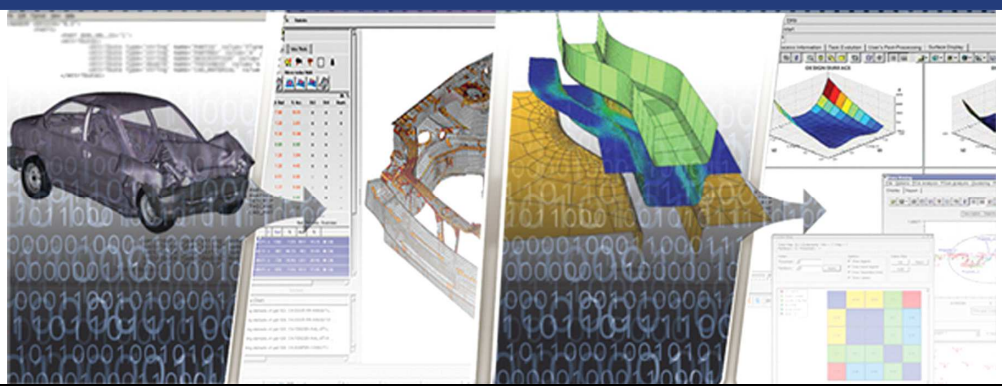


RADIOSS THEORY MANUAL

Version 2017 – January 2017

Large Displacement Finite Element Analysis

Chapter 8

Altair Engineering, Inc., World Headquarters: 1820 E. Big Beaver Rd., Troy MI 48083-2031 USA
Phone: +1.248.614.2400 • Fax: +1.248.614.2411 • www.altair.com • info@altair.com

CONTENTS

Chapter 8

INTERFACES

8.0 INTERFACES

8.1 Introduction

Interfaces solve the contact and impact conditions between two parts of a model. Contact-impact problems are among the most difficult nonlinear problems to solve as they introduce discontinuities in the velocity time histories. Prior to the contact, the normal velocities of the two bodies which come into contact are not equal, while after impact the normal velocities must be consistent with the impenetrability condition. In the same way, the tangential velocities along interfaces are discontinuous when stick-slip behavior occurs in friction models. These discontinuities in time complicate the integration of governing equations and influence performance of numerical methods.

Central to the contact-impact problem is the condition of impenetrability. This condition states that bodies in contact cannot overlap or that their intersection remains empty. The difficulty with the impenetrability condition is that it cannot be expressed in terms of displacements as it is not possible to anticipate which parts of the bodies will come into contact. For this reason, it is convenient to express the impenetrability condition in a rate form at each cycle of the process. This condition can be written as:

$$\gamma_N = v_N^A - v_N^B \leq 0 \qquad \text{EQ. 8.1.0.1}$$

on the contact surface Γ_C common to the two bodies.

v_N^A and v_N^B are respectively the normal velocities in the two bodies in contact. γ_N is the rate of interpenetration.

EQ. 8.1.0.1 simply expresses that when two bodies are in contact, they must either remain in contact and $\gamma_N = 0$, or they must separate and $\gamma_N < 0$.

On the other hand, the tractions must observe the balance of momentum across the contact interface. This requires that the sum of the tractions on the two bodies vanish:

$$t_N^A + t_N^B = 0 \qquad \text{EQ. 8.1.0.2}$$

Normal tractions are assumed compressive, which can be stated as:

$$t_N = t_N^A = -t_N^B < 0 \qquad \text{EQ. 8.1.0.3}$$

EQ 8.1.0.1 and EQ. 8.1.0.2 can be combined in a single equation stating that, $t_N \gamma_N = 0$. This condition simply expresses that the contact forces do not create work. If the two bodies are in contact, the interpenetration rate vanishes. On the other hand, if the two bodies are separated $\gamma_N < 0$ but the surface tractions vanish. As a result, the product of the surface tractions and the interpenetration rate disappear in all cases.

If we notice that the impenetrability condition is expressed as an inequality constraint, the condition:

$$t_N \gamma_N = 0 \qquad \text{EQ. 8.1.0.4}$$

can also be seen as the Kuhn-Tucker condition associated with the optimization problem consisting in minimizing the total energy (EQ. 2.10.0.5) subject to the inequality constraint 8.1.0.1.

In practice, the solution to a contact problem entails in three steps:

- First, it is necessary to find for each point those points in the opposite body which will possibly come into contact. This is the geometrical recognition phase.
- The second phase is to check whether or not the bodies are in contact and, if the bodies are in contact, if they are sticking or slipping. This step makes use of the geometrical information computed in the first phase.
- The last step will be to compute a satisfactory state of contact.

The geometrical recognition phase is dependent on the type of interface. This will be discussed below in parallel with the description of interfaces. On the other hand, structural problems with contact-impact conditions lead to constrained optimization problems, in which the objective function to be minimized is the virtual power subject to the contact-impact conditions. There are conventionally two approaches to solving such mathematical programming problems:

- the Lagrange multiplier method
- the Penalty method.

Both methods are used in *RADIOSS*.

8.1.1 Lagrange Multiplier Method

Lagrange multipliers can be used to find the extremum of a multivariate function $f(x_1, x_2, ..., x_n)$ subject to the constraint $g(x_1, x_2, ..., x_n) = 0$, where *f* and *g* are functions with continuous first partial derivatives on the open set containing the constraint curve, and $\nabla g \neq 0$ at any point on the curve (where ∇ is the gradient).

To find the extreme, we write:

$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + ... + \frac{\partial f}{\partial x_n} dx_n = 0 \qquad \text{EQ. 8.1.1.1}$$

But, because *g* is being held constant, it is also true that

$$dg = \frac{\partial g}{\partial x_1} dx_1 + \frac{\partial g}{\partial x_2} dx_2 + ... + \frac{\partial g}{\partial x_n} dx_n = 0 \qquad \text{EQ. 8.1.1.2}$$

So multiply EQ. 8.1.1.2 by the as yet undetermined parameter λ and add to EQ. 8.1.1.2,

$$\left(\frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1}\right) dx_1 + \left(\frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2}\right) dx_2 + ... + \left(\frac{\partial f}{\partial x_n} + \lambda \frac{\partial g}{\partial x_n}\right) dx_n = 0 \qquad \text{EQ. 8.1.1.3}$$

Note that the differentials are all independent, so any combination of them can be set equal to 0 and the remainder must still give zero. This requires that:

$$\left(\frac{\partial f}{\partial x_k} + \lambda \frac{\partial g}{\partial x_k}\right) dx_k = 0 \qquad \text{EQ. 8.1.1.4}$$

for all *k* = 1, ..., *n*, and the constant λ is called the *Lagrange multiplier*. For multiple constraints, $g_1 = 0$, $g_2 = 0$, ...,

$$\nabla f = \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2 + ... \qquad \text{EQ. 8.1.1.5}$$

The Lagrange multiplier method can be applied to contact-impact problems. In this case, the multivariate function is the expression of total energy subjected to the contact conditions:

$$f(x_1, x_2, ..., x_n) \equiv \Pi(x, \dot{x}, \ddot{x}) \qquad \text{EQ. 8.1.1.6}$$

$$g(x_1, x_2, ..., x_n) \equiv Q(x, \dot{x}, \ddot{x}) = 0 \qquad \text{EQ. 8.1.1.7}$$

where $x, \dot{x}, \ddot{x}$ are the global vectors of d.o.f. The application of Lagrange multiplier method to the previous equations gives the weak form as:

$$M\ddot{x} + f_{int} - f_{ext} + L\lambda = 0 \qquad \text{EQ. 8.1.1.8}$$

with $Lx = b$ EQ. 8.1.1.9

This leads to:

$$\begin{bmatrix} K & L^T \\ L & 0 \end{bmatrix} \begin{Bmatrix} x \\ \lambda \end{Bmatrix} = \begin{Bmatrix} f \\ 0 \end{Bmatrix} \qquad \text{EQ. 8.1.1.10}$$

The Lagrange multipliers are physically interpreted as surface tractions. The equivalence of the modified virtual power principle with the momentum equation, the traction boundary conditions and the contact conditions (impenetrability and surface tractions) can be easily demonstrated [28].

It is emphasized that the above weak form is an inequality. In the discretized form, the Lagrange multiplier fields will be discretized and the restriction of the normal surface traction to be compressive will result from constraints on the trial set of Lagrange multipliers.

8.1.2 Penalty method

In the solution of constrained optimization problems, penalty methods consist in replacing the constrained optimization problem with a sequence of unconstrained optimization problems. The virtual power continues to be minimized so as to find the stationary condition, but a penalty term is added to EQ. 2.10.0.5 so as to impose the impenetrability condition:

$$\delta Q = \int_{\Gamma^c} \rho\phi(\gamma_N) d\Gamma \qquad \text{EQ. 8.1.2.1}$$

where:

$\phi(\gamma_N) = 0$ if $\gamma_N = 0$

$\phi(\gamma_N) > 0$ if $\gamma_N < 0$

ρ is an arbitrary parameter known as the penalty parameter. The penalty function ϕ is an arbitrary function of the interpenetration and its rate. It is emphasized that the weak form, including the virtual power and the penalty term EQ. 8.1.2.1 is not an inequality form. The penalty function will be defined in the description of interfaces.

8.2 Interface overview

There are several different interface types available in RADIOSS. A brief overview of the different types and their applications is shown below.

Interface Descriptions

Type	Description
1	Tied contact (boundary) between an ALE part and a Lagrangian part.
2	Tied contact.
3	Used to simulate impacts and contacts on shell and solid elements. Surfaces should be simply convex.
5	Used to simulate impacts and contacts between a master surface and a list of slave nodes. Best suited for beam, truss and spring impacts on a surface.
6	Used to simulate impacts and contacts between two rigid surfaces.
7	A general interface that removes the limitations of types 3 and 5.
8	Drawbead contact for stamping applications
9	ALE lagrange with void opening and free surface.
10	Tied after impact with or without rebound.
11	Edge to edge or line to line impact.
12	Connects 2 fluid meshes with free, tied or periodic options ALE or EULER or LAG/ ALE or EULER or LAG.
14	Ellipsoidal surfaces to nodes contact.
15	Ellipsoidal surfaces to segments contact.
18	Coupling between a Lagrangian material and an ALE material.
19	General contact interface. Node to segment contact and Edge to Edge contact. Equivalent to one interface type 7 + one interface type 11.
20	Single surface, surface to surface with optional Edge to Edge contact. No time step condition with soft penalty
21	Specific interface between a non-deformable master surface and a slave surface designed for stamping

Each of these interfaces was developed for a specific application field. However, this application field is not the only selection criteria. Some limitations of the different algorithms used in each interface can also determine the choice.

The algorithm limitations concern mainly the search of the impacted segment. This search may be performed directly (type 7, 10, and 11 interfaces), or via a search of the nearest node (type 3, 5, and 6 interfaces).

Apart from the limitation of the nearest node search, some limitations exist for the choice between the segments connected to the nearest node. These limitations are the same for type 3, 5 and 6 interfaces.

Type 3, 5 and 6 interfaces also have some limitations due to the orientation of the normal segment.

Type 7 interface was written to emulate type 3 and 5 interfaces without algorithm limitations. With this interface, each node can impact one or more segments, on both sides, on the edges or on the corners of the segments. The only limitation to this interface concerns high impact speed and/or small gap. For these situations the interface will continue to work properly, but the time step can decrease dramatically.

Types 3, 5 and 7 interfaces are compatible with all RADIOSS kinematic options.

Type 1 interface is a special option used with solid elements to provide mesh transition. Type 1 interface is used to connect a Lagrangian and an A.L.E. mesh.

Type 2 interface is used to connect a fine and a coarse Lagrangian mesh.

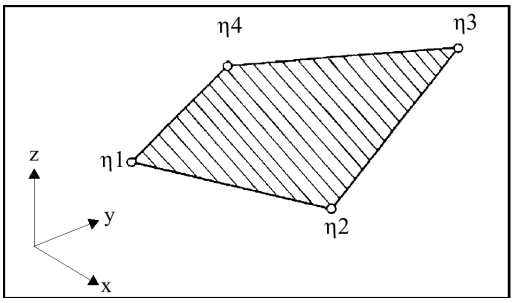
All other interface types (3, 4, 5, 6 and 7) are used to simulate impacts and contacts.

A node may belong to several interfaces.

8.2.1 Surface (Segment) Definition

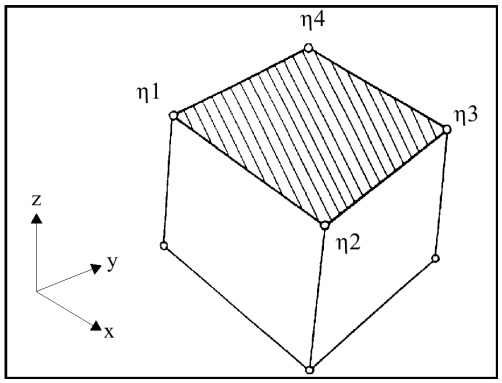
Surfaces or segments may be defined in different ways, depending on the type of element being used. For a four node three dimensional shell element, an element is a segment.

Figure 8.2.1 Shell Elements Segment



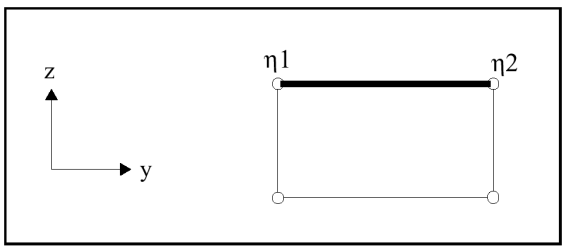
For a brick element, a segment is one face of the brick.

