

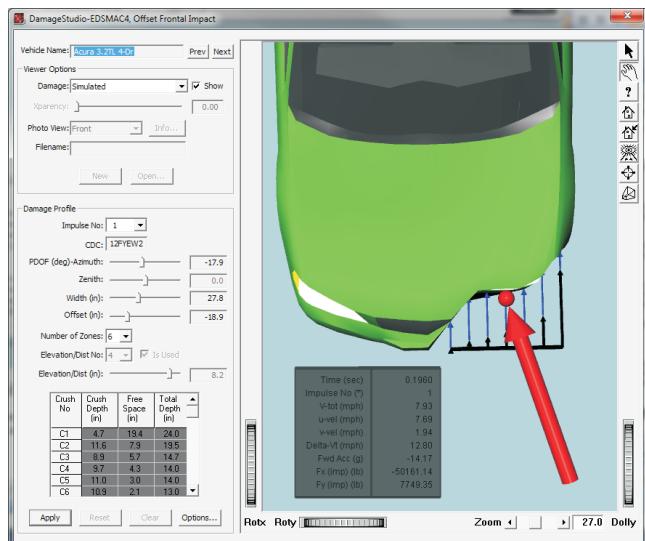
Technical Newsletter

Available on-line in the EDC Library at www.edccorp.com

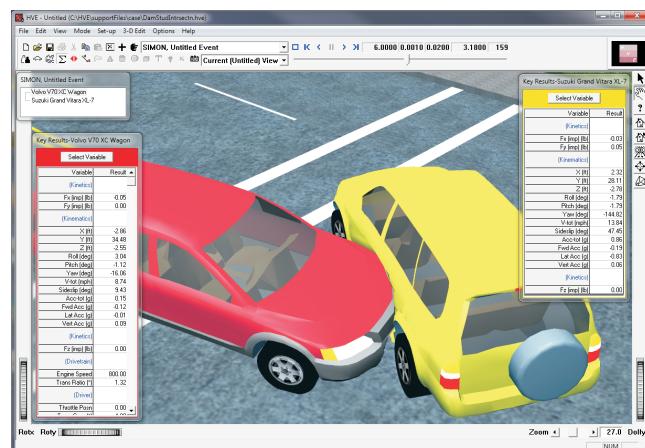
Version 8.20 - Now Available!

HVE, HVE-2D & HVE-CSI Version 8.20 were released right on schedule on June 20th! Here are a few of the new features and enhancements found in Version 8.20:

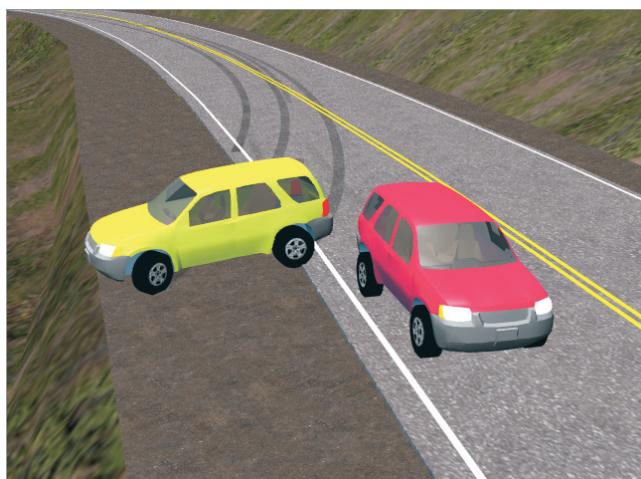
DamageStudio® for HVE-2D & EDSMAC4 - DamageStudio is now available as an option for **HVE** and **HVE-2D** users. In addition to *SIMON/DyMESH* simulations, **EDSMAC4** supports DamageStudio.



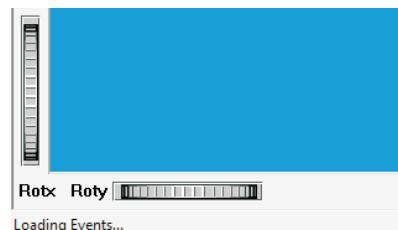
Color Matched Key Results Windows - Key Results windows now have a background color that matches the vehicle color, making it easy to quickly match a vehicle with its Key Results window.



HVE Electronic Stability Systems Model - The new **HVE** Electronic Stability Systems (ESS) Model in **HVE** Version 8.20 provides accident reconstructionists with the ability to investigate and reconstruct crashes involving vehicles fitted with traction control and electronic stability control systems.



Enhanced Status Bar Reporting - The Status Bar now displays current activity (e.g., Loading Humans, Auto-Saving File). This information can be very useful while waiting for **HVE** to accomplish a lengthy task.



