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

Numerical Integration in VS Math Models

The basic operation of a VehicleSim (VS) Math Model is described in the *VS Math Models Reference Manual*. This tech memo provides more detail about the numerical integration calculations performed within a VS Math Model.

In a VehicleSim product, a simulation is executed when the VS Math Model performs calculations at closely spaced intervals of time, using equations that define a math model. The current state of the model is defined by a set of independent variables called *state variables*. Traditionally, the state variables for a multibody vehicle model have been defined exclusively with differential equations. Indeed, many of the state variables in a VS model are defined by a set of ordinary differential equations (ODEs).

Note Additional variables are defined algebraically to "remember" conditions such as friction force. They are needed to define the state of the model, and are therefore considered state variables, but they do not have associated differential equations.

Machine-generated documentation from VS Math Models lists all state variables; those associated ODEs will include "ODE" in the text description and are called ODE state variables in documentation.

VS Math Models have been set up to work with several numerical integration algorithms that are described in this memo. The memo also discusses sources of error that involve communication

between the basic model and other systems running simultaneously, such as other software or hardware that is being tested through real-time hardware-in-the-loop (HIL) simulation.

Overview of Numerical Integration Methods

The ODE state variables are copied to an array q. To simplify the following descriptions, the symbol q is used for a single ODE state variable. Subscripts are used to indicate a value of q in a sequence of values calculated by numerical integration.

In theory, each ODE state variable is a continuous function of time: q(t). However, in a simulation run, calculations are made at discrete intervals of time. Figure 1 shows the history for a state variable q at discrete values of time separated by a constant step ΔT, where ΔT is the time-step specified with the parameter `TSTEP`. At each step, identified with a count i, there is a corresponding value of q_i and its derivative $\dot{q}(t)$, indicated as q'_i in the figure.

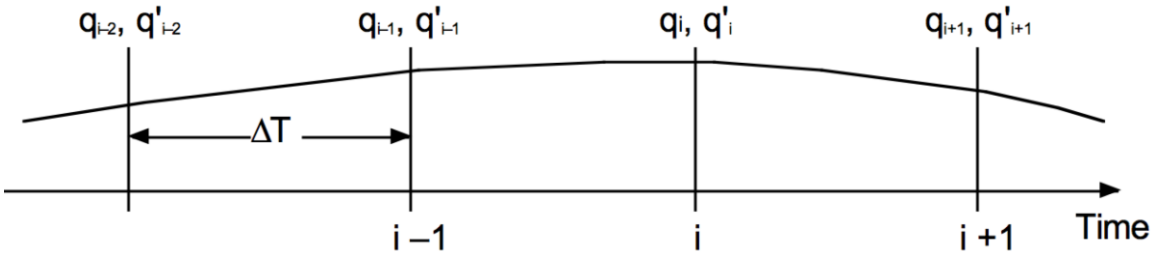

Figure 1. Discrete samples of a state variable q and its derivative q'.

The VS Math Model computes the values of each output variable in the VS Math Model at time t, using current values of the state variables and possibly their derivatives, where each derivative is calculated from an ODE in the VS Math Model. The ODE state variables are in turn calculated using numerical integration.

The correct value of q(t) at the next interval (i+1) is the integral of $\dot{q}(t)$:

$$q_{i+1} = q(T_i + \Delta T) = q_i + \int_{T_1}^{T_1+\Delta T} \dot{q}(t)\,dt \qquad (1)$$

where $\dot{q}(t)$ is a continuous function of q(t) and other variables in the VS Math Model, plus variables that may be imported into the math model from other software or measurements from HIL. The behavior of the many variables that influence $\dot{q}(t)$ are not known over the interval. Therefore, approximations are used to calculate the new value of q_{i+1} via numerical integration.

The simplest method of numerical integration, called Euler integration, is to project the last known derivative forward to estimate the new value of q:

(Euler) $$q_{i+1} = q_i + \Delta T\, q'_i \qquad (2)$$

Euler integration does not account for any change in the derivative over the time interval and is not accurate unless ΔT is so small that changes in the derivative are negligible over the step.