Figure 8.2.2 Brick Element Segment



For a two dimensional element, a segment is one side.

Figure 8.2.3 2D Element Segment



8.3 Tied interface (type 2)

Refer to Chapter 6 ‘Kinematic Constraints’ for a detailed description.

8.4 Auto contacts

8.4.1 Introduction

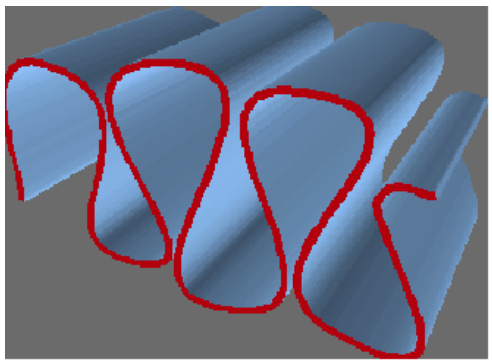
The physics of contacts is involved in various phenomena, such as the impact of two billiard balls, the contact between two gears, the impact of a missile, the crash of a car, etc. While the physics of the contact itself is the same in all these cases, the main resulting phenomena are not.

In the case of billiard balls, it is the shock itself that is important and it will then be necessary to simulate perfectly the wave propagation. In the case of gears, it is the contact pressure that has to be evaluated precisely.

The quality of these simulations depends mainly on the quality of the models (spatial and temporal discretization) and on the choice of the integration scheme. In structural crash or vehicle crash simulations, the majority of the contacts result from the buckling of tubular structures and metal sheets. Modeling the structure using shell and plate finite elements, the physics of the contact cannot be described in a precise way. The reflection of the waves in the thickness is not captured and the distribution of contact pressures in the thickness is not taken into account. The peculiarity of the contacts occurring during the crash of a structure lies more in the complexity of the structural folding and the important number of contact zones than in the description of the impact or the contact itself.

During a contact between two solid bodies, the surface in contact is usually continuous and only slightly curved. On the other hand, during the buckling of a structure, the contacts, resulting from sheet folding, are many and complex. Globally, the contact is no longer between two identified surfaces, but in a surface impacting on itself. The algorithms able to describe this type of contact are “auto-impacting” algorithms. Especially adapted to shell structures, they still can be used to simulate the impact of the external surface of a solid (3D element) on itself.

The main capabilities of the auto-contact can be summarized by the following functionalities:



· capacity to make each point of the surface impact on itself

· capacity to impact on both sides of a segment (internal and external)

· possibility for a point of the surface to be wedged between an upper and a lower part

· processing of very strong concavities (will complete folding)

· reversibility of the contact, thereby authorizing unfolding after folding or the simulation of airbag deployment.

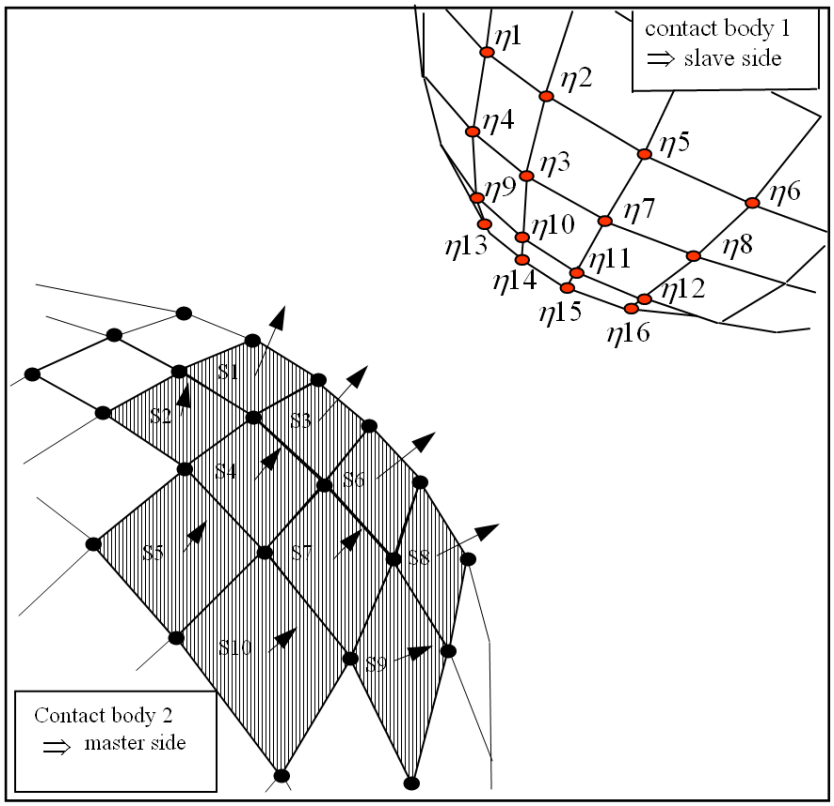
8.4.2 Modeling of contacts

The contacts occurring between two surfaces of a finite element mesh can be modelled in different ways:

· Contact nodes to nodes

The contact is detected based on the criteria of distance between the two nodes. After detection of contact, a kinematic condition or penalty formulation method prevents the penetration attaining the rebound point ("pin-ball" formulation).

Figure 8.4.1 Contact nodes to surface

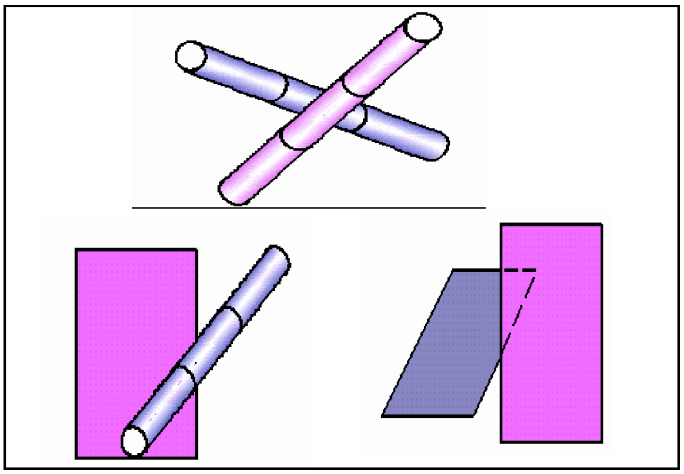


The contact is detected based on the criteria of distance between a group of nodes and a meshed surface. The distance between a node and the surface of a triangular or quadrangular segment is evaluated, locally.

Symmetrized contact nodes to surface

The symmetrization of the previous formulation makes it possible to model a contact between two surfaces, as the group of nodes of the first surface can impact the second group and vice-versa.

Figure 8.4.2 Contact edges to edges



This formulation makes it possible to model contacts between wire framed structures or between edges of two-dimensional structures. Contact is detected based on the criteria of distance between two segments. This formulation can also be used to describe in an approximate way the surface to surface contact.

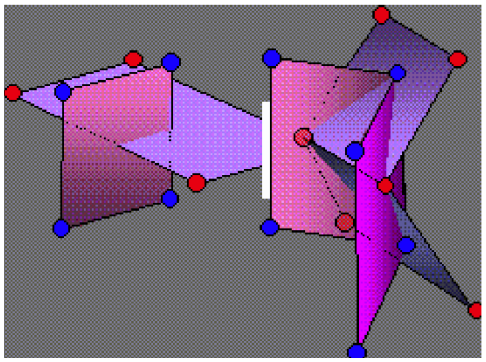
Surface to surface contact

Several approaches can be used to detect the contact between two surfaces. If the two surfaces are quadrangular, the exact calculation of the contact can be complex and quite expensive. An approximate solution may be made by combining the two previous formulations. By evaluating all the contacts of nodes to surface, as well as the edge to edge contacts, the only approximation is the partial consideration of the segment curvature.

8.4.2.1 Choice of a formulation for auto impact

In the case of a surface impacting on itself, it is possible to use one of the previous formulations if considering certain specificities of the auto-contact. The choice of a formulation will depend on two essential criteria: the quality of the description of the contact and the robustness of the formulation. The selected formulation has to provide results that are as precise as possible in a normal operational situation, while still working in a satisfying way in extreme situations. The node-to-node method provides the best robustness, but the quality of the description is not sufficient enough to simulate in a realistic way those contacts occurring during the buckling of a structure.

The node-to-surface contact is the best compromise. However, it has some limitations, the main one being that it cannot detect contacts occurring on the edges of a segment. The most critical situation occurs when this leads to the locking of a part of one surface to another. This phenomenon, being irreversible, might create irrelevant behavior during the deployment of a structure (e.g. airbag deployment). An example of a locking situation is shown in the above images. To correct this, it is possible to associate a node-to-surface formulation to an edge-to-edge formulation.



8.4.3 Algorithms of search for impact candidates

During the impact of one part on another, it is possible to predict which node will impact on a certain segment. In the case of the buckling of a shell structure, such as a tube, it is impossible to predict where different contacts will occur. It is thus necessary to have a fairly general and powerful algorithm that is able to search for impact candidates.

The detail of the formulation of an algorithm, able to search for impact candidates, will depend on the choice of the contact formulation described in the previous chapter. In a node-to-node contact formulation, it is necessary to find for each node the closest node, whose distance is lower than a certain value. In the case of the edge-to-edge formulation, the search for neighboring entities concerns the edges and not the nodes. However, we should note that in some algorithms, the search for neighboring edges or segments is obtained by a node proximity calculation. Moreover, an algorithm designed to search for proximity of nodes can be adapted in order to transform it into a search for proximity of segments or even for a mixed proximity of nodes and segments.

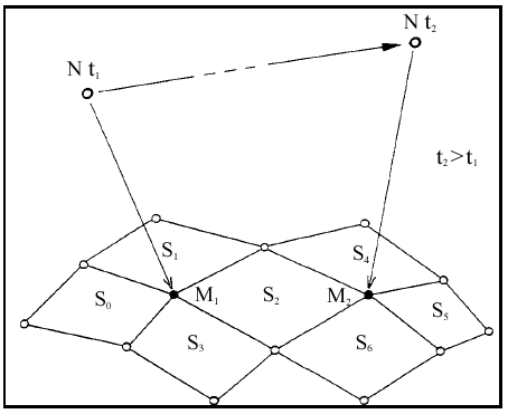
It is possible to distinguish four main types of search for proximity:

- direct search
- topologically limited search
- algorithms of sorting by boxes (bucket sort)
- algorithms of fast sort (octree, quick sort)

When using direct search, at each cycle the distance is calculated from each entity (node, segment, edge) to all others. The quadratic cost (N*N) of this algorithm makes it unusable in case of auto-contact.

8.4.3.1 Topologically limited search

In a simulation in fast dynamics, geometrical modifications of the structure are not very important during one cycle of calculation. It is then possible to consider neighborhood search algorithms using the information of the previous cycle of computation. If for a node the nearest segment is known at the previous cycle, it is then possible to limit the search for this node to the segments topologically close to the previous one (the segments having at least one common node). Furthermore, if an algorithm based on the search of neighboring nodes is used, then the search may be limited to the nodes of the segments connected to the previous closest node.



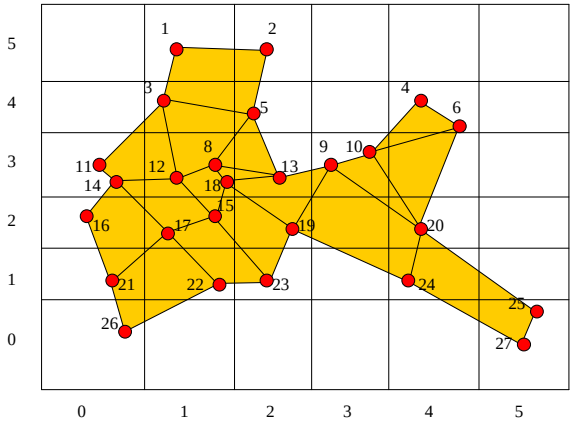
It remains however necessary to do a complete search before the first cycle of calculation. We will also see that this algorithm presents significant restrictions that limit its use to sliding surfaces (it cannot be used for auto-impacting surfaces).

The cost of this algorithm is linear (N), except at the first cycle of computation, during which it is quadratic (N*N). The combination of this algorithm with one of the two following is also possible.

8.4.3.2 Bucket sort

Sorting by boxes consists in dividing space in to steady boxes (not necessarily identical) in which the nodes are placed. The search for closest nodes is limited to one box and the twenty-six neighboring boxes. The cost of this sorting is linear (N) for regular meshes. For irregular meshes, an adaptation is possible but its interest becomes

less interesting compared with the next solution. This three-dimensional sorting is of the same kind as one-way sorting with direct addressing or needle sort.



In order to limit the memory space needed, we first count the number of nodes in each box, thereby making it possible to limit the filling of those boxes that are not empty. In the two-dimensional example shown above, nodes are arranged in the boxes as described in the following table. With this arrangement, the calculation of distances between nodes of the same box is not a problem. On the other hand, taking into account the nodes of neighboring boxes is not straightforward, especially in the horizontal direction (if the arrangement is first made vertically, as in this case). One solution is to consider three columns of boxes at a time. Another solution, more powerful but using more memory, would be for each box to contain the nodes already located there plus those belonging to neighboring boxes. This is shown in the third series of columns in the following table. Once this sorting has been performed, the last step is to calculate the distances between the different nodes of a box, followed by the distance between these nodes and those of the neighboring box. In box 0,3 for example, fifteen distances must be calculated.

