.,2867.76
13.,2860.82,14.,2841.23,15.,2805.02,16.,2753.97,17.,2689.97,18.,2628.29
19.,2555.78,20.,2490.25,21.,1952.42,22.,1935.77,23.,1914.37,24.,1887.40
25.,1857.46,26.,1829.31,27.,1800.02,28.,1764.91,29.,1731.13,30.,1592.05
31.,1565.95,32.,1092.49,33.,1091.43,34.,1089.64,35.,1087.38,36.,1084.45
37.,1081.,38.,1076.,39.,1070.23,40.,1063.92,41.,1056.28,42.,1048.02
43.,1038.99,44.,1029.48,45.,1019.94,46.,1011.12,47.,1002.23,48.,993.245
49.,982.456,50.,971.171,51.,960.411,52.,951.206,53.,943.081,54.,743.197
55.,741.438,56.,739.354,57.,737.271,58.,734.430,59.,731.477,60.,728.117
61.,724.351,62.,720.551,63.,716.469,64.,712.388,65.,708.084,66.,703.667
67.,699.333,68.,695.412,69.,691.490,70.,687.272,71.,682.872,72.,678.390
73.,673.305,74.,668.220,75.,663.168,76.,658.126,77.,653.581,78.,649.846
79.,646.112,79.25,0.0
1 1 360 8 17.3 251
0.7,40.9,0.7,0.7,29.9
1,2,2,0,0,0
5,0,0,0,0,0,2.630,11.2
-15.0,2.629,2.630,11.2,23.85
-5125,-1000,1000,3355,14100
5,0,0,0,0,0,0.190,9.13
-15.12,0.189,0.190,9.13,21.78
-5200,-380,380,3250,14100
9,0,0,0,0,0
-234,-31,-16,-6,0,6,16,31,234
-1330,-770,-342,-158,0,80,280,652,1586
9,0,0,0,0,0
-234,-31,-16,-6,0,6,16,31,234
-1536,-976,-542,-194,0,108,428,988,1922
0,0,0,2,0,0,1
5.7,-13.9
170,15,0,0
5545,41300,27.18,0.0,39.1,101.3,77.2,0.008
18.15,250,57.7,17.10,2,4000,1512.4,1,0,12.3,10.25,2,1,0,0,-1
18.15,250,-72.3,17.22,2,4000,1260.1,1,0,12.3,10.25,2,1,0,0,-1
1,1,1,0,0
2,2,2,0,0
```

Figure 50. M998 vehicle data file.

```
0.1 750.
23.4 170.8
3.0
6
180.00,23.8,178.00,23.8,160.00,16.0,31.50,16.0,11.75,23.4,0.0,23.4
28.6
```

Figure 51. M998 UID data file.

```
M998 HMMWV (2/28/92)
  MAXIMUM TRACTIVE FORCE: 3568.809999999999
  WEIGHT (LBS): 5545.
  DRAWBAR COEFFICIENT: 0.6436086564472496
  MAXIMUM SLOPE (%) BASED ON DRAWBAR:  75.0

  MAXIMUM SLOPE (%) WITH DYNAMICS: 81.2
```

Figure 52. Slope climbing output data file.

Problem 7: Maximum slope climb for 4×4 LMTV over 300-RCI SM-type deformable soil

The seventh problem to be demonstrated is also a maximum slope climb problem, but the vehicle is a 4×4 light medium tactical vehicle (LMTV) and the terrain is a 300-RCI SM-type surface. Using Table 2 as a guide, the required input files for this type of run, called "Maximum Slope Climbing Determination on Deformable" in the table, are the vehicle data file (lmtv.vd4), control data file (control.dat), standard UID data file (lmtv.uid), and deformable soil data VMI file (vmi.dat). The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 53.

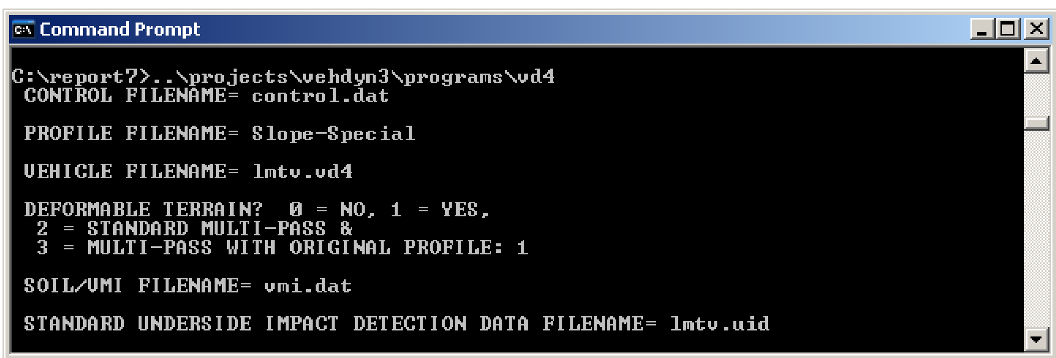


Figure 53. Screen capture for problem 7.

The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 54.

The second line in the control file (LET7AVG) is again irrelevant, as in the previous two problems, as the course is autogenerated by VEHDYN 4.0 as needed.

The vehicle data file, lmtv.vd4, for the 4×4 LMTV is shown in Figure 55.

```
LMTV
LET7AVG
0.1,0.001,0,60.0,0.0,0.5,0.01
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,1,10.0
```

Figure 54. Control data file.

```
LMTV
S&S M1078 Light 4x4 MTV (2.5 Ton) --- Loaded (3/2/94)
 107
0.,31550.0,0.5,29002.0,1.,26454.0,1.5,23906.0,2.,21358.0,2.5,19098.0
3.,17025.3,3.5,15104.3,4.,13300.0,4.5,11700.9,5.,10654.3,5.5,9669.17
6.,8791.0,6.5,8171.39,7.,7551.78,7.5,6932.18,8.,6600.0,8.5,6218.0
9.,5836.0,9.5,5454.0,10.,5072.0,10.5,4877.33,11.,4732.61,11.5,4587.89
12.,4568.0,12.5,4497.0,13.,4426.0,13.5,4355.0,14.,4284.0,14.5,4166.0
15.,4048.0,15.5,3930.0,16.,3812.0,16.5,3683.75,17.,3555.5,17.5,3427.25
18.,3299.0,18.5,3157.46,19.,3015.92,20.5,2972.75,21.,2900.5,21.5,2828.25
22.,2756.0,22.5,2680.75,23.,2605.5,23.5,2530.25,24.,2455.0,24.5,2373.0
25.,2291.0,26.5,2253.19,27.,2219.13,27.5,2185.06,28.,2151.0,28.5,2114.0
29.,2077.0,29.5,2040.0,30.,2003.0,30.5,1965.0,31.,1927.0,31.5,1889.0
32.,1851.0,32.5,1811.5,33.,1772.0,33.5,1732.5,34.,1693.0,34.5,1649.5
35.,1606.0,35.5,1562.5,37.5,1550.88,38.,1529.0,38.5,1508.25,39.,1487.5
39.5,1466.75,40.,1446.0,40.5,1425.0,41.,1404.0,41.5,1383.0,42.,1362.0
42.5,1340.75,43.,1319.5,43.5,1298.25,44.,1277.0,44.5,1255.0,45.,1233.0
45.5,1211.0,46.,1189.0,46.5,1165.0,47.,1141.0,47.5,1117.0,48.,1093.0
48.5,1070.64,49.,1048.28,49.5,1025.92,51.5,1011.0,52.,993.0,52.5,973.75
53.,954.5,53.5,935.25,54.,916.0,54.5,896.25,55.,876.5,55.5,856.75
56.,837.0,56.5,818.683,57.,800.366,57.5,674.339,57.6300,641.0
1,1,403,6,22.56,635
0.75,51.3,0.85,1.2,36.0
1,2,1,0,0,0
15,0,0,-0.47,9.17
-1.67,-1.47,-1.27,-1.07,-.87,-.67,-0.47,0,9.17,9.37
9.57,9.77,9.97,10.17,10.37
-7099,-4377,-3366,-2525,-1903,-1342,-591,0,11524,
12275,12837,13458,14299,15311,18032
14,0,0,0,9.65
-1.2,-1.,-0.8,-0.6,-0.4,-0.2,0,9.65,9.85,10.05,10.25,
10.45,10.65,10.85
-6508,-3786,-2775,-1934,-1312,-751,0,8646,9397,
9958,10580,11421,12432,15154
11,0,0
-20.63,-15.47,-10.31,-5.16,-2.05,0,2.05,5.16,10.31,
15.47,20.63
-2405,-2001,-1574,-989,-270,0,169,348,641,899,1124
0,0,0,2,0,0,1
79.15,-1.53
0.,0.,0,0
22740.,280788.,57.0,26.21,53.08,129.3,125.0,0.
23.,1114.,72.2,21.61,1.94,6570.,6005.,1,.1,15.3,11.1,2,1,0,0,-1
23.,1015.,-81.3,21.53,1.42,5430.,5365.,1,.1,15.3,11.1,2,1,0,0,-1
1,1,1,0,0
2,2,1,0,0
```

Figure 55. LMTV data file.

The next input file, lmtv.uid, contains the description of the underside geometry of the vehicle; this UID data file is shown in Figure 56 for the LMTV.

```
.1,750.
39.6,201.8
3.0
10
254.3,44.9,254.5,36.8,236.7,22.4,225.6,23.4,176.7,22.2
113.6,23.2,70,23,24.4,23,10.4,24,0,39.6
47.3
```

Figure 56. LMTV UID data file.

The last input file, vmi.dat, contains the material properties for the deformable terrain surface; this vmi data file is shown in Figure 57.

```
1
-5000 50000  300  'SM'
0
```

Figure 57. Vehicle-media interaction file.

The single output file, called Slope.out, is shown in Figure 58.

```
S&S M1078 Light 4x4 MTV (2.5 Ton) --- Loaded (3/2/94)
  MAXIMUM TRACTIVE FORCE: 14851.698000000004
  WEIGHT (LBS): 22740.
  DRAWBAR COEFFICIENT: 0.6531089709762534
  MAXIMUM SLOPE (%) BASED ON DRAWBAR:  76.5

  MAXIMUM SLOPE (%) WITH DYNAMICS: 64.8
```

Figure 58. Slope climbing output data file.

Problem 8: Gap crossing for LAV 8×8 wheeled vehicle over deformable gap obstacles

The eighth problem to be demonstrated involves the movement of the 8×8 LAV (see Problem 1) across a series of ditch-like deformable (300-RCI SM soil material) gaps. Using Table 2 as a guide, the required input files for this type of run, called “Gap Crossing” in the table, are the vehicle data file (lav.vd4), control data file (control.dat), a special-format profile data file that is used to generate the gap sequence (obs_geometry.dat), standard UID data file (lav.uid), and VMI soil data file (vmi.dat). The VEHDYN 4.0

program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 59.

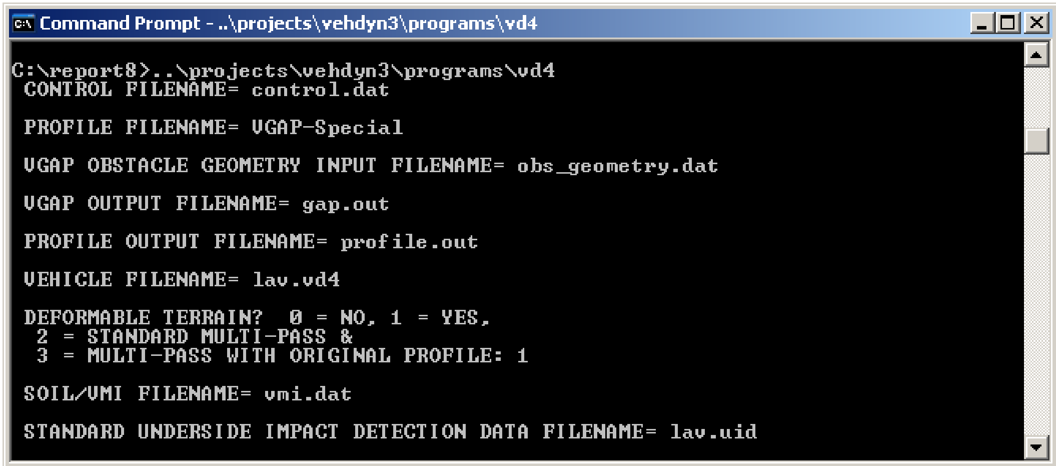


Figure 59. Screen capture for problem 8.

To tell VEHDYN 4.0 that this is to be a gap-crossing problem, a special profile name must be input "VGAP-Special." The program recognizes this case-sensitive text and will then run the sequence of profiles as defined in the profile geometry data file given above as "obs_geometry.dat." The format for this special-format profile data file is given in Chapter 4. The output for this run will be collected primarily in two output files. The VGAP output file, gap.out, collects the GO-NOGO information for each of the generated gaps while the profile output file, profile.out, collects a set of profile data for each of the generated gaps in a format consistent with standard profile data.

The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 60.

