Lecture 7

Constraints and Connections

Overview

- Introduction
- Surface-Based Coupling Constraints
- Connector Elements
- Surface-Based Ties
- Mesh-Independent Point Fasteners
- Mesh-Independent Surface Connections
- Tips for Diagnosing Constraint and Connection Errors

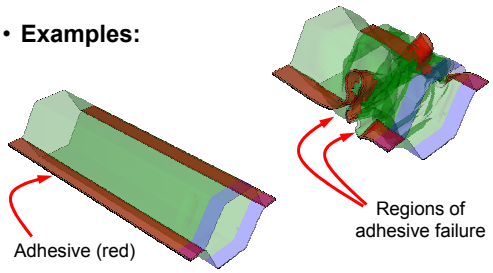
Introduction

Introduction

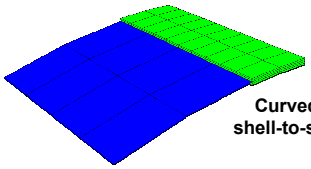
- **Constraints vs. connections**
  - **Constraints** partially or fully eliminate degrees of freedom of a group of nodes and couple their motion to the motion of a master node (or nodes).
  - Examples of constraints include:
    - Rigid Bodies
    - Coupling Constraints
    - Shell-to-Solid Coupling
    - Surface-Based Ties
  - **Connections** model actual connections between parts, such as: ball joints, springs, dampers, bushings, links, hinges, spot welds, rivets, adhesives, bonds, etc.
  - Examples of connections include:
    - Connector Elements
    - Mesh-Independent Point Fasteners
    - Mesh-Independent Surface Connections

Introduction

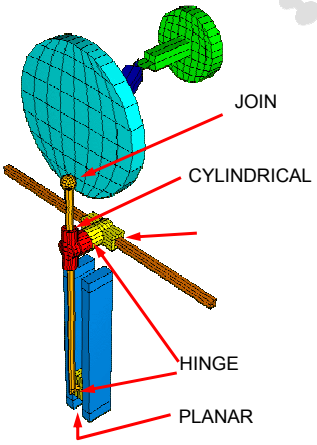
- **Examples:**



Rail crush with failing adhesive layer



Curved plate with shell-to-solid coupling



multibody mechanism with connector elements

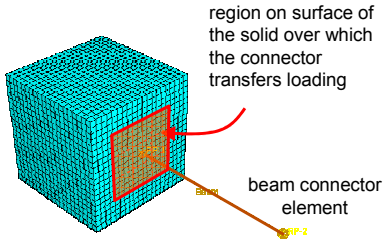
ABAQUS

ABAQUS

Surface-Based Coupling Constraints

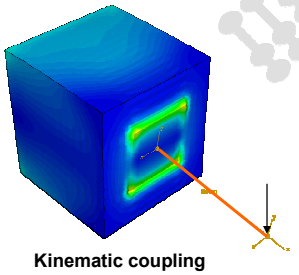
Surface-Based Coupling Constraints

- **Kinematic and distributing couplings**
  - Coupling interactions provide a constraint between a reference node and the nodes on a surface (the coupling nodes).
  - The coupling constraint is useful when a group of coupling nodes is constrained to the rigid body motion of a single node.
  - Typical applications:
    - To apply loads or boundary conditions to a model
    - To model end conditions
    - To model interactions with other constraints, such as connector elements



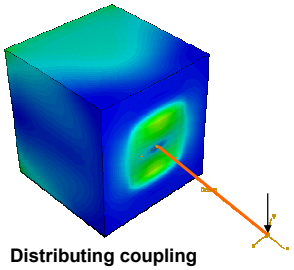
Surface-Based Coupling Constraints

- **Kinematic coupling**
  - Rigid constraints between the master nodes and the coupling nodes
  - Provides a solution for lack of rotational stiffness at nodes connected to solid elements



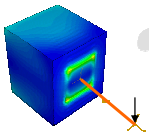
Kinematic coupling

- **Distributing coupling**
  - The constraint is enforced in an average sense
  - Coupling node weight factors provide control of the load transmission.

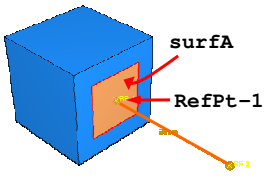


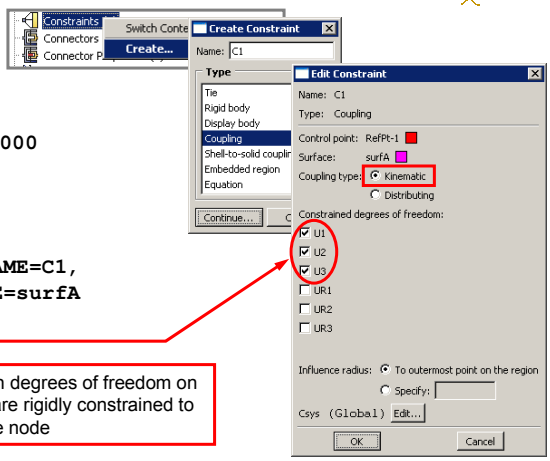
Distributing coupling

Surface-Based Coupling Constraints



- Kinematic coupling definition

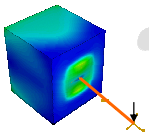




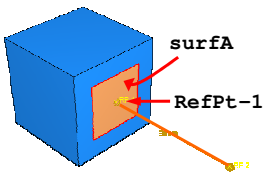

```
*COUPLING, CONSTRAINT NAME=C1,
  REF NODE=1000, SURFACE=surfA
*KINEMATIC
 1, 3
```

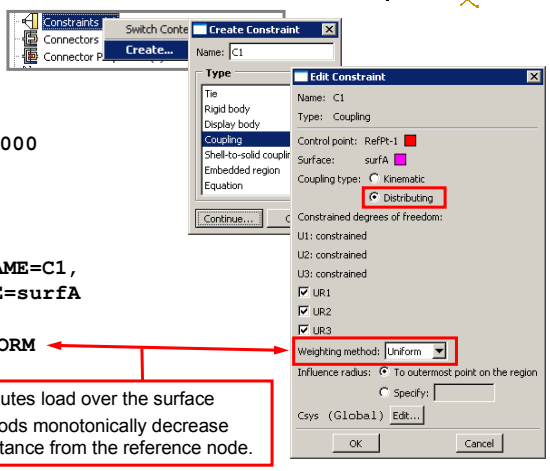
specify which degrees of freedom on the surface are rigidly constrained to the reference node

Surface-Based Coupling Constraints



- Distributing coupling definition



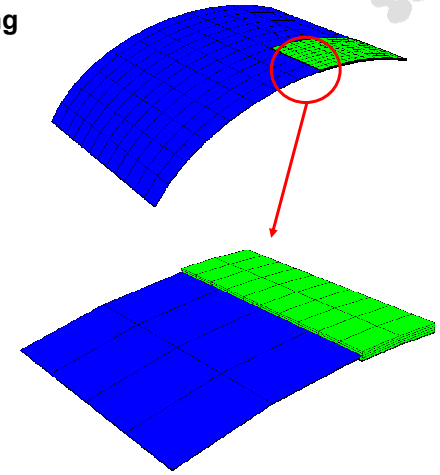



```
*COUPLING, CONSTRAINT NAME=C1,
  REF NODE=1000, SURFACE=surfA
*DISTRIBUTING,
  WEIGHTING METHOD=UNIFORM
 1, 6
```

Uniformly distributes load over the surface
Other weighting methods monotonically decrease loading with radial distance from the reference node.