Many numerical integration methods are available for solving equations of motion for multibody systems. Overall, a method is satisfactory if it can calculate all the ODE state variables with acceptable accuracy over the range of time needed by the user, with an acceptably fast computation

time. Most example simulation datasets provided with VehicleSim products calculate new values for all variables with an internal time step of 0.0005s. This gives acceptable computation speed (faster than real-time for BikeSim, CarSim, and most TruckSim models) and valid numerical results. (By "valid numerical results," we mean that the same results are obtained as when a much smaller time-step is used.) The parts of the model that are most likely to have numerical errors if the time-step is too large are the wheel spin rates with the vehicle running at low speeds, and possibly some powertrain spin rates.

Note When a numerical error is reported, it is usually because some internal numbers have huge values that cause failure of subsequent calculations involving the vehicle position on a road (S and L coordinates cannot be found) or calculations of angles in the steering system model (kingpin constraint equations).

If these types of errors are reported when attempting a simulation, you should try running with a smaller time-step. In addition, set the output interval to match the calculation time-step, to view potentially unstable oscillations in plots, as described later (page 7).

Although the time-step for updating the calculations is typically 0.0005s in VehicleSim products, the variables are often written to file at a larger interval, such as 0.025s (40 Hz). Here, the "best" time-step is one that is as large as possible while allowing all the dynamic behavior of interest to be viewed in plots and animations. Using a larger time-step means that files are smaller, and post-processing tools show results more quickly because fewer numbers are handled. On the other hand, the output sample interval can be set to match the calculation interval for validation and debugging analyses, as will be discussed later.

Following are some of the factors that are involved in different methods of numerical integration.

Order of the Method

For certain sinusoidal functions, the error involved in the calculation of q_{i+1} is proportional to ΔT^n, where n is the "order" of the method. Euler integration is a first-order method. Most methods in common use are second-order or higher. If the differential equations are continuous (smooth) with many levels of derivatives, then higher-order methods can be more efficient. However, if the equations involve discontinuities or changes that violate the assumptions underlying the higher-order methods, then higher-order methods are better.

Discontinuities can be a significant part of the math model, when on-off changes are triggered. These include abrupt gear changes, clutch engagement/disengagement, ABS controllers, etc.

Other discontinuities are subtler. For example, table lookup functions may be set to use straight-line interpolation between points. The transition from one region in a table to another can abruptly change slope and derivatives in ODEs, violating the assumptions of the higher-order integration methods. Another form of discontinuity occurs with HIL when sampled measurements change abruptly, also violating the assumptions of higher-order methods.

The equations in most VS Math Models work best with methods of order 2 with the datasets that are included with a typical installation.

Fixed and Variable-Step Methods

The time-step ΔT associated with numerical integration is varied in some methods as a simulation run proceeds. When variables change rapidly, a small time-step might be needed to control numerical error; when variables change slowly, a larger time-step can be used to reduce the computation time. When a variable-step integration algorithm determines that the time-step should be cut, it will typically discard current values and go backward in time to perform new calculations. This can cause problems with some user-supplied equations that are written under the assumption that numerical integration always proceeds forward.

In general, a fixed time-step is preferred when merging equations from different sources (e.g., CarSim and Simulink) to avoid problems with synchronization and initialization. A fixed time-step is usually necessary when running in real-time with HIL, to maintain consistent communication with hardware.

All integration methods provided in VS Math Models use fixed time-steps.

Implicit and Explicit Methods

Implicit methods use iteration to confirm that the calculated values of the state variable and their derivatives are consistent. The implicit methods are helpful for "stiff systems" with high-frequency dynamics that exist as unwanted side effects of some modeling methods. The extra computation needed at each time-step to iterate several times and obtain consistent values for the state variables and their derivatives allows much larger time-steps to be used, with the result that the overall amount of computation is reduced.

In general, implicit methods do not work well for real-time HIL systems. As with variable-step methods, implicit methods can also complicate the merging of equations from different sources. However, implicit methods are required for some classes of equations, such as differential-algebraic equations (DAE) that are used with some numerical multibody programs (e.g., ADAMS).

Explicit methods have a clear sequence of calculations that rely only on current and past values of q and q', such as Euler's method (equation 2).

The equations for vehicle models in VehicleSim products are well-behaved ODEs that do not require the extra complexity of implicit solution methods. Accordingly, all integration methods provided in VS Math Models are explicit.

Sequence of Derivative Calculations (Predictor/Corrector Options)

Most explicit fixed-step methods involve calculations of predicted variables other than the state variables at the specified time-steps. For example, a set of state variables might be predicted for use in computing derivatives that are then used to obtain better estimates of the variables at the new time.