```
LAV
Slice 0
0.5,0.001,0.0,30.0,0.0,0.5,0.01
5.,30.,0.
1,1
0.6,0.7,1.0,1.0
0 1 2.0
```

Figure 60. Control data file.

The data on the second line of the control file are, again, irrelevant as the actual profile data are autogenerated by VEHDYN 4.0 according to the data in the special-format obstacle geometry data file obs_geometry.dat. This special-format file containing the obstacle geometry is shown in Figure 61.

```
3
6  12  24
2
72  120
1
4.71
```

Figure 61. Obstacle geometry data file.

The data in this obstacle geometry file tell VEHDYN 4.0 that the desired gaps have three heights (6, 12, and 24 in.), two widths (72 and 120 in.), and one approach angle (4.71 rad), for a total of 3 × 2 × 1 = 6 gaps.

The vehicle data file for the 8×8 LAV, called lav.vd4, and the corresponding UID underside data file, lav.uid, are the same as for Problem 1 earlier in this chapter and can be viewed in that section of the report.

After the program completes its run, the two output files can be examined. The VGAP output file gap.out is shown in Figure 62.

```
NOHGT
3
NANG
1
NWDTH
2
 <GO> 6. 269.8631238 72.39996
 <GO> 12. 269.8631238 72.79991999999999
 <GO> 24. 269.8631238 73.59984
 <GO> 6. 269.8631238 120.39996
 <GO> 12. 269.8631238 120.79991999999999
 NOGO 24. 269.8631238 121.59984
```

Figure 62. VGAP output data file.

The data in this file indicate whether the LAV was able to cross each generated gap. The data after each GO/NOGO indicator are the height (in.), angle (deg), and width (in.) of the generated gap. The profile output file is shown in Figure 63.

```
OBS 1
 Generated Obstacle
 4 0
        0.000000        0.000000
        0.199980       -6.000000
       72.199980       -6.000000
       72.399960        0.000000
OBS 2
 Generated Obstacle
 4 0
        0.000000        0.000000
        0.399960      -12.000000
       72.399960      -12.000000
       72.799920        0.000000
OBS 3
 Generated Obstacle
 4 0
        0.000000        0.000000
        0.799920      -24.000000
       72.799920      -24.000000
       73.599840        0.000000
OBS 4
 Generated Obstacle
 4 0
        0.000000        0.000000
        0.199980       -6.000000
      120.199980       -6.000000
      120.399960        0.000000
OBS 5
 Generated Obstacle
 4 0
        0.000000        0.000000
        0.399960      -12.000000
      120.399960      -12.000000
      120.799920        0.000000
OBS 6
 Generated Obstacle
 4 0
        0.000000        0.000000
        0.799920      -24.000000
      120.799920      -24.000000
      121.599840        0.000000
```

Figure 63. Profile output data file.

Problem 9: Obstacle crossing for M1 tank over a nondeformable series of obstacles

The ninth problem to be demonstrated involves the movement of the M1 Abrams tank from Sample Problem 2 across a series of nondeformable obstacles. Using Table 2 as a guide, the required input files for this type of run, called “ObsMod” in the table, are the vehicle data file (m1.vd4), control data file (control.dat), a standard UID data file (m1.uid), and a special format profile data file (obs_geometry.dat). The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 64.

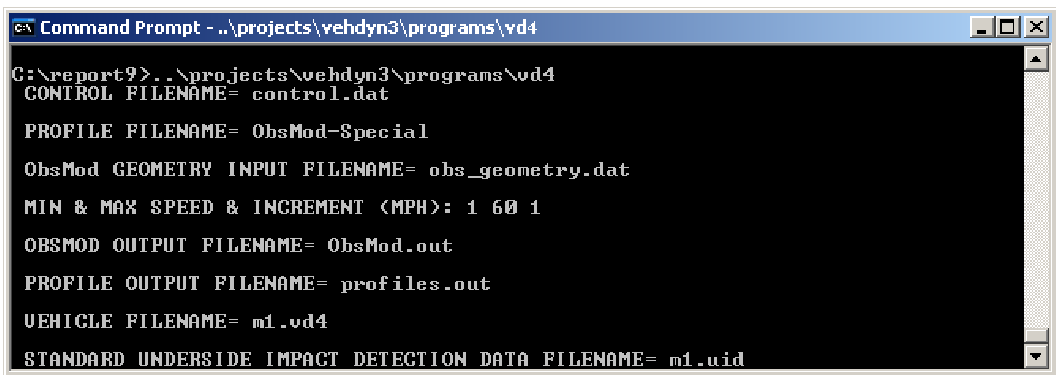


Figure 64. Screen capture for problem 9.

The vehicle file (m1.vd4) and its associated UID underside data file (m1.uid) are the same as listed for Problem 2 earlier in this chapter.

To initiate the obstacle crossing (i.e., ObsMod) run, the special case-sensitive name “ObsMod-Special” must be input for the profile filename. The program understands this to mean that the ObsMod run sequence must be run and VEHDYN 4.0 then asks for the “ObsMod GEOMETRY INPUT FILENAME” to be input (obs_geometry.dat).

The control file (control.dat) has the format shown in Figure 65.

```
M1
Slice 0
0.5,0.001,0.0,30.0,0.0,0.5,0.01
5.,30.,0.
1,1
0.6,0.7,1.0,1.0
0 1 2.0
```

Figure 65. Control data file.

Again, as with problems 5 through 8, the data on the second line have no real meaning here, as the actual profile(s) to be run are autogenerated by VEHDYN 4.0 using the data in the special format profile data file (obs_geometry.dat - see Chapter 4 for the input guide) shown in Figure 66.

```
2
12  36
2
72  120
1
4.71
```

Figure 66. Obstacle geometry data file.

These data define two obstacle heights (12 and 36 in.), two obstacle widths (72 and 120 in.), and one approach angle (4.71 rad) for a total of 2 × 2 × 1 = 4 obstacles.

The output is collected in two files. ObsMod.out, which is in ObsMod format, is shown in Figure 67.

```
NOHGT
2
NANG
1
NWDTH
2
CLRMIN    FOOMAX     FOO     HOVALS    AVALS     WVALS    SPEEDS (MPH) FOR
INCHES    POUNDS    POUNDS   INCHES   RADIANS    INCHES    1G   2.5G   4G
 14.59    78851.0    665.8    12.00      4.71     72.00    2.37  8.64 60.00
  8.84  109178.2    645.9    36.00      4.71     72.00    1.12  1.69 60.00
  8.96  637255.0    979.8    12.00      4.71    120.00    0.60  2.14  3.09
  0.00  637255.0    865.0    36.00      4.71    120.00    0.02  0.05  0.08
```

Figure 67. Obstacle geometry data file.

The file profiles.out has the standard format profile data for each of the generated obstacles, as shown in Figure 68.

Problem 10: Ride performance for Caterpillar 3030 tractor over a series of nondeformable ride courses

The tenth problem to be demonstrated involves the performance computations of a Caterpillar 3030 Tractor over a series of nondeformable ride courses. Using Table 2 as a guide, the required input files for this type of run, called "Ride Performance" in the table, are the profile data file (bigprf.dat), vehicle data file (cat3030.vd3), control data file (control.dat), and a file containing profile names and starting speed guesses. The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown as Figure 69.

```
OBS 1
 Generated Obstacle
 4 0
       0.000000       0.000000
       0.399960     -12.000000
      72.399960     -12.000000
      72.799920       0.000000
OBS 2
 Generated Obstacle
 4 0
       0.000000       0.000000
       1.199880     -36.000000
      73.199880     -36.000000
      74.399760       0.000000
OBS 3
 Generated Obstacle
 4 0
       0.000000       0.000000
       0.399960     -12.000000
     120.399960     -12.000000
     120.799920       0.000000
OBS 4
 Generated Obstacle
 4 0
       0.000000       0.000000
       1.199880     -36.000000
     121.199880     -36.000000
     122.399760       0.000000
```

Figure 68. Profile output data file.

```
Command Prompt - ..\projects\vehdyn3\programs\vd4

C:\report10>..\projects\vehdyn3\programs\vd4
 CONTROL FILENAME= control.dat

 PROFILE FILENAME= Ride-Special

 ABSORBED POWER LEVEL (WATTS): 6

 MAX SPEED (MPH) & SPEED INCREMENT (MPH): 60 1

PROFILE FILENAME= bigprf.dat

 FILE OF PROFILE NAMES AND STARTING SPEEDS (MPH)= profile_complete.dat

 RIDE QUALITY OUTPUT FILENAME= rqual.out

 RIDE PERFORMANCE OUTPUT FILENAME= rperf.out

 VEHICLE FILENAME= cat3030.vd3
```

Figure 69. Screen capture for problem 10.

The special profile name "Ride-Special" is input to VEHDYN 4.0 to tell the program that this will be a ride performance series of runs. The program then asks the user what absorbed power level of computation is to be made (6 W in our example), maximum vehicle speed (60 mph), and speed increment for each run (1 mph). The profile file "bigprf.dat" contains many profiles stacked one on the other; the file "profile_complete.dat" contains a list of which profiles from bigprf.dat the user wants to run along with a starting speed guess for each profile.

The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 70.

```
3030EST
FC8_DTCH
2.5,0.001,0.,15.0,0.,5.,0.01
5.,30.,0.
0 0
0.6,0.7,0.0,0.0
0 -1 0.0
```

Figure 70. Control data file.

As with problems 5 through 9, the second line is irrelevant, as the actual list of courses is contained in the file of profile names and starting speeds.

The vehicle data file for the Caterpillar 3030 is different from previous vehicle files in these sample problems in that it is in an older VEHDYN 3 format that VEHDYN 4.0 still supports. The main difference is that a VEHDYN 3 vehicle file does not have the tractive force-speed data, the engine data, the aerodynamic drag coefficient, vehicle frontal area, maximum braking coefficient, hydrodynamic drag coefficient, and maximum fording depth. The resulting rather lengthy cat3030.vd3 vehicle file is listed in Figures 71, 72, and 73.

```
3030EST
Caterpillar 30/30 Engineer Support Tractor (12/90)
1,4,5,0,0,5
22,0,0.,-1.4,7.6
-2.7,-2.4,-1.7,-1.0,-0.4,0.1,0.6,1.1,1.6,2.1,2.6,3.1,3.6,4.1,4.6,5.1
5.6,6.1,6.6,7.1,7.6,8.6
-85070.,-30824.,-10164.,-3429.,-114.,1916.,3352.,4485.,5455.,6343.
7198.,8056.,8946.,9892.,10921.,12061.,13344.,14811.,16514.,18522.
20928.,55000.
17,0,0.,-2.15,4.15
-4.15,-2.15,-2.,-1.5,-1.,-.5,0.,.5,1.,1.5,2.,2.5,3.,3.5,4.,4.15,6.05
-30000.,21731.,22069.,23264.,24574.,26015.,27609.,29377.,31348.,33558.
36048.,38874.,42101.,45817.,50133.,51565.,100000.
23,0,0.,-3.5,7.7
-4.5,-3.5,-3.2,-2.7,-2.2,-1.7,-1.1,-0.6,0.,0.6,1.2,1.8,2.4,3.,3.6,4.2
4.9,5.5,6.1,6.7,7.4,7.7,8.7
-15000.,2881.,2956.,3092.,3243.,3412.,3602.,3817.,4059.,4335.,4651.
5015.,5437.,5931.,6514.,7208.,8047.,9073.,10350.,11968.,14069.
15649.,35000.
22,0,0.,-4.36,8.85
-5.35,-4.36,-3.72,-3.06,-2.38,-1.69,-0.97,-0.24,0.49,1.24,1.99,2.74
3.49,4.24,4.94,5.72,6.44,7.15,7.84,8.52,8.85,9.85
-10000.,10211.,10447.,10688.,10937.,11196.,11465.,11748.,12046.,12362.
12699.,13058.,13443.,13858.,14307.,14795.,15326.,15908.,16549.,17257.
17640.,40000.
45,0,0.
-150.,-143.,-135.,-128.,-120.,-113.,-105.,-98.,-90.,-83.,-75.,-68.,-60.,-53.
-45.,-38.,-30.,-23.,-15.,-8.,-3.,-1.5,0.,1.5,3.,8.,15.,23.,30.,38.,45.,53.
60.,68.,75.,83.,90.,98.,105.,113.,120.,128.,135.,143.,150.
-101560.,-91810.,-82560.,-73810.,-65560.,-57810.,-50560.,-43810.,-37560.
-31810.,-26560.,-21810.,-17560.,-13810.,-10560.,-7810.,-5560.,-3810.
-2560.,-1810.,-1600.,-1410.,0.,1410.,1600.,1810.,2560.,3810.,5560.,7810.
10560.,13810.,17560.,21810.,26560.,31810.,37560.,43810.,50560.,57810.
65560.,73810.,82560.,91810.,101560.
41,0,0.
-100.,-95.,-90.,-85.,-80.,-75.,-70.,-65.,-60.,-55.,-50.,-45.,-40.,-35.
-30.,-25.,-20.,-15.,-10.,-5.,0.,5.,10.,15.,20.,25.,30.,35.
40.,45.,50.,55.,60.,65.,70.,75.,80.,85.,90.,95.,100.
-26000.,-23465.,-21060.,-18785.,-16640.,-14625.,-12740.,-10985.,-9360.
-7865.,-6500.,-5265.,-4160.,-3185.,-2340.,-1625.,-1040.,-585.,-260.
-65.,0.,65.,260.,585.,1040.,1625.,2340.,3185.,4160.,5265.,6500.,7865.
9360.,10985.,12740.,14625.,16640.,18785.,21060.,23465.,26000.
```

Figure 71. Top of Caterpillar 3030 vehicle data file.

The file bigprf.dat contains many profiles all stacked one after another and is a good file to use for this type of problem as VEHDYN 4.0 requires whatever profiles are to be used in the ride performance computation to coexist in a single file. Excerpts from bigprf.dat are shown in Figure 74. This particular problem actually computes results for the cat3030 running over 26 ride courses, as listed in the profile file, complete.dat, shown as Figure 75.