11-14, 11-3, 11-8, 11-12, 11-15, 11-16, 11-17, 11-18, 14-3, 14-8,

14-12, 14-15, 14-16, 14-17, 14-18.

Box	**Nodes**			**Nodes of the neighboring box**										
0,0	26			21										
0,1	21			15	16	17	22	26						
0,2	16			8	11	12	14	15	17	21	22	26		
0,3	11	14		3	8	12	15	16	17	18				
1,1	22			15	16	17	19	21	23	26				
1,2	15	17		8	11	12	13	14	18	16	19	21	22	23
1,3	8	12	18	3	5	11	13	14	15	16	17	19		
1,4	3			1	2	5	11	12	13	14	18			
1,5	1			2	3	5								
2,1	23			15	17	19	22							
2,2	19			8	9	10	12	13	15	17	18	22	23	
2,3	13			3	5	8	9	10	12	15	17	18	19	
2,4	5			1	2	3	8	9	10	12	13	18		

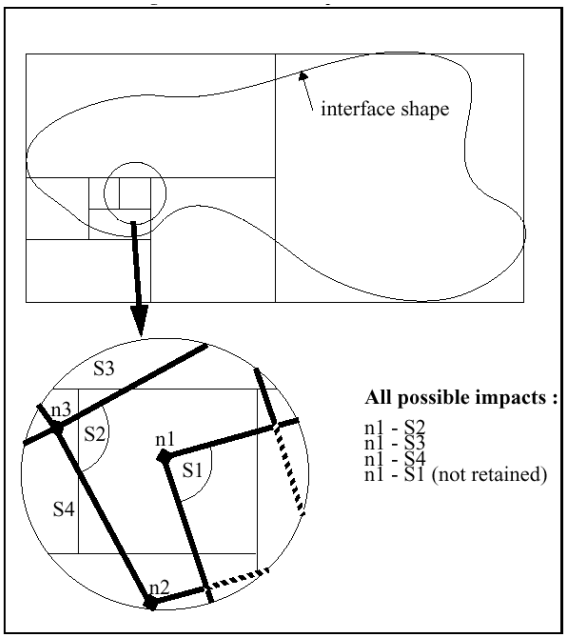
2,5	2			1	3	5								
3,3	9	10		4	5	6	13	19	20					
4,1	24			20	25	27								
4,2	20			9	10	24								
4,4	4	6		9	10									
5,0	25	27		24										

In this example we have considered a search based on nodal proximity. It is possible to adapt this algorithm in order to arrange segments or even edges in the boxes. It will be necessary however to reserve more memory, for a segment can overlap several boxes.

8.4.3.3 Quick sort

The octree is a three-dimensional adaptation of fast sorting. Space is divided into eight boxes, each one being subdivided into eight boxes. In this way a tree with eight branches per node is obtained. An alternative to the "octree", closer to the quick sort, consists in successively dividing space in two equal parts, according to directions X, Y, or Z. This operation is renewed for each of the two resulting parts as long as some segments or nodes are found in the space concerned. The main advantage to this algorithm, as compared to sorting by boxes, lies in the fact that its performance is affected neither by the irregularity of the mesh, nor by the irregularity of the model. The cost of this algorithm is logarithmic (N*Log(N)).

Figure 8.4.3 Possible Impacts of Node n_1



In order to illustrate the 3D quick sort, let us consider a search for node-segment proximity. At each step, space is successively divided into two equal parts, according to directions X, Y or Z. The group of nodes is thus separated into two subsets. We thereby obtain a tree organization with two branches per node. After each division, a check is made to determine whether the first of the two boxes must be divided. If so, it will be divided similar to the previous one. If not, the next branch is then checked. This recursive algorithm is identical to the regular fast sorting one.

The segments are also sorted using the spatial pivot. The result of the test can lead to three possibilities: the segment is on the left side of the pivot, on the right side of the pivot or astride the pivot. In the first two cases, the segments are treated similar to the nodes, but in the third situation, the segment is duplicated and placed on both sides.

Among the different criteria that can be used to stop the division are the following situations:

- the box does not contain any nodes
- the box does not contain any segments
- the box contains sufficiently few elements that the calculation of distance of all the couples is more economical
- the dimension of the box is smaller than a threshold

8.4.4 Contact processing

After the choice of a good sorting algorithm, a formulation for the handling of the contact has to be selected. One can distinguish three techniques ensuring the conditions of continuity during the contact:

- kinematic formulation of type master/slave.

In a contact node to segment, the slave node transmits its mass and force to the master segment and the segment transmits its speed to the node. This formulation is particularly adapted to an explicit integration scheme, provided that the nodes do not belong to a master segment. A node cannot be at the same time slave and master. This approach then cannot be used in the case of auto-contact.

- the Lagrange multipliers ensure kinematic continuity at contact.

There is no restriction as in the previous formulation but the system of equations cannot be solved in an explicit way. The Lagrange multiplier matrix has to be reversed at each cycle of computation. In the case of auto-contact, the number of points in contact can become significant and this formulation then becomes quite expensive.

- penalty methods do not ensure kinematic contact continuity, but they add springs at the contact spots.

The first advantage to this formulation is its natural integration in an explicit code. Each contact is treated like an element and integrates itself perfectly into the code architecture, even if the programming is vectorial and parallel. Contrary to the kinematic formulations, the penalty method ensures the conservation of momentum and kinetic energy during impact.

The formulation used in RADIOSS is a penalty type formulation. The choice of the penalty factor is a major aspect of this method. In order to respect kinematic continuity, the penalty spring must be as rigid as possible. If the impedance of the interface becomes higher than those of the structures in contact, some numerical rebounds (high frequency) can occur. To ensure the stability of the integration diagram, without having additional constraints, this rigidity must be low. With a too low penalty, the penetration of the nodes becomes too strong and the geometrical continuity is no longer ensured.

The compromise selected consists in using a stiffness of the same order of magnitude than the stiffness of the elements in contact. This stiffness is nonlinear and increases with the penetration, so that a node is not allowed to cross the surface.

These choices provide a clear representation of physics, without numerical generation of noise, but require the contact stiffness in the calculation of the criteria of stability of the explicit scheme to be taken into account.

8.4.5 Contact detection

After identifying the candidates for the impact, it is necessary to determine whether contact takes place and its precise localization. If for a formulation of node to node contact the detection of the contact is quite easy, it becomes more complex in the case of a node to segment or edge to edge contact. In the case of edge-to-edge contacts, a direct solution is possible if the segments are planar. If not, it is better to triangulate one of the segments, which would then turn it into a node-to-segment contact problem.

The search algorithm for candidates is uncoupled from the rest of the processing of the interfaces. This is not the case with regard to the detection, localization and processing of the contact. These last three tasks significantly overlap with each other so we will limit ourselves to the processing of the contact by penalty for simplicity.

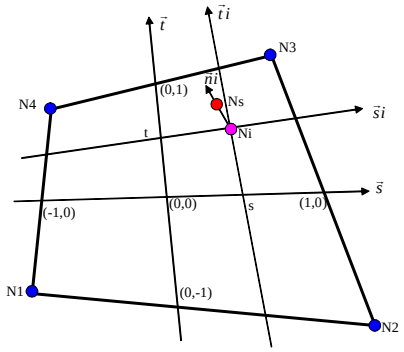
In the case of contact between two solid bodies modeled with 3D finite elements, contact can only take place on the segments of the external surface. This external surface has a certain orientation and the impact of a node can come only from the outside. Most of the node-to-surface contact algorithms use this orientation to simplify detection of contact. In the case of impact of a two-dimensional structure modeled with shell or plate finite elements, contact is possible on both sides of the surface.

For an oriented surface, it is necessary to consider contacts of the positive dimension side of the surface on itself, contacts of the negative dimension side on itself, and the contact of the “positive” side on the “negative” one. This last situation, which is quite rare, can occur in the case of a surface rolled up into itself or during the impact of a twisted surface.

8.4.5.1 Node to segment contact

The use of a "gap" surrounding the segment is one way of providing physical thickness to the surface and makes it possible to distinguish the impacts on the top or on the lower part of the segment. The contact is activated if the node penetrates within the gap or if the distance from the node to the segment becomes smaller than the gap.

To calculate the distance from the node to the segment, we make a projection of this node on the segment and measure the distance between the node and the projected point.



The projected point is calculated using isoparametric coordinates for a quadrangular segment and barycentric coordinates for a triangular segment. In the case of any quadrangular segment, the exact calculation of these coordinates leads to a system of two quadratic equations that can be solved in an iterative way. The division of the quadrangular segment into four triangular segments makes it possible to work with a barycentric coordinate system and gives equations that can be solved in an analytical way.

From the isoparametric coordinates (*s*,*t*) of the projected point (*Ni*), we have all the necessary information for the detection and the processing of contact. The relations needed for the determination of *s* and *t* are as follows: the vector *NiNs* is normal to the segment at the point *Ni*; and the normal to the segment is given by the vectorial product of the vectors *si* and *ti*.

$$H1 = (1-s)(1-t)/4$$
$$H2 = (1+s)(1-t)/4$$
$$H3 = (1+s)(1+t)/4$$
$$H4 = (1-s)(1+t)/4$$
$$\overrightarrow{ONi} = H1\overrightarrow{ON1} + H2\overrightarrow{ON2} + H3\overrightarrow{ON3} + H4\overrightarrow{ON4}$$
$$\vec{s}i = (1-t)\overrightarrow{N1N2} + (1+t)\overrightarrow{N4N3}$$
$$\vec{t}i = (1-s)\overrightarrow{N1N4} + (1+s)\overrightarrow{N2N3}$$
$$\vec{n}i = (\vec{s}i \times \vec{t}i)/(\overrightarrow{\vec{s}i \times \vec{t}i})$$
$$\overrightarrow{NiNs} = a\,\vec{n}i$$

After bounding the isoparametric coordinates between +1 or -1, the distance from the node to the segment and the penetration are calculated:

$$D = \overrightarrow{NiNs}$$
$$P = \max(0, Gap - D)$$

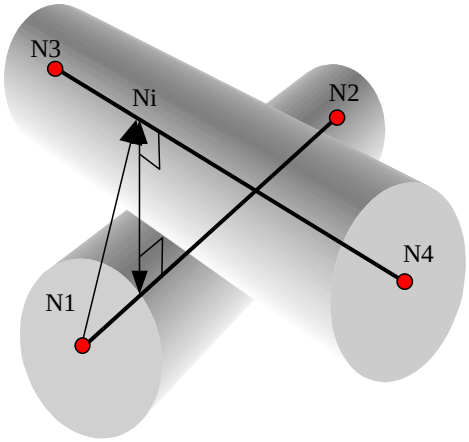
A penalty force is deducted from this. It is applied to the node *Ns* and distributed between the four nodes (*N1*, *N2*, *N3*, *N4*) of the segment according to the following shape functions:

8.4.5.2 Edge to edge contact

The formulation of edge-to-edge contact is similar to that of node-to-segment contact. The gap surrounding each edge defines a cylindrical volume. The contact is detected if the distance between the two edges is smaller than the sum of the gaps of the two edges. The distance is then calculated as follows:

$$2\,\overrightarrow{N1Ni} = (1-s)\overrightarrow{N1N3} + (1+s)\overrightarrow{N1N4}$$
$$D = \left|\overrightarrow{N1Ni} - \left(\overrightarrow{N1Ni}\cdot\overrightarrow{N1N2}/\overrightarrow{N1N2}^2\right)\overrightarrow{N1N2}\right|$$
$$\partial D/\partial s = 0$$

The force of penalty is calculated as in node-to-segment contact. It is applied to the nodes N1, N2, N3, N4 and



therefore ensures the equilibrium of forces and moments.

8.5 Type 3 - Solid and Shell Element Contact - No Gap

The main use of this interface is with shell or solid plates that are initially in contact.

There are no master and slave surfaces in this interface. Each surface is considered as if it were a slave.

8.5.1 Limitations

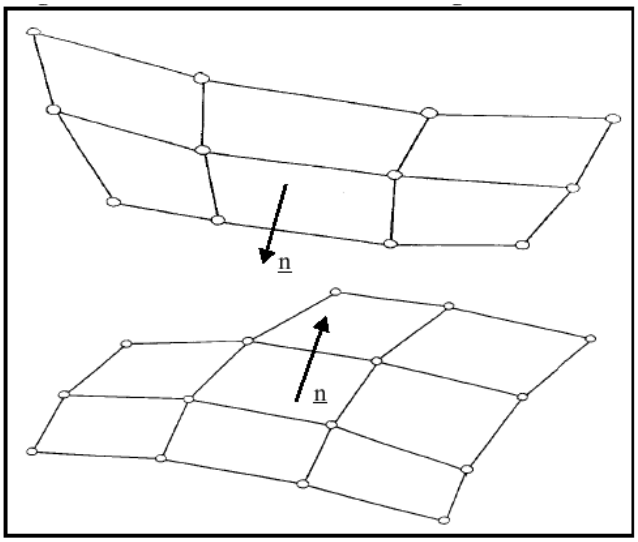
The main limitations of type 3 interfaces are:

- The two surfaces should be simply convex.
- The surface normals must face each other.
- A node may not exist on the master and slave side of an interface simultaneously.
- Surfaces must consist of either shell or brick elements.

