Mechanical Simulation
755 Phoenix Drive, Ann Arbor MI, 48108, USA
Phone: 734 668-2930 • Fax: 734 668-2877 • Email: info@carsim.com

GPS Coordinates

Using GPS in CarSim, TruckSim, and BikeSim

The global positioning system (GPS) is a worldwide, satellite-based technology that provides precise location and timing information to any device capable of receiving GPS signals. GPS location information has two uses in BikeSim, CarSim, and TruckSim:

1. Creating paths and roads.
2. Reporting a vehicle or moving object's location.

Creating Paths and Road from GPS Data

GPS location data is commonly given in the form of latitude, longitude, and altitude. This can be measured with a GPS receiver (such as in a smartphone) or exported from online mapping services. A typical smartphone GPS receiver is accurate, on average, to 4.9 meters (van Diggelen, 2015). For this reason, GPS is often not the most accurate way to measure a road or track the motion of a vehicle. However, the accuracy of GPS is acceptable for many uses.

Mechanical Simulation Corporation has a web application, *Atlas*, located at `atlas.carsim.com`, which may be used to retrieve GPS data for public roads. Both VS Scene (`.vsscene`) and CSV (comma-separated values, `.csv`) filetypes containing GPS location data are available from Atlas. The primary difference is that the VS Scene file includes altitude while the CSV file does not.

GPS data in the form of a CSV file may be imported into CarSim on a **Path: X-Y Coordinates**, **Path: X-Y Coordinates for Segment**, or **Scene: External Import** screen. The latter also supports GPS data in the form of a VS Scene file. Datasets created with these screens may be used to build roads, such as by linking them from a **Road: 3D Surface (All Properties)** screen.

GPS Output Variables

For vehicles and moving objects, output variables for latitude, longitude, and altitude are available. This is like having a real-life GPS receiver located in a vehicle. This could allow, for example, a CarSim vehicle's trajectory to be saved and overlaid on a virtual map using suitable software.

How do BikeSim, Car Sim, and TruckSim work with GPS data? The VS math models used in these products have equations based on Newtonian physics, with the modeling assumption of a uniform gravitational field. The multibody equations of motion include the calculation of locations using the rectangular coordinate systems defined in ISO 8855 and JSAE 670, where the Z-axis is upwards, and the curvature of the Earth is ignored. To convert GPS coordinates to VehicleSim coordinates, or vice-versa, the directions are aligned so positive X is east and positive Y is north. The Z coordinate is directly given by altitude.

Both import and output calculations of GPS data rely on a designated GPS reference point. This value is key to the coordinate conversion calculations, as it allows a change in latitude or longitude to be converted to a change in meters in the Y or X, directions, respectively. This is done by using the radii of Earth in those directions, at that reference point. If desired, the reference point can be manually defined. The reference point is specified with the parameters `GPS_REF_LAT`, `GPS_REF_LONG`, and `GPS_REF_ALT`, with default values for the location of the Mechanical Simulation offices in Ann Arbor, Michigan (42.231° north latitude, -83.727° east longitude, 250 meters elevation).

To improve the accuracy of the coordinate conversions, the GPS reference point will reset when the path (in the case of import) or the vehicle (in the case of output) travels a certain distance from the reference point (5000 meters north or south). This is the case even when an initial reference point has been manually defined.

> **Note** Roads composed of multiple GPS paths, which require the same reference point to be manually defined for each individual path, should be relatively small in geographic area for best results. This prevents the reference point from being inadvertently updated.

An in-depth description of the GPS coordinate conversion is provided below, along with a subsequent section with observations of the effects of measurement precision on GPS coordinates.

Technical Details

The conversion method makes use of some properties of the WGS 84 system, described below.

WGS 84

In addition to the system of satellites themselves, calculation of precise coordinates for GPS applications requires a definition of a standard world geodetic system. This the most recent (2004) revision of the *World Geodetic System* (WGS 84), originally developed by the United States Department of Defense. WGS 84 consists primarily of a coordinate system, an ellipsoid, and a geoid (National Imagery and Mapping Agency, 2000).

The coordinate system is an Earth-centered, Earth-fixed Cartesian coordinate system. This coordinate system is used to formulate equations which are solved to locate the GPS receiver in space. To relate the resulting receiver coordinates to latitude and longitude requires a reference ellipsoid.

To this end, WGS 84 provides a precise description of an ellipsoidal shape that very closely approximates the oblate shape of the Earth. Latitude — strictly-speaking, *geodetic latitude* — and *longitude* locate a point on the surface of the ellipsoid. In general, a GPS receiver does not lie on the surface, so it is necessary to add a third parameter, *geodetic height*, to give the receiver's position with respect to the reference ellipsoid.