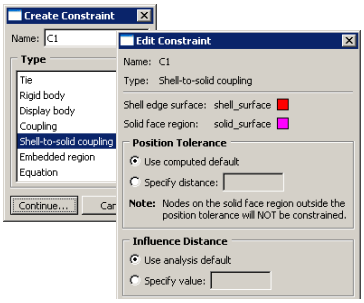
Surface-Based Coupling Constraints

- **Surface-based shell-to-solid coupling**
  - Allows for a transition from shell element modeling to solid element modeling
  - Useful when local modeling requires 3D solid elements but other parts of the structure can be modeled as shells
  - Couples the motion of a "line" of nodes along the edge of a shell model to the motion of a set of nodes on a solid surface
    - Uses a set of internally defined distributing coupling constraints



Surface-Based Coupling Constraints

- **Defining Surface-based shell-to-solid coupling**

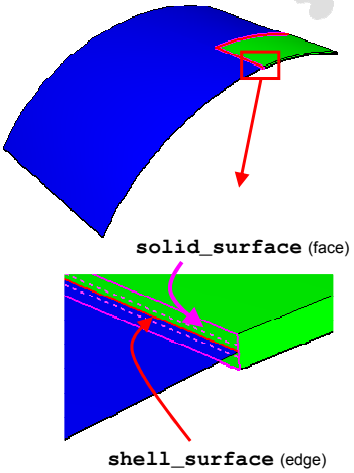



```
*SHELL TO SOLID COUPLING, CONSTRAINT NAME=C1
shell_surface, solid_surface
```

The shell surface must be **edge based**

```
*SURFACE, TYPE=ELEMENT, NAME=shell_surface
shell_surface_E1, E1
```

an edge identifier



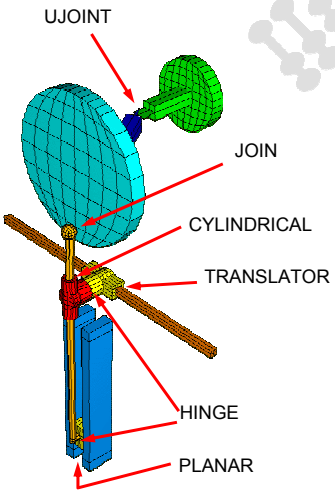
ABAQUS

Connector Elements

Connector Elements

Video Clip

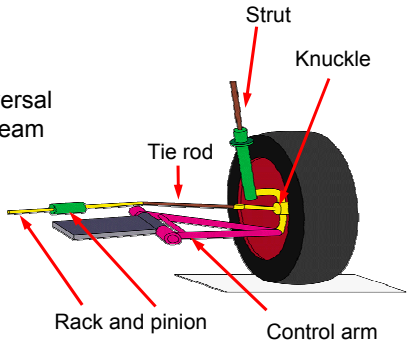
- **Introduction**
  - Connector elements model discrete (point-to-point) physical connections between deformable or rigid bodies.
  - Connectors impose kinematic constraints. For example:
    - hinges
    - constant velocity joints
    - pin-in-slot constraints
  - This section provides a basic introduction to connector elements.
    - Connector elements are discussed in detail in the *Flexible Multibody Systems with ABAQUS* lecture notes.



Connector elements in a multibody mechanism

Connector Elements

- Connectors may include (nonlinear) force-versus-displacement (or velocity) behavior in their unconstrained relative motion components.
  - Example: axial behavior of shock absorbing struts in an automotive suspension systems.
- ABAQUS has an extensive library of:
  - Connection types
    - For example: axial, hinge, weld, universal joints, constant velocity joints, link, beam connectors, etc.
  - Kinetic behaviors
    - Uncoupled or coupled response
    - Elasticity and damping
    - Plasticity and damage
    - Friction
    - Stops, locks, failure



Typical connections in automotive suspension

Connector Elements

Assembled	Basic translational	Basic rotational
BEAM	LINK	ALIGN
WELD	JOIN	REVOLUTE
HINGE	SLOT	UNIVERSAL
UJOINT	SLIDE-PLANE	CARDAN
CVJOINT	CARTESIAN	EULER
TRANSLATOR	RADIAL-THRUST	CONSTANT VELOCITY
CYLINDRICAL	AXIAL	ROTATION
PLANAR	PROJECTION CARTESIAN	FLEXION-TORSION
BUSHING		PROJECTION FLEXION-TORSION

Connector Elements

- **Example: Truck door hinges – keyword interface**

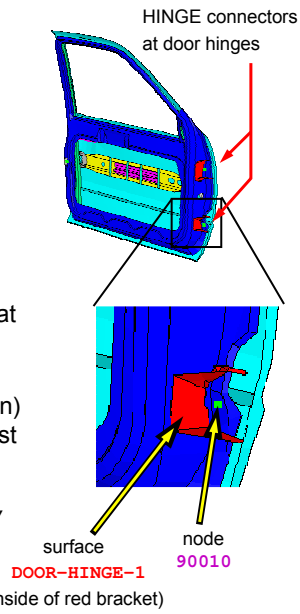
1. Define connector elements

```
*ELEMENT, TYPE=CONN3D2, ELSET=CONN_DOOR_HINGE
620601, 90009, 90010
620602, 90011, 90012
```

2 nodes per element

– In this example, each connector joins two nodes that are reference nodes for distributing coupling constraints.
  - One distributes the hinge load to the door (shown) and the other distributes the hinge load to the rest of the truck body (not shown)

```
*COUPLING, REF NODE=90010, SURFACE=DOOR-HINGE-1,
 CONSTRAINT NAME=DOOR-HINGE-1
*DISTRIBUTING
 1,6
```