```
45,0,0.
-100.,-95.,-90.,-85.,-80.,-75.,-70.,-65.,-60.,-55.,-50.,-45.,-40.,-35.
-30.,-25.,-20.,-15.,-10.,-5.,-2.,-1.,0.,1.,2.,5.,10.,15.,20.,25.,30.,35.
40.,45.,50.,55.,60.,65.,70.,75.,80.,85.,90.,95.,100.
-9500.,-8720.,-7980.,-7280.,-6620.,-6000.,-5420.,-4880.,-4380.,-3920.
-3500.,-3120.,-2780.,-2480.,-2220.,-2000.,-1820.,-1680.,-1580.,-1520.
-1503.,-1350.,0.,1350.,1503.,1520.,1580.,1680.,1820.,2000.,2220.,2480.
2780.,3120.,3500.,3920.,4380.,4880.,5420.,6000.,6620.,7280.,7980.
8720.,9500.
45,0,0.
-100.,-95.,-90.,-85.,-80.,-75.,-70.,-65.,-60.,-55.,-50.,-45.,-40.,-35.
-30.,-25.,-20.,-15.,-10.,-5.,-2.,-1.,0.,1.,2.,5.,10.,15.,20.,25.,30.,35.
40.,45.,50.,55.,60.,65.,70.,75.,80.,85.,90.,95.,100.
-2551.,-2473.,-2399.,-2329.,-2263.,-2201.,-2143.,-2089.,-2039.,-1993.
-1951.,-1913.,-1879.,-1849.,-1823.,-1801.,-1783.,-1769.,-1759.,-1753.
-1751.,-1575.,0.,1575.,1751.,1753.,1759.,1769.,1783.,1801.,1823.,1849.
1879.,1913.,1951.,1993.,2039.,2089.,2143.,2201.,2263.,2329.,2399.,2473.
2551.
45,0,0.
-100.,-95.,-90.,-85.,-80.,-75.,-70.,-65.,-60.,-55.,-50.,-45.,-40.,-35.
-30.,-25.,-20.,-15.,-10.,-5.,-2.,-1.,0.,1.,2.,5.,10.,15.,20.,25.,30.,35.
40.,45.,50.,55.,60.,65.,70.,75.,80.,85.,90.,95.,100.
-8168.,-7681.,-7218.,-6781.,-6368.,-5981.,-5618.,-5281.,-4968.,-4681.
-4418.,-4181.,-3968.,-3781.,-3618.,-3481.,-3368.,-3281.,-3218.,-3181.
-3170.,-2850.,0.,2850.,3170.,3181.,3218.,3281.,3368.,3481.,3618.,3781.
3968.,4181.,4418.,4681.,4968.,5281.,5618.,5981.,6368.,6781.,7218.,7681.
8168.
13,0,0.,2.836,3.218
2.796,2.836,2.876,2.916,2.955,2.993,3.032,3.070,3.107,3.145,3.182
3.218,3.255
-350000.,41877.,46582.,52123.,58735.,66741.,76603.,89003.,104986.
126228.,155581.,198274.,613975.
4,0,0.,1.22173,1.91986
0.22173,1.22173,1.91986,2.91986
-4.0E7,0.,0.,4.0E7
4,0,0.,0.994838,2.146755
-0.005162,0.994838,2.146755,3.146755
-4.0E7,0.,0.,4.0E7
6,0,0.,2.836,3.3
2.796,2.836,3.036,3.2,3.3,3.35
-1.E7,-70000.,0.,0.,,50000.,1.E7
13,0,0.,2.836,3.218
2.796,2.836,2.876,2.916,2.955,2.993,3.032,3.070,3.107,3.145,3.182
```

Figure 72. Middle of Caterpillar 3030 vehicle data file.

File complete.dat is an ordered list of the 26 profiles to be run. Each of these profiles must exist in the profile data file bigprf.dat. Following each profile name, there is a single number designating an initial guesstimate for starting speed in mph. This speed should be such that, when the vehicle is run over this course at that speed, the absorbed power value is less than the target absorbed power of 6.0 W. If the initial speed is too great, VEHDYN 4.0 cuts the speed in half and will start the run sequence again.

```
3.218,3.255
-5.E7,-324200.,-27710.,-2217.,-1556.,-755.,230.,1470.,3068.
26000.,81280.,123970.,5.E7
1,0,1,1,0,1,1
55.,26.77
0.,0.,0,0
37000.,2.1E5,35.1,6.75,14.90,90.,90.,0.
16.87,1500.,47.0,16.55,0.,0.,3800.,1,0,12.3,1.25,3,1,0,0,-1
9.0,600.,25.0,8.8,0.,0.,2450.,0,0,12.3,1.25,3,1,0,0,-1
9.0,600.,11.0,8.8,0.,0.,2450.,0,0,12.3,1.25,3,1,0,0,-1
9.0,450.,-8.0,8.81,0.,0.,2300.,0,0,12.3,1.25,3,1,0,0,-1
9.0,450.,-22.0,8.81,0.,0.,2300.,0,0,12.3,1.25,3,1,0,0,-1
16.87,800.,-47.5,16.44,0.,0.,5200.,1,0,12.3,1.25,3,1,0,0,-1
1,1,6,2
25.0,0,1.86,183.1,187.5,1,17.0
13500.,0.
16.,-73.42,35.5,0.52,4400.,0,0,0,-1
18.,51.95,30.,33.52,18.,14.12,-8.7,29.9,30.,19.9
1,1,0,0
0,2,0,0,2
2,3,0,0,4
2000.,15.,30.,1500.
6,4,5,0,0
3,4,0,0,3
-15.,30.,1500.
4,0,0,0,-15.,-15.
5,0,0,0,-15.,-15.
```

Figure 73. Bottom of Caterpillar 3030 vehicle data file.

```
apg_11  1.3209 RMS
APG COURSE 11, 400 FEET LONG (1.32 IN. RMS) XM=6.2220 SD=10.4075 L=400 ft
     401     12.
     .00     .24     .48     .48    1.20    1.20     .36    -.12     .96    1.20
     .72     .84     .84    1.20    1.08    1.20    2.16    2.88    3.36    4.20
    5.64    5.76    5.52    5.04    4.20    3.36    4.08    4.44    4.80    4.80
-
-
-
    6.24    6.72    6.96    6.84    7.20    7.80    8.76    9.12    9.00    9.00
    8.40    8.16    7.92    7.32    6.84    6.12    5.76    5.40    4.56    4.56
    4.68
apg_12   1.7179 RMS
APG COURSE 12, 300 FEET LONG (1.718 IN. RMS) XM=-39.6279 SD=20.5498 L=300 ft
     301     12.
     .00    -.36    -.48   -1.20   -1.20   -1.44   -1.44   -1.92   -2.76   -3.00
   -3.60   -3.84   -4.56   -5.40   -7.44   -9.48  -10.80   -7.44  -11.52  -11.40
-
-
-
  -82.20  -82.80  -80.28  -79.20  -79.56  -80.04  -79.32  -79.92  -80.16  -80.64
  -80.40
APG_12*  1.6427 RMS
APG COURSE 12, 291 FEET LONG (1.643 IN. RMS)  XM=-38.3852 SD=19.5872 L=291 ft
     292     12.
     .00    -.36    -.48   -1.20   -1.20   -1.44   -1.44   -1.92   -2.76   -3.00
   -3.60   -3.84   -4.56   -5.40   -7.44   -9.48  -10.80   -7.44  -11.52  -11.40
  -11.76  -12.36  -12.84  -13.44  -14.28  -15.24  -16.68  -17.52  -17.40  -16.32
-
-
-
```

Figure 74. Excerpts from the bigprf.dat file.

```
let21t
30
ftkn12a
30
let11t
30
apg_9
48
LET4LT
46
let51t
30
apg_11
30
let4rt
25
SR3RT82
27
LET5LT
20
apg_12
25
ftkn56a
18
LET6LT
20
YPG04
20
APG_27
19
let6rt
20
apg_IV
19
apg_29a
18
ftkn910
19
SR5RT731
19
LET7LT
15
APG_14*
7
ftkn78a
16
SR4RT731
14
LET16
6
LET7LTD
6
```

Figure 75. Data file containing list of ride courses.

For each profile, the cat3030 will begin running at the speed listed in this file. Then, assuming the result is an absorbed power less than the target value of 6.0 W, it will increment by the inputted speed increment given during the run sequence (1 mph) and will continue running problems up to the maximum vehicle speed, also given in the run sequence (60 mph), or at least until the target absorbed power (6.0 W) is reached. Two output files, a ride quality file (rqual.out) and a ride performance file (rperf.out), collect the resulting run results. The ride quality file contains data results from each run that is computed, whereas the ride performance file contains only the resulting speeds that give the target absorbed power value as well as the RMS roughness index for each course. A partial listing of the ride quality file rqual.out is shown in Figure 76.

```
-------------------------------------------------
     Profile= let2lt    (RMS Roughness=  0.615 in.)
     -----------------------------------------------
    Speed (mph)    Avg Abs Power (W)   Filt RMS Acc (G's)
    -----------    -----------------   ------------------
      2.50             0.133               0.0416
      3.50             0.398               0.0663
      4.50             0.927               0.0910
      5.50             1.628               0.1130
      6.50             2.135               0.1307
      7.50             2.238               0.1420
      8.50             2.419               0.1590
      9.50             2.565               0.1730
     10.50             2.682               0.1887
     11.50             2.806               0.2056
     12.50             2.868               0.2204
     13.50             3.161               0.2361
     14.50             3.517               0.2496
     15.50             3.815               0.2623
     16.50             4.155               0.2750
     17.50             4.503               0.2845
     18.50             5.136               0.3008
     19.50             5.733               0.3173
     20.50             5.984               0.3253
     21.50             6.445               0.3338
-------------------------------------------------
     Profile= ftkn12a   (RMS Roughness=  0.688 in.)
     -----------------------------------------------
    Speed (mph)    Avg Abs Power (W)   Filt RMS Acc (G's)
    -----------    -----------------   ------------------
     15.00             3.345               0.2430
     16.00             3.697               0.2514
     17.00             3.836               0.2565
     18.00             4.095               0.2626
     19.00             4.292               0.2722
     20.00             4.618               0.2795
     21.00             5.035               0.2903
     22.00             5.803               0.3037
     23.00             6.173               0.3152
-------------------------------------------------
     Profile= let1lt    (RMS Roughness=  0.862 in.)
     -----------------------------------------------
```

Figure 76. Partial listing of ride quality output file.

A complete listing of the ride performance file rperf.out is shown as Figure 77.

