*RAIL

Two keywords are defined in this section.

*RAIL_TRACK

*RAIL_TRAIN

*RAIL_TRACK

Purpose: Wheel-rail contact algorithm intended for railway applications but can also be used for other purposes. The wheel nodes (defined on *RAIL_TRAIN) represent the contact patch between wheel and rail. A penalty method is used to constrain the wheel nodes to slide along the track. A track consists of two rails, each of which is defined by a set of beam elements.

Card Sets. For each track include one pair of Cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	BSETID1	NORGN1	LCUR1	OSET1	SF1	GA1	IDIR
Type	I	I	I	I	F	F	F	I
Default	none	none	none	↓	0.0	1.0	0.0	0

Card 2	1	2	3	4	5	6	7	8
Variable		BSETID2	NORGN2	LCUR2	OSET2	SF2	GA2	
Type		I	I	I	F	F	F	
Default		none	none	↓	0.0	1.0	0.0	

VARIABLE	DESCRIPTION
ID	Track ID
BSETID1,2	Beam set ID for rails 1 and 2 containing all beam elements that make up the rail; see *SET_BEAM.
NORGN1,2	Reference node at one end of each rail, used as the origin for the roughness curve. The train will move in a direction away from this node.
LCUR1,2	Load curve ID (see *DEFINE_CURVE) defining track roughness (vertical displacement from line of beam elements) of the rail as a function of distance from the reference node NORIGIN. Distance

VARIABLE	DESCRIPTION
	from reference node is the x-axis of the curve while roughness is on the y-axis. Default: no roughness.
OSET1,2	Origin of curve LCUR is shifted by distance OSET towards the reference node.
SF1,2	Roughness values are scaled by SF. Default: 1.0.
GA1,2	Shear stiffness of rail per unit length (used to calculate local rail shear deformation within each beam element). GA = shear modulus × cross-sectional area. Default: local shear deformation is ignored.
IDIR	Contact forces are calculated in local directions relative to the plane containing the two rails at the contact point. IDIR determines which side of the plane is "up", that is, the direction in which the wheel can lift off the rail. "Up" is either **c** or −**c**,where **c** = **a** × **b**. **a** is the direction along rail 1 heading away from node NORGN1 and **b** is the vector from rail 1 to rail 2. Both **a** and **b** are determined locally. EQ.0: Whichever out of **c** or −**c** has a positive global Z component is "up" (default). EQ.1: −**c** is "up". EQ.-1: **c** is "up".

Remarks:

*RAIL_TRACK and *RAIL_TRAIN were written by Arup to represent wheel-rail contact. They have been used to generate loading on models of bridges for vibration predictions, for stress calculations, and for estimating accelerations experienced by passengers. Other non-railway uses are possible: the algorithm causes the "train" nodes to follow the line defined by the "rail" beam elements and transfers forces between them. In some cases (especially vibration modeling), double precision versions of LS-DYNA may give superior results because of the small relative deflections between wheel and rail.

Track Modeling:

The rails of the track should be modeled by two parallel lines of beam elements. The track can be curved or straight and the rails can be modeled as deformable or rigid. If required, rail pads, sleepers and ballast may also be modeled – typically with spring, damper and beam elements. It is also possible to use this algorithm to control the motion