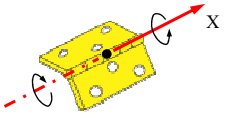


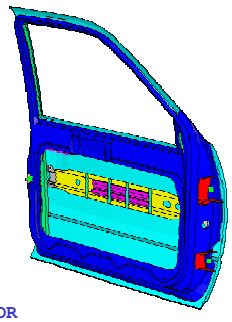
Connector Elements

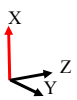
- **Example (cont'd): Truck door hinges – keyword interface**

2. Define connector orientation
   - Orientations are used to define local directions for connection types that use local directions.
     – Orientation directions rotate with the rotation of the nodes of the element.
   - Hinge connectors require an orientation which will be associated with the first connector node.
     – The hinge axis is aligned with the orientation X-direction.

```
*ORIENTATION,NAME=ORI_CONN_DOOR
0.,0.,1., 1.,0.,0.
3,0
```







ORI_CONN_DOOR

Connector Elements

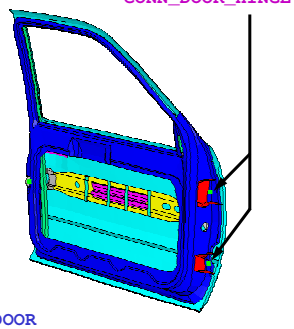
- **Example (cont'd): Truck door hinges – keyword interface**

3 Define connector section

```
*ORIENTATION,NAME=ORI_CONN_DOOR
 0.,0.,1., 1.,0.,0.
 3,0
*ELEMENT, TYPE=CONN3D2, ELSET=CONN_DOOR_HINGE
 620601,9000009,9000010
 620602,9000011,9000012
*CONNECTOR SECTION, ELSET=CONN_DOOR_HINGE
HINGE
 ORI_CONN_DOOR
```

Assembled connection type;
one can also use basic connection types:
JOIN, REVOLUTE

Connector elements
CONN_DOOR_HINGE



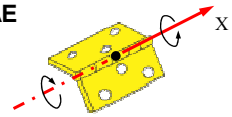
ORI_CONN_DOOR

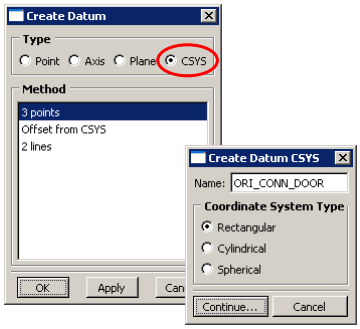
Connector Elements

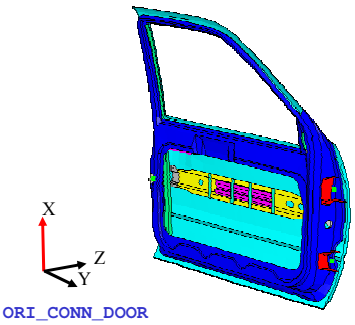
- **Example (cont'd): Truck door hinges – ABAQUS/CAE**

1 Define connector orientation

- In ABAQUS/CAE orientations are defined using datum coordinate systems.





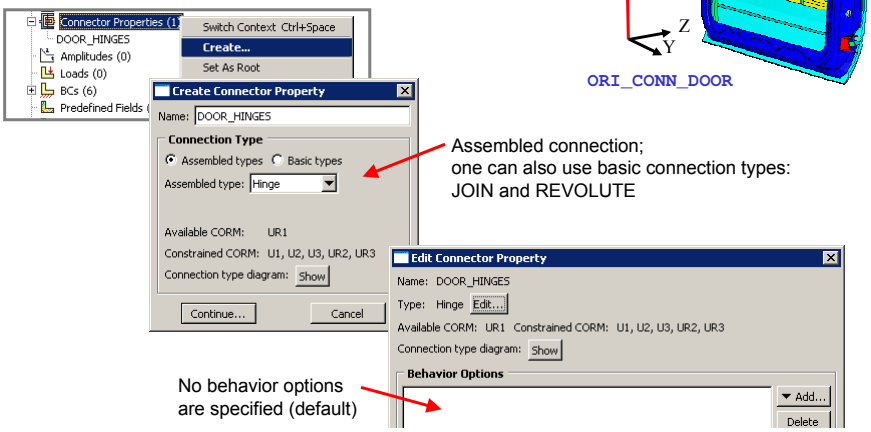


ORI_CONN_DOOR

Connector Elements

- **Example (cont'd): Truck door hinges – ABAQUS/CAE**

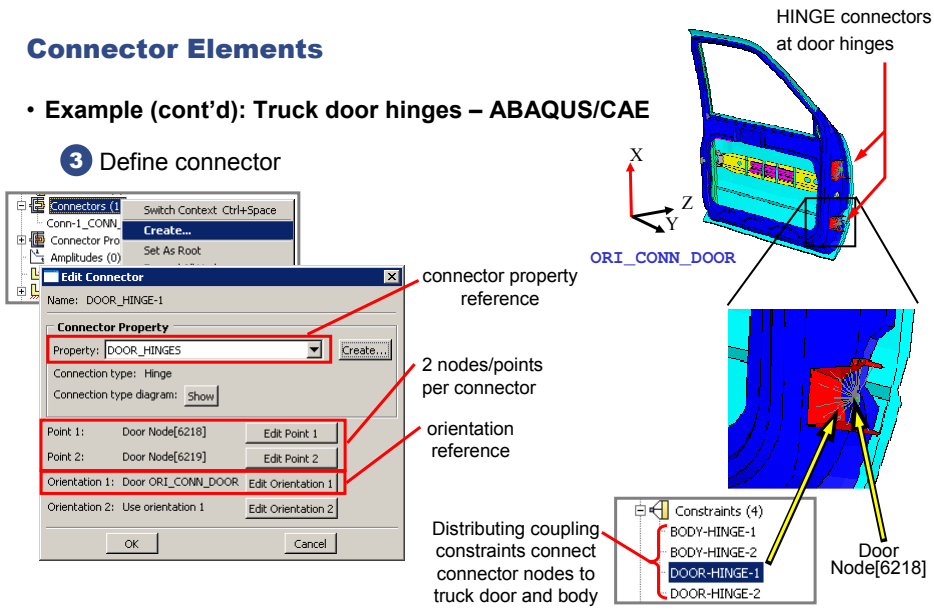
2 Define connector property



Connector Elements

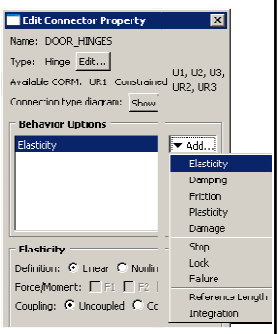
- **Example (cont'd): Truck door hinges – ABAQUS/CAE**