The geodetic height, which is the height of the GPS receiver, is not a height above sea level. This is because the ellipsoid is a poor model of Earth's mean-sea-level (MSL). Rather, MSL is modeled by a surface called a geoid, which is practically given as the height of MSL relative to the reference ellipsoid at any given point. WGS 84 uses the EGM96 geoid, with more details in (F. G. Lemoine, 1998). The geoid allows more accurate reporting of height above MSL (elevation) than otherwise possible.

Altogether, WGS 84 provides the tools needed for a GPS receiver to solve for its location and then report that location in terms of the familiar latitude, longitude, and elevation (altitude).

GPS Coordinate Conversions

Note	Details of the GPS to X-Y coordinate conversion are given from the perspective of importing the data. For output, the inverse of this calculation is performed.

Path data, used to create roads, is input as GPS coordinates of geodetic latitude ϕ, longitude λ, and, optionally, elevation H along the path. In order to use a path in this form, it is necessary to convert the GPS coordinates into Cartesian coordinates X, Y, and Z. This conversion happens automatically when a GPS path is imported. The conversion relies primarily on the reference point, which is defined by the parameters `GPS_REF_LAT`, `GPS_REF_LONG`, and `GPS_REF_ALT` (the last referring to altitude, also known as elevation). The default values of these correspond to the first point of the input path.

The GPS coordinate conversion relies on the WGS 84 ellipsoid to model Earth's surface. This ellipsoid is defined by its semi-major axis $a = 6378137.0$ meters and polar flattening $f = 1/298.257223563$ (National Imagery and Mapping Agency, 2000). Other useful characteristics of an ellipsoid are (Rapp, 1991):

- Semi-minor axis $b = a(1 - f)$
- First eccentricity $e = \sqrt{a^2 - b^2}/a$
- Radius of curvature in the meridian $M = a(1 - e^2)/(1 - e^2\, sin^2\, ϕ)^{3/2}$
- Radius of curvature in the prime vertical section $N = a/(1 - e^2\, sin^2\, ϕ)^{1/2}$
- Radius of curvature of the parallel $p = N\, cos\, ϕ$.

The meridian is the section of Earth through the given reference point and the two poles (north/south section). The parallel is the section of Earth of constant latitude which passes through the given reference point (east/west section).

The assumption that Earth is flat is justifiable on the basis that M and p at a given reference point are very large — millions of meters — as a straight line as infinite radius. Radius of curvature in the meridian M remains quite large for all values of latitude ϕ, but radius of curvature of the parallel p becomes relatively small as latitude approaches ±90° (as the reference point approaches the poles, sections of constant latitude become smaller in radius). Thus, the validity of this flat-Earth assumption weakens toward the poles.

Because Earth is treated as flat, a change in latitude Δϕ (radians) from the reference point is considered a change in Y-coordinate. The change in Y-coordinate may be computed from the meridian arc length s for the particular change in latitude Δϕ. A true meridian arc length formula is quite involved; it is typical to derive approximations appropriate for the desired accuracy (Rapp, 1991). Here, perhaps the simplest-possible approximation is used:

$$s \approx M\,\Delta\phi.$$

This approximation is analogous to the familiar expression for arc length on a circle ($s = r\theta$, where r is the radius of the circle and θ is the angle, in radians, which the arc subtends at the center of the circle).

Similarly, a change in longitude $\Delta\lambda$ (radians) is considered a change in X-coordinate. In this case, the parallel is a circle — the WGS 84 ellipsoid is a surface of revolution — and so the parallel arc length L is given exactly by

$$L = p\Delta\lambda.$$

Lastly, a change in altitude ΔH from the reference point is considered a change in Z-coordinate, with the reference point having Z equal zero. There is some approximation error in X and Y which become clear from stating this assumption; this locates the reference point on the surface of the WGS 84 ellipsoid, when in fact the reference point should be slightly above the ellipsoid. Additionally, as stated in the previous section, the ellipsoid is not at MSL. In the case where elevation along the path is not provided, the Z-coordinate is set to zero for the entire path.

The GPS reference point resets when the change in Y-coordinate from the current reference point exceeds a magnitude of 5000 meters. This updates the radii used to convert a change in latitude or longitude to Y or X, improving the accuracy of the conversion.

Ultimately, there is no canonical conversion from GPS coordinates to Cartesian coordinates. The assumptions made here are intended to allow a practical implementation with an appropriate level of accuracy for vehicle dynamics simulations.

GPS Outputs

GPS coordinates may be output from simulation using the following output variables:

- `GPSlongA`. Absolute longitude (relative change plus reference point value).
- `GPS_Altitude`. Absolute altitude (relative change plus reference point value).
- `GPS_Lat`. Relative latitude.
- `GPS_LatA`. Absolute latitude (relative change plus reference point value).
- `GPS_Long`. Relative longitude.
- `Zo`. Z-coordinate of vehicle’s sprung mass origin (the relative altitude of the vehicle).