It is recommended that the two surface meshes be regular with a good aspect ratio. The interface gap should be kept small, if not zero.

There are some search problems associated with this interface.

Figure 8.5.1 Surfaces 1 and 2 with Facing Normals

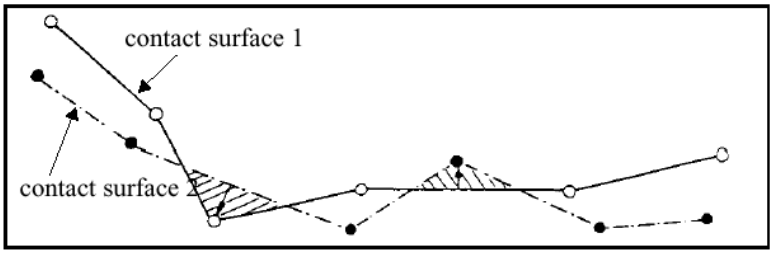


8.5.2 Computation Algorithm

The computation and search algorithms used for type 3 interface are the same as for type 5. However, type 3 interface does not have a master surface, so that the algorithms are applied twice, one for each surface.

The surfaces are treated symmetrically, with all nodes allowed to penetrate the opposing surface. The interface spring stiffness applies the opposing penetration reduction force.

Figure 8.5.2 Contact Surfaces Treated Symmetrically



Because the computation algorithm is performed twice, accuracy is improved over a type 5 interface. However, the computational cost is increased.

The first pass solution solves the penetration of the nodes on surface 1 with respect to segments on surface 2. The second pass solves surface 2 nodes with respect to surface 1 segments.

8.5.3 Interface Stiffness

When two surfaces contact, a massless stiffness is introduced to reduce the penetration's nodes of the other surface into the surface.

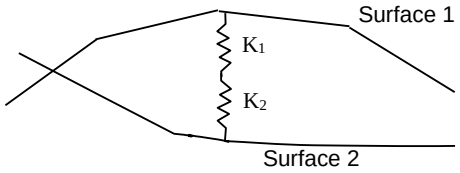
The nature of the stiffness depends on the type of interface and the elements involved.

The introduction of this stiffness may have consequences on the time step, depending on the interface type used.

The type 3 interface spring stiffness *K* is determined by both surfaces. To retain solution stability, stiffness is limited by a scaling factor which is user defined on the input card. The default value (and recommended value) is 0.2.

The overall interface spring stiffness is determined by considering two springs acting in series.

Figure 8.5.3 Interface Springs in Series



The equation for the overall interface spring stiffness is:

$$K = s\frac{K_1 K_2}{K_1 + K_2} \qquad \text{EQ. 8.5.3.1}$$

where s = Stiffness Scaling Factor. Default is 0.2.

K_1 = Surface 1 Stiffness

K_2 = Surface 2 Stiffness

K = Overall Interface Spring Stiffness

The scale factor, s, may have to be increased if:

$K_1 << K_2$ or $K_2 << K_1$

The calculation of the spring stiffness for each surface is determined by the type of elements.

For example:

$$K_1 = \frac{1}{10}K_2 \text{ implies } K = \frac{s}{11}K_2 \text{ or } K = \frac{10s}{11}K_1$$

8.5.3.1 Shell Element

If the master interface segment is a set of shell elements, the stiffness is calculated by:

$$K = 0.5sEt \qquad \text{EQ. 8.5.3.2}$$

where E is the Modulus of Elasticity

t is the Shell Thickness

The stiffness does not depend on the shell size.

8.5.3.2 Brick Element

If the master interface segment is a set of brick elements, the stiffness is calculated by:

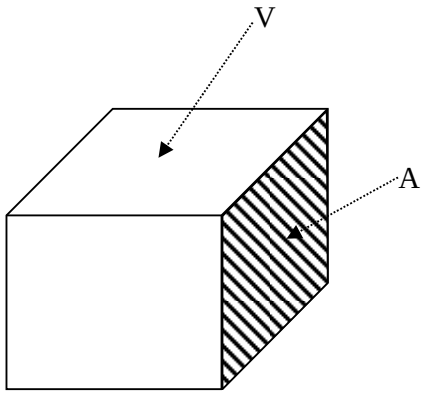
$$K = 0.5\frac{sBA^2}{V} \qquad \text{EQ. 8.5.3.3}$$

where B is the Bulk Modulus

A is the Segment Area

V is the Element Volume

Figure 8.5.4 Brick element



8.5.3.3 Combined Elements

If a segment is a shell element that is attached to the face of a brick element, the shell stiffness is used.

8.5.4 Interface Friction

Type 3 interface allows sliding between contact surfaces. Coulomb friction between the surfaces is modelled. The input card requires a friction coefficient. No value (default) defines zero friction between the surfaces.

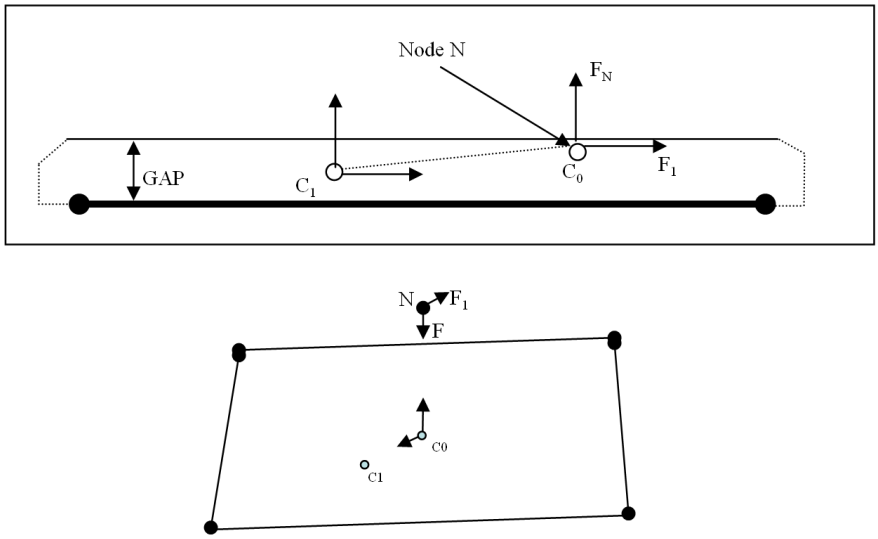
The friction on a surface is calculated by:

$$\Delta\vec{F}_t = \frac{K}{10}\overrightarrow{C_1C_0} \qquad \text{EQ. 8.5.4.1}$$

where K = Interface Spring Stiffness

$\overrightarrow{C_1C_0}$ = contact node displacement vector

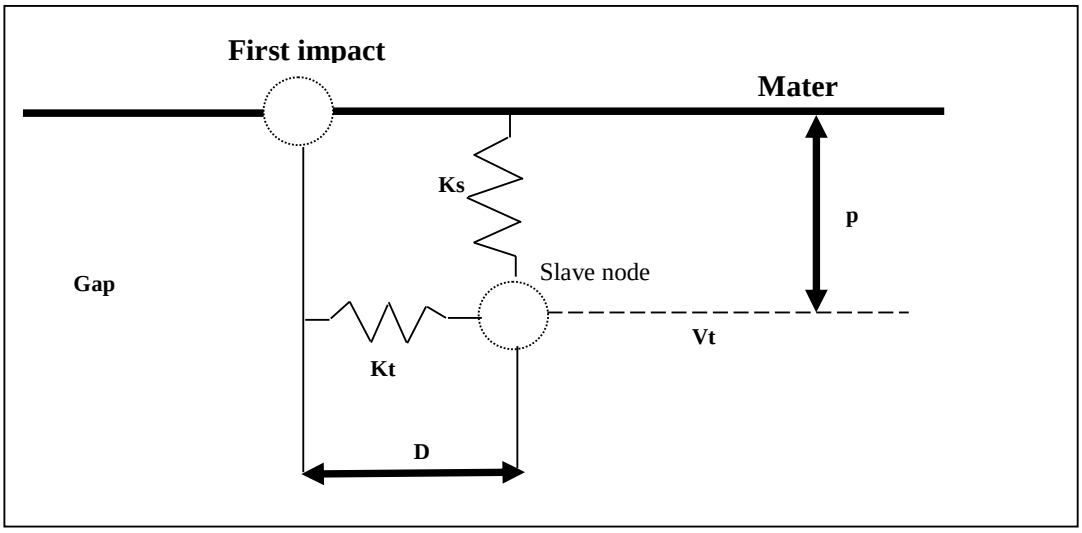
Figure 8.5.5 Coulomb Friction



C_0= Contact point at time t

C_1= Contact point at time $t + \Delta t$

Figure 8.5.6 Friction on interface type 3



The normal force computation is given by:

$$F_n = K_s P \qquad \text{EQ. 8.5.4.2}$$

where $K_s = K_0 \left(\frac{Gap}{Gap - P} \right)$

K_0 is the Initial interface spring stiffness (as in type 5)

The tangential force computation is given by:

$$F_t = K_t D \qquad \text{EQ. 8.5.4.3}$$

where $K_t = \frac{K_n}{10}$

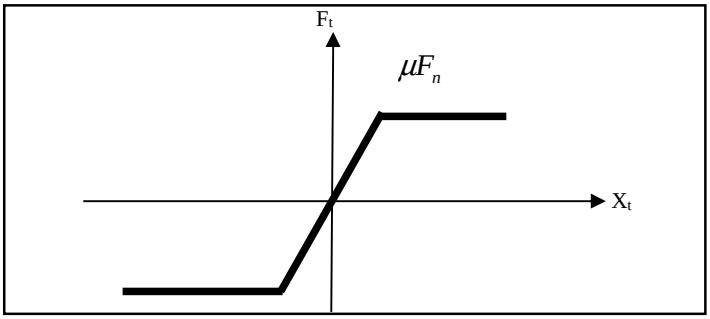
If the friction force is greater than the limiting situation, $|F_t| > \mu|F_n|$, the frictional force is reduced to equal the limit, $|F_t| = \mu|F_n|$, and sliding will occur. If the friction is less than the limiting condition, $F_t \le \mu F_n$, the force is unchanged and sticking will occur.

Time integration of the frictional forces is performed by:

$$\vec{F}_t^{new} = \vec{F}_t^{old} + \Delta\vec{F}_t \qquad \text{EQ. 8.5.4.4}$$

where $\Delta\vec{F}_t$ = result from equation 8.5.4.1

Figure 8.5.7 Friction on type 3 interface



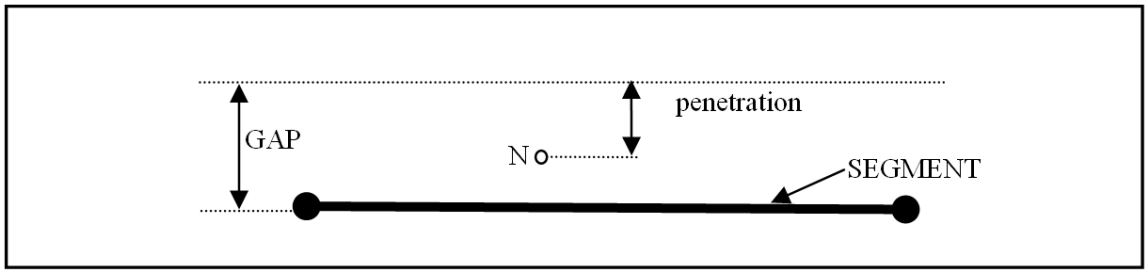
8.5.5 Interface Gap

Type 3 interfaces have a gap that determines when contact between two segments occurs. This gap is user defined, but some interfaces will calculate an automatic default gap. The gap determines the distance for which the segment interacts with the three nodes. If a node moves within the gap distance, such as nodes 1 and 2, reaction forces act on the nodes.

Type 3 interface have a gap:

- only normal to the segment, as shown in Figure 8.5.8.
- on the contact side of the segments, which is defined by the surface normal. The size of the gap defined for certain interface types is critical. If the gap is too small, the solution time step may be dramatically reduced or a node may move across the entire gap in one time step. However, if the gap is too large, nodes not associated with the direct contact may become involved.

Figure 8.5.8 Interface Gap

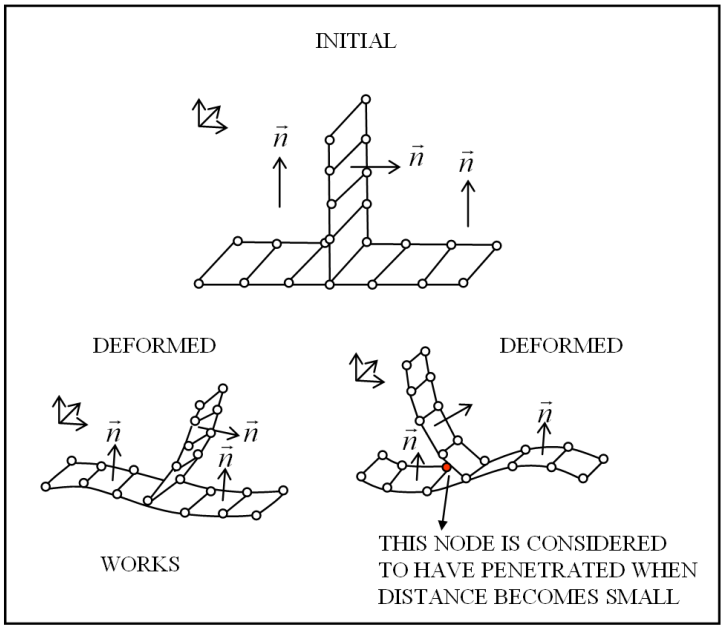


8.5.6 Interface Failure Examples

There are a number of situations in which type 3 elements may fail. A couple of these are shown below.