3 Define connector



Connector Elements

- **Connector behavior**
  - Kinetic behavior can be specified only for available components of relative motion
    - Both uncoupled and coupled behavior can be defined
  - A wide variety of behaviors are possible:
    - spring-like elastic behavior (*CONNECTOR ELASTICITY)
    - rigid-like elastic behavior (*CONNECTOR ELASTICITY, RIGID)
    - dashpot-like (damping) behavior (*CONNECTOR DAMPING)
    - friction (*CONNECTOR FRICTION)
    - plasticity (*CONNECTOR PLASTICITY)
    - damage (*CONNECTOR DAMAGE)
    - stops (*CONNECTOR STOP)
    - locks (*CONNECTOR LOCK)
    - failure (*CONNECTOR FAILURE)



Connector Elements

- **Example: Truck door hinges with rotational springs – keyword interface**

Connector definition and kinematics

```
*ORIENTATION,NAME=ORI_CONN_DOOR
0.,0.,1., 1.,0.,0.
3,0

*ELEMENT, TYPE=CONN3D2, ELSET=CONN_DOOR_HINGE
620601,9000009,9000010
620602,9000011,9000012

*CONNECTOR SECTION, ELSET=CONN_DOOR_HINGE,
 BEHAVIOR=CONN_DOOR_HINGES_BEH
HINGE
ORI_CONN_DOOR
```

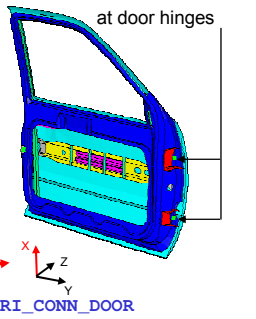
Connector behavior

```
*CONNECTOR BEHAVIOR, NAME=CONN_DOOR_HINGES_BEH

*CONNECTOR ELASTICITY,COMPONENT=4
10.0,
```

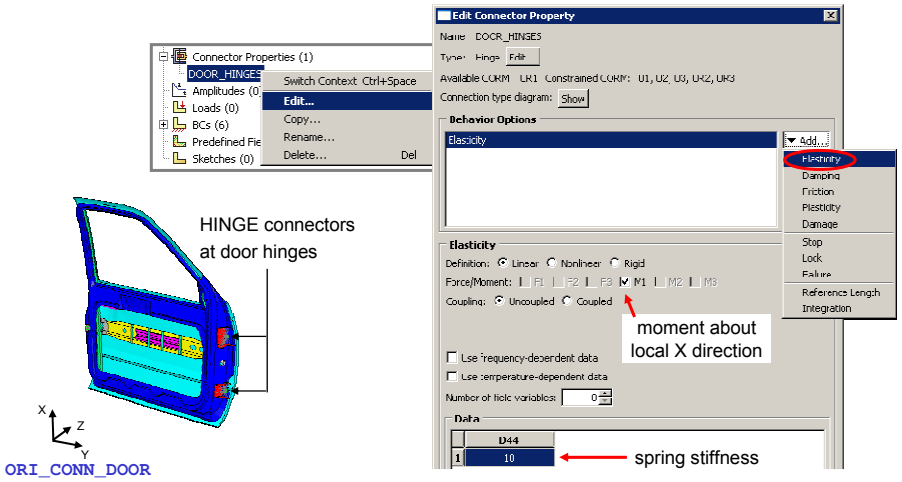
spring stiffness

rotation about local X direction



Connector Elements

- **Example: Truck door hinges with rotational springs – ABAQUS/CAE**



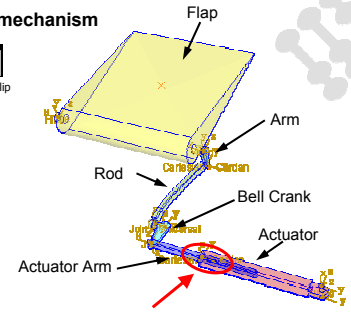
Connector Elements

- **Connector actuation**
  - Connector actuation can be used to drive any available component of relative motion
    - by a prescribed displacement (rotation) or
    - by a specified force (moment).
  - It can also be used to fix available components of relative motion.
  - The prescribed relative motions and loads are in the local directions associated with the connector.

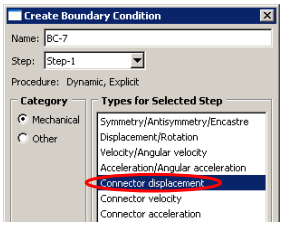
```
*CONNECTOR MOTION, AMPLITUDE=AMP-2
TRANSLATOR, 1, 1.0
```

Flap mechanism

Video Clip



TRANSLATOR connector between Actuator and Actuator Arm

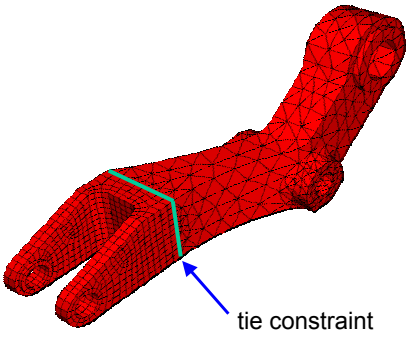


ABAQUS

Surface-Based Ties

Surface-Based Ties

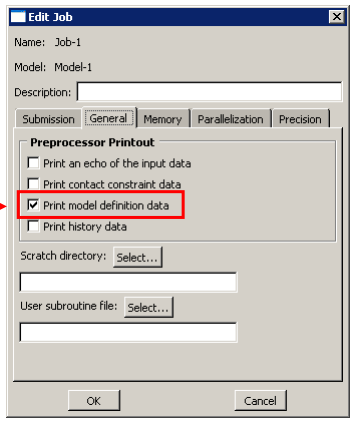
- **In ABAQUS fully constrained contact behavior is defined using tie constraints.**
  - A tie constraint provides a simple way to bond surfaces together permanently.
    - Easy mesh transitioning.
  - Surface-based constraint using a master-slave formulation.
  - The constraint prevents slave nodes from separating or sliding relative to the master surface.