More information about these and other enhancements available in **HVE**, **HVE-2D** and **HVE-CSI** v8.20 can be found at www.edccorp.com/Version820.

SAVE THE DATE!
"Laissez Les HVE Cours de Formation Rouler!"
2012 HVE Forum - New Orleans, LA
February 27 - March 3, 2012



Technical Session

Pole impact simulation is the subject of this Technical Session. *EDSMAC* and *EDSMAC4* (hereafter *EDSMAC(4)*) and *SIMON/DyMESH* (hereafter just *DyMESH*) all simulate vehicle-to-vehicle collisions, and to some degree, are amenable to vehicle-to-pole collision simulations as well. But there are issues to address when setting up and executing pole impact simulations, and in this Technical Session we will step through these issues in detail.

Barrier Basics

To keep this discussion reasonable in length, we will begin by assuming the reader is familiar with the modeling methods used by *EDSMAC(4)* and *DyMESH* to calculate collision forces and moments. If you are not, please first review the references at the bottom of the page before continuing.

Since a pole is assumed to be a rigid barrier with narrower dimensions (compared to a vehicle), our discussion begins with an overview of barrier collision simulation. The process of simulating the collision force begins by detecting an overlap between two colliding objects. Based on that overlap, the surface of each object is displaced back from its original position, and the force is calculated based on that displacement according to the mechanical (i.e., stiffness) properties of the colliding objects. Right away, you can see a problem: A rigid barrier, by definition, has an infinite stiffness, so there is no displacement.

EDSMAC

To get around this issue using *EDSMAC*, the surface must be slightly deformable. Users of *EDSMAC* will need to select a vehicle to act as our surrogate pole (*EDSMAC* assumes all objects have two axles, so we cannot start with a pole for an *EDSMAC* barrier

References:

Day, T., Hargens, R., "An Overview of the Way *EDSMAC* Calculates Delta-V," SAE Technical Paper No. 880069, Society of Automotive Engineers, Warrendale, PA 1988 (EDC Lib. Ref. No. 0004).

Day, T., "An Overview of the *EDSMAC4* Collision Simulation Model," SAE Technical Paper No. 1999-01-0102, Society of Automotive Engineers, Warrendale, PA 1999 (EDC Lib. Ref. No. 0026).

Day, T., York, A., "The *DyMESH* Method for 3-Dimensional, Multi-Vehicle Collision Simulation," SAE Technical Paper No. 1999-01-0104, Society of Automotive Engineers, Warrendale, PA 1999 (EDC Lib. Ref. No. 0024).

simulation) and increase its weight. Next, increase the K_v stiffness to a high, but tolerable, value. Experience shows that 750-1000 lb/in² is reasonable. But as a result of this increase, the collision vector adjustment length, Δp , and allowable force imbalance, λ (see *EDSMAC* Calculation Options dialog), must be modified to ensure a stable solution. The default values for Δp and λ are 0.2 inches and 15 lb/in, respectively. The relationship between K_v , Δp and λ is

$$\lambda = K_v(\max) \times \Delta p$$

where $K_v(\max)$ is the larger of the vehicle stiffnesses. To ensure a stable solution, the minimum value for λ is

$$\lambda = 1000 \text{ lb/in}^2 \times 0.2 \text{ in} = 200 \text{ lb/in}.$$

This value of λ is too large by more than an order of magnitude! We cannot simply increase λ from 15 to 200 because this would permit a potential for a huge error in force calculations (remember, λ is an allowable force imbalance; we don't want Newton to get on our case!). We can solve this problem by reducing Δp to, say, 0.02 inches (reducing Δp too much might cause more than 200 rho vector iterations, resulting in program termination). We must next adjust λ according to the above relationship,

$$\lambda = 1000 \text{ lb/in}^2 \times 0.02 \text{ in} = 20 \text{ lb/in}.$$

which is reasonably close to our default value (15 lb/in; see above). With these changes, *EDSMAC* provides a reasonable approximation for barrier collisions.

EDSMAC4

EDSMAC4 automates the above process for barriers. When a barrier is included as one of the objects (*EDSMAC4* can simulate objects with no axles, so we can select a barrier directly in the Vehicle Editor), *EDSMAC4* automatically sets the barrier A and B stiffness coefficients to 1000 lb/in and 1000 lb/in², respectively, sets Δp to 0.01 inches and increases the number of rho vector iterations from 200 to 3500. The value for λ is calculated internally,

$$\lambda = 1000 \text{ lb/in}^2 \times 0.01 \text{ in} = 10 \text{ lb/in}.$$

As one might expect, because the simulated barrier is not perfectly rigid, a small amount of deformation always occurs to a barrier in *EDSMAC* and *EDSMAC4* barrier collision simulations. The amount of deformation is smaller in *EDSMAC4* than *EDSMAC* because its force vs. displacement model uses an A+Bδ model (the presence of the A coefficient produces a large initial force with ~nil displacement).