Some methods involve repeating calculations at the same point in time. For example, calculate an estimate of q_{i+1}, q^e_{i+1}, which is then used to calculate a derivative $\dot{q}^e_{i+1}$, which is then used to calculate a corrected (more accurate) value of q_{i+1}. Other methods advance time in smaller

increments, such as a half-step ΔT/2. For example, one of the most popular numerical integration methods is a 4th-order Runge-Kutta method that calculates derivatives twice at each full step and twice at each half-step. However, this method is not compatible with real-time systems because it is not possible to jump back in physical time to repeat a measurement for a second iteration.

VS Math Models support several predictor/corrector methods. All are so-called "real-time formulations" in which all calculations of derivatives are performed with increasing values of time that can be synchronized to measurements made in real time.

Numerical Integration Methods

VS Math Models support six methods for numerical integration, specified with the parameter `OPT_INT_METHOD`. Two of these calculate derivatives once per time step, and the other four calculate derivatives twice per time step.

Methods That Calculate Derivatives Once Per Time Step

The following two methods performs all calculations at the main time-step ΔT, with no additional calculations

Euler 1st Order Method

The Euler equation presented earlier is repeated here:

(Euler) $$q_{i+1} = q_i + \Delta T\, q'_i \tag{2}$$

Overall, the Euler method is the least accurate of the methods available in a VS Math Model. It is provided in support of advanced users who might want this benchmark method for testing and diagnosing problems.

The Euler method is only method presented that does not assume any continuity, as is the case for all the other methods that use saved derivatives. If abrupt changes in variable derivatives are expected (as with brake pressure subject to ABS controller), this method may be helpful in validation and other research activities.

Adams-Bashforth 2nd Order Method

The Adams-Bashforth second-order method uses one old derivative to predict some curvature and provides better accuracy than Euler integration when ODE derivatives are continuous.

(AB-2) $$q_{i+1} = q_i + \frac{\Delta T}{2}(3q'_i - q'_{i-1}) \tag{3}$$

AB-2 is recommended as the first choice in VehicleSim products. The recommended value for the time-step ΔT is 0.0005s in most cases.

The only time that this method is not recommended is when connecting the VS Math Model to external models (Simulink, LabVIEW, FMI, etc.) in which communication does not have to be rapid, and a slower communication rate (e.g., 0.001s) is preferred. In these cases, the AM-2 method described in the next section is recommended.

Methods That Calculate Derivatives Twice Per Time Step

Four of the methods involve calculations at a half-step, at interval i+1/2 (Figure 2). With these methods, the values produced at the half-step i+1/2 are predictions that are not as accurate as the corrected values produced at the full step i+1.

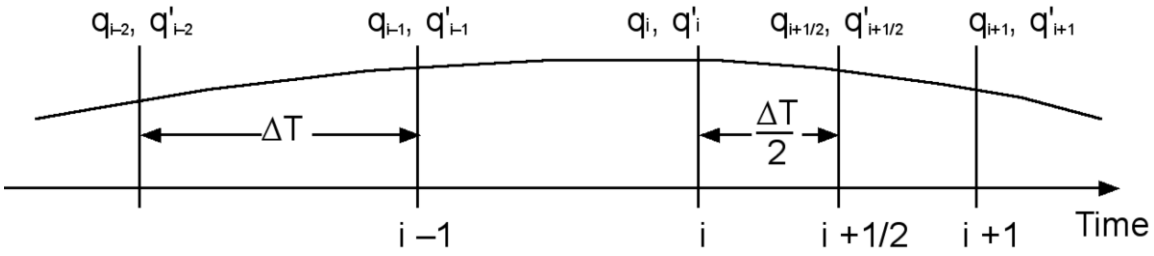

Figure 2. Use of a half-step for numerical integration.

Adams-Moulton 2nd Order Method

VS Math Models support three Adams-Moulton methods. In the second-order method, the derivatives from the previous step are used to account for some curvature in predicting the state variables at the half-step. The two equations are:

(AM-2)
$$q_{i+1/2} = q_i + \frac{\Delta T}{8}(5q'_i - q'_{i-1}) \quad (4)$$

$$q_{i+1} = q_i + \Delta T\, q'_{i+1/2} \quad (5)$$

This method is recommended when the external code is mainly providing controls at a lower frequency. In this case, communication with the external code can be set to occur only at the full step, improving the speed of the external code. For example, when `TSTEP` is set to 0.001s, this is the frequency that the VS Math Model communicates with the external code. Internally, the internal ODE calculations are made at 0.0005s.