```
 Vehicle= 3030EST
Caterpillar 30/30 Engineer Support Tractor (12/90)
 ------------------------------------------------------------
    Profile   RMS (in)    6.0-Watt Speed (mph)
    -------   --------   ---------------------
    let2lt     0.615             20.53
    ftkn12a    0.688             22.53
    let1lt     0.862             23.01
    apg_9      1.037             16.40
    LET4LT     1.038             17.11
    let5lt     1.262             15.56
    apg_11     1.322             15.15
    let4rt     1.351             13.58
    SR3RT82    1.454             14.95
    LET5LT     1.690             12.59
    apg_12     1.715             13.40
    ftkn56a    1.738             14.61
    LET6LT     1.746             11.95
    YPG04      1.817             10.74
    APG_27     1.886             13.37
    let6rt     2.103             12.07
    apg_IV     2.185             11.03
    apg_29a    2.232             12.33
    ftkn910    2.308             16.61
    SR5RT731   2.325             16.89
    LET7LT     2.502              9.78
    APG_14*    3.287              8.33
    ftkn78a    3.367             12.09
    SR4RT731   3.890             11.82
    LET16      3.976              7.46
    LET7LTD    5.005              6.22
```

Figure 77. Listing of ride performance output file.

Problem 11: Shock performance for M998 HMMWV over a series of nondeformable half-round obstacles

The eleventh problem to be demonstrated involves the movement of the M998 HMMWV 4×4 wheeled vehicle from Problem 6 over a series of nondeformable half-round obstacles to determine the vehicle's shock performance. Using Table 2 as a guide, the required input files for this type of run, called "Shock Performance" in the table, are the vehicle data file (m998.vd4 – same file as in Problem 6), control data file (control.dat), and the M998's standard UID data file if desired. Because we do not expect the vehicle underside to impact the obstacles, we elected not to include the UID file. The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 78.

To tell VEHDYN 4.0 that this problem is to be a shock performance run, the user must input the special case-sensitive profile filename "Shock-Special." The program understands this set of characters and initiates the

```
Command Prompt - ..\projects\vehdyn3\programs\vd4
C:\report11>..\projects\vehdyn3\programs\vd4
 CONTROL FILENAME= control.dat

 PROFILE FILENAME= Shock-Special

 FILTERED MAX ABS ACC LEVEL (G'S),
 MIN & MAX OBSTACLE HEIGHTS (IN) AND
 OBSTACLE HEIGHT INCREMENT (IN): 2.5 5 20 0.25

 SHOCK QUALITY OUTPUT FILENAME= squal.out

 SHOCK PERFORMANCE OUTPUT FILENAME= sperf.out

 VEHICLE FILENAME= m998.vd4
```

Figure 78. Screen capture for problem 11.

appropriate question-answer sequence shown in the figure above for this type of problem. The next required input is the peak acceleration level for which we wish to determine speeds (2.5 g's), as well as the minimum (5.0-in.) and maximum (20.0-in.) obstacle heights and obstacle height increment (0.25 in.). VEHDYN 4.0 generates the half-round obstacle profiles as needed to run the M998 HMMWV across. The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 79.

```
M998
FC8_DTCH
2.5,0.001,0.,15.0,0.,5.,0.01
5.,30.,0.
0 0
0.6,0.7,0.0,0.0
0 0 0.0
```

Figure 79. Control data file.

As with Problems 5 through 10, the second line of data in the control file is not used and, therefore, could be anything. The actual profiles are auto-generated by VEHDYN 4.0 as the program runs. The vehicle file m998.vd4 is the same one as listed under Problem 6 earlier in this chapter. The output for a shock performance run is collected in a similar fashion as for the ride performance run. VEHDYN 4.0 writes shock quality data to the user-inputted filename “squal.out” and shock performance data to the filename “sperf.out.” A partial listing of the lengthy shock quality file is shown in Figure 80. The resulting shock performance file “sperf.out,” containing the table of 2.5-g speeds for all of the obstacles, is shown in Figure 81.

```
---------------------------------------------------------
     Obstacle Ht.=   5.00 in.
     ------------------------
    Speed (mph)    Peak Abs Filt G's
    -----------    -----------------
      8.00              1.409
      9.00              1.509
     10.00              1.563
     11.00              1.590
.
.
.
     57.00              1.575
     58.00              1.570
     59.00              1.551
     60.00              1.511
---------------------------------------------------------
     Obstacle Ht.=   5.25 in.
     ------------------------
    Speed (mph)    Peak Abs Filt G's
    -----------    -----------------
      7.00              1.318
      8.00              1.402
      9.00              1.506
     10.00              1.579
.
.
.
     57.00              1.580
     58.00              1.560
     59.00              1.510
     60.00              1.476
.
.
.
---------------------------------------------------------
     Obstacle Ht.=  19.75 in.
     ------------------------
    Speed (mph)    Peak Abs Filt G's
    -----------    -----------------
      3.00              1.147
      4.00              1.324
      5.00              2.988
---------------------------------------------------------
     Obstacle Ht.=  20.00 in.
     ------------------------
    Speed (mph)    Peak Abs Filt G's
    -----------    -----------------
      3.00              1.136
      4.00              1.333
      5.00              3.020
```

Figure 80. Partial listing of shock quality output file.

Vehicle= M998
M998 HMMWV (2/28/92)

Obstacle Ht (in)	2.5-G Speed (mph)
5.00	60.00
5.25	60.00
5.50	60.00
5.75	60.00
6.00	60.00
6.25	60.00
6.50	60.00
6.75	60.00
7.00	60.00
7.25	60.00
7.50	60.00
7.75	35.01
8.00	29.00
8.25	15.92
8.50	15.53
8.75	14.93
9.00	14.54
9.25	14.22
9.50	13.92
9.75	13.40
10.00	12.68
10.25	11.81
10.50	11.04
10.75	10.75
11.00	10.48
11.25	10.08
11.50	9.83
11.75	9.65
12.00	9.45
12.25	9.31
12.50	9.11
12.75	8.92
13.00	8.75
13.25	8.60
13.50	8.44
13.75	8.33
14.00	8.25
14.25	8.19
14.50	8.13
14.75	8.08
15.00	8.04
15.25	7.60
15.50	7.02
15.75	5.88
16.00	5.39
16.25	5.00
16.50	4.97
16.75	4.94
17.00	4.92
17.25	4.89
17.50	4.87
17.75	4.85
18.00	4.83
18.25	4.81
18.50	4.79
18.75	4.77
19.00	4.75
19.25	4.74
19.50	4.72
19.75	4.71
20.00	4.69

Figure 81. Listing of shock performance output file.

Problem 12: Drawbar pull determination for M2 BFV on a deformable 200-RCI SM soil surface

The last problem to be demonstrated involves the determination of drawbar pull for the M2 BFV from Problem 5. This type of problem is a simulation of a drawbar pull test and computes through a series of iterations the horizontal resistance required to bring a moving vehicle to a dead stop on a given deformable surface (200-RCI SM soil in this problem). The vehicle is initially accelerated to its maximum speed, then a resisting load applied, and the run continues until a stable constant speed is achieved via the tractive force-speed relations. Then, the resisting load is increased and the process repeated, until the stable constant speed is as close to zero as possible (i.e., within a preset tolerance). Each time a new load is applied, output is sent to the main output file. Using Table 2 as a guide, the required input files for this type of run, called “Drawbar Pull Test” in the table, are the vehicle data file (m2.vd4), control data file (control.dat), standard UID data file (m2.uid), and the VMI soil data file (vmi.dat). The VEHDYN 4.0 program is run from a command line in a DOS window; a screen capture of what this would look like for this problem is shown in Figure 82.

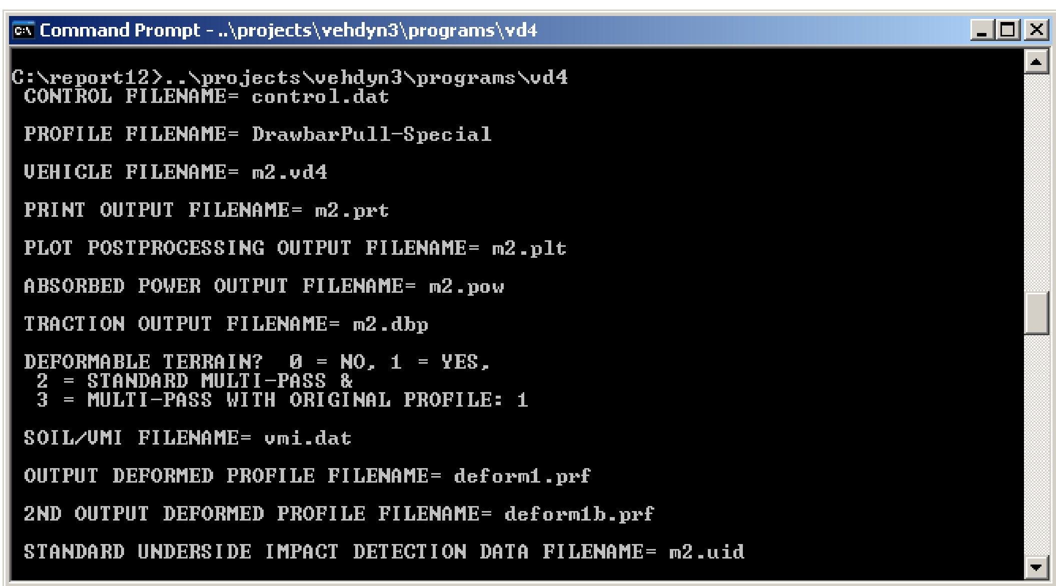


Figure 82. Screen capture for problem 12.

The input files all must be created prior to running VEHDYN 4.0. According to the format guides given in Chapter 4, the control data file, named control.dat, is shown in Figure 83.

```
M2
LET7AVG
0.1,0.001,0,600.0,0.0,0.5,0.05
0.5,30,0
1,1
0.6,0.7,1.0,1.0
0,1,40.0
```

Figure 83. Control data file.

As with Problems 5 through 11, the second data line is irrelevant as the actual course used is a flat autogenerated terrain. The vehicle file (m2.vd4) and its corresponding UID underside data file (m2.uid) are the same as in Problem 5 earlier in this chapter. Also, the formatted output file (m2.prt), the plot postprocessing output file (m2.plt), the absorbed power output file (m2.pow), and the two deformed profile output files (deform1.prf and deform1b.prf) are generated and are similar to files of that type in earlier sample problems in this chapter. What is of interest in this problem is the drawbar pull test data stored in the traction output file "m2.dbp." These data are shown in Figure 84.