DyMESH

In *DyMESH*, only the slave object deforms during any single timestep. This fact is used to advantage for barriers: When a barrier object is included in a simulation, it is always the master vehicle (*SIMON* swaps master and slave objects if necessary). As a result of this design, all of the issues related to barrier deformation are eliminated.

Poles

Compared to a typical vehicle or barrier, the smaller dimensions of a pole present an additional set of issues for collision simulation. Although the issues for *EDSMAC(4)* and *DyMESH* are similar, the specifics are different, so we will discuss them separately.

EDSMAC and *EDSMAC4*

The first observation is that *EDSMAC* and *EDSMAC4* assume the pole has a rectangular shape, defined by its exterior dimensions (CG to front, right, back and left). The fact that the pole is not modeled as a circular object is probably not a big issue.

EDSMAC and *EDSMAC4* both use the deformation of rho vectors to calculate the collision force. These rho vectors originate at the vehicle CG and emanate outwards radially to the object's exterior. The rho vectors are equally spaced; the default spacing between rho vectors, $\Delta\psi$, is 2 degrees.

Figure 1 shows a typical vehicle with rho vectors in the vicinity of an approaching pole. Let's assume this pole is 12 inches in diameter (more correctly, a 12 x 12 rectangle). For a typical passenger sedan the distance from the CG to the front of the vehicle is about 75-100

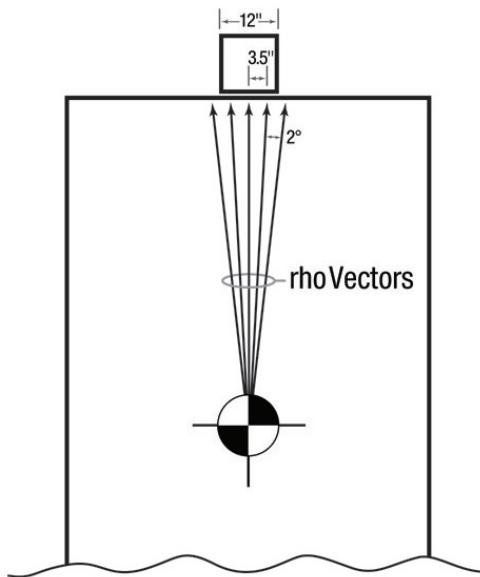


Figure 1 - Vehicle vs. pole geometrical relationship in *EDSMAC* and *EDSMAC4*

inches, maybe more. Using a 2 degree angular spacing means the linear distance, d , between two rho vectors at the front of the vehicle, assuming $x_f=100$ inches is

$$d = 2 \times X_f \sin\left(\frac{\Delta\psi}{2}\right) = 2 \times 100 \sin\left(\frac{2}{2}\right) = 3.5 \text{ inches}$$

Because the pole is 12 inches wide, only 3 rho vectors will be included in the force calculations. As you can see, a wider pole will allow more rho vectors to be included; a narrower pole will have fewer.

EDSMAC and *EDSMAC4* both allow the user to reduce $\Delta\psi$. Reducing $\Delta\psi$ to 1 degree will obviously double the number of rho vectors involved in the force calculation. But the same rho vector spacing is also used by the pole. *EDSMAC* allows a force to be applied to only 100 rho vectors. *EDSMAC* terminates if more than 100 rho vectors are required, *EDSMAC4* allows a force on up to 360 rho vectors, thus avoiding this potential program termination.

The next issue is vehicle penetration into the pole (more about pole penetration into the vehicle later). Rho vectors are not allowed to shorten to zero length. If this occurs, the run terminates. In the above example with a 12 inch diameter pole, the maximum penetration into the pole without termination is 6 inches, assuming the CG is in the middle of the pole. The use of high stiffness coefficients (see above discussion of K_v and A and B), helps to prevent termination, but success is not guaranteed considering the forces involved.

Now, let's consider the pole's penetration into the vehicle. Figure 2 shows the pole and its relationship to the vehicle's rho vectors. In cases where the pole

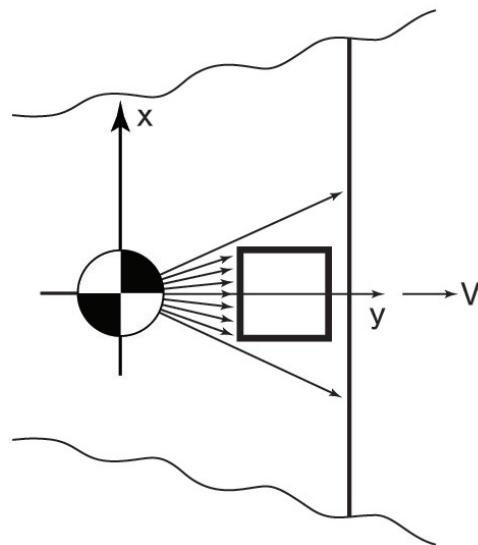


Figure 2 - Pole penetration direction aligned with vehicle rho vectors in *EDSMAC* and *EDSMAC4*