On the other hand, if the external code requires high-frequency communication with the VS Math Model, the AB-2 method (described earlier) is recommended.

> **Note** If an AM or RK two-step method is chosen, a system parameter `OPT_IO_UPDATE` can be used to communicate with external software at intervals of `TSTEP`/2. This option is not recommended; if variables are to be exchanged rapidly, then the AB-2 method should be selected (with half the time step).

Adams-Moulton 3d Order Method

The third-order version accounts for more curvature by going back two steps. The equations are:

(AM-3)
$$q_{i+1/2} = q_i + \frac{\Delta T}{24}(17q'_i - 7q'_{i-1} + 2\,q'_{i-2}) \quad (6)$$

$$q_{i+1} = q_i + \frac{\Delta T}{18}(20q'_{i+1/2} - 3q'_i + q'_{i-1}) \quad (7)$$

This method gives performance like the AM-2 method. It can be slightly better for conditions that have high-frequency vibrations, and low-level maneuvers in general. On the other hand, simulations of maneuvers that involve controls or road geometries with abrupt changes in slope can run more efficiently with AM-2.

Adams-Moulton 4th Order Method

The fourth-order version goes back three steps to provide more accuracy if the variables change smoothly over several steps. The equations are:

(AM-4)
$$q_{i+1/2} = q_i + \frac{\Delta T}{384}(297q'_i - 187q'_{i-1} + 107\,q'_{i-2} - 25q'_{i-3}) \tag{8}$$

$$q_{i+1} = q_i + \frac{\Delta T}{30}(36q'_{i+1/2} - 10q'_i + 5q'_{i-1} - q'_{i-2}) \tag{9}$$

This method does not work as well with most vehicle simulations as the AM-2 or AM-3. However, there might be cases where the extra nonlinearity will help. This method is provided in the VS Math Models if some conditions are simulated that are better suited to a higher-order method.

Runge-Kutta 2nd Order Method

This Runge-Kutta method is like the AM-2 method. The equations are:

(RK-2)
$$q_{i+1/2} = q_i + \frac{\Delta T}{2}q'_i \tag{10}$$

$$q_{i+1} = q_i + \Delta T\, q'_{i+1/2} \tag{11}$$

Notice that equations 5 and 11 are identical. However, instead of using derivatives from a previous time-step (equation 4), RK-2 uses Euler integration to estimate the state variables at the half-step (equation 10).

This method is sometimes called “real-time Runge-Kutta” because it always marches forward in time, unlike other Runge-Kutta methods that perform corrections of the derivatives at a repeated point in time.

Because it does not use old calculated derivatives, RK-2 is less affected by discontinuous inputs, and might be preferable over AM-2 when dealing with sampled data in HIL systems or slope discontinuities in tables.

Time-step for Output Files (Plotting and Animation)

The time step ΔT is set in a VS Math Model using the system parameter `TSTEP`.

Be aware that `TSTEP` is associated with the validity of the numerical integration method. In most vehicle simulations, it should not be set larger than about 0.002s for the half-step methods such as AM-2; 0.001s is the recommended value for half-step methods, and 0.0005s is the recommended value for the single-step methods Euler and AB-2.

The time interval used for writing outputs to a VS, ERD, or CSV file for plotting and animation is a multiple of `TSTEP`, specified in the VS Math Model with a system parameter `IPRINT`. For example, if `IPRINT` is set to 40 and `TSTEP` is 0.001, then the time interval for the output file is 0.04s (25 Hz). When running from the VS GUI, you normally specify the output frequency or time-step, and the VS Browser automatically calculates `IPRINT` and writes the value in the Parsfile that

will be read by the VS Math Model. If the simulation involves only low-frequency behavior, then the output frequency can be lowered to reduce the size of output files. For example, set `IPRINT` = 100 to generate outputs with a 0.1s time-step (10 Hz).