Surface-Based Ties

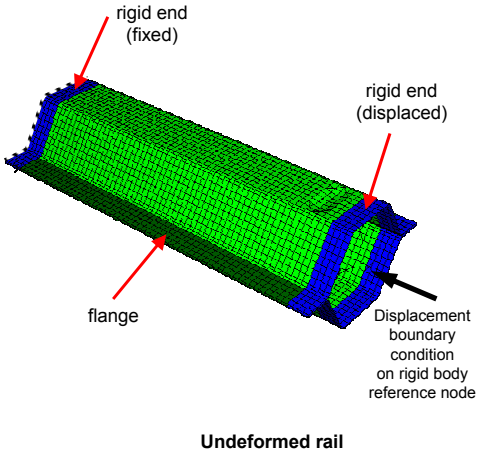
- Slave nodes that contact the master surface at the start of the simulation will be tied to it.
  - Slave nodes not initially tied will remain unconstrained throughout the analysis.
    - They will never "see" the master surface and will be able to penetrate it.
  - A table is printed in the data (`.dat`) file listing each slave node and the master surface nodes to which it will be tied if the preprocessor printout of the model data is requested:

```
*PREPRINT, MODEL=YES
```



Surface-Based Ties

- **Example: Crushing of a rail**
  - The dynamic collapse of a rail is modeled.
  - The ends of the rail are rigid bodies.
    - One end is fully constrained.
    - The other end is loaded with a prescribed displacement.
  - In this section, the flanges are attached using a tie constraint.
    - Different connections types will be used to connect the two halves of the rail later in this lecture.

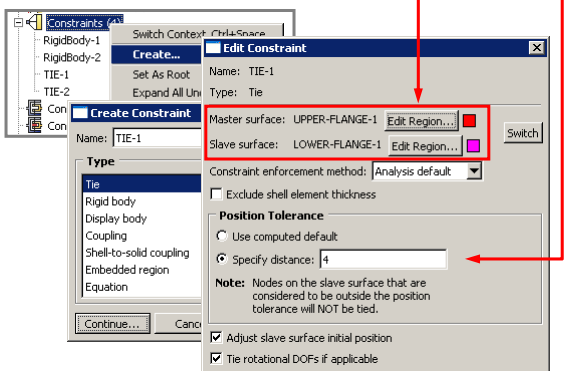


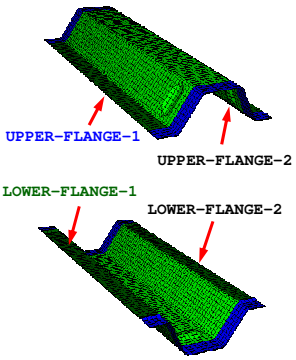
Undeformed rail

Surface-Based Ties

- **Example: Rail crush with tie constraints**

```
*TIE, NAME=TIE, POSITION TOLERANCE=4.0
LOWER-FLANGE-1, UPPER-FLANGE-1
LOWER-FLANGE-2, UPPER-FLANGE-2
```

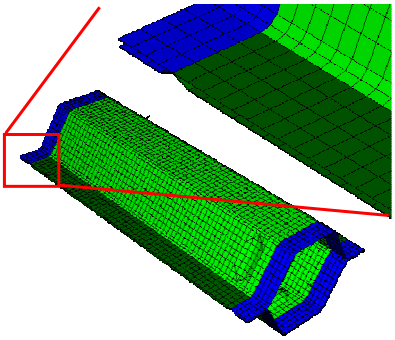




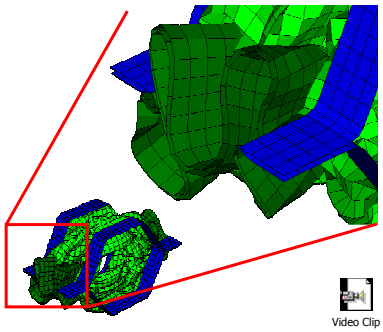
exploded view of rail assembly surfaces are labeled

Surface-Based Ties

- **Example (cont'd): Rail crush with tie constraints**
  - Flange surfaces remain tied throughout the analysis.
    - By default, slave nodes are adjusted to the master surface at the start of the analysis with shell thicknesses accounted for.



undeformed rail assembly

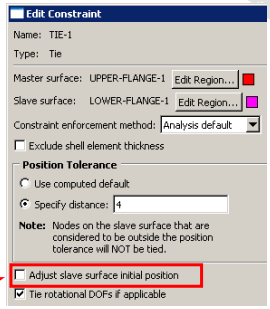


crushed rail assembly

Surface-Based Ties

- The repositioning of slave nodes in the tolerance region onto the master surface is strain-free.
  - In general automatic adjustments are recommended, especially if neither surface has rotations.
    - In this case a constant offset vector is used, which is not appropriate for noncoincident tied surfaces that undergo rigid body rotations.
  - Adjustments can be deactivated

```
*TIE, NAME=TIE, ADJUST=NO
```

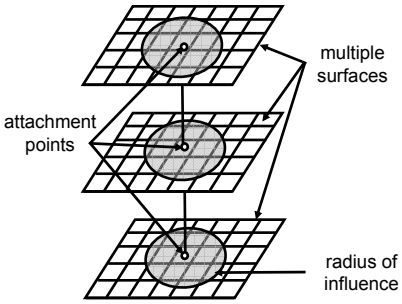


- Avoid applying boundary conditions to the slave nodes of a tie constraint;
  - doing so will overconstrain the model at those nodes.
  - The constraints will be enforced in an average sense using a penalty method, and the analysis computational cost may increase.

Mesh-Independent Point Fasteners

Mesh-Independent Point Fasteners

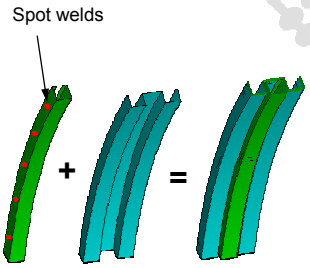
- **Mesh-independent point fasteners allow you to conveniently define point-to-point connections between surfaces.**
  - Fasteners can model spot welds, rivets, screws, bolts, or other types of fastening mechanisms.
  - They can be located anywhere between surfaces.
    - They need not be defined at nodal locations.
  - They can connect multiple layers;
    - i.e., the number of connected surfaces is not restricted.
  - The fastener acts over a specified radius of influence.



Mesh-Independent Point Fasteners

- The fastener capability combines either:
  - connector elements or
  - beam multi-point constraints

  with distributing coupling constraints to define a connection.
- Translation and rotation of the attachment points are related to the translation of nodes on the shell surface.
  - Twist is conveyed by the fastener independently of the shell element drill stiffness.