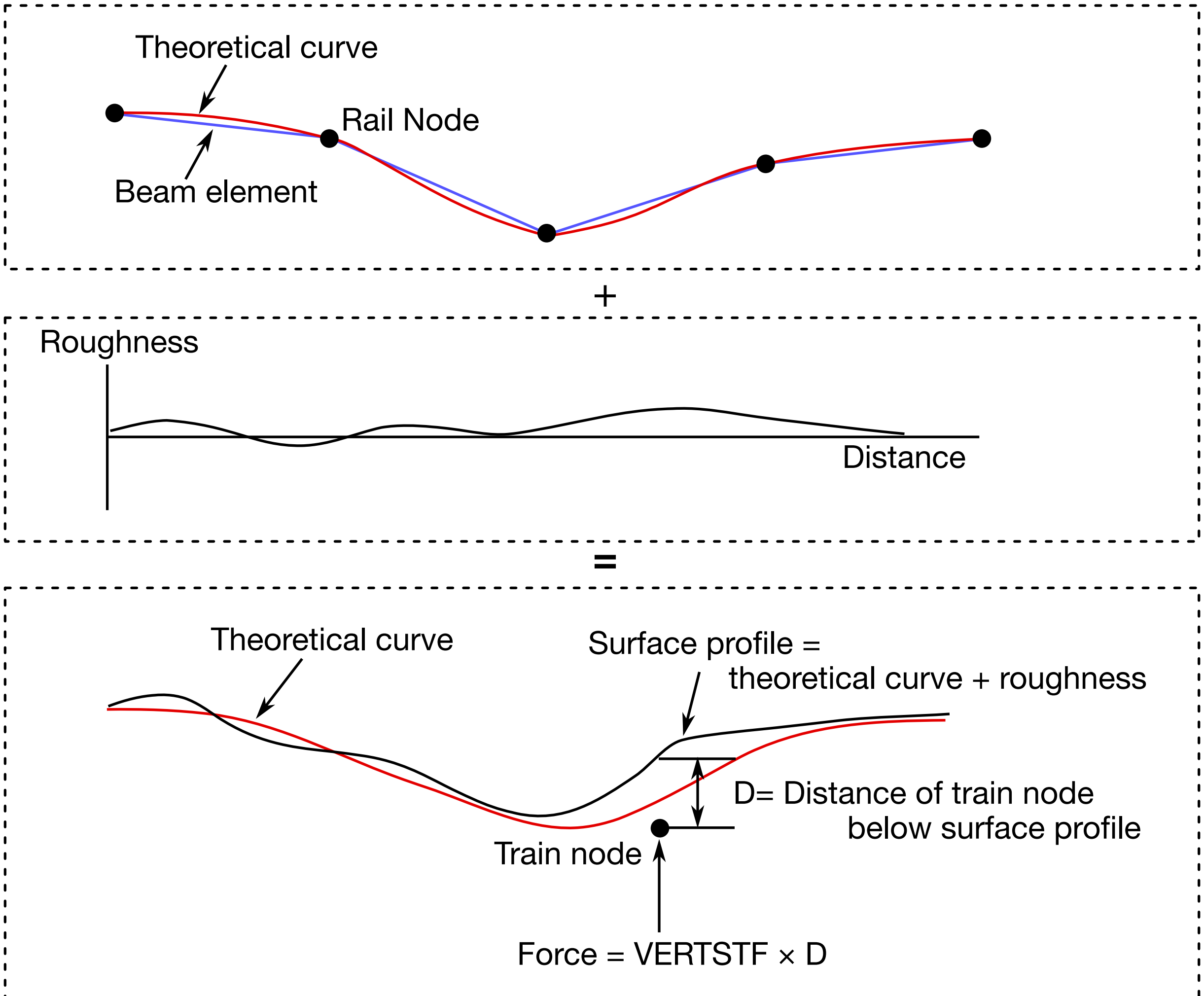

Figure 39-1. Track Model

of simple road vehicle models: beam element "rails" made of null material can be embedded in the road surface. It is recommended that the mesh size of the two rails should be similar: LS-DYNA calculates a local coordinate system for each train node based on the alignment of the currently contacted beam element and the nearest node on the other rail.

Because wheel-rail contact stiffness is generally very high, and wheel masses are large, small deviations from a straight line or smooth curve can lead to large transient forces. It is recommended that great care be taken in generating and checking the geometry for the track, especially where the track is curved. Some pre-processors write the coordinates with insufficient precision to the LS-DYNA input file which can cause unintended roughness in the geometry. For the same reason, if the line of the track were taken as straight between nodes, spurious forces would be generated when the wheel passes from one rail element to the next. This is avoided because the *RAIL algorithm calculates a theoretical curved centerline for the rail element to achieve continuity of slope from one element to the next. Where the length of the rail elements is similar to or shorter than the maximum section dimension, shear deformation may be significant, and it is possible to include this in the theoretical centerline calculation to further reduce spurious forces at the element boundaries (inputs GA1, GA2).

Roughness (small deviations in the vertical profile from a perfect straight line) does exist in real life and is a principal source of vibration. *RAIL allows the roughness to be modeled by a load curve giving the vertical deviation (in length units) of the rail surface from the theoretical centerline of the beam elements as a function of distance along the track from the origin node of the rail. The roughness curve is optional. Ideally, roughness profiles measured from both rails of the same piece of track should be used so that the relationship between bump and roll modes is correctly captured.

Whether roughness is included or not, it is important to select as the origin nodes (NORIGIN1 and NORIGIN2) the nodes at the end of the rails away from which the train will be traveling. The train can start at any point along the rails but must travel away from the origin nodes.