It is sometimes helpful to see values for every time-step when diagnosing unexplained results. If you set `IPRINT` = 1, then the output time-step matches the specified integration time-step. However, if you are using a half-step method such as AM-2, this will not include calculations made at the half step. To include the half-step calculations, set `IPRINT` = 0. When this parameter is set to zero, the VS Math Model will write to the output file at smallest interval possible.

If the simulation involves slow changes (e.g., a series of quasi-static conditions as done for steady-turning analysis), there are options for controlling the writing to output files independently of the integration time-step. A system parameter `OPT_WRITE` specifies whether variables are being written to file; it can be changed during the simulation with VS Commands to skip writing during transitions that are of no interest.

Working with External Models or HIL

VS Math Models can be used alone or together with other software, such as Simulink, LabVIEW, and FMI. Import and export variables are transferred between the VS Math Model and the external model and/or HIL every time-step, and possibly at the half-step.

The math model equations within a VS Math Model are *causal*: they are organized such all of the calculations at any time-step are performed in a valid sequence. However, when parts of the vehicle model are replaced with components from external software, the organization might become *acausal*. The practical effect is that some variables used in the model might have values that were calculated in a previous time-step. The effect on the overall model behavior is negligible if the time-step is very small compared to the dynamic response time. However, if the response time is also quick, then the delay of a time-step can degrade the validity.

Four conventions are available:

1. If either the AB-2 or Euler method is selected, then variables are exchanged at intervals of `TSTEP` (there is no half-step). Note that the time-step for AB-2 or Euler must usually be shorter than the time-step for the AM and RK methods by about a factor of two.

2. When the AB-2 or Euler method is selected, then two options are available for the timing of how import and export variables are updated for communication with other software. If the system parameter `OPT_IO_SYNC_FM` = 0, then the values of time within the VS block (e.g., VS S-Function, VS FMU, etc.) is fixed for the entire step. However, if `OPT_IO_SYNC_FM` is nonzero, then the communication with external software occurs in the middle of the time step, such that the external software receives a preview of the kinematical output variables for the next time step, relative to variables that involve forces and accelerations. This is recommended when parts of the multibody model are defined externally that calculate force/moment values based on kinematical information, such as a tire model, springs, etc. The forces and moments imported to the VS Math Model are synchronized to the rest of the model.

3. If an AM or RK method is chosen and the system parameter OPT_IO_UPDATE = 0, then variables are exchanged at intervals of TSTEP. The VS Math Model performs calculations at the half-step for internal use, but those values are not shared. This option is helpful if the external model cannot run with a time step of 0.0005s (a fixed time step of 0.001s is a common standard for vehicle ECU hardware/software).
4. If an AM or RK method is chosen and the system parameter OPT_IO_UPDATE is nonzero, then variables are exchanged at intervals of TSTEP/2. This option is no longer recommended; if variables are to be exchanged rapidly, then the AB-2 method should be selected (with half the time step).

The Echo file generated for each simulation run includes a calculated parameter T_DT that is the interval between calculations. This is set automatically to either TSTEP or TSTEP/2, depending on the integration method. This parameter is provided for advanced users who may use it in VS Commands.

Each of the first three conventions has potential advantages that depend on the properties of the external model and/or hardware.

When the dynamics in the external model and/or hardware involve high frequencies that match the dynamics in the VS Math Model, then the AB-2 method is recommended, with a shorter time-step than would be used with other methods (e.g., 0.0005s).

When the external model includes a physical part of the vehicle (tire, steering, powertrain, etc.), the second option can sometimes maintain the proper sequence of calculations, maintaining the causality of the model.

When the external model cannot communicate at a rapid frequency, then an AM method can be used, in which the VS Math Model uses half-step calculations internally but communicates at the full step interval TSTEP.

The fourth convention supports rapid communication using the AM and RK methods. However, in most these cases, the AB-2 method is recommended with TSTEP set to a smaller value (e.g., 0.0005s).

Changing State Variable Values During a Run

About half of the state variables in a VS Math Model are defined with ODEs; other state variables are needed to define conditions of interest in the model, such as whether a clutch is locked or slipping, or the current value of a force that includes friction hysteresis.

ODE State Variables

ODE State variables are given initial values at the start of the run; all future values are normally determined by integrating the differential equations. However, there are occasions where it can be useful to reset some of the state variables during a run, overriding the values obtained by numerical integration. For example, several test conditions can be simulated in one run by resetting a vehicle position for the start of the next test, such that it jumps instantly to a new position. One way to do this is by using VS Events that read new settings from one or more Parsfiles that will be loaded

when the Event is triggered. Another way is to add equations with VS Commands that assign new values to state variables.