```
M2 Bradley Fighting Vehicle --- 1985      WEIGHT (LBS):     66000.00
 SOIL TYPE: SM    RCI:  200.
 SURFACE: CONDITION- NFG  MOISTURE- NOR
  DEPTH-  99.9   DENSITY-  0.
  FROST DEPTH-  0.   THAW DEPTH-  0.
                                                            VELOCITY
    ELAPSED                        APPLIED      MOTION      DEPENDENT
     TIME    VELOCITY   SLIP        FORCE     RESISTANCE   RESISTANCE
    (SECS)    (MPH)      %          (LBS)       (LBS)        (LBS)
     0.0000  37.34419   0.281        0.000     3728.32      208.794  U
    20.0380  13.01574   0.993     8250.000     2998.17       25.364  S
    27.7420   7.72798   1.885    15015.000     3050.22        8.941  S
    32.0850   5.02716   2.934    20562.300     3130.43        3.784  S
    34.4300   4.04364   4.124    25111.086     3173.74        2.448  S
    36.6100   3.38803   5.382    28841.091     3101.09        1.719  S
    38.5750   2.86184   6.815    31899.694     3127.14        1.226  S
    40.3760   2.49090   8.355    34407.749     3140.12        0.929  S
    41.8940   2.21257  10.161    36464.354     3146.71        0.733  S
    43.2300   2.01420  12.093    38150.771     3110.88        0.607  S
    44.5310   1.85336  13.660    39533.632     3066.49        0.514  S
    45.9000   1.69427  15.210    40853.632     3070.28        0.430  S
    47.5320   1.51967  19.670    42173.632     3107.09        0.346  S
    49.2490   1.34237  25.033    43493.632     3133.24        0.270  S
    51.0550   1.16685  30.342    44813.632     3146.16        0.204  S
    52.8900   0.99005  35.690    46133.632     3168.92        0.147  S
    56.0210   0.78172  44.559    47453.632     3171.66        0.091  S
    60.4800   0.37328  73.527    48773.632     3206.14        0.021  S
    62.5250   0.00000 100.000    48773.632     3206.14        0.000  U
```

Figure 84. Listing of drawbar pull output file.

This output file reports data such as the time, speed, wheel slip, applied load, and the motion resistance after each leg of the test (i.e., at the point when the speed became stable after an additional load was applied).

A depiction of the M2 BFV from the VEHDYN 4.0 animation post-processor VD4Animator is shown as Figure 85.

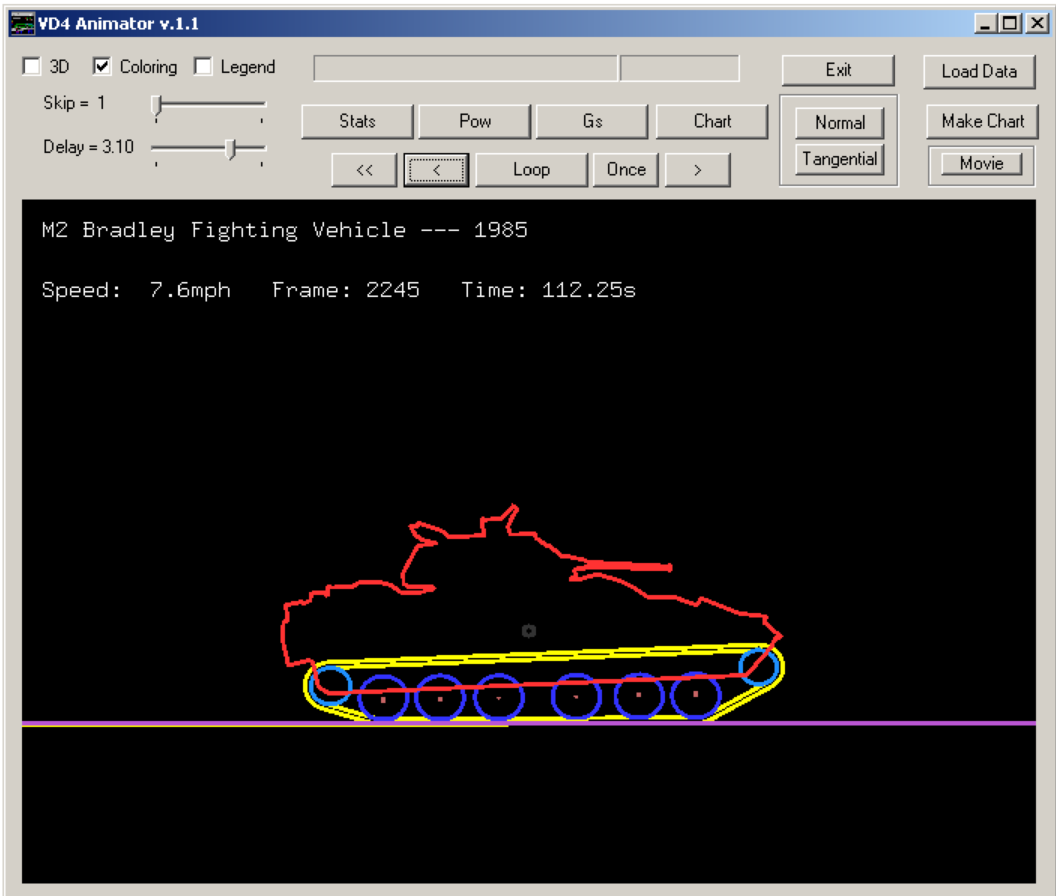


Figure 85. Snapshot of M2 animation.

6 Conclusions and Recommendations

Conclusions

This report has described VEHDYN 4.0, which is the latest version of the longstanding series of two-dimensional (2-D) vehicle dynamics programs developed and maintained by the ERDC-GSL's Mobility Systems Branch. The current version of VEHDYN is able to both execute many different types of runs (12 of which are listed in Table 2) as well as provide a user with several enhancements over the earlier versions of VEHDYN. These enhancements include the following:

- the direct input of the settled configuration of the vehicle
- rotational springs for enhanced suspension beam modeling
- a band track model with local and uniform tension modes
- a wheel damping model for computing an additional normal force component to represent the wheel's resistance to change
- a vehicle-terrain interface model to predict soil deformation and wheel/track slip
- a vehicle underside interference model that in conjunction with the VTI model computes soil sinkage, the normal forces, and the drag forces that occur when the underside of a vehicle's sprung mass comes into contact with the terrain profile
- a variable-speed mode that allows the vehicle to accelerate and decelerate based on its tractive force versus speed curve
- a water-vehicle buoyancy and resistance model that is invoked when the vehicle attempts to cross a water-filled ditch
- a new postprocessor for displaying an animation of the results called VD4ANIMATOR

Recommendations

Based on the information presented in this study, it is recommended to

- enhance the model to allow for multiple unit vehicle configurations, thus enabling the modeling of vehicles with trailers,
- Perform validation and verification on the new effect models to confirm that the best and most accurate models are being incorporated into VEHDYN.

References

Creighton, D. C. 1986. *Revised vehicle dynamics module: user's guide for computer program VEHDYN II*. Technical Report SL-86-9. Vicksburg, MS: U.S. Army Engineer Waterways Experiment Station.

Creighton, D. C. 1987. *A preprocessor for the revised vehicle dynamics module: user's guide for computer program PREVDYN2*. Technical Report SL-87-9. Vicksburg, MS: U.S. Army Engineer Waterways Experiment Station.

Murphy, N. R., and Ahlvin, R. B. 1976. *AMC-74 vehicle dynamics module*. Technical Report M-76-1. Vicksburg, MS: U.S. Army Engineer Waterways Experiment Station.

Sedgewick, R. 1983. *Algorithms*. Reading, MA: Addison-Wesley.

Appendix A: Wheel Damping Model

A new feature in this version of VEHDYN is that damping has been added to the continuous spring tire model and applies to all types of wheels including the underside wheels. The damping is an additional normal force component added to the continuous spring model's current normal force and represents the tire's resistance to change in average normal deflection as a function of time. Figure A1 represents a typical situation for a wheel as it traverses a terrain profile.

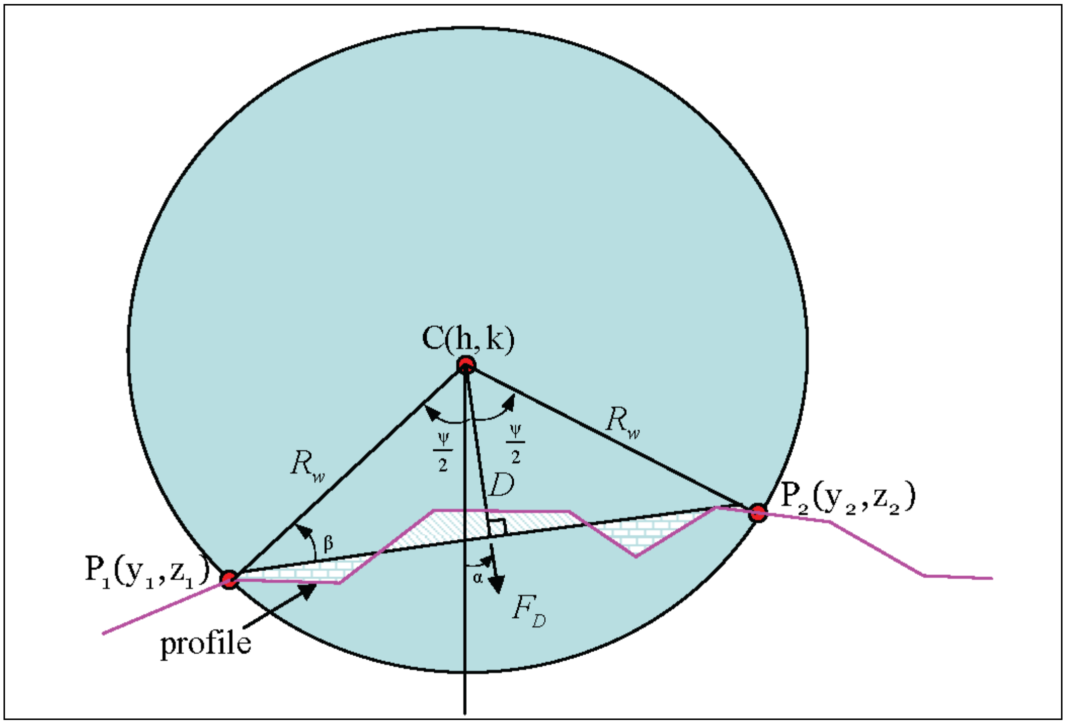


Figure A1. Typical wheel during terrain profile traversal.

Points $P_1(y_1,z_1)$ and $P_2(y_2,z_2)$ represent the intersection of the terrain profile with the wheel undeflected circle while point C (h,k) is the circle's center. The undeflected radius of the wheel is R_w. A bisector of angle P_1CP_2 (ψ in the figure) is constructed of length D such that the normal to the bisector cuts through the profile in such a way as to exactly balance the areas above and below the normal to the bisector. The area above the line is shown as the area shaded with diagonal lines and the area below the line is shown as the brick pattern. If D_{now} represents the average normal length from the wheel center to the intersecting profile (length D in the figure) for the current time step and D_{old} represents the average normal length from the wheel center to the intersecting profile for the previous time step, then the normal resistance due to damping (F_D) as the length D varies from time step to time step can be represented as

$$F_D = C_D v_D = C_D \left(\frac{D_{now} - D_{old}}{dt} \right)$$

where:

C_D = damping coefficient (a user input quantity)
υ_D = speed of movement of the normal to the bisector
dt = time step.

Once the resistance F_D has been determined, it is applied to the horizontal and vertical wheel center force components as follows

$$F_D^Y = \text{horizontal component of } F_D = F_D \sin\alpha$$
$$F_D^Z = \text{vertical component of } F_D = -F_D \cos\alpha$$

where:

$$\alpha = \frac{\psi}{2} - \left(\frac{\pi}{2} - \beta \right) = \frac{\psi}{2} + \beta - \frac{\pi}{2}$$

$$\beta = \tan^{-1}\left(\frac{k - z_1}{h - y_1} \right)$$

Appendix B: Underside Impact Detection Model

Another new feature that has been added to the current VEHDYN is interaction resistance between the underside of the vehicle chassis and the terrain profile referred to herein as the underside impact detection (UID) model. When modeling a vehicle attempting to cross a piece of off-road ground, an important consideration is the determination of the vehicle's ability to cross certain rough areas, particularly mounds and ditches. A new interaction relationship has been added to the current VEHDYN program to account for the normal and drag forces that occur when the underside of a vehicle's sprung mass comes in contact with the terrain profile. This appendix discusses this new UID model in terms of a figure of the model representation along with the corresponding adjustments required to VEHDYN 4.0's equations of motion resulting from any contact between the underside and the terrain.