Train Modeling:

The vehicles are typically modeled using spring, damper and rigid elements, or simply a point mass at each wheel position. Each node in the set referred to on *RAIL_TRAIN represents the contact patch of one wheel (note: not the center of the wheel). These nodes should be initially on or near the line defined by either of the two rails. LS-DYNA will move the train nodes initially onto the rails to achieve the correct initial wheel-rail forces. If the results are viewed with magnified displacements, the initial movements can appear surprising.

Wheel roughness input is available. This will be applied in addition to track roughness. The input curve must continue for the total rolled distance – it is not assumed to repeat with each wheel rotation. This is to avoid problems associated with ensuring continuity between the start and end of the profile around the wheel circumference, especially since the profiles might be generated from roughness spectra rather than taken directly from measured data.

Wheel-Rail Interface:

The wheel-rail interface model is a simple penalty function designed to ensure that the train nodes follow the line of the track. It does not attempt to account for the shape of the rail profile. Vertical and lateral loads are treated independently. For this reason, the algorithm is not suitable for rail vehicle dynamics calculations.

Wheel-rail contact stiffness is input on *RAIL_TRAIN. “Vertical” means normal to the plane containing the two rails at the contact point. “Horizontal” means the direction in the plane of the two rails perpendicular to the contacted rail. Vertical contact works like a normal penalty-based contact. A linear force-deflection relationship is assumed in

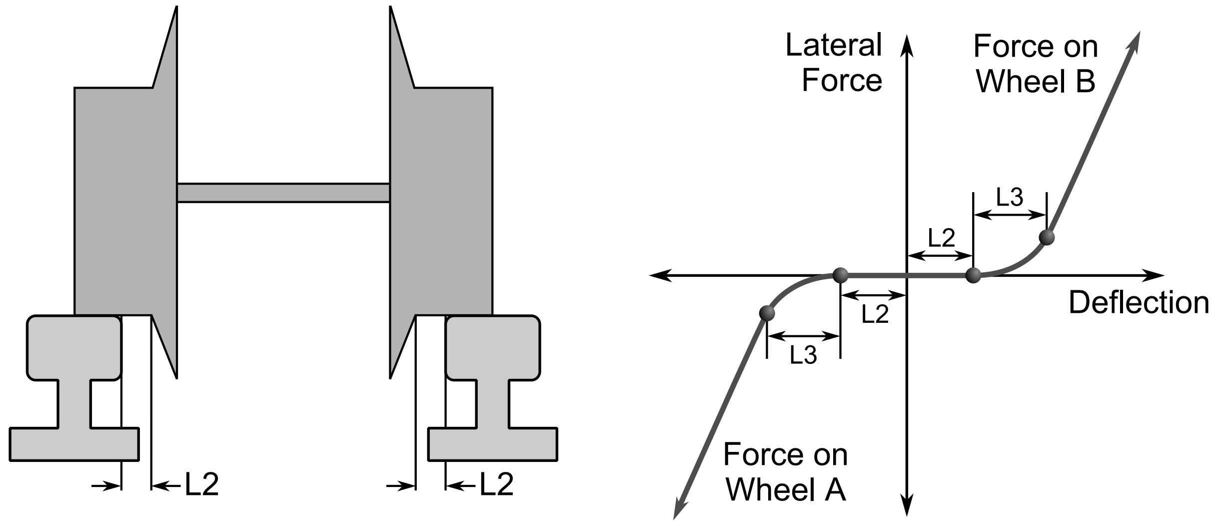

Figure 39-2. Illustration of lateral drift parameters L2 and L3 from *RAIL_-TRAIN. Diagram shows configuration for LATDIR = 0.

compression (see VERTSTF on *RAIL_TRAIN); no tensile force is generated if the train wheel lifts up off the rail. The direction treated as "up" is controlled by IDIR. Typical contact stiffness is 2 MN/mm. Lateral deflections away from the theoretical centerline of the rail beams are penalized by a linear stiffness LATSTF (see *RAIL_TRAIN).

Optionally, a "gap" can be defined in input parameter L2 such that the wheel-set can drift laterally by L2 length units before any lateral force is generated. A further option is to allow smooth transition between "gap" and "contact" by means of a transition distance input as parameter L3. Figure 39-2 illustrates the geometry of parameters L2 and L3. A further option is FRIC on *RAIL_TRAIN. The friction force resists lateral motion of the train wheels relative to the track and is applied along with the forces calculated from LATSTF, L2 and L3 described above.