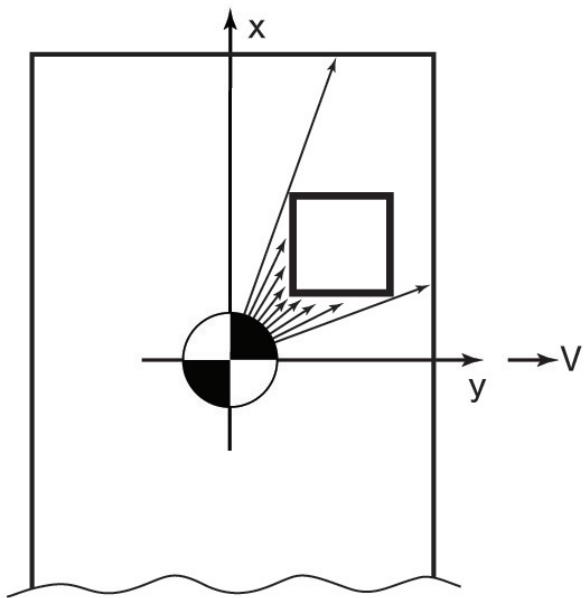


Figure 3 - Pole penetration direction not aligned with vehicle rho vectors in *EDSMAC* and *EDSMAC4*

penetration direction is aligned with the rho vectors, the model works pretty well. But consider the case of a side-slipping or rotating vehicle, where the direction of pole penetration is not aligned with the vehicle's rho vectors, as shown in Figure 3. The result is that the pole contacts the sides of the rho vectors, conceptually chopping them off. This causes a very non-physical result. The main issues here are the radial force model and the fact that the model does not include induced damage. *EDSMAC* and *EDSMAC4* are not useful for simulating this collision.

DyMESH

DyMESH uses the vertices in the vehicle mesh to calculate the collision force. The mesh may be cylindrical in shape to simulate a pole.

Each mesh vertex is assigned mechanical force vs. displacement properties. Unlike *EDSMAC(4)*, the direction of the force is not radial. Rather, the force is 3-dimensional and its direction is dictated by the direction of deformation. The distance between any two adjacent vertices is determined by the resolution of the mesh. 10 to 20 inches is not uncommon, but a generic vehicle may have 60 inches between vertices.

Figure 4 shows a typical vehicle in the vicinity of an approaching pole. Again, let's assume the pole is 12 inches in diameter. Note that there are only a few vertices in a position to interact with the pole. If the pole hit a few inches to the left, there may be no interacting vertices!

Fortunately, *HVE* provides a solution for this situation: super-tessellation. This option (see Set-up menu, Vehicle Mesh option) provides a tool that allows the

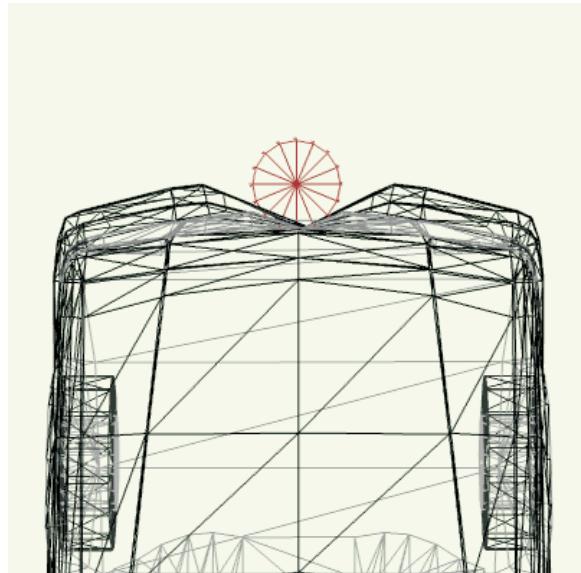


Figure 4 - Pole penetration in *DyMESH*

user to increase the number of vertices of any vehicle mesh, including generic vehicles. By setting the vehicle's super-tessellation to 2 inches in the vicinity of the pole impact location, the vehicle now has dozens (maybe even hundreds) of vertices in a position to interact with the pole, as shown in Figure 5.

The issue of vehicle penetration into the pole is eliminated because the pole never deforms (remember, barriers are always the master object in *DyMESH*).