Similar outputs are available for moving objects. The output calculation is the inverse of the import. The practicality of inverting the chosen GPS to Cartesian conversion is a point in its favor.

When no GPS data has been imported, it is still possible to output GPS coordinates of a vehicle or moving object provided a reference point is given. As noted earlier, the default values of `GPS_REF_LAT`, `GPS_REF_LONG`, and `GPS_REF_ALT` are the location of the Mechanical Simulation offices in Ann Arbor, Michigan (42.231° north latitude, -83.727° east longitude, 250 meters elevation).

Note GPS reference coordinates `GPS_REF_LAT`, `GPS_REF_LONG`, and `GPS_REF_ALT` do not appear in a VS GUI unless a GPS path is being imported. To set a different reference location when only GPS output is needed, type the name of each reference parameter followed by the location in decimal degrees or meters into a miscellaneous data field. After making a run, you can verify the reference location (or find out what reference location was used in a run previously made) by opening the Echo file with initial conditions with the View button on the Run Control screen.

Additional parameters associated with the GPS outputs and viewable in an Echo file are

- `GPS_RANGE_Y`. Change in Y-coordinate from initial value allowed before updating the reference point.
- `GPS_REF_X`. The reference point's X value.
- `GPS_REF_Y`. The reference point's Y value.

> **Note** For path or scene import, the parameters `GPS_RANGE_Y`, `GPS_REF_X` and `GPS_REF_Y` appear in an Echo file to verify the values associated with the GPS coordinate conversion. If they are modified, such as with VS Commands, the GPS output coordinates of the vehicle or moving object will not necessarily align with the original road data. For this reason, these values should not be overwritten unless no GPS data has been imported.

Output-Format Precision

Bear in mind that one degree of longitude (at the equator) or latitude is over 110 km, so GPS outputs show only very small relative changes. The number of digits available with a 32-bit floating-point number provides a resolution of several centimeters in North America, Japan, and Europe (the exact resolution depends on latitude). A 64-bit format provides much better resolution. For this reason, output files are best specified using the VS file format with 64-bit data. With this choice, the GPS latitude and longitude angles can be viewed with acceptable precision for any application. For example, Figure 1 shows the path of a vehicle in CarSim for a simulated double lane change. The two plots show data from two files: one with 32-bit data (baseline) and one with 64-bit data. Note the effect of the limited precision when using data from the 32-bit file (Baseline, red line).

Internally, all calculations in the VS Math Model are performed with 64-bit variables and calculations, so the calculated vehicle behavior is identical. Most outputs are represented OK in the legacy 32-bit ERD file. For example, Figure 2 shows identical vehicle trajectories using the X-Y coordinates rather than GPS coordinates. For applications involving third-party tools like mapping and navigation software that use GPS data, the limited precision of 32-bit data might be acceptable. If not, then be sure to set the output file format to be **Output: VS 64-bit**.

Prior to the release of version 9 for CarSim and TruckSim, all output files followed the 32-bit ERD output file that has been in use since 1984. In order to provide GPS coordinates within this limit, VS Math Models in past versions provided relative GPS outputs `GPS_Lat` and `GPS_Long`, defined as:

```
GPS_Lat  = GPS_LatA - GPS_REF_LAT
GPS_Long = GPSlongA - GPS_REF_LONG
```

The improvement associated with this approach is clear in Figure 3.

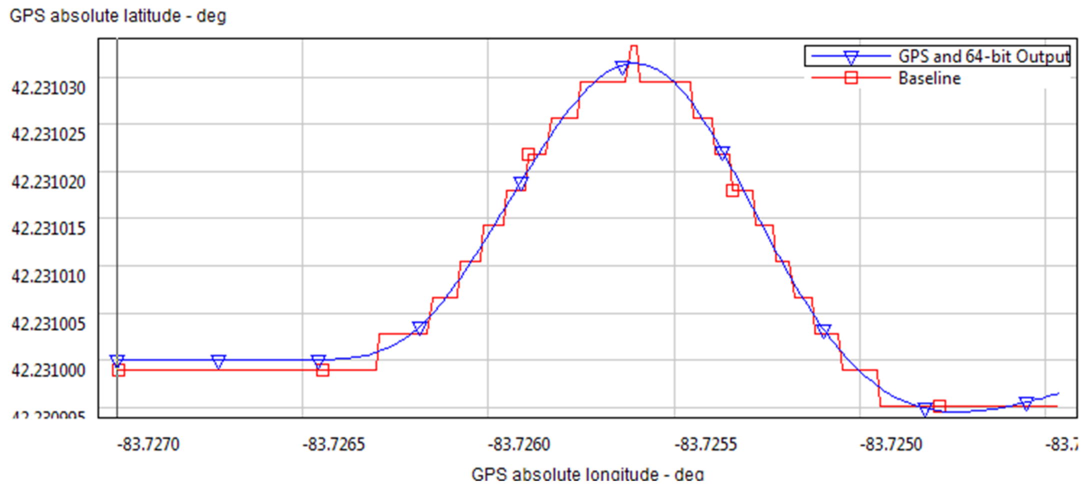

Figure 1. Vehicle trajectory scaled using absolute GPS angles using 32-bit and 64-bit numbers.