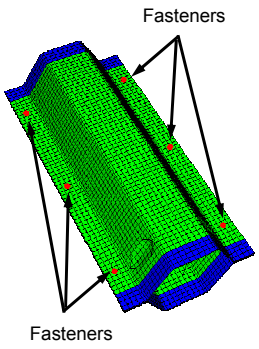
assembly with spot welds

- Fasteners can model either rigid, elastic, or inelastic connections with failure by using the generality of connector behavior definitions.
- Fasteners are not currently supported by ABAQUS/CAE

Mesh-Independent Point Fasteners

- **Example: Rail crush with rigid fasteners**
  - The dynamic rail collapse is analyzed again.
    - Flanges now are connected using six rigid fasteners modeled with connector elements.
  - Procedure to define the fasteners:

1. Define a connector section referencing an element set name with no user-defined elements.
   - Define connector behavior properties as usual.




```
*CONNECTOR SECTION, ELSET=Fasteners
BEAM,
```

Beam connector type to define a rigid fastener

empty element set

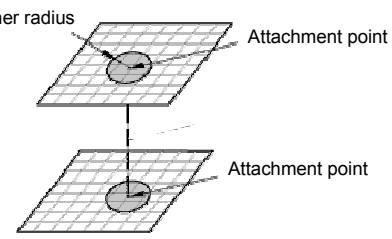
ABAQUS will automatically generate the connector elements and include them in the set named **Fasteners**

Mesh-Independent Point Fasteners

- **Example (cont'd): Rail crush with rigid fasteners**

2. Define the fastener properties.
   - Fasteners are assumed to have a circular projection onto the connected surfaces.
   - You are required to specify the radius of the fastener.

```
*FASTENER PROPERTY, NAME=FASTENER-RIGID
4.0,
```



Mesh-Independent Point Fasteners

- **Example (cont'd): Rail crush with rigid fasteners**

3 Define the fasteners.

- All six fasteners are defined using one *FASTENER option

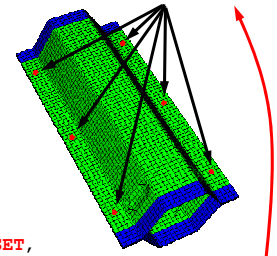
```
*CONNECTOR SECTION, ELSET=Fasteners
BEAM,

*FASTENER PROPERTY, NAME=FASTENER-RIGID
4.0,

*FASTENER, INTERACTION=F1, ELSET=Fasteners,
PROPERTY=FASTENER-RIGID, REFERENCE NODE SET=FASTENER_NSET,
NUMBER OF LAYERS=1, SEARCH RADIUS=7.0000E+00
```

Same empty element set

FASTENER_NSET



One less than the number of surfaces to be joined (2 surfaces − 1 = 1)

Fasteners will connect only to surfaces that are within this distance of reference nodes (surfaces identified automatically)

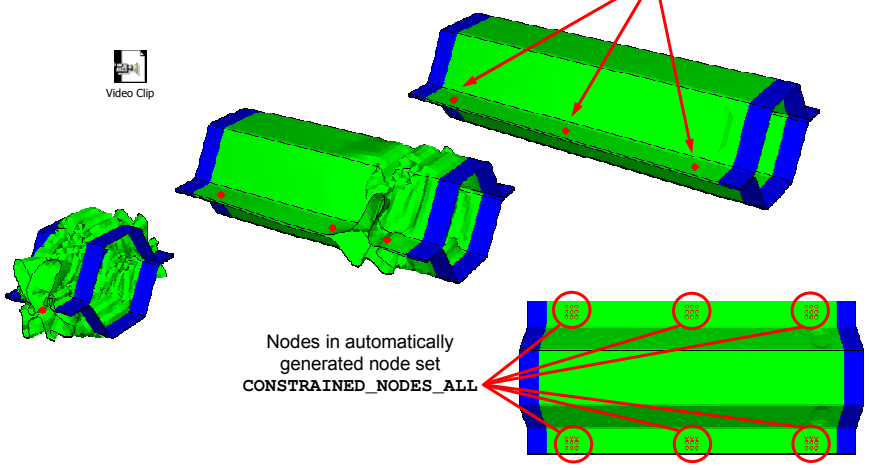
Reference nodes used to locate the fasteners

Mesh-Independent Point Fasteners

- **Example (cont'd): Rail crush with rigid fasteners**

Connector elements

Video Clip



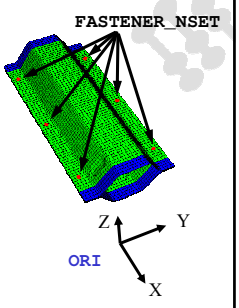
Nodes in automatically generated node set CONSTRAINED_NODES_ALL

Mesh-Independent Point Fasteners

- **Alternative options for fastener creation**
  - Fastened surfaces can be specified directly using surface names.
    - In some cases the automatic surface identification method is not appropriate.
      - The automatic approach does not distinguish between coincident facets.
      - A number of surfaces may lie within the reference node search radius, but it may not be desirable to connect all of them.
  - Connector elements can be defined directly.
    - This option provides more control over the connector definitions;
      - however, it also requires more user input.
  - Rigid fasteners can be modeled with beam multi-point constraints (MPCs) rather than beam connector elements.
    - Connectors are often recommended because they can be easily modified to model more complex connection behaviors.

Mesh-Independent Point Fasteners

- **Example: Rail crush with deformable fasteners**
  - The rail flanges now are connected using fasteners that allow for plasticity, damage, and failure.



Modifications to the rigid fastener definition are in **black**:

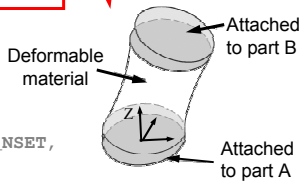
```
*CONNECTOR SECTION, ELSET=Fasteners, BEHAVIOR=CONN_PLAS_FAIL
BUSHING
ORI,

*ORIENTATION, NAME=ORI
1,0,0, 0,0,-1

*FASTENER PROPERTY, NAME=FASTENER-RIGID
4.0,

*FASTENER, INTERACTION=F1, ELSET=Fasteners,
PROPERTY=FASTENER-RIGID, REFERENCE NODE SET=FASTENER_NSET,
NUMBER OF LAYERS=1, SEARCH RADIUS=7.0000E+00
```

BUSHING = PROJECTION CARTESIAN + PROJECTION FLEXION-TORSION
It does not impose any constraints on the 6 components of relative motion.