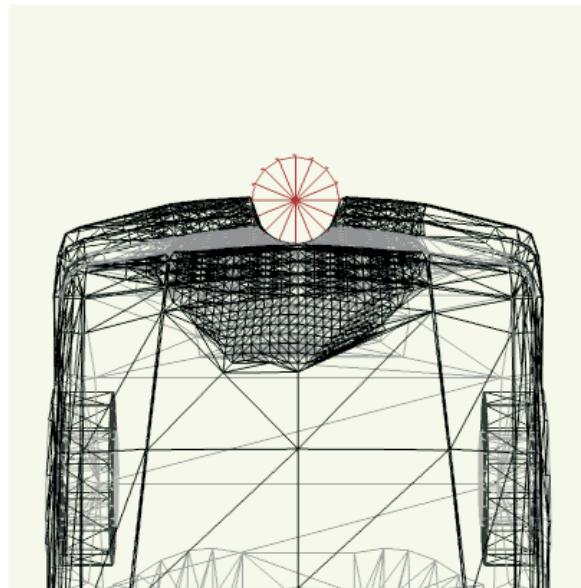


Figure 5 - Pole penetration in *DyMESH* with a vehicle using 2-inch super-tessellation

When considering the pole's penetration into the vehicle, the problem is similar to that of *EDSMAC* and *EDSMAC4*: Direct penetration normal to the undeformed surface and collisions that do not result in vehicle rotation are modeled quite well. Penetration not exceeding about one pole diameter is also modeled quite well. Beyond these cases, however, the pole tends to become entrapped by the vehicle mesh and the results become non-physical. The root issue is that *DyMESH* does not include induced damage.

Initial Contact

Consider a vehicle approaching a 12 inch diameter pole at a speed of 50 mph. That's 73.3 ft/sec, or 880 in/sec. Figure 6 shows the vehicle just before impact, when it is 1 inch from making initial contact with the pole. There is no contact, so there is no collision calculation.

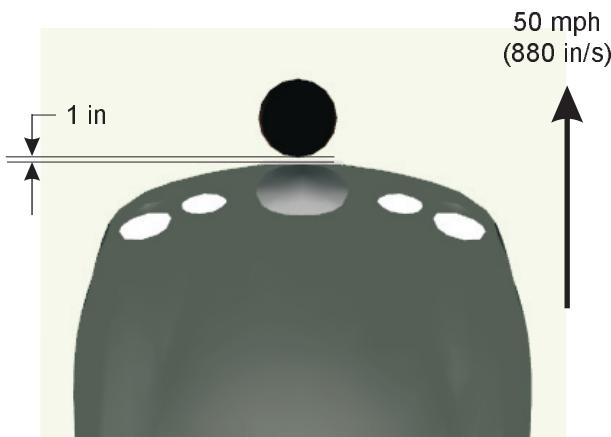


Figure 6 - Geometrical relationship of vehicle and pole one timestep before collision

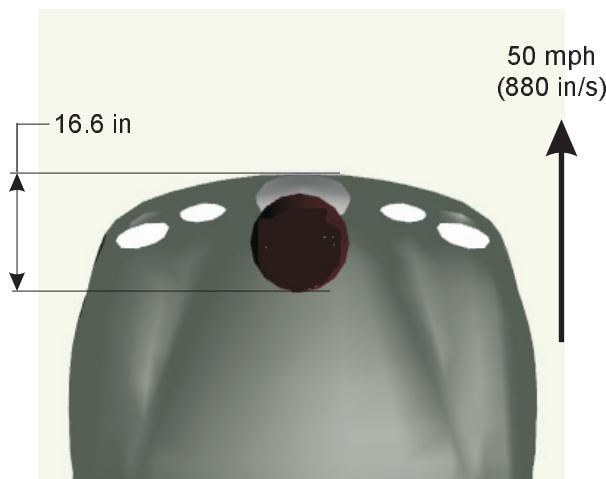


Figure 7 - Geometrical relationship of vehicle and pole one timestep later (compare with Figure 6)

For *EDSMAC* and *EDSMAC4*, the default integration timestep for the trajectory phase is 0.02 seconds. Thus, the vehicle will travel a distance of 17.6 inches during the next timestep. Figure 7 shows the result: By the time contact is identified, the vehicle has traveled completely through the pole. The run terminates for a variety of reasons (rho vector lengths, too many rho vectors, ...). This problem is easily resolved by setting the separation and trajectory timesteps equal to the collision timestep, 0.001 seconds. Using this smaller timestep, the vehicle travels only 0.88 inches per timestep, thus eliminating the problem.

It should be noted that the scenario described above is a worst-case scenario, i.e., one timestep before collision the front of the vehicle is a small distance from the pole. In the general case, the front of the vehicle may be several inches from the pole. In this case, the initial penetration into the pole that occurs during the following timestep will be much smaller and the problem described above may not occur.