Figure B1 is a picture of how the underside is modeled using the new UID model. Prior to VEHDYN 4.0, there were two types of wheels—towed and driven. To model the vehicle underside, a new wheel type was created—the UID wheel. It is similar to a driven wheel, with both a normal and traction component, except that the traction component drags rather than pushes because it is a wheel that cannot turn. The entire portion of underside that could possibly interact with the terrain as the vehicle traverses the profile is modeled with a continuous set of small touching UID wheels (the orange circles in Figure B1). These tiny UID wheels are placed along the underside of the vehicle frame and await impact with the ground. Each UID wheel is like a drag or fully braked wheel as the tangential force involved is opposite to the direction of the corresponding force component of a driven wheel. The pertinent force components and resulting adjustments to the chassis equations of motion are illustrated in Figure B2.

In the current implementation of the model, a user is required to input a maximum traction coefficient for normal drive wheels (to define the maximum available traction) and a traction coefficient for the UID wheels to be applied when one or more UID wheels impacts the ground surface. Each terrain profile segment is treated like profile segments that interact with regular road wheels using the continuous spring model as outlined

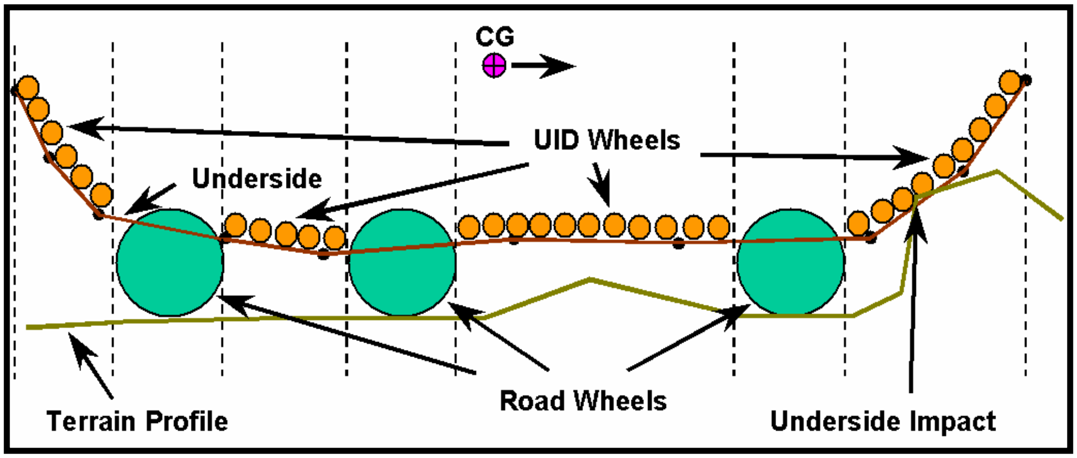


Figure B1. Vehicle underside impact detection model for obstacle crossing.

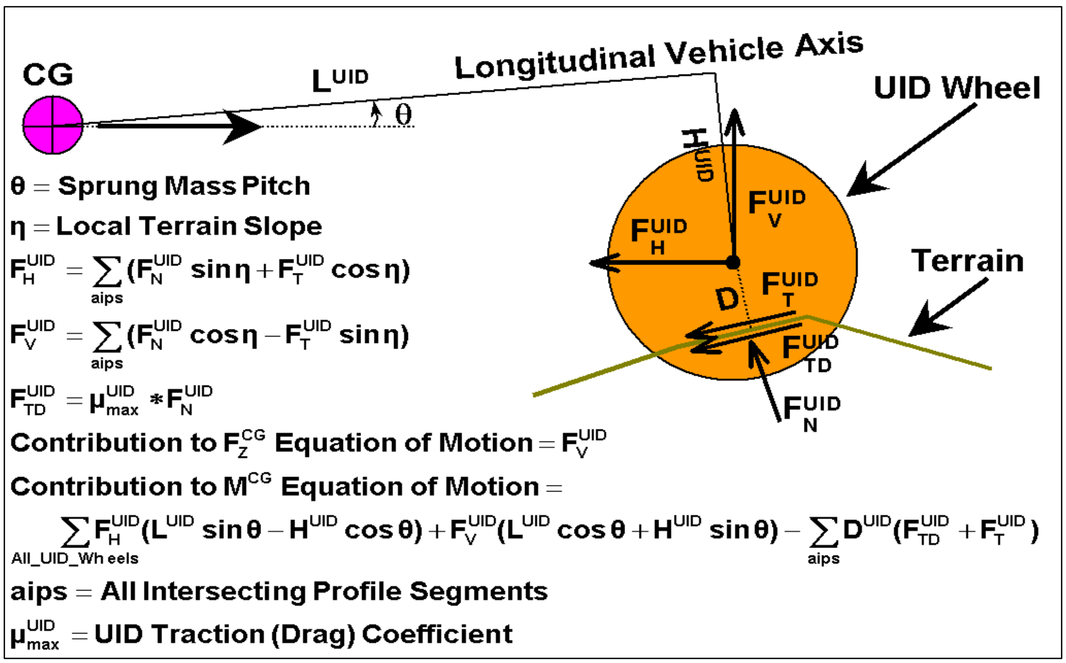


Figure B2. Forces and relevant contributions to chassis equations of motion resulting from UID wheel interactions.

in the VEHDYN II User's Guide (Creighton 1986) including the new wheel damping model outlined in Appendix A. Normal and tangential force components at the tire-profile interface are calculated, and then resolved to vertical and normal force components applied at the UID wheel's center. These force components are then used to adjust the two equations of motion for the vehicle chassis (vertical and pitch motions) according to the equations in Figure B2. For run types that maintain a constant speed as the vehicle proceeds down the course, the required traction (push) coefficient (*C*) to maintain the user-selected constant speed is calculated for the driven road wheels. The equation used to calculate *C* is:

$$C = \frac{\sum_{Road\_Wheels} F_Y + \sum_{Track\_Model} F_Y^T + \sum_{UID\_Wheels} F_Y^{UID} + \mu_{max}^{UID} \sum_{UID\_Wheels} \sum F_N \cos\eta}{\sum_{Drive\_Wheels} \sum F_N \cos\eta}$$

The numerator of the right-hand side is the total horizontal resistance, while the denominator is that portion of drive-wheel normal force that is acting in the horizontal direction. The double summation symbols in the equation above indicate the sum over all intersecting profile segments in each case. As defined in Figure B2, the subscript *Y* represents the horizontal direction, the subscript *N* represents the normal direction, and the variable η represents a profile segment's slope. On flat surfaces, this traction coefficient would be zero because all horizontal resisting force components are zero and there are no UID wheel-terrain interactions. But as the vehicle proceeds up and down hills and valleys, nonzero horizontal wheel resistances develop and need to be balanced with computed traction components and applied to each driven road wheel. User-supplied values of UID-wheel drag coefficient μ_{max}^{UID} and maximum available traction coefficient for the drive wheels μ_{max}^{D} are used at each time step to determine whether an underside-ground interaction is sufficient to stop the vehicle. A comparison of resisting forces versus available traction is made to determine if the available traction is sufficient to overcome the resistance.

Appendix C: Vehicle-Terrain Interface Model

The ERDC-GSL Mobility Systems Branch has been developing the methodologies used in the VTI system for several decades. The algorithms describing the VTI presented in this appendix describe the vehicle's performance by modeling the various components that provide the collective force and movement on a single traction element and the entire vehicle. The current state of the art for terrain mechanics is insufficient to utilize a finite element method (a popular approach for tire modeling on hard surfaces) due to the slowness of the required calculations. Rather, the existing partitioning and physical properties of a traction element are based on single (whole traction element) quantities, and the resulting reactions are disassociated to provide reactions for the individual terrain nodes that are in contact with the traction element.

The chosen method for creating a terrain mechanics model was derived from testing in laboratory soil bins. This method estimates a vehicle's traction element performance based on a dimensionless numeric, dependent on the vehicle's traction element parameter sand soil strength. This method of predicting vehicle mobility performance has been designated as the "WES numeric." The WES numeric relates vehicle/traction element performance to a combination shear and bearing strength and plasticity of the soil as determined by the penetration resistance of a standard cone at a prescribed penetration rate. The cone index (CI) is adjusted for cohesive soils, which have an observed tendency to reduce the mobility capability of vehicles as the soil is displaced from its in situ state by vehicle traction. This apparent reduction in strength, termed remold index (RI), is quantified by a standard field test involving the application of a standard amount of "work" to a sample of soil using a field test apparatus. The rating cone index (RCI) is determined by multiplying the in situ CI by the RI. This resulting RCI soil strength has become a reliable indication of vehicle performance and was used to relate measurable vehicle parameters to traction element test results to create the WES numeric.

The WES numeric was designed to predict the fundamental traction forces of drawbar, soil motion resistance, drive torque, slip, and sinkage of a single traction element. Since the era of extensive laboratory dynamometer testing and the development of the WES numeric, many aspects of the numeric method have been revised to better reflect observed vehicle field results.

The fundamental traction element forces for fine-grained and coarse-grained soils are estimated from specifically derived numerics. The clay numeric (ΠC) is a function of the RCI, traction element nominal contact length and width; tire section width, diameter, and deflection. The sand numeric (ΠS) is a function of the CI and a similar set of measurable traction element items. A variant of the clay numeric (ΠCZ) is used to estimate wheel sinkage.

Clay Numeric	Sand Numeric
$\Pi_C = \dfrac{RCI(bd)}{w\left(1-\dfrac{\delta}{h}\right)^{\frac{3}{2}}\left(1+\dfrac{b}{d}\right)^{\frac{3}{4}}}$	$\Pi_S = \dfrac{CI(bd)^{\frac{3}{2}}}{w\left(1-\dfrac{\delta}{h}\right)^{3}}$

where:

RCI, CI = soil strength
b = tire section width
d = nominal wheel diameter
w = weight beneath tire
δ = tire deflection
h = tire section height.

The vehicle traction element sinkage (z) for powered (p) and nonpowered or towed elements (u) is calculated by

Clay	Sand
Towed: $\dfrac{z_U}{d} = \dfrac{5}{\Pi_{Cz}^{\ 2}}$ $\Pi_{Cz} = \dfrac{RCI(bd)}{w\left(1-\dfrac{\delta}{h}\right)^{\frac{3}{2}}}$	Towed: $\dfrac{z_U}{d} = \dfrac{22}{\Pi_{Sz}^{\ \frac{9}{8}}}$ $\Pi_{Cz} = \Pi_S \bullet \dfrac{1}{1+\dfrac{b}{d}}$

Powered:	Powered:
$\frac{z_P}{d} = \frac{5}{\left(\frac{\Pi_{CZ}}{\text{slip}_{SP}^{\frac{1}{5}}}\right)^{\frac{5}{3}}}$	$\frac{z_P}{d} = \frac{14}{\Pi_{SZ}}$

The data presented in Figure C1 show the relationship between the HMMWV tire sinkage in clay soils and RCI soil strengths. The sinkage relationship shows the sinkage difference between the powered and nonpowered wheels for the same soil strengths. The results show that the powered wheels have larger sinkage values for the same soil strengths.