Connection diagram

Mesh-Independent Point Fasteners

- **Example (cont'd): Rail crush with deformable fasteners**
  - The options used to define the connector behavior:

```
*CONNECTOR BEHAVIOR, NAME=CONN_PLAS_FAIL
```

Elastic behavior:
```
*CONNECTOR ELASTICITY
```

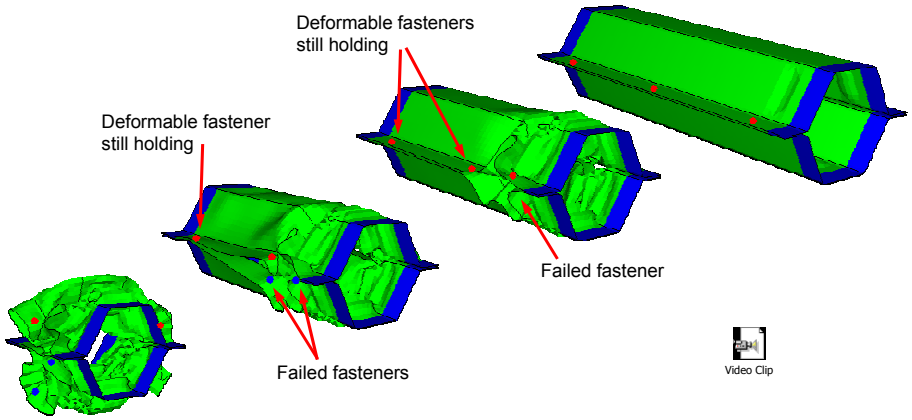
Plastic behavior:
```
*CONNECTOR DERIVED COMPONENT
*CONNECTOR PLASTICITY
*CONNECTOR POTENTIAL
*CONNECTOR HARDENING
```

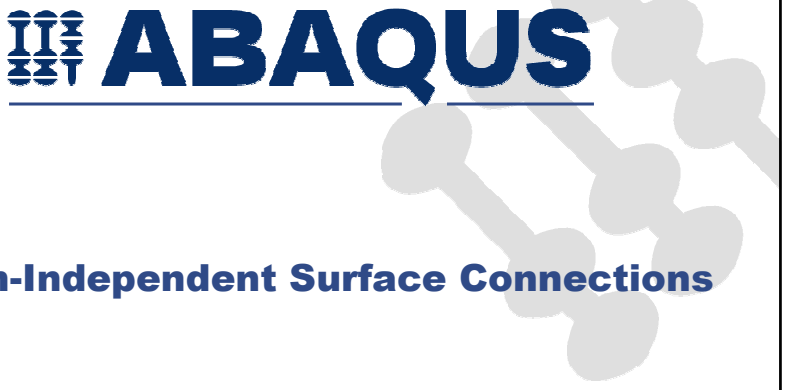
Damage and failure:
```
*CONNECTOR DAMAGE INITIATION, CRITERION=PLASTIC MOTION
*CONNECTOR DAMAGE EVOLUTION
```

  - Connector damage and failure will be discussed in Lecture 9, *Material Damage and Failure*.
  - For more information on other connector behavior options refer to the *ABAQUS Analysis User's Manual* or the *Flexible Multibody Systems with ABAQUS* lecture notes

Mesh-Independent Point Fasteners

- **Example (cont'd): Rail crush with deformable fasteners**

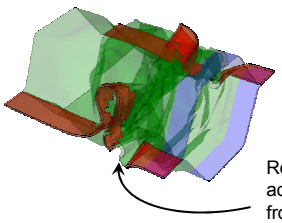




Mesh-Independent Surface Connections

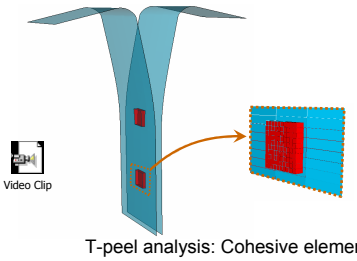
Mesh-Independent Surface Connections

- **Introduction**
  - ABAQUS uses cohesive elements to model the behavior of adhesives or interfaces
  - Cohesive elements model separation between two initially bonded surfaces
  - Specific applications include
    - Progressive failure of adhesives
    - Delamination in composites
    - Detailed spot-weld modeling



Video Clip

Regions of failed adhesive are eliminated from the analysis



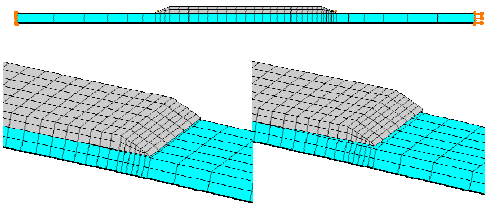
Video Clip

T-peel analysis: Cohesive elements are used for modeling spot-welds

Mesh-Independent Surface Connections

– Complex fracture mechanisms are idealized with a macroscopic "cohesive law," which relates the traction across the interface to the separation

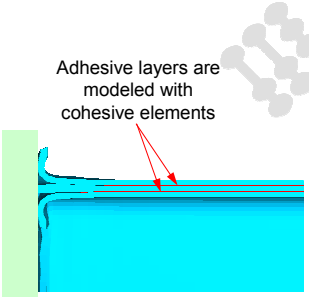
– Particularly attractive when interface strengths are relatively weak compared to the adjoining materials



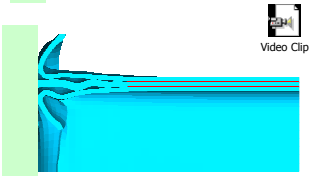
Beginning of separation After separation

Skin/stiffener debonding in a composite structure

Video Clip



Video Clip

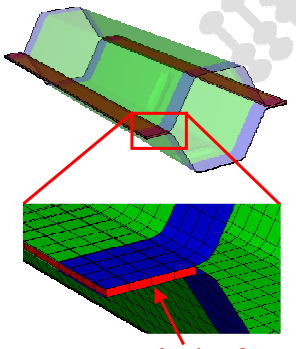


Composite delamination

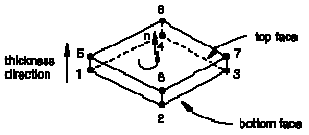
Mesh-Independent Surface Connections

- **Example: Rail crush with adhesive interface**
  - The rail flanges now are connected by an adhesive modeled with cohesive elements.

1. Define cohesive elements.
   - Cohesive element response is unique in the thickness and tangential directions.
   - By default, thickness direction is determined by the nodal connectivity.
     - Similar to continuum shell elements described in Lecture 2, *Elements*.



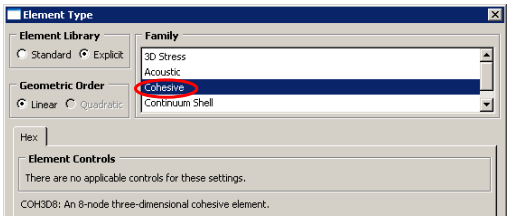

```
*ELEMENT, TYPE=COH3D8, ELSET=cohesive_layer
273, 1036, 1037, 1042, 1041, 691, 692, 697, 696
274, 1037, 1038, 1043, 1042, 692, 693, 698, 697
⋮
```