The default integration timestep for a *SIMON* simulation is 0.0025 seconds, which translates into 2.2 inches per timestep. When *DyMESH* timesteps are used, the default is 0.001 seconds and the issue is eliminated.

Summary

Pole collision simulations present numerous challenges for *EDSMAC(4)* and *DyMESH*. However, they can be accomplished in some cases. These cases include:

- Pole impacts resulting in a crush depth less than one pole diameter
- Pole impacts that result in little or no rotation
- Impacts with larger poles

Set-up is important, especially for *EDSMAC*. Set-up issues include selecting appropriate stiffness and collision parameters, as well as confirming the mechanics of initial contact.

Induced damage needs to be incorporated in order to support general pole impact simulation.

These methods have not been validated against actual pole impact tests. However, if your scenario looks like a good candidate (based on the above criteria), give it a try and carefully scrutinize the results.

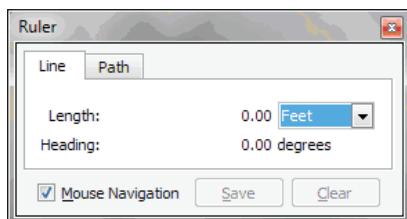
Rate This Tech Session

Please go to www.edccorp.com/TechSessionRating to tell us if you liked this Technical Session and to suggest other topics you'd like to see in future technical sessions in the EDC Technical Newsletter. Thank you!

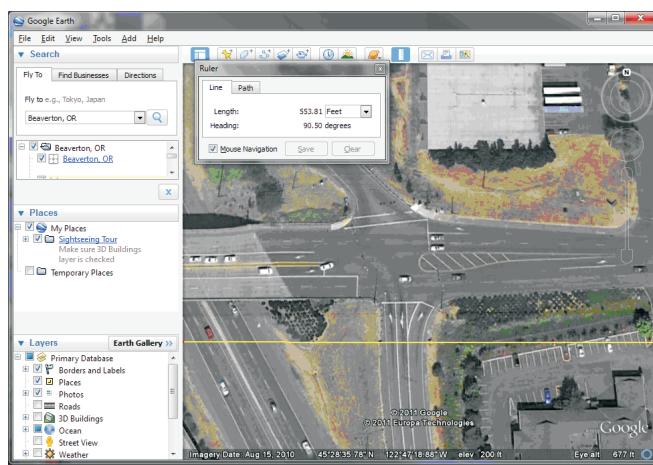
How To Get Your Google Earth Images Into HVE

The ability to quickly import an "Aerial Photo" as the environment model is a powerful feature for *HVE*, *HVE-2D* and *HVE-CS* users. Images used as aerial photos may be actual photographs of the roadway taken from an airplane or satellite, or overhead views of CAD drawings of the actual crash site. The concepts of importing an aerial photo into *HVE* are the same, regardless of the source. Here are the basic procedures for using a Google Earth image:

1. Start Google Earth and identify the location. Adjust the elevation and coordinate settings until you have the exact image you want displayed in the viewer. Make sure that you are directly overhead in order to not distort the scene when using it in *HVE*.
2. Locate and click the *Ruler* button on the Google Earth toolbar. In the Ruler dialog that appears, set the Length to report in units of Feet (or Meters).



3. Now position the mouse cursor at the left edge of the image and click the left mouse button. Move the mouse over to the right edge of the image, making sure that the line drawn by the "ruler" is as straight as possible, and click the left mouse button again. You can now read distance in the *Ruler* dialog which measures from left edge to right edge and record the distance. In this example it is 553.81 feet.

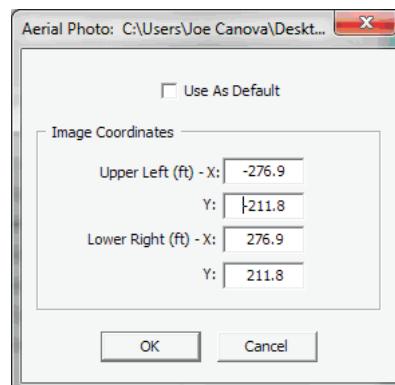


4. Repeat the steps to measure from top to bottom and record the distance. In this example it is 423.56 feet.

5. Select *File*, *Save* and *Save Image* on the main menu to save the image on your computer. This action saves the exact image that you measured in steps 3 and 4.

6. Start *HVE* (*HVE-2D* or *HVE-CS*) and proceed to the Environment Editor. Locate and click the Add New Environment (+) button on the toolbar to add an environment. Click the *Open* button on the lower portion of the Environment Information dialog to open a file browser and select the image saved from Google Earth in Step 5. (Note: It was a .jpg image, so make sure your files of type selection is set to .jpg format.) Press *Open* after selecting the image.