Resetting state variables during a run can be tricky, and is considered an advanced technique. There are at least two theoretical inconsistencies that might be introduced:

1. If an AM or RK method is used (with half-step calculations), the VS Math Model discards values of the state variables computed at the half-step, after the half-step derivatives are calculated. This means that any replacement values for state variables are lost, and values of state variables for the end of the time-step are based on values at the previous full step, combined with the saved derivatives.
2. The VS Math Model keeps track of previous values of the derivatives of all state variables. The saved values of the derivatives will still be applied as described in equations 3 – 11, even if the current values of state variables are no longer linked to the recorded histories of the derivatives.

Both effects only apply when equations are added at runtime to replace values of state variables.

The first effect means that a correction made to state variables at the half-step calculations would be lost. For this reason, the VS Command `EQ_FULL_STEP` should be used to add equations that reset state variables. Equations added with this command are only applied at the full step. Another option is to use VS Events, which are only applied at the full step.

The second effect is less significant; it depends on the integration method (it applies only to the AB-2 and AM methods), and ends as soon as the old saved derivatives have been replaced in one or two time-steps.

State Variables Calculated Directly

In addition to the ODE state variables, a VS Math Model has other variables that are needed to define the state of the model, but which are not defined with ODEs. Instead, these are calculated directly.

The VS Math Model also includes hundreds (or potentially thousands) of output variables for use in post-processing, such as plotting and animation. These output variables are all available for exporting to other software, such as Simulink.

Whenever the VS Math Model needs derivatives for the ODEs, values are calculated for three groups of variables:

1. ODE state variables are calculated by numerical integration using the values from the most recent full time-step plus values of their derivatives as calculated at one or more previous time-steps, including half-steps.
2. Other state variables are calculated directly using the current values of (potentially) all model parameters and all state variables (including themselves).
3. Output variables are calculated directly using the current values of (potentially) all model parameters and all state variables.

A distinction is that state variables calculated by numerical integration (group 1) are calculated based on values from the most recent full time-step. Values of these variables calculated at the half-

step are used to calculate derivatives and other variables (groups 2 and 3), and are then discarded. On the other hand, values of variables from groups 2 and 3 are handled the same at the half-step as they are at the full step.

Summary and Recommendations

VS Math Models have two system parameters for defining the numerical integration method: `TSTEP` (time-step) and `OPT_INT_METHOD` — an integer parameter that specifies a method as listed in Table 1.

Table 1. Numerical integration methods available in VS Math Models.

Name	Method	Order	Half-Step	Old Steps	When to Use
Euler	-1	1	No	0	Research and diagnostic applications.
AB-2	0	2	No	1	Recommended as default method with `TSTEP` = 0.0005s whenever possible.
AM-2	2	2	Yes	1	When using external models that cannot update at 0.0005s. In these cases, set `TSTEP` to 0.001s. (The VS Math Model will then calculate variables internally at 0.0005s steps.)
AM-3	3	3	Yes	2	For continuous conditions where a larger time step might be possible.
AM-4	4	4	Yes	3	For scenarios with exceptional continuity.
RK-2	1	2	Yes	0	For scenarios with discontinuities that degrade AM-2 performance.

In the table, AM indicates an Adams-Moulton method, AB is Adams-Bashforth, and RK is Runge-Kutta. “Method” indicates the value of the system parameter `OPT_INT_METHOD`. “Order” is the theoretical relationship between the time-step and numerical error for smooth equations, e.g., the error varies as ΔT^2 for a second-order method. “Old Steps” indicates the number of old sets of derivatives used to account for curvature in the variables, assuming smooth and continuous equations. The three methods whose names are shown in bold are the one that are used most often.

The Euler method is not recommended for routine simulation but is sometimes useful for research and diagnosing problems with a model.

The AB-2 method is recommended as the default, with `TSTEP` = 0.0005s. If there is ever doubt about the validity of a time-step, then you can duplicate the top-level **Run Control** dataset and choose a smaller time-step (e.g., divide by 2) and repeat the run. If the plots overlay with acceptable agreement (typically about 4 significant digits), then the original time-step was OK. If not, consider using a smaller time-step for the vehicle and/or procedures.