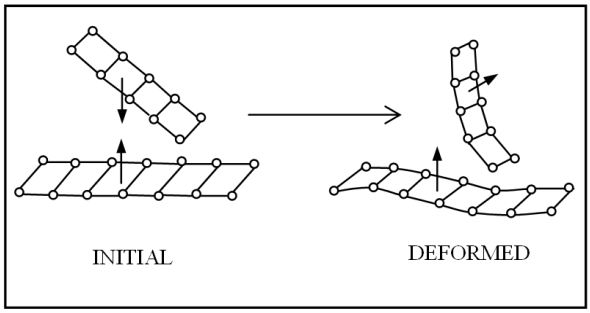
Care must be taken when defining contact surfaces with large deformation simulations. If the normal definitions of the contact surfaces are incorrect, node penetration will occur without any reaction from either surface.

Figure 8.5.9 Improper Normal Direction



Referring to Figure 8.5.9, the first situation shows the mesh deforming in a way that allows the normals to be facing each other. However, in the second case, the deformation moves two surfaces with normals all facing the same direction, where contact will not be detected. Large rotations can have a similar effect, as shown in Figure 8.5.10.

Figure 8.5.10 Initial and Deformed Mesh (Before Impact)



Kinematic motion may reposition the mesh so that normals do not correspond. It is recommended that possible impact situations be understood before a simulation is attempted.

8.6 Type 5 - General Purpose Contact

This interface is used to simulate the impact between a master surface and a list of slave nodes, as shown in Figure 8.6.1.

The penetration is reduced by the penalty method.

Another method is possible: Lagrange multipliers. But spatial distribution force is not smooth and induces hourglass deformation.

This interface is mainly used for:

- Simulation of impact between beam, truss and spring nodes on a surface.
- Simulation of impact between a complex fine mesh and a simple convex surface.
- A replacement for a rigid wall.

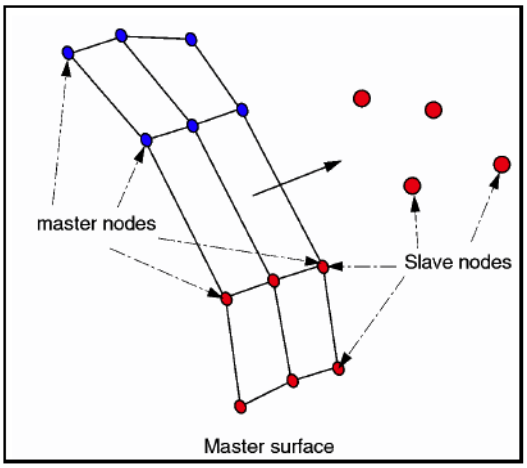
8.6.1 Limitations

The main limitations of type 5 interface are:

- The master segment normals must be oriented from the master surface towards the slave nodes.
- The master side segments must be connected to solid or shell elements.
- The same node is not allowed to exist on both the master and slave surfaces.
- Some search problems (refer to the problems A1, B1, B2, B3, C1, and D1 in chapter)

It is recommended that the master surface mesh be regular, with a good aspect ratio and that a small or zero gap be used to detect penetration.

Figure 8.6.1 Surface 1 (nodes) and surface 2 (segments)



8.6.2 Computation Algorithm

The computation and search algorithms used for type 5 are the same as for type 3. Refer to Chapter 8.5.2.

8.6.3 Interface Stiffness

When two surfaces contact, a massless stiffness is introduced to reduce the penetration's nodes of the other surface into the surface.

The nature of the stiffness depends on the type of interface and the elements involved.

The introduction of this stiffness may have consequences on the time step, depending on the interface type used.

For a type 5 interface, the spring stiffness *K* is determined by the master side only. The stiffness scaling factor default value (and recommended value) is 0.2.

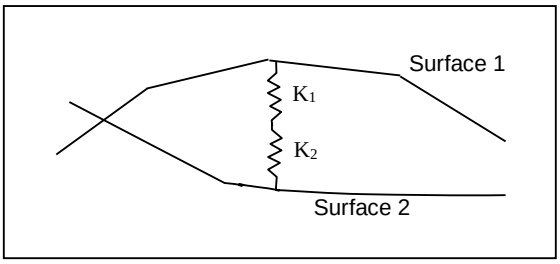
For a soft master surface material and stiff slave surface, the stiffness scaling factor should be increased by the elastic modulus ratio of the two materials.

The calculation of the spring stiffness' is the same as in a type 3 interface.

If a segment is a shell as well as the face of brick element, the shell stiffness is used.

The overall interface spring stiffness is determined by considering two springs acting in series.

Figure 8.6.2 Interface Springs in Series



The equation for the overall interface spring stiffness is:

$$K = s\frac{K_1 K_2}{K_1 + K_2} \qquad \text{EQ. 8.6.3.1}$$

where s = Stiffness Scaling Factor. Default is 0.2.

K_1 = Surface 1 Stiffness

K_2 = Surface 2 Stiffness

K = Overall Interface Spring Stiffness

The scale factor, s, may have to be increased if:

$$K_1 << K_2 \text{ or } K_2 << K_1$$

The calculation of the spring stiffness for each surface is determined by the type of elements.

For example:

$$K_1 = \frac{1}{10}K_2 \text{ implies } K = \frac{s}{11}K_2 \text{ or } K = \frac{10s}{11}K_1$$

8.6.3.1 Shell Element

Refer to Chapter 8.5.3.1.

8.6.3.2 Brick Element

Refer to Chapter 8.5.3.2.

8.6.3.3 Combined Elements

Refer to Chapter 8.5.3.3.

8.6.4 Interface Friction

Type 5 interface allows sliding between contact surfaces. Coulomb friction between the surfaces is modelled. The input card requires a friction coefficient. No value (default) defines zero friction between the surfaces.

The friction computation on a surface is the same as for type 3 interface. Refer to Chapter 8.5.4.

Darmstad and Renard models for friction are also available:

Darmstad law:

$$\mu = C_1 \cdot e^{(C_2 V)} \cdot p^2 + C_3 \cdot e^{(C_4 V)} \cdot p + C_5 \cdot e^{(C_6 V)} \qquad \text{EQ. 8.6.4.1}$$

Renard law:

$$\mu = C_1 + (C_3 - C_1) \cdot \frac{V}{C_5} \cdot \left(2 - \frac{V}{C_5}\right) \qquad \text{if } V \in [0, C_5] \qquad \text{EQ. 8.6.4.2}$$

$$\mu = C_3 - \left((C_3 - C_4) \cdot \left(\frac{V - C_5}{C_6 - C_5}\right)^2 \cdot \left(3 - 2 \cdot \frac{V - C_5}{C_6 - C_5}\right)\right) \qquad \text{if } V \in [C_5, C_6]$$

$$\mu = C_2 - \frac{1}{\frac{1}{C_2 - C_4} + (V - C_6)} \qquad \text{if } V \geq C_6$$

where, $C_1 = \mu_s, C_2 = \mu_d$

$C_3 = \mu_{\max}, C_4 = \mu_{\min}$

$C_5 = V_{cr1}, C_6 = V_{cr2}$

Possibility of smoothing the tangent forces via a filter:

$$4F_{\mathrm{T}} = \alpha \cdot F_{\mathrm{T}}^{t} + (1 - \alpha) \cdot F_{\mathrm{T}}^{t-1} \qquad \text{EQ. 8.6.4.3}$$

where the coefficient a depends on the Ifiltr flag value.

8.6.5 Interface Gap

Refer to Chapter 8.5.5 for type 3 interface.

8.6.6 Interface Algorithm

The algorithm used to calculate interface interaction for each slave node is:

1. Determine the closest master node.
2. Determine the closest master segment.
3. Check if the slave node has penetrated the master segment.
4. Calculate the contact point.
5. Compute the penetration.
6. Apply forces to reduce penetration.

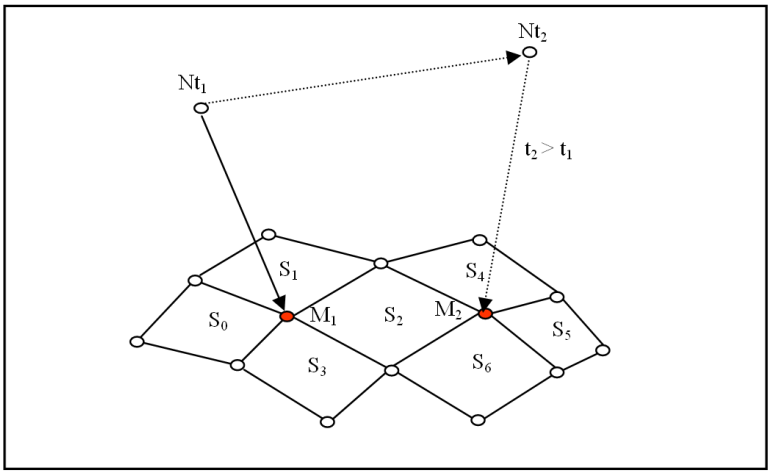
For more information, refer to Chapter 8.4.

8.6.6.1 Detection of Closest Master Node

The RADIOSS Starter and Engine use different methods to determine the closest master node to a particular slave node.

The Starter searches for the master node with the minimum distance to the particular slave node. The Engine carries out the following algorithm, referring to Figure 8.6.3.

Figure 8.6.3 Search Method



1. Get the previous closest segment, S_0, to node N at time t_1.
2. Determine the closest node, M_1, to node N which belongs to segment S_0.
3. Determine the segments connected to node M_1 (S_0, S_1, S_2 and S_3).
4. Determine the new closest master node, M_2, to node N at time t1. The new master node must belong to one of the segments S_0, S_1, S_2 or S_3.
5. Determine the new closest master segment (S_2, S_4, S_5 and S_6).

The Starter CPU cost is calculated with the following equation:

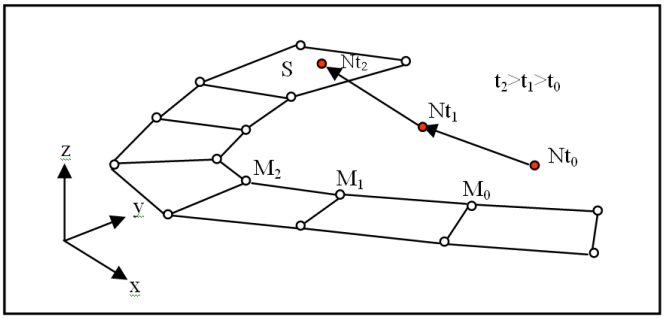
CPUstarter = a x Number of Slave Nodes x Number of Master Nodes

The Engine CPU cost is calculated with the following equation:

CPUengine = b x Number of Slave Nodes

The algorithm used in the engine is less expensive but it does not work in some special cases. In Figure 8.6.4, if node N is moving from $N(t_0)$ to $N(t_1)$ and then to $N(t_2)$, the closest master nodes are M_0 and M_1. When the final node movement to $N(t_2)$ is taken, the impact on segment S will not be detected since none of the nodes on this segment are considered as the closest master node.

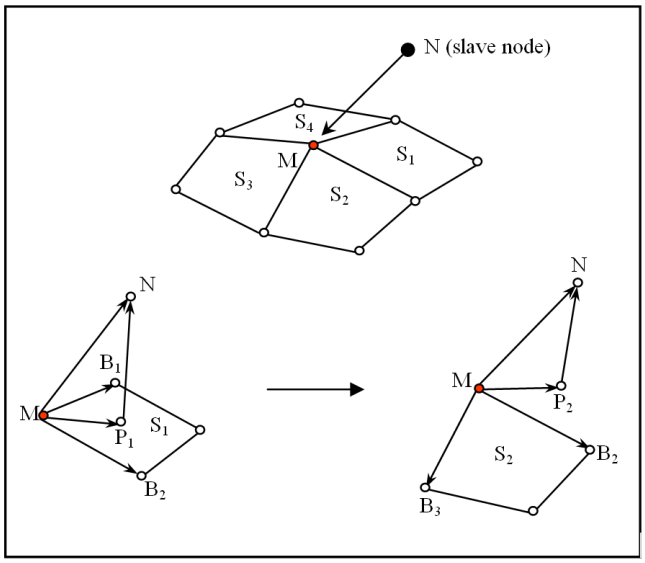
Figure 8.6.4 Undetected Impact



8.6.6.2 Detection of Closest Master Segment

The closest master segment to a slave node N, shown in Figure 8.6.5, is found by determining a reference quantity, A.

Figure 8.6.5 Closest Master Segment Determination Method



The A value for segment S_1 is given by:

$$A = \left(M\vec{B}_2 \times M\vec{P}_1\right)\left(M\vec{B}_1 \times M\vec{P}_1\right) < 0 \qquad \text{EQ. 8.6.6.1}$$

where P_1 = Projection of N on plane (M B_1 B_2)

B_1 and B_2 = Tangential Surface Vectors along segment S_1 edges.