7. You are now prompted for Image Coordinates, which are used to size the image and identify the origin (0,0) of the environment. To center the image, divide the left to right distance in half and the top and bottom distance in half and enter those values for the coordinates, as shown below for our example. Press *OK* and *OK* once again to close the dialogs.



8. Your aerial image from Google Earth now appears in the Environment Editor and is the correct scale to match the dimensions of the vehicles used in your simulations and reconstructions.

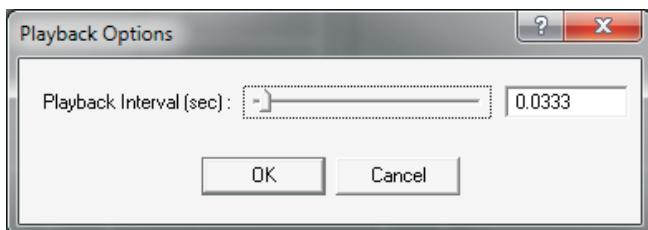


HVE and HVE-2D F.A.Q.

This section contains answers to frequently asked questions submitted to EDC Technical Support staff by HVE and HVE-2D users.

Q. I am trying to export the Variable Output report for my simulation, but the output time seems to be stuck at 0.0333 seconds. I went back to my event and changed the times in my Simulation Controls, but that doesn't seem to change the value in the Variable Output table in Playback. I want to export the values at every 0.001 second so that I can look closely at the collision pulse. How can I do this?

A. You are very close to answering this question yourself! What you would like to do is to adjust the Time Interval used in the Variable Output report. By default, the time interval used for the Variable Output and Trajectory Simulation reports and also the Playback Window in the Playback Editor is 0.0333 seconds (1/30th of a second) which corresponds with NTSC video output of 30 frames per second. To change the Time Interval from the default value, select *Options, Playback* from the main menu to display the Playback dialog as shown below. Change the Playback Interval to your desired value and press the OK button to implement your change. That's all you need to do!



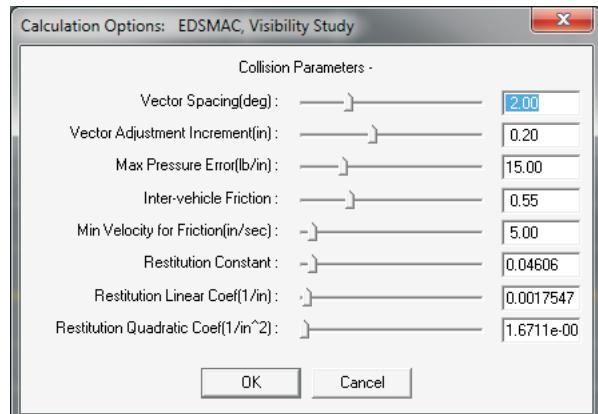
Playback Options dialog, used to assign the Playback Interval for the Variable Output and Traj Sim reports and also the Playback Window.

Q. I want to increase the inter-vehicle friction between two vehicles colliding with each other in my EDSMAC4 event. I used to find this in the Calculation Options for the event, but I don't see it listed there. Where do I make the adjustment?

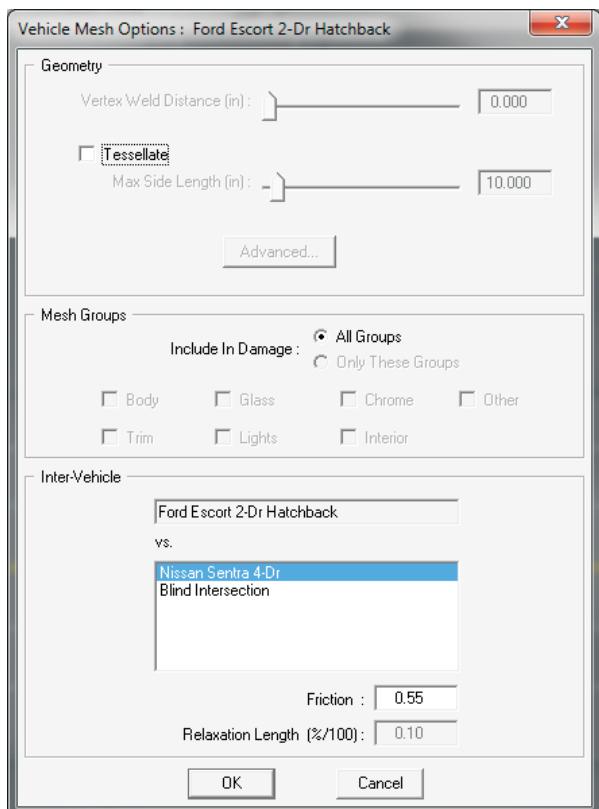
A. If you are using EDSMAC for a crash simulation, the adjustment for Inter-vehicle Friction is located by selecting *Options, Calculation Options* on the main menu. You will clearly find the adjustment of Inter-vehicle Friction in the Calculation Options dialog.