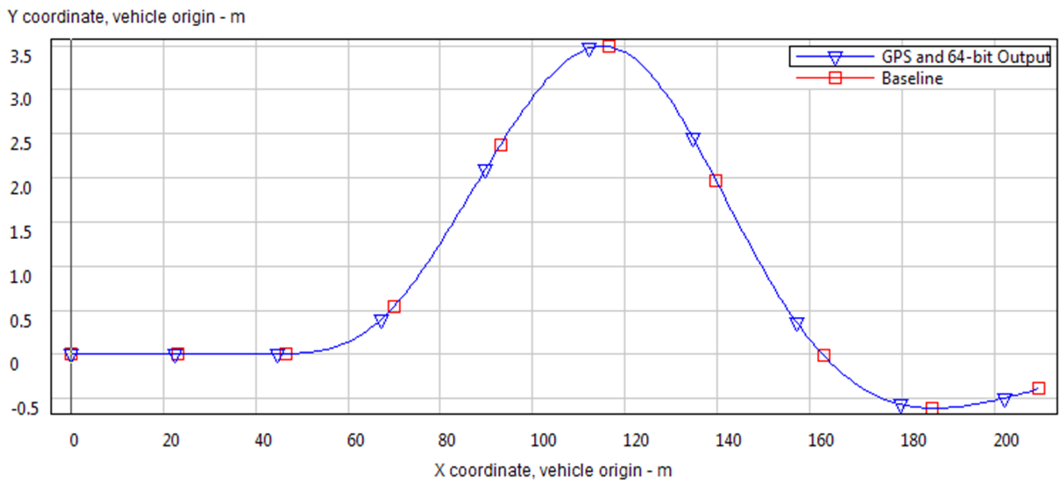

Figure 2. Vehicle trajectory scaled with multibody model X and Y coordinates.

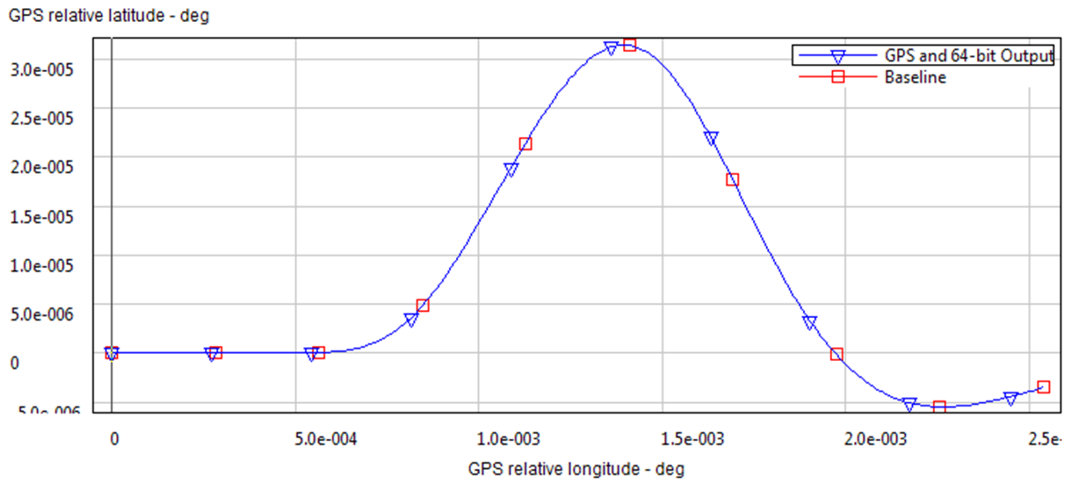

Figure 3. Vehicle trajectory scaled using relative GPS angles using 32-bit and 64-bit numbers.

References

F. G. Lemoine, S. C. (1998). *The Development of the Joint NASA GSFC and NIMA Geopotential Model EGM96.* Greenbelt, MD: NASA Goddard Space Flight Center.

National Imagery and Mapping Agency. (2000). *Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems.* National Imagery and Mapping Agency, Geodesy and Geophysics Department, St. Louis, MO.

Rapp, R. H. (1991). *Geometric Geodesy Part I.* Columbus, OH: The Ohio State University, Department of Geodetic Science and Surveying.

van Diggelen, F. E. (2015). The World's first GPS MOOC and Worldwide Laboratory using Smartphone. Tampa, FL: Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015).