The A value for segment S_2 is given by:

$$A = \left(M\vec{B}_3 \times M\vec{P}_2\right)\left(M\vec{B}_2 \times M\vec{P}_2\right) > 0 \qquad \text{EQ. 8.6.6.2}$$

where P_2 = Projection of N on plane (M B_2 B_3)

The same procedure is carried out for all master segments that node M is connected to.

The closest segment is the segment for which A is a minimum.

In some special cases (curved surfaces), it is possible that:

1. All values of A are positive.

2. More than one value of A is negative.

8.6.6.3 Detection of Penetration

Penetration is detected by calculating the volume of the tetrahedron made by slave node N and the master nodes of the corresponding master segment, as shown in Figure 8.6.6.

For a given normal n, the sign of the volume shows if penetration has occurred.

Figure 8.6.6 Tetrahedron used for Penetration Detection

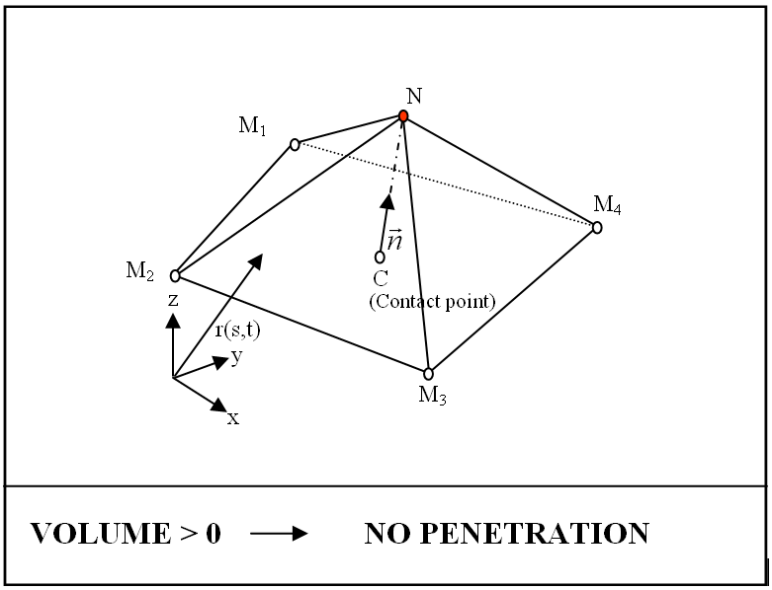
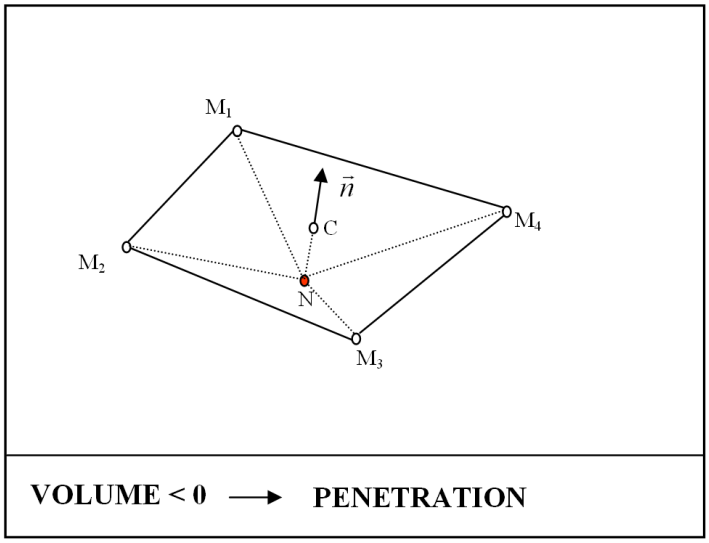


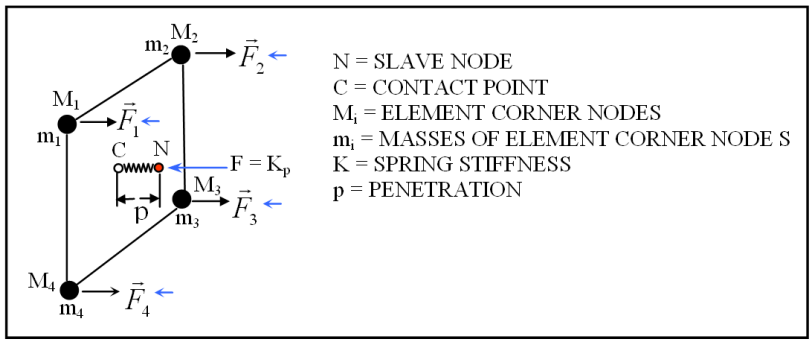
Figure 8.6.7 Negative Volume Tetrahedron - Penetrated Node



8.6.6.4 Reduction of Penetration

The penetration, p, is reduced by the introduction of a massless spring between the node, N, and the contact point, C.

Figure 8.6.8 Forces Associated with Penetration



The force applied on node N in direction $\overrightarrow{NC}$ is:

$$F = Kp \qquad \text{EQ. 8.6.6.3}$$

where K = Interface Spring Stiffness

Reaction forces F_1, F_2, F_3 and F_4 are applied on each master node (as shown in Figure 8.6.8) in the opposite direction to the penetration force, such that:

$$F_1 + F_2 + F_3 + F_4 = -F \qquad \text{EQ. 8.6.6.4}$$

Forces F_i (i = 1, 2, 3, 4) are functions of the position of the contact point, C. They are evaluated by:

$$F_i = \frac{d_i}{\sum_{j=1}^{4} d_j} F \qquad \text{EQ. 8.6.6.5}$$

where $d_i = \dfrac{Ni(S_c, t_c)}{\sum_{j=1}^{4} N_j^2(S_c, t_c)} m_i$

or more simply:

$$F_i = N_i(S_c, t_c) F \qquad \text{EQ. 8.6.6.6}$$

where N_i = Standard linear quadralateral interpolation functions

S_c and t_c = Isoparametric coordinates contact point

The penalty method is used to reduce the penetration. This provides:

- Accuracy
- Generality
- Efficiency
- Compatibility

8.7 Type 6 - Rigid Body Contact

This interface is used to simulate impacts between two rigid bodies. It works like type 3 interface except that the total interface force is a user defined function of the maximum penetration. The input and computational algorithms are the same as for type 3 interfaces. This interface is used extensively in vehicle occupant simulations, eg. knee bolsters.

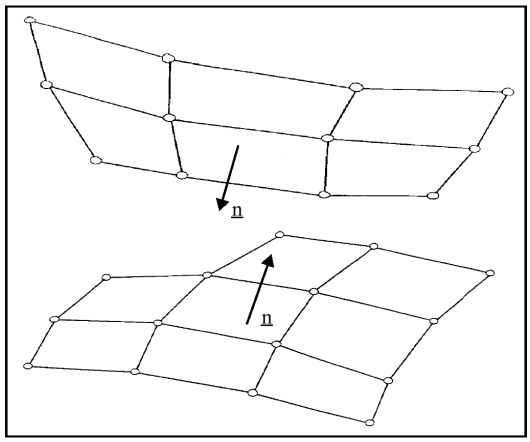
8.7.1 Limitations

Some of the main limitations for this interface type are:

- Surface 1 must be part of one and only one rigid body.
- Surface 2 must be part of one and only one rigid body.
- The interface stiffness (user defined function) can reduce the time step.

Other limitations are the same as for type 3 interfaces.

Figure 8.7.1 Surfaces 1 and 2 with Facing Normals



8.7.2 Interface Stiffness

When two surfaces contact, a massless stiffness is introduced to reduce the penetration's nodes of the other surface into the surface.

The nature of the stiffness depends on the type of interface and the elements involved.

If a segment is a shell as well as the face of brick element, the shell stiffness is used.

8.7.3 Interface Friction

Type 6 interface allows sliding between contact surfaces. Coulomb friction between the surfaces is modelled. The input card requires a friction coefficient. No value (default) defines zero friction between the surfaces. The friction computation on a surface is the same as for type 3 interface (refer to Chapter 8.4.4).

8.7.4 Interface Gap

Refer to Chapter 8.4.5 for type 3 interface.

8.7.5 Time Step Calculation

The stable time step used for time integration equations is computed by:

$$\Delta t = 0.1\sqrt{\frac{M}{K}} \qquad \text{EQ. 8.7.5.1}$$

where M = min(M *rigid body 1* , M *rigid body 2*)

K = Tangent of user force function

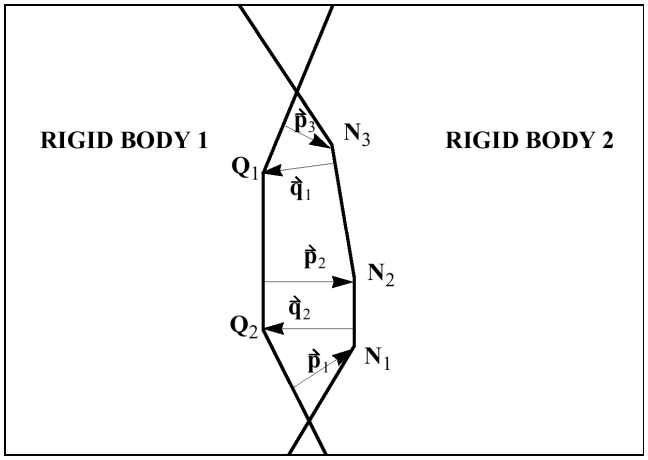
Time step Δt is affected by the actual stiffness derived from function *f*:

$$\Delta t \le \sqrt{\frac{M}{f'(p)}} \qquad \text{EQ. 8.7.5.2}$$

The function *f* refers to a function number given in input and must be provided by user.

8.7.6 Contact force

Figure 8.7.2 Specially suited for rigid bodies



$$F_{N_i} = \frac{-\vec{p}_i + \vec{q}_{N\,i}}{\left\|\sum \vec{p}_j - \sum \vec{q}_k\right\|} f\left(MAX\left(\left\|\vec{p}_k\right\|, \left\|\vec{q}_j\right\|\right)\right) \qquad \text{EQ. 8.7.6.1}$$

$\vec{q}_{N_i}$ is the contribution to node N_i of vector $\vec{q}_i$ distributed on the segment penetrated by node Q_i.

8.8 Type 7 - General Purpose Contact

This interface simulates the most general type of contacts and impacts. Type 7 interface has the following properties:

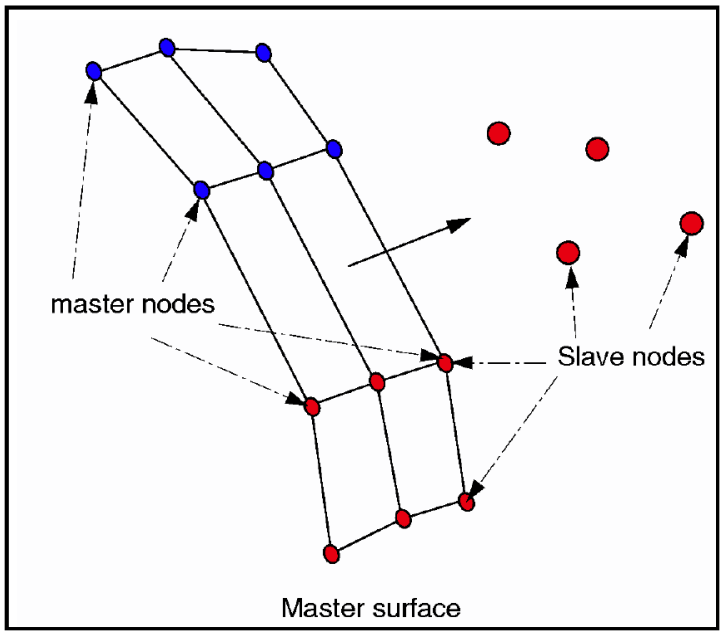
1. Impacts occur between a master surface and a set of slave nodes, similar to type 5 interface.
2. A node can impact on one or more master segments.
3. A node can impact on either side of a master surface.
4. Each slave node can impact each master segment except if it is connected to this segment.
5. A node can belong to a master surface and a set of slave nodes, as shown in Figure 8.8.1.
6. A node can impact on the edge and corners of a master segment. None of the previous interfaces allow this.
7. Edge to edge contacts between master and slave segments are not solved by this interface.

8.8.1 Limitations

All limitations encountered with interface types 3, 4 and 5 are solved with this interface.

It is a fast search algorithm without limitations.

Figure 8.8.1 Slave and Master Node Impact



There are no search limitations with this interface concerning node to surface contacts. All possible contacts are found.

There is no limitation on the use of large and small segments on the same interface. This is recommended to have a good aspect ratio elements or a regular mesh to obtain reasonable results; however, it is not an obligation.

There is no limitation to the size of the spring stiffness factor. The spring stiffness is much greater than interfaces 3 and 5, with the default stiffness factor set to 1.0. This is to avoid node penetrations larger than the gap size, removing problems that were associated with the other interfaces.

8.8.2 Interface Stiffness

When two surfaces contact, a massless stiffness is introduced to reduce the penetration of one surface node to the other surface.

The nature of the stiffness depends on the type of interface and the elements involved.

The introduction of this stiffness may have consequences on the time step, depending on the interface type used.

The interface spring stiffness calculation is not as simple as for types 3, 4 and 5. The initial stiffness is calculated using the methods for type 3 interfaces. However, after initial penetration, the stiffness is given as a function of the penetration distance and the rate of penetration.