If you are using EDSMAC4 or SIMON/DyMESH, the adjustment for Inter-vehicle Friction is found by first selecting a vehicle, then selecting *Set-up, Vehicle Mesh*. The third section of the Vehicle Mesh Options

dialog allows the user to assign the Inter-Vehicle Friction value between the current vehicle and other vehicles in the simulation.



EDSMAC Calculation Options dialog, used for two-vehicle simulations



Vehicle Mesh dialog, used to assign inter-vehicle friction for multi-vehicle events, e.g., EDSMAC4

**Visit the Support section of
www.edccorp.com for the latest
Downloads and answers to F.A.Q.s**

EDC Training Courses

EDC Reconstruction & EDC Simulations

EDC offers excellent one-week courses on the use of the *EDCRASH* reconstruction program or the use of simulation programs, such as *EDSMAC*, *EDSMAC4*, *EDSVS* and *EDVTS*. The **EDC Reconstruction** and the **EDC Simulations** courses are designed to fully investigate the inner workings of the physics programs. Lectures are full of helpful hints gained from years of experience. During the course, students will use the physics programs to complete several workshops highlighting the capabilities of each program discussed in the course.

All users of *HVE* and *HVE-2D* agree that these courses are extremely beneficial and challenging. It's the fastest way to learn what you really need to know – how to effectively use the physics programs and get the right results. *Note: These courses focus on the physics programs, not on the user interface. For courses on using HVE, HVE-2D or HVE-CSI, check out the HVE Forum.*

Vehicle Dynamics

The **Theoretical & Applied Vehicle Dynamics** course extends the scope of a general vehicle dynamics discussion by including several direct applications using the *SIMON* vehicle dynamics simulation program within *HVE* and providing a solid theoretical background for such simulations. The course is focused towards engineers and safety researchers with an interest in an understanding of vehicle dynamics and automotive chassis systems development.

Engineering Dynamics Corporation Training Course Schedule

EDC Simulations

Los Angeles, CA January 2013
Miami, FL November 7 - 11, 2011

EDC Reconstruction

Los Angeles, CA January 23 - 27, 2012
Miami, FL November 9 - 13, 2012

Theoretical & Applied Vehicle Dynamics

Upon Request

2012 HVE FORUM

New Orleans, LA Feb 27 - Mar 3, 2012

HVE Forum

The **HVE Forum** offers workshops designed to help *HVE*, *HVE-2D* and *HVE-CSI* users improve their modeling and application skills. By participating in workshops, attendees learn new techniques and also how to use the latest advancements in the software. The *HVE* Forum is also a great opportunity to meet other users and expand your network of resources.

Course Registration

To register for a course, download a registration form from the Training page at edccorp.com or contact EDC Customer Service at 503.644.4500 or by email to training@edccorp.com. All courses are eligible for Continuing Education Units and ACTAR credits.

HVE Training Partners

HVE, *HVE-2D* and *HVE-CSI* users looking to improve their skills, but unable to attend one of EDC's regularly scheduled courses, can contact an *HVE* Training Partner for assistance. *HVE* Training Partners are experienced *HVE* and *HVE-2D* users who offer introductory and custom training courses on the use of *HVE*, *HVE-2D*, *HVE-CSI* and compatible physics programs.

HVE Discussion Groups

Websites hosted by experienced *HVE* Users offer information about using *HVE* as well as moderated online discussions with other users. Be sure to visit:

Yahoo - tech.groups.yahoo.com/group/HVErecon - Discussion group hosted by Roman Beck of Beck Forensics, Inc.

DiscoverHVE.com - Online training and discussion group hosted by Wes Grimes of Collision Engineering Associates

Engineering Dynamics Corporation
8625 SW Cascade Blvd, Suite 200
Beaverton, Oregon 97008 USA
Phone 503.644.4500 / FAX 503.526.0905
Email: info@edccorp.com
Website: www.edccorp.com

EDCRASH, *EDSMAC*, *EDSMAC4*, *EDSVS*, *EDVTS*, *EDHIS*, *EDVDS*, *EDGEN*, *EDVDB*, *HVE*, *HVE-2D*, *HVE-CSI*, *HVE* Brake Designer and *GetSurfaceInfo()* are trademarks of Engineering Dynamics Corporation. All Rights Reserved.

SIMON, *DyMESH* (Patent number 6,195,625) and *DamageStudio* are registered trademarks of Engineering Dynamics Corporation. All Rights Reserved.

GATB, *GBF*, *DiscoverHVE*.com and *PhotoIntoHVE* are trademarks of Collision Engineering Associates, Inc.