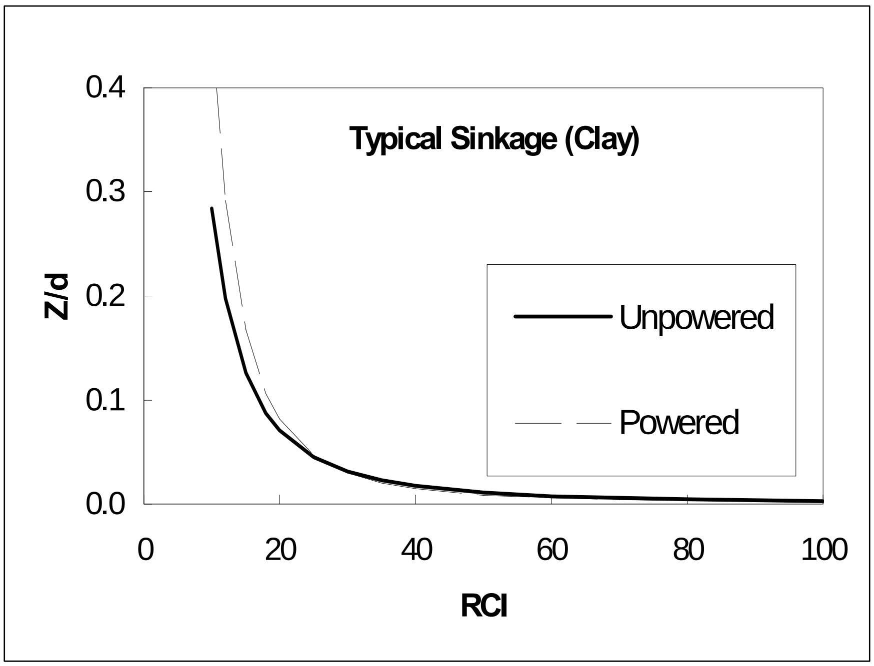


Figure C1. Sinkage of the HMMWV in clay based on the WES numeric.

A major assumption for this approach was that the performance predictions for the "whole" traction element could be broken down into small contact lengths based on the normal load. This is justified in that the equations are normalized by normal force, i.e., coefficient form. Thus, when the total force on a traction element is combined within the vehicle dynamics code, the resulting force will be representative of a whole traction element calculation.

The implementation of the sinkage relationship is based on the dynamic model time step, the chord length across the tire, the total traction element sinkage, and the wheel travel. The chord is calculated such that the area between it and the perimeter of the circle representing the undeflected tire diameter is equal to the area between the profile segments and the same circle. The diagram in Figure C2 shows the tire and terrain interface.

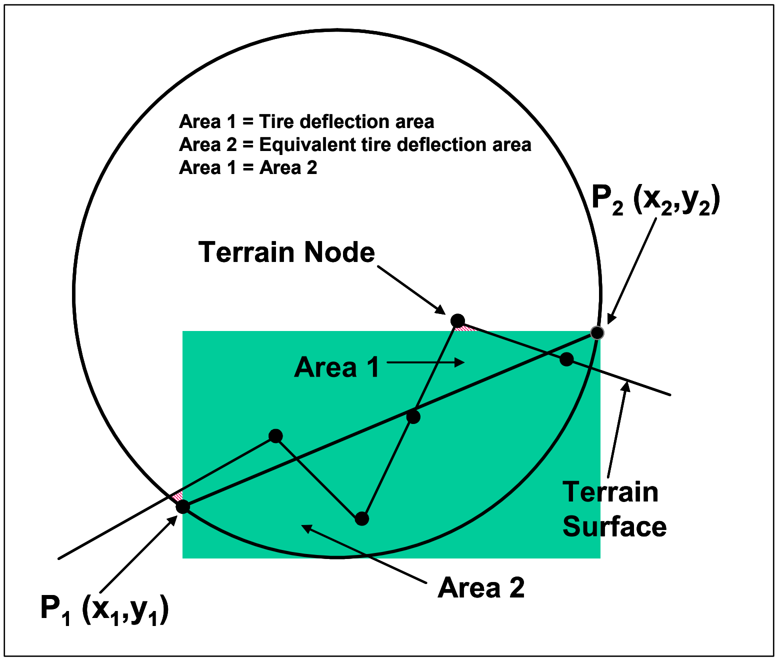


Figure C2. Exaggerated diagram of tire and terrain node interface.

The traction element sinkage is determined by the appropriate relationship and combined with the following equation to determine the amount of the total calculated traction element sinkage to be applied at each terrain node in contact with the traction element for each model time step.

$$S_c = (V \times T)/C \times S$$

where:

S_c = sinkage (in.) this time step
V = vehicle's instantaneous velocity (in./sec)
T = time step (sec)
C = chord length (in.) from P1 to P2
S = predicted total sinkage (in.) for entire wheel.

The sinkage for a tracked vehicle is determined by the same equation, but the length of the track in contact with the ground is used instead of the chord length.

The vehicle traction is determined by using the theoretical tractive force and speed relationship, as shown in Figure C3. The theoretical tractive force developed from the vehicle's propulsion system is adjusted based on the soil properties and empirical relationships to predict the available drawbar pull and soil resistance available to the vehicle. The available traction on the terrain is determined by the vehicle's current speed and the soil adjusted tractive force relationship. The available traction is applied at the wheels to determine the forward motion of the vehicle. If greater traction is required than is available from the tractive force speed relationship for the vehicle to continue forward, the vehicle will experience a "NOGO" situation or a reduced speed. To achieve a constant speed, the amount of tractive force applied and the amount of forward resistance calculated by the vehicle dynamics model must be in balance.

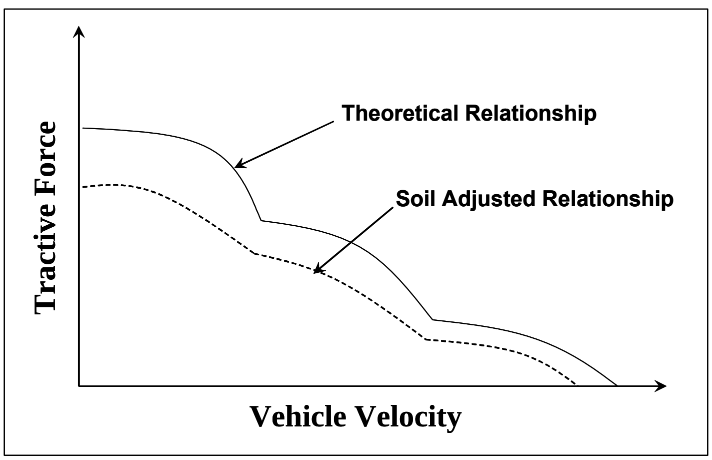


Figure C3. Vehicle tractive force speed relationship.

The vehicle dynamics wheel model is designed as a continuous spring. The wheel model takes into account the horizontal resistance from the soil as the tire tries to climb up and out of the rut created by the sinking of the traction element. This resistance and the available traction from the soil adjusted tractive force speed curve determine the vehicle's acceleration and resulting horizontal velocity.

Appendix D: Band Track Model

VEHDYN 4.0 comes with several choices for which track model to apply during the run. The choices are band track, with uniform or localized tension or interconnecting spring with profile smoothing and with or without band track output. The purpose of this appendix is to describe the methodology that goes into the computation scheme for the band track model.

The computation of the band track begins with the creation of a convex hull around the discrete points that represent the road wheels and SPRIDLERs. The convex hull of a set of points in the plane is defined to be the smallest convex polygon containing them all (Sedgewick 1983). A convex polygon is a polygon with the property that any line connecting any two points inside the polygon must itself be inside the polygon. Figure D1 shows the resulting convex hull for a set of points representing the road wheels and SPRIDLERs of the M1 tank.

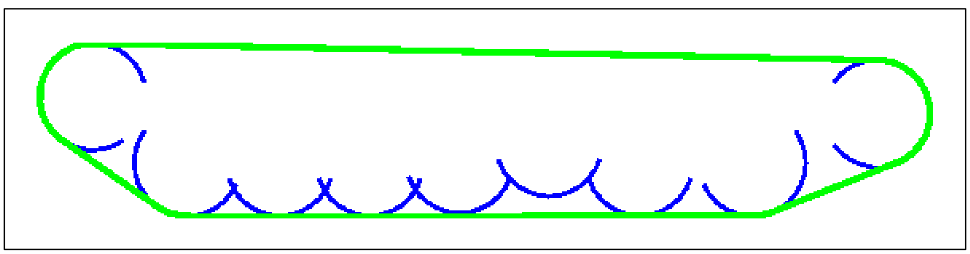


Figure D1. Convex hull around discrete points representing M1 road wheels and SPRIDLERS.

The next step in band track computation involves the deflection of the convex hull by the profile points. The result on the convex hull for the M1 is shown in Figure D2.

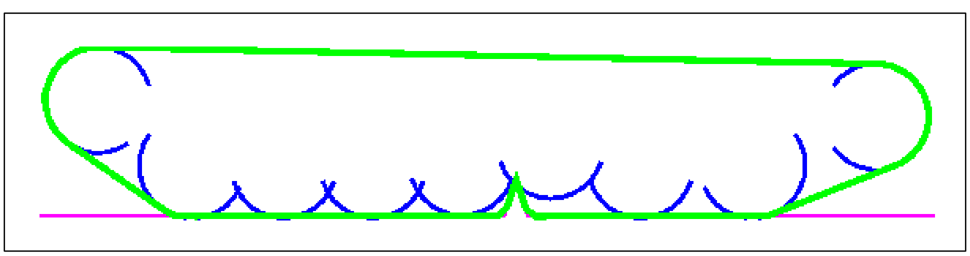


Figure D2. Convex hull after deflection by profile points.

Some points along the edge of the convex hull would not be touched by the track due to high points in the profile. These points are now eliminated. The result is shown in Figure D3.

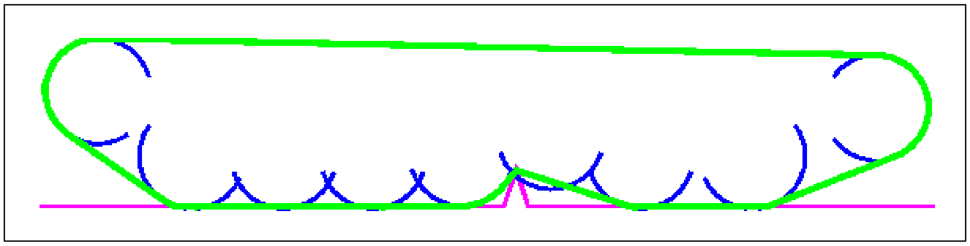


Figure D3. Convex hull after elimination of unnecessary points following profile.

As depicted in the example shown as Figure D3, the modified convex hull may now strike road wheels that did not contribute to the initial convex hull. The routine adjusts this modified convex hull to properly strike these intermediate road wheels. The final result of these computations, which represents the band track, is shown in Figure D4.

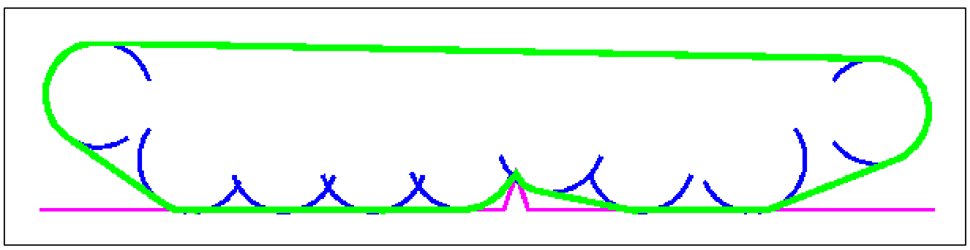


Figure D4. Final track with track no longer running through third road wheel.

When the variable ITRKTYPE is set to 2, the program applies tension forces to the road wheels and SPRIDLERs in the local area where the track is stretched. Points offset from the center of each road wheel and SPRIDLER are computed for the track in the vehicle's initial state. The distances between these points along the track are used to compute track stretching. These points for the M1 are shown in Figure D5.

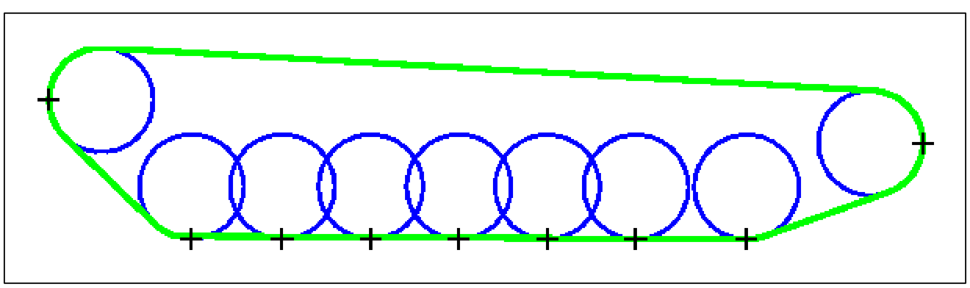


Figure D5. Initial track showing points used to compute local tension.

An example of local track stretching is shown in Figure D6. The distance along the track between the two designated points would be computed. The initial distance between the corresponding two points in Figure D5 would be subtracted from this new distance and would yield the amount the track is stretched between the SPRIDLER and the road wheel. The track tension is computed using the same equations as for uniform track tension, except that the static and instantaneous track lengths are for adjacent road wheels and SPRIDLERs rather than for the entire track.

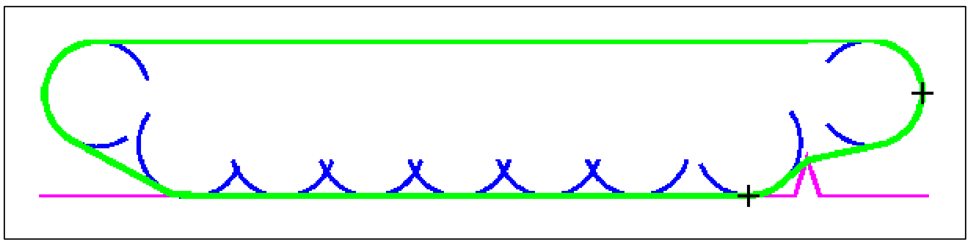


Figure D6. Track striking obstacle between SPRIDLER and road wheel.

Appendix E: Variable-Speed Model

The vehicles traveled at a constant velocity in the earlier versions of VEHDYN. In VEHDYN 4.0, the vehicle's ability to accelerate and brake has been added. The constant velocity mode has been retained and is essential to the creation of ride and shock curves. The variable-speed mode allows for the use of the model as a GO/NOGO predictor for obstacles such as mounds and ditches. The basic driving force for this modeling is an input tractive force versus speed curve (TFSC). The only required braking input is the vehicle's braking coefficient. This input TFSC is modified for wheel/track slip for each soil strength and soil type combination that the vehicle encounters on the profile. When the vehicle's running gear strikes more than one soil type and/or strength, the average strength is used for slip computations and one of the soil types is chosen arbitrarily. The computation of slip is described in Appendix C. Figure E1 shows the effect of slip on the input TFSC for pavement and a relatively soft soil.

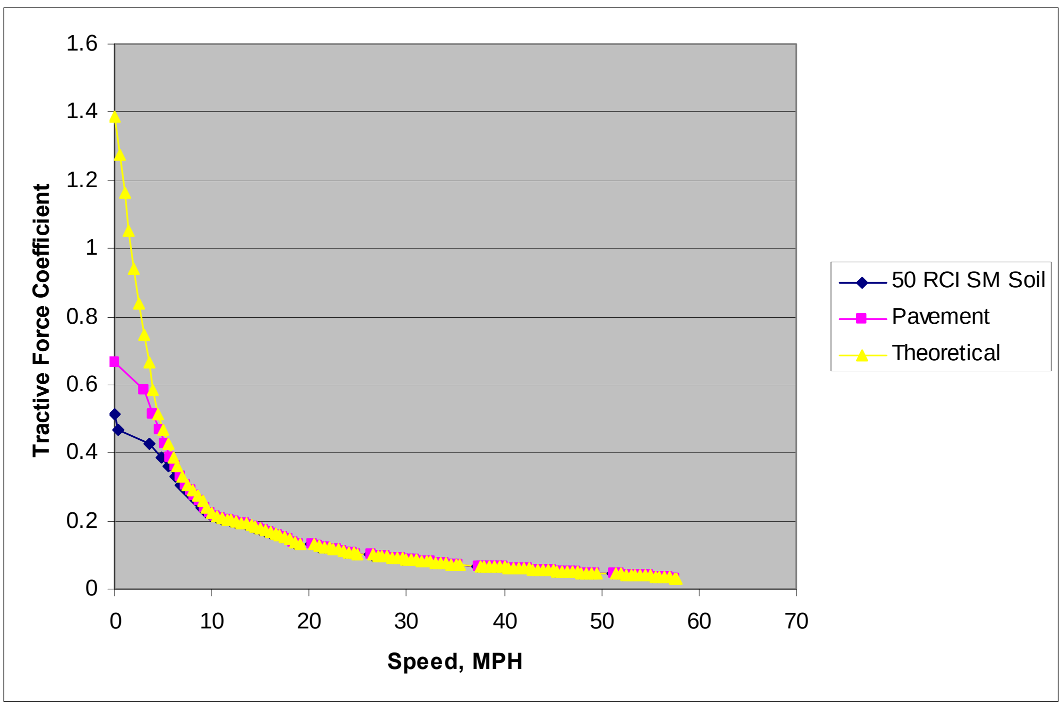


Figure E1. Tractive force versus speed curves for LMTV.