A critical viscous damping coefficient given on the input card (*visc*) allows damping to be applied to the interface stiffness.

$$F = f(p) + visc\sqrt{2KM}\,\frac{dp}{dt} \qquad \text{EQ. 8.8.2.1}$$

The stiffness is much larger than the other interfaces to accommodate high speed impacts with minimal crossing of surfaces. The consequence of this is that a stable time step is calculated to maintain solution stability.

8.8.3 Interface Friction

Type 7 interface allows sliding between contact surfaces. Coulomb friction between the surfaces is modelled. The input card requires a friction coefficient. No value (default) defines zero friction between the surfaces.

In type 7 interface a critical viscous damping coefficient is defined, allowing viscous damped sliding.

The friction on a surface may be calculated by two methods. The first method suitable for contact tangential velocity greater that *1m/s* consist in computing a viscous tangential growth by:

$$\Delta\vec{F}_t = C_t\vec{V}_t \qquad \text{EQ. 8.8.3.1}$$

In the second method an artificial stiffness K_S is input. The change of tangent force F_t is obtained the following equation:

$$\Delta F_t = K_S\ \delta_t \qquad \text{EQ. 8.8.3.2}$$

where δ_t is the tangent displacement.

The normal force computation is given by:

$$F_n = K_s P + C\frac{dP}{dt} \qquad \text{EQ. 8.8.3.3}$$

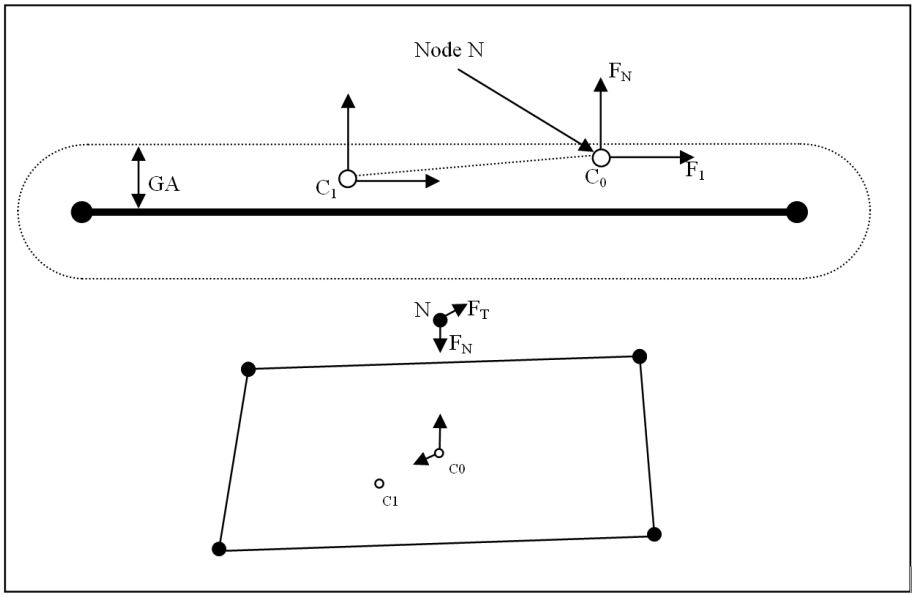
where $K_s = K_0\left(\frac{Gap}{Gap - P}\right)$

$$C = VIS_s\sqrt{2K_sM}$$

K_0 = Initial interface spring stiffness (as in type 5)

VIS_s = Critical damping coefficient on interface stiffness (input)

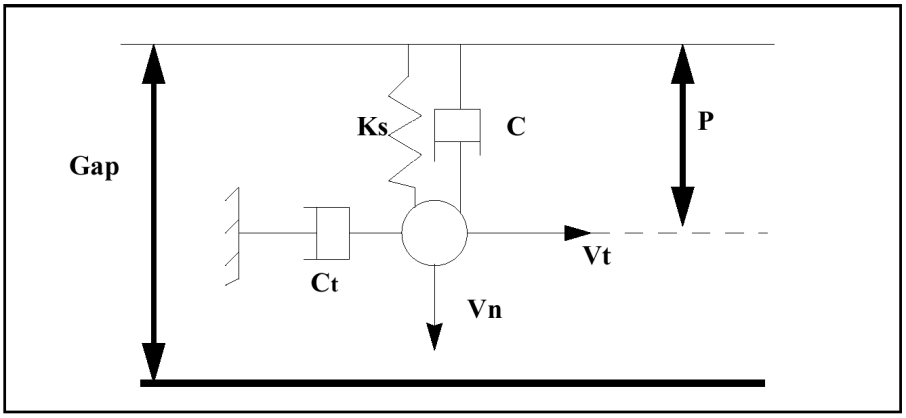
Figure 8.8.2 Coulomb Friction



C0 = Contact Point at time t

C1 = Contact Point at time $t = t + \Delta t$

Figure 8.8.3 Friction on type 7 interface - scheme



The tangential force computation is given by:

$$F_t = \min\left(FRIC * F_n, F_{ad}\right) \qquad \text{EQ. 8.8.3.4}$$

where $F_{ad} = C_t V_t$

$C = VIS_F \sqrt{2K_s M}$

F_{ad} = adhesion force

VIS_F = Critical damping coefficient on interface friction (input)

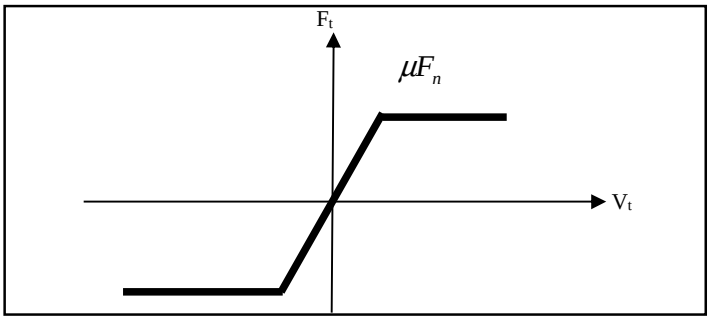
If the friction force is greater than the limiting situation, $|F_t| > \mu |F_n|$, the frictional force is reduced to equal the limit, $|F_t| = \mu |F_n|$, and sliding will occur. If the friction is less than the limiting condition, $F_t \le \mu F_n$, the force is unchanged and sticking will occur. Note that the friction coefficient μ may be obtained by Coulomb, Darmstad and Renard models as described in section 8.6.4.

Time integration of the frictional forces is performed by:

$$\vec{F}_t^{new} = \vec{F}_t^{old} + \Delta\vec{F}_t \qquad \text{EQ. 8.8.3.5}$$

where $\Delta\vec{F}_t$ is obtained from EQ. 8.8.3.1 or EQ. 8.8.3.2.

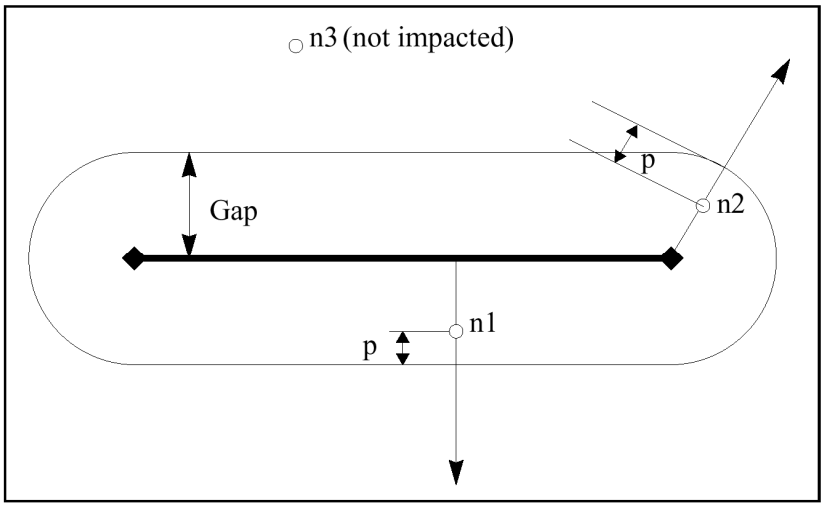
Figure 8.8.4 Friction on type 7 interface



8.8.4 Interface Gap

Type 7 interfaces have a gap that determines when contact between two segments occurs. This gap is user defined, but some interfaces will calculate an automatic default gap. Shown in Figure 8.8.2 is a segment type 7 interface with three nodes in close proximity. The gap, as shown, determines the distance for which the segment interacts with the three nodes. If a node moves within the gap distance, such as nodes 1 and 2, reaction forces act on the nodes.

Figure 8.8.5 Type 7 Interface Gap



Type 7 interface has a gap that covers both edges of the segments, as shown in Figure 8.8.5.

8.8.5 Time Step

A time step is calculated to maintain stability when a type 7 interface is used.

The kinematic or interface time step is calculated if $\frac{dP}{dt} > 0$ by:

$$\Delta t_{\min} = 0.5\left[\frac{Gap - p}{\frac{dP}{dt}}\right] \qquad \text{EQ. 8.8.5.1}$$

The stable time step or nodal time step is given by:

$$\Delta t_{nod} = \sqrt{\frac{2M}{K}} \qquad \text{EQ. 8.8.5.2}$$

where, M = Nodal mass

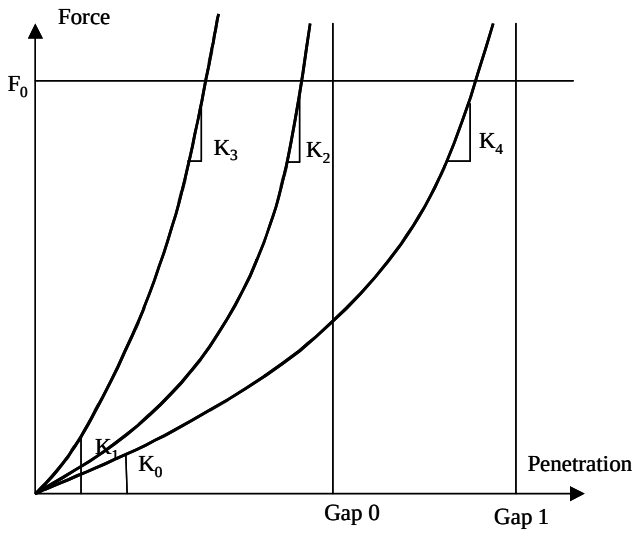
$K = \sum\left(K_{\text{int}er} + K_{el}\right)$ = Nodal stiffness

The time step used for the interface is the smaller of the two. If the interface spring stiffness is too great, the time step can be reduced dramatically. If the two materials involved in the contact are the same, the default interface stiffness factor can be retained. This is the case when modelling sheet metal. However, the stiffness factor may need adjustment if the two materials stiffness' vary too much; for example, steel and foam.

8.8.5.1 Methods to Increase Time Step

The time step can be altered by two different methods, by altering the size of the gap and by increasing the initial stiffness. Figure 8.8.6 shows three force-penetration curves for a type 7 interface. Both methods change the nature of the stiffness which affects the time step.

Figure 8.8.6 Force - Penetration Curves



Using a larger gap size, curves 1 and 2 keep the same initial stiffness; hence the initial time step remains the same. Since the impact slowing force is applied over a greater distance, the stiffness is not changed as much, but increases.

Increasing the initial interface stiffness, although decreasing the time step initially; will increase the time step if penetration is large.

8.8.6 Detection and Gap Size

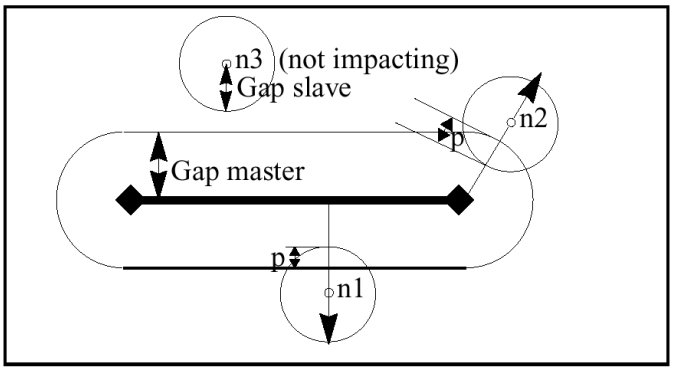
A slave node can be detected near a master segment from all directions, as shown in Figure 8.8.7.

The size of the gap can be user defined, but RADIOSS automatically calculates a default gap size, based on the size of the interface elements. For shell elements, the computed gap is the average thickness. For brick elements it is equal to one tenth of the minimum side length.

8.8.7 Variable gap

By default the gap is constant on all master segments. If the variable gap option is activated, a different gap is used for each contact taking into account the physical thickness on the master and slave sides.

Figure 8.8.7 Variable gap



For shell elements, the master gap is equal to one half of the shell thickness. The slave gap is equal to one half of the largest thickness of all connected shell elements.

For solid elements, the master gap is zero. If the slave node is only connected to solid elements, the slave gap is zero.

For beam or truss elements connected to the slave node, the slave gap is one half of the square root of the section area.

If a slave node is connected to different elements (shell, brick, beam, and truss) the largest gap value is used.