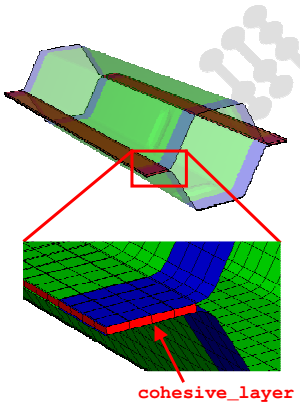


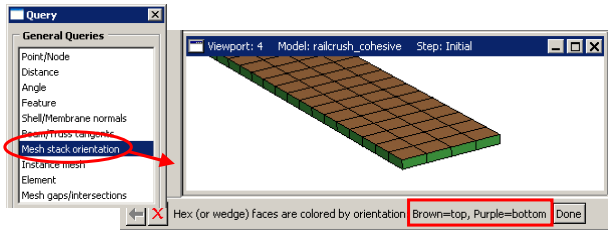
Default thickness direction

Mesh-Independent Surface Connections

- **Example (cont'd): Rail crush with adhesive interface**



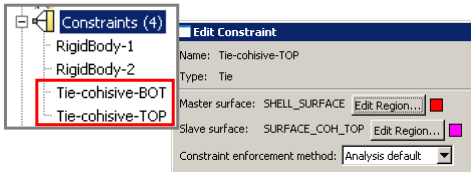


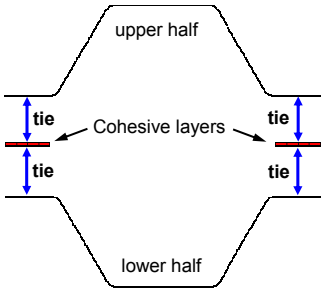


Mesh-Independent Surface Connections

- **Example (cont'd): Rail crush with adhesive interface**
  - **2** Tie the cohesive elements to the flange surfaces.
    - Cohesive elements may connect to other model components through:
      - shared nodes,
      - contact conditions, or
      - tie constrains (used in this example).

```
*TIE, NAME=cohesive_tie, ADJUST=NO
surface_coh_top, shell_surface
surface_coh_bot, shell_surface
```





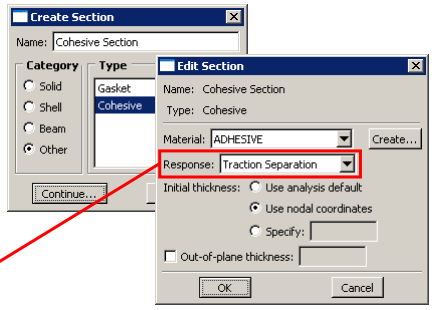
Exploded view of rail cross-section

Mesh-Independent Surface Connections

- **Example (cont'd): Rail crush with adhesive interface**
  - 3 Define the cohesive elements section properties.

```
*ELEMENT, TYPE=COH3D8, ELSET=cohesive_layer
 273,1036,1037,1042,1041,691,692,697,696
  :

*COHESIVE SECTION, ELSET=cohesive_layer,
   MATERIAL=adhesive, THICKNESS=GEOMETRY,
   RESPONSE = TRACTION SEPARATION
```



Constitutive model specified directly in terms of **traction versus separation**

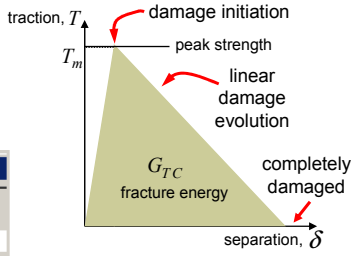
Intended for bonded interfaces where the interface thickness is negligibly small

Use the **continuum** response when the adhesive joint connects two bodies has a finite thickness.

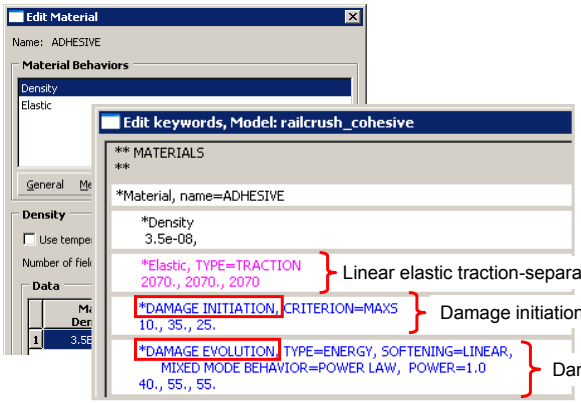
Use the **gasket** response to define a uniaxial stress-based constitutive model

Mesh-Independent Surface Connections

- **Example (cont'd): Rail crush with adhesive interface**
  - 4 Define the cohesive elements constitutive behavior.



Traction-separation response



Linear elastic traction-separation behavior

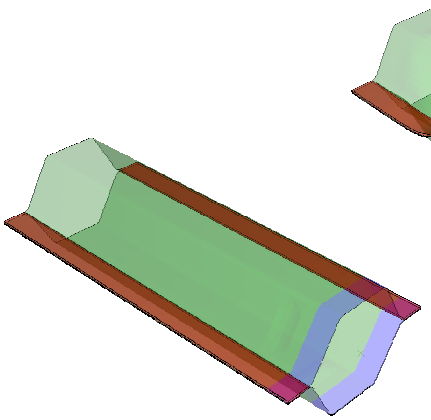
Damage initiation criteria

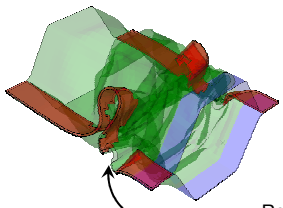
Damage evolution behavior

Mesh-Independent Surface Connections

- Example (cont'd): Rail crush with adhesive interface

Video Clip





Regions of failed adhesive are eliminated from the analysis

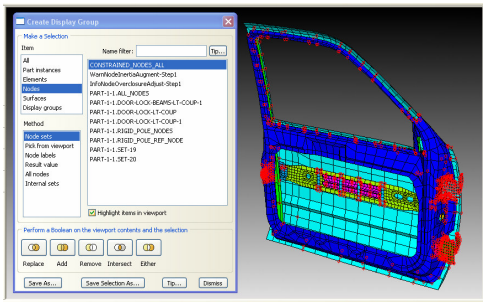
ABAQUS

ABAQUS

Tips for Diagnosing Constraint and Connection Errors

Tips for Diagnosing Constraint and Connection Errors

- **Viewing constrained nodes**
  - Constrained nodes can be viewed by displaying node set CONSTRAINED_NODES_ALL using display groups.



Constrained nodes displayed in ABAQUS/Viewer

Tips for Diagnosing Constraint and Connection Errors

- **Overconstraints**
  - An overconstraint means applying multiple kinematic constraints to a single degree of freedom.
    - overconstraints are **consistent** if all the constraints are compatible with each other.
      - All consistent constraints are resolved automatically by ABAQUS/Explicit.
    - overconstraints are **inconsistent/conflicting** if the constraints are incompatible with each other.
      - A correct **solution does not exist**.
        - The model is ill-posed.
      - Conflicting constraints may lead to inaccurate solutions or termination of the analysis.