When running with Simulink or other external simulation environments and the intent is to communicate as frequently as possible, the AB-2 method provides each set of export variables from the VS Math Model has the same level of accuracy.

In general, small time-steps are needed when simulating tests that involve the vehicle starting from zero speed or braking to a stop, or when including effects of high-speed controllers. When operating only at normal road speeds (20 km/h and up), larger time-steps (e.g., 0.001s for AB-2) are usually adequate.

The AM-2 method is recommended with `OPT_IO_UPDATE` = 0 when using external models that do not need rapid communication, or which are required to run at 0.001s (a common standard for controller models). Note that when `TSTEP` is set to 0.001s, the VS Math Model calculates variables internally every 0.0005s (the half step).

The higher-order methods (AM-2, AM-3) might sometimes allow better computational performance (a larger time step) for simulations with continuous conditions.

Tips for Diagnosing Instability in VS Math Models

If model parameters are set to define a numerically unstable system, the model can generate numerical values that are huge or undefined. Three likely messages are:

1. Error calculating path S-L coordinates from global X-Y coordinates.
2. Error in table lookup, usually because the simulation generated invalid numbers such as `nan` (not a number) or `inf` (infinity).
3. Hitch Deflection limits are exceeded. This error involves translational springs in trailer hitches but is often caused by bad choices in parameter values unrelated to the hitch.

When the error occurs due to numerical instability, here are several steps to take in diagnosing the problem.

- Set the parameter `IPRINT` to 0 (use the Miscellaneous yellow field on the **Run Control** screen) to plot outputs with the smallest sampling intervals.
- Check the math model time step size. Recommendations are 1000 Hz (0.001s) for AM-2 integration and 2000 Hz (0.0005s) for AB-2 integration, but smaller time steps might be needed for some vehicle descriptions. If the simulation can proceed with smaller time-steps, that implies the system is numerically stiff. Either use the smaller time step or find the parameter settings that make the system unstable (usually the problem is masses or inertias that are too small, or springs that are too stiff).
- Check to see if the data for the vehicle and procedure make sense.
- If the cause of the failure is not obvious, create a vehicle with 50/50 weight distribution, suspension kinematics and compliances set to zero, steering kingpin geometry parameters set to 0, etc., then run the vehicle on a flat, level road and check for the occurrence of error.
- Incrementally add detail back into the vehicle until the error occurs.

Revision History

The options for numerical integration have changed over the history of Mechanical Simulation. Following are the major changes made in the numerical integration methods and communication with external software (e.g., Simulink).

Early versions (prior to version 6.0, 2004): VS Math Models used only the RK-2 method. Communication with Simulink or other external software was done at the major time step. E.g., if the time step was specified as `TSTEP` = 0.001s, the communication with Simulink would be at 0.001s intervals.

Version 6 (2004): The parameter `OPT_IO_UPDATE` was added to support more frequent communication with Simulink and other external software.

Version 7.0 (2007): VS Commands and the VS API were added to provide more options for extending the math model, including the additions of equations applied along with the built-in equations.

Version 7.01 (2007): The parameter `OPT_INT_METHOD` was added, in support of Adams-Moulton methods (AM-2, AM-3, and AM-4) and the Adams-Bashforth 2nd order method (AB-2). Unfortunately, the AB-2 option had a bug and users were soon advised not to use it.

Version 7.1 (2008): The bug in AB-2 was fixed.

Version 2016.0 (2016): During development of this version, it was discovered that the AB-2 method had been inadvertently disabled at some time, such that it was actually applying the Euler method. The Euler integration method was added as an explicit alternative.

Version 2018.1 (2018): The parameter `OPT_IO_SYNC_FM` was added to eliminate lag when using external software (e.g., Simulink) to replace parts of the vehicle multibody model such as tires, powertrain, etc. All documentation describing numerical integration or communicating with external software was updated. In reviewing existing usage of the software, the documentation and existing examples in shipping databases were modified to recommend two settings for most conditions, repeated here:

1. The AB-2 method with time step of 0.0005s (2000Hz) is recommended for all setups except when using external software where this 2000Hz frequency is not possible.
2. The AM-2 method with time step of 0.001s (1000Hz) is recommended when using external software when the external software cannot update at 2000Hz. Be sure the parameter `OPT_IO_UPDATE` is set to zero in this case.