Generally, with straight tracks a simple linear stiffness is sufficient. With curved tracks, a reasonable gap and transition distance should be defined to avoid unrealistic forces being generated in response to small inaccuracies in the distance between the rails.

Gravity loading is expected, in order to maintain contact between rail and wheel. This is normally applied by an initial phase of dynamic relaxation. To help achieve convergence quickly, or in some cases avoid the need for dynamic relaxation altogether, the initial force expected on each train node can be input (field FINIT on *RAIL_TRAIN). LS-DYNA positions the nodes initially such that the vertical contact force will be FINIT at each node. If the suspension of the rail vehicles is modeled, it is recommended that the input includes pre-compression of the spring elements; if this is not done, the train model may take a long time to reach equilibrium under gravity loading.

The *RAIL algorithm ensures that the train follows the rails but does not provide forward motion. This is generally applied using *INITIAL_VELOCITY, or for straight tracks, *BOUNDARY_PRESCRIBED_MOTION.

Output:

LS-DYNA generates an additional output file train_force.csv, containing force time-histories for each train node, output at the same time intervals as the binary time history file (DT on *DATABASE_BINARY_D3THDT). R12 and previous versions of LS-DYNA generated a file in a different ASCII format, train_force_*n*, where *n* is an integer. For backward compatibility of post-processing tools that format is still available by setting FMT = 1 on *RAIL_TRAIN.

Checking:

We recommend testing the track and train models separately before adding the *RAIL cards. You should check that the models respond stably to impulse forces and that they achieve equilibrium under gravity loading. Most problems we have encountered have been due to unstable behavior of train or track. Often, these are first detected by the *RAIL algorithm, and an error message will result.

*RAIL_TRAIN

Purpose: Define train properties. A train is defined by a set of nodes in contact with a rail defined by *RAIL_TRACK. See description under *RAIL_TRACK.

Card Sets. For each train include one pair of Cards 1 and 2. This input ends at the next keyword ("*") card.

Card 1	1	2	3	4	5	6	7	8
Variable	ID	NSETID	FMT	FINIT	VGAP	TRID	LCUR	OFFS
Type	I	I	F	F	F	I	I	F
Default	none	none	0.0	0.0	0.0	0	↓	0.0

Card 2	1	2	3	4	5	6	7	8
Variable	VERTSTF	LATSTF	V2	V3	L2	L3	LATDIR	FRIC
Type	F	F	F	F	F	F	F	F
Default	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

VARIABLE	DESCRIPTION
ID	Train ID
NSETID	Node set ID containing all nodes that are in contact with rails
FMT	Format of output file containing force time-histories for all train nodes: EQ.0.0: CSV format EQ.1.0: ASCII file, same as R12 and previous versions
FINIT	Estimate of initial vertical force on each wheel (optional) – speeds up the process of initial settling down under gravity loading.
VGAP	Initial gap between wheel and track. If VGAP is defined, FINIT must be zero.

VARIABLE	DESCRIPTION
TRID	ID of track for this train (see *RAIL_TRACK)
LCUR	Load curve ID (see *DEFINE_CURVE) containing wheel roughness (distance of wheel surface away from perfect circle) as a function of distance traveled. The curve does not repeat with each rotation of the wheel – the last point should be at a greater distance than the train is expected to travel. Default: no wheel roughness.
OFFS	Offset distance used to generate different roughness curves for each wheel from the roughness curve LCUR. The curve is offset on the x-axis by a different whole number multiple of OFFS for each wheel.
VERTSTF	Vertical stiffness of rail contact
LATSTF	Lateral stiffness of rail contact
V2, V3	Unused variables – leave blank.
L2	Lateral clearance from rail to wheel flange (see Figure 39-2 of *RAIL_TRACK). Lateral force is applied to a wheel only when it has moved more than L2 laterally relative to the rail in the direction determined by LATDIR.
L3	Further lateral distance before full lateral stiffness applies (force-deflection curve follows a parabola up to this point). See Figure 39-2 of *RAIL_TRACK.
LATDIR	Determines the lateral direction (relative to the track) in which wheel movement is resisted by flange contact. If two wheels are fixed to an axle, lateral force is generally applied to one or other of the two wheels, depending on the direction of lateral movement. EQ.0.0: Wheel flanges run on inside faces of rails EQ.1.0: Wheel flanges run on outside faces of rails EQ.2.0: Wheel flanges on both faces of rails (both wheels resist lateral motion in both directions).
FRIC	Coefficient for additional friction force resisting lateral motion of wheel relative to rail.