The modeling of acceleration consists of looking up the available tractive force for the vehicle's current speed and distributing this force to the drive wheels on the ground. This distribution is based on the contact pressure of each wheel. Wheels with more pressure on them will have a higher percentage of the available tractive force applied to them. If no wheels are on the ground, no forces will be added to the wheels. The hard-surface motion resistance (MR) is subtracted from the tractive force before it is applied to the drive wheels. The hard-surface MR is used regardless of the soil strength, since the soil MR is accounted for by the amount of sinkage. The resulting acceleration curves for the LMTV on the soft soil and pavement are shown in Figure E2.

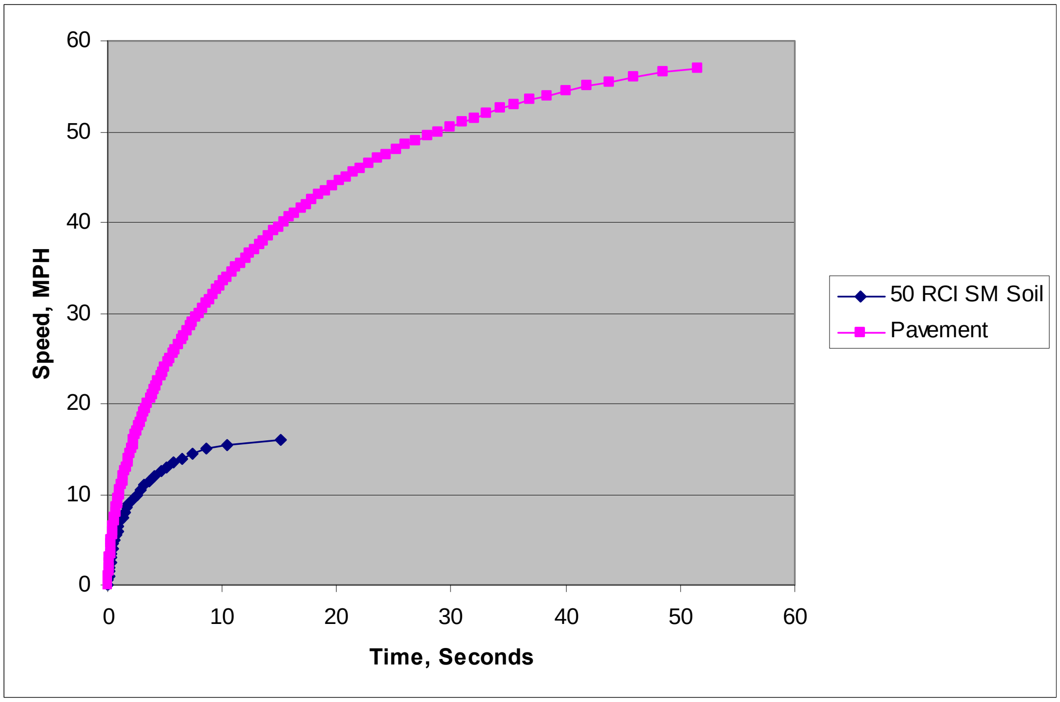


Figure E2. Acceleration curves for LMTV on two surfaces.

The slower acceleration and lower maximum speed for the LMTV on the soil is due to the increased slip and the sinkage in the soft soil. When the vehicle sinks, it in essence has to climb out of its ruts, thus producing a soil MR. When the vehicle reaches a speed greater than the objective speed, the maximum available braking force is applied to the wheels in the opposite direction. This continues until the speed falls below the objective speed. If the vehicle's speed falls to zero due to sinkage or obstacle interference, the run is deemed a NOGO.

Appendix F: Buoyancy and Drag Effect Water Model

Another effect that has been added to VEHDYN is that of the buoyancy and drag effects that occur as a vehicle crosses a volume of water. Figure F1 is a display of the pertinent vehicle and terrain parameters for this buoyancy and drag effect water model.

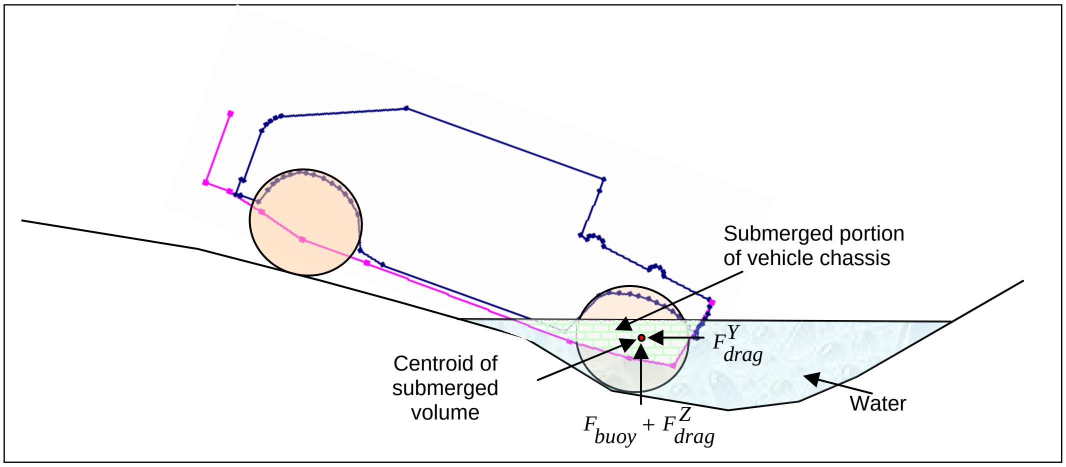


Figure F1. VEHDYN 4.0 buoyancy and drag effect water model.

In the figure, the pink line represents the data from the UID data file that define the underside of the vehicle to be used to determine vehicle-terrain interactions (as discussed in Appendix C). If the underside data are also to be used for buoyancy effects, as in Figure F1, the ends of the data must be extended upward enough such that when the first and last points are connected, a realistic closed volume is formed for the submergence activity. VEHDYN 4.0 connects the front and rear end points forming a closed volume for the submergence if the first and last points in the UID data file defining the underside are different.

As the vehicle enters a region of water, a submerged volume (the brick-pattern volume in Figure F1) begins to grow in size. Each time step, an approximate centroid is computed for this submerged volume and two forces that act on the centroid are computed. The first is the uplifting buoyancy force defined by

$$F_{buoy} = \rho_{water} \bullet Vol_{submerged}$$

The two terms on the right side of the equation are the water density and the volume of the submerged portion of the vehicle. This submerged volume is approximated by the cylinder whose cross section is defined by the two-dimensional submerged area multiplied by the vehicle's belly width. The second force is a drag resistance, having both a horizontal component F_{drag}^{Y} and vertical component F_{drag}^{Z}, which develops as the submerged volume tries to move through the water. The two components are defined as

$$F_{drag}^{Y} = \frac{1}{2}\rho_{water} \bullet C_{drag} \bullet A_{proj}^{Y} \bullet V_{Y}^{2}$$
$$F_{drag}^{Z} = \frac{1}{2}\rho_{water} \bullet C_{drag} \bullet A_{proj}^{Z} \bullet V_{Z}^{2}$$

The terms on the right side of this equation are water density, drag coefficient, horizontally (Y) and vertically (Z) projected area of the submerged portion of the vehicle, and the instantaneous CG horizontal (Y) and vertical (Z) speed components. The determination of the horizontally projected area A_{proj}^{Y} is approximated by taking the deepest submerged vehicle point and subtracting that point's vertical coordinate from the water level's vertical coordinate to get a height multiplied by the vehicle's belly width to obtain a rectangular area. The determination of the vertically projected area A_{proj}^{Z} is approximated by taking the difference between the horizontal components of the minimum and maximum submerged points on the vehicle chassis (in a horizontal sense) multiplied by the vehicle's belly width to obtain a rectangular area. The buoyancy force tends to grow as the submerged volume ($Vol_{submerged}$) grows and tends to push up the front end of the vehicle as it plows through the water. The drag force tends to both lift the front end and slow the vehicle down at a faster rate for faster moving vehicles and for vehicles with a large underwater projected area.

Also determined is the vehicle's ability to resist being swept away by a stream cross current. For each body of water, a cross-current water speed V_X is also input. As the vehicle enters the water, a water force due to the rushing water is calculated using the following equation:

$$\text{Water Force} = F_{drag}^{X} = \frac{1}{2}\rho_{water} \bullet C_{drag} \bullet A_{proj}^{X} \bullet V_{X}^{2}$$

where A_{proj}^{X} is the submerged projected vehicle area in the direction of the water current.

If the vehicle is intersecting more than one body of water during a given time step, the water forces for all interacting water bodies are summed, and the total water force is compared with the current maximum vehicle tractive resistance. If the total water force is greater than this vehicle traction, the vehicle simulation is considered to be a NOGO in the sense that the vehicle has been swept away by the water current.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
May 2009

2. REPORT TYPE
Final report

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

Enhanced Vehicle Dynamics Module

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Daniel C. Creighton, George B. McKinley, Randolph A. Jones, and Richard B. Ahlvin

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Engineer Research and Development Center
Geotechnical and Structures Laboratory
3909 Halls Ferry Road, Vicksburg, MS 39180-6199;
RBA Technical Services
3 Shadow Wood Drive, Vicksburg, MS 39180-9741

8. PERFORMING ORGANIZATION REPORT NUMBER

ERDC/GSL TR-09-8

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The vehicle dynamics module (VEHDYN 4.0) includes many enhancements to the previously documented and released VEHDYN II model and the internally developed, but not released, VEHDYN 3.0 model. The VEHDYN 3.0 model accepted the vehicle in a settled, or equilibrium, configuration instead of the "zero-force" configuration (the force acting on each suspension spring is assumed to be zero) required by VEHDYN II. VEHDYN 3.0 also included a special suspension and a band track model that were required to model the Caterpillar 30/30 engineering vehicle. VEHDYN 3.0 was further enhanced to add several other effects, including (1) a vehicle underside interference model that computes the normal and drag forces that occur when the underside of a vehicle's sprung mass comes in contact with the terrain profile, (2) a wheel damping model that computes an additional normal force component to represent the wheel's resistance to change, (3) a vehicle-terrain interface model to allow the terrain to deform, (4) accelerating and decelerating the vehicle using tractive force-speed relations and the soil-slope resistance, (5) rotational springs for enhanced suspension beam modeling, (6) a water-vehicle buoyancy and drag model to be used when a vehicle crosses a water-filled terrain obstacle, (7) numerous new run-type options including multi-run ride and shock performance sequences, slope climbing, obstacle crossing, etc., and (8) new post-processors for animation. This report documents the VEHDYN 4.0 program, including the new enhancements, the input and output files, and 12 example problems.

15. SUBJECT TERMS

Gap crossing
Mobility
Vehicle dynamics
Vehicle terrain interaction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED		138	