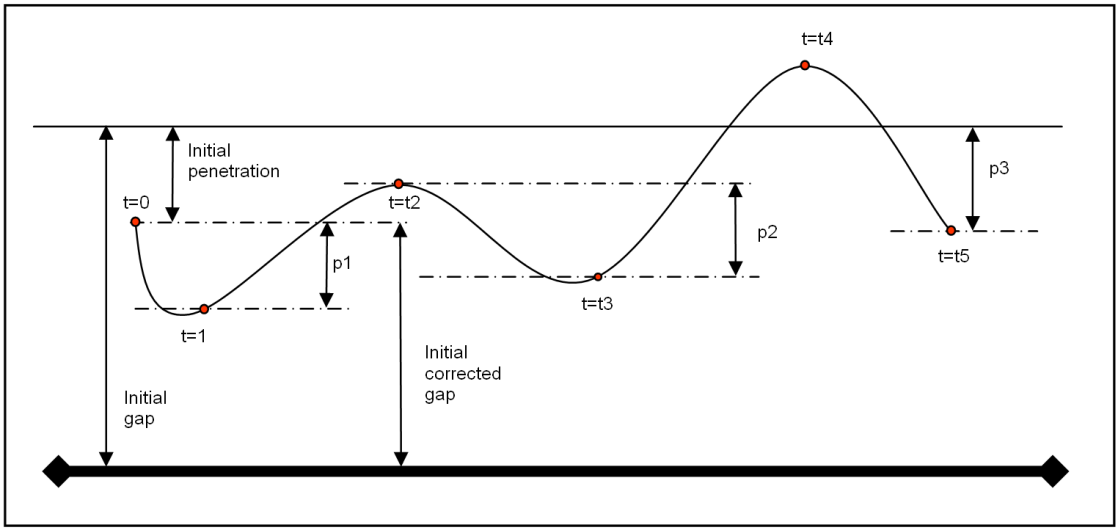
The total gap is the sum of the slave and master gaps. The total gap cannot be smaller than a minimum gap (user input gap).

8.8.8 Gap correction for nodes with initial penetration

Type 7 interface is very sensitive to initial penetrations. One method for solving the resulting problems is to use an automatic gap correction (INACTI = 5).

With automatic gap correction the effective gap is corrected to take into account the initial penetration. The correction is only applied to the initially penetrated nodes. If the node penetration decreases, the correction is reduced. The computed penetration is illustrated in Figure 8.8.8.

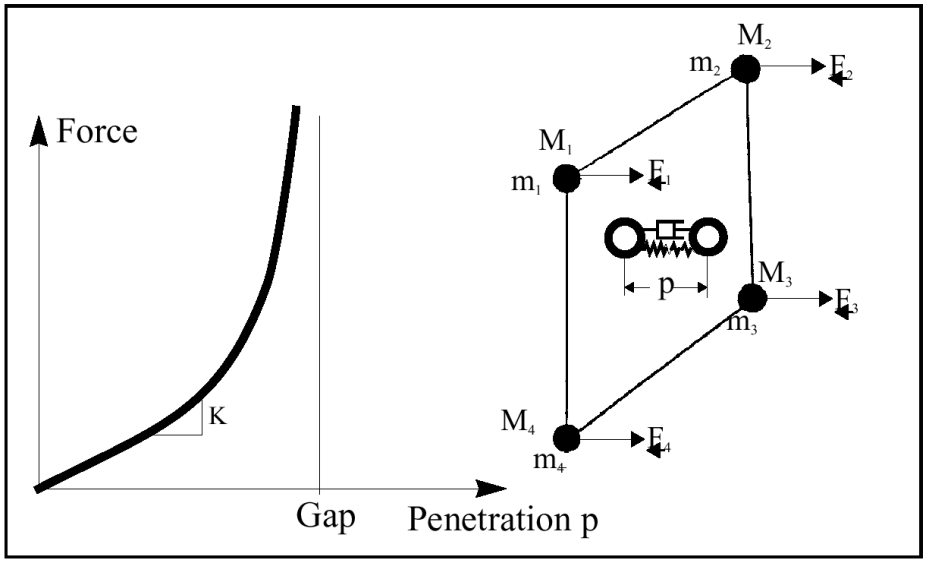
Figure 8.8.8 Corrected gap



8.8.9 Penetration Reaction

Like the other interface types, type 7 has a spring stiffness as a slave node penetrates the interface gap (previous section). However, there are some fundamental differences in the determination of the reaction force. Figure 8.8.9 shows a graph of force versus penetration of a node on a master segment. This figure also shows a pictorial diagram of node penetration and the associated forces.

Figure 8.8.9 Penetration Reaction Force



The reaction force is not a linear relation like the previous interfaces. There is a viscous damping which acts on the rate of penetration.

The force computation is given by:

$$F = K_s P + C\frac{dP}{dt} \qquad \text{EQ. 8.8.9.1}$$

where $K_s = K_0\left(\frac{Gap}{Gap - P}\right)$

$$C = VIS_s\sqrt{2K_sM}$$

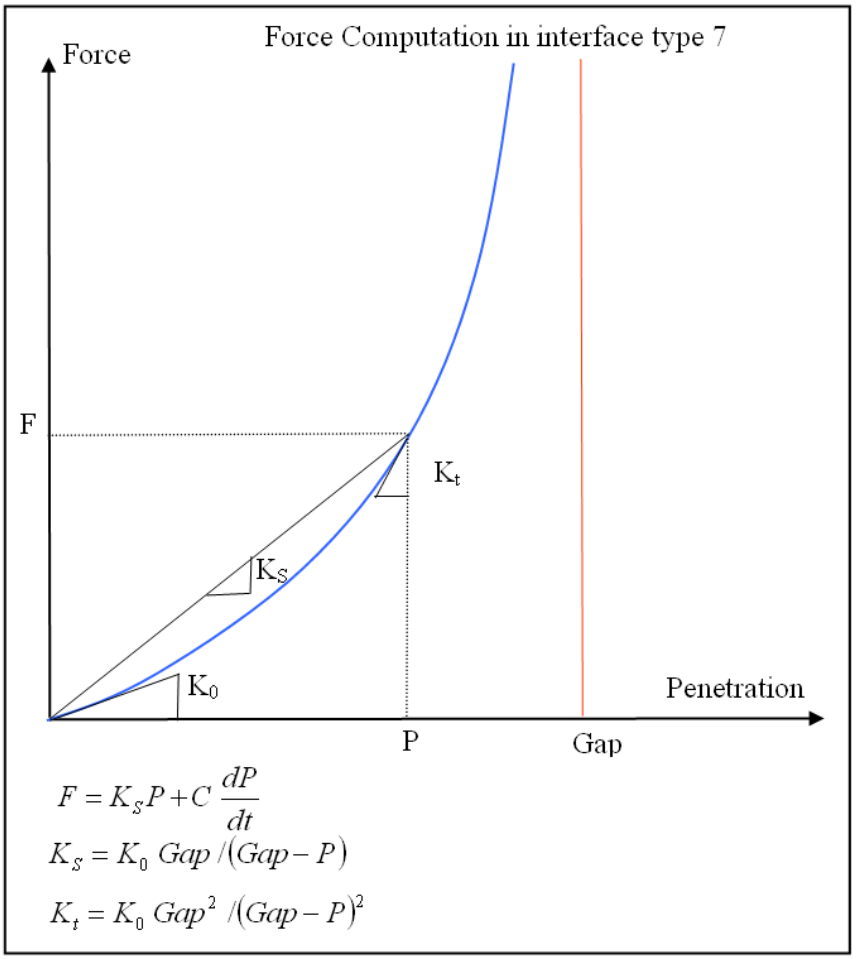
K_0 = Initial interface spring stiffness (as in type 5)

VIS_s = Critical damping coefficient on interface stiffness (input)

The instantaneous stiffness is given by:

$$K_t = K_0\left(\frac{Gap^2}{(Gap - P)^2}\right) \qquad \text{EQ. 8.8.9.2}$$

Figure 8.8.10 Force - Penetration Graph

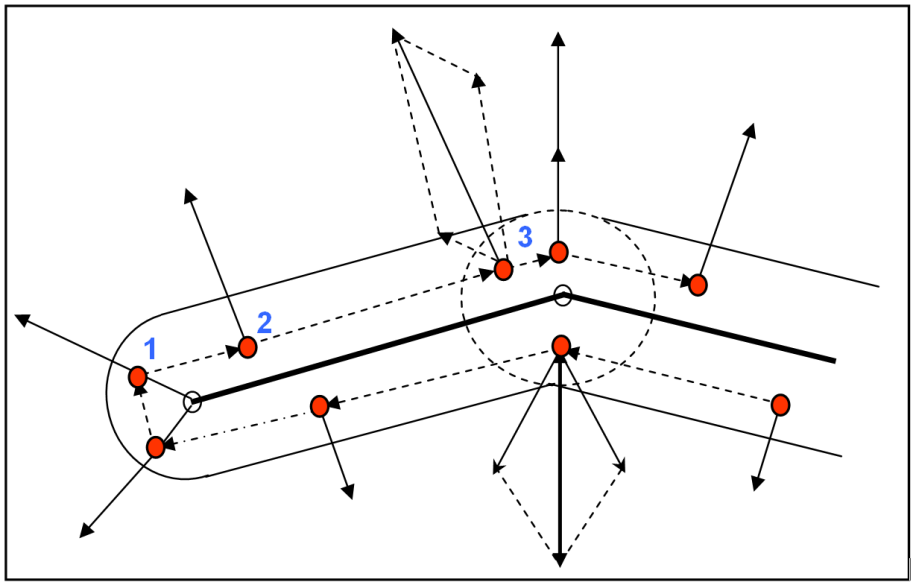


A critical viscous damping is required to be defined on the type 7 input card for both damping on the spring stiffness and for interface friction damping.

8.8.10 Force Orientation

Due to the gap on a type 7 interface extending around the edges of a segment, the reaction forces over a surface will be smooth. Figure 8.8.10 shows the reaction forces on a node at various positions around two adjoining segments.

Figure 8.8.11 Force Orientation



Position 1 in Figure 8.8.11 shows the force acting radially from the edge of the segment. The size of the force depends on the amount of penetration. At position 2 the force is normal to the segment surface. In position 3 two segments intersect and their gaps overlap. The result is that each segment applies a force to the node, normal to the respective segment, this may double the force for the distance of gap overlap.

8.8.11 Interface hints

8.8.11.1 Main problem

One main problem remains namely:

- deep penetrations are not easily tolerated

They lead to high penalty forces and stiffness', and consequently to a drop in time step.

When such a problem occurs, you may see:

- a very small time step
- an infinite loop message (most likely due to divergence)
- a large contact force vector in animation

Deep penetrations (i.e. close to gap value) cannot be avoided in most car crash simulations.

They occur in the following cases:

- initial penetrations of adjacent plates
- edge impacts: wrong side contacts
- full collapse of one structural region
- rigid body impact on another rigid body or on fixed nodes or on very stiff structures
- impact between heavy stiff structures
- high impact speed
- small gap

The elastic contact force is calculated with the formulation:

$$F = k \cdot \frac{gp}{(g-p)} \qquad \text{EQ. 8.8.11.1}$$

With k = 0.5 STFAC * E * t

The elastic contact energy is calculated with the formulation:

$$CE = kg\left(-p - gLn\frac{(g-p)}{g} \right) \qquad \text{EQ. 8.8.11.2}$$

When node to element mid-plane distance is smaller then 10^{-10} gap, the node is deactivated.

The maximum potential contact energy is:

elastic contact energy CE = 23 kg

Drastic time step dropping is mostly due to cases where node is forced into the gap region.

8.8.11.2 Remedies to the main problem

There are several ways to resolve this problem:

1- Increase Gap

Increasing the gap is the best remedy, but check that no initial penetrations result from this.

2- Increase Stiffness

Increase STFAC dimensionless stiffness factor or provide an appropriate effective global stiffness value (v23 and up).

3- DT/INTER/DEL (Engine option)

Some nodes will be allowed to cross the impacted surface freely before penetration reaches (1-10^{10}) gap.

4- /DT/INTER/CST (Engine option)

Nodal mass will be modified to maintain time step constant. This option should be avoided when rigid body slave nodes are slaves of a type 7 interface.

The initial penetrations are mostly due to discretization and modelization problems.

They result in high initial forces that should be avoided.

5- Modify geometry

New coordinates are proposed in the listing file for all initially penetrated nodes. These are the coordinates used in the automatic coordinate change option. However, this might not suffice. Several iterations are sometimes necessary. RADIOSS will create a file *RootD0A* containing the modified geometry.

6- Reduce gap

When there are only small penetrations with a gap, this should be reduced; otherwise care should be taken as this will reduce potential contact energy.

7- Deactivate node stiffness

This solution is the simplest. It will generally not unduly affect your results. For sake of precision, use this option only for initial penetrations remaining after geometrical adjustments.

Edge contact problem

A special algorithm is developed for this purpose.

Modelization should eventually be adapted to prevent situations where 2 nodes of an element move to opposite sides of a surface.

For solid to solid contacts, the external closed surfaces may be used.

8.9 Type 14 - Ellipsoidal surface to node contact

This interface simulates impacts between a hyper-ellipsoidal rigid master surface and a list of slave nodes. This interface is used for MADYMO to RADIOSS coupling.

The hyper-ellipsoidal surface is treated as an analytical surface (hyper-ellipsoidal surfaces are discretized only for post-processing).

The interface allows user defined behavior.

- User defines total elastic force versus maximum penetration of nodes.

- A local friction coefficient is computed at each impacted node, depending upon elastic force computed at its location by scaling the total elastic force by the following factor: penetration of the node divided by sum of node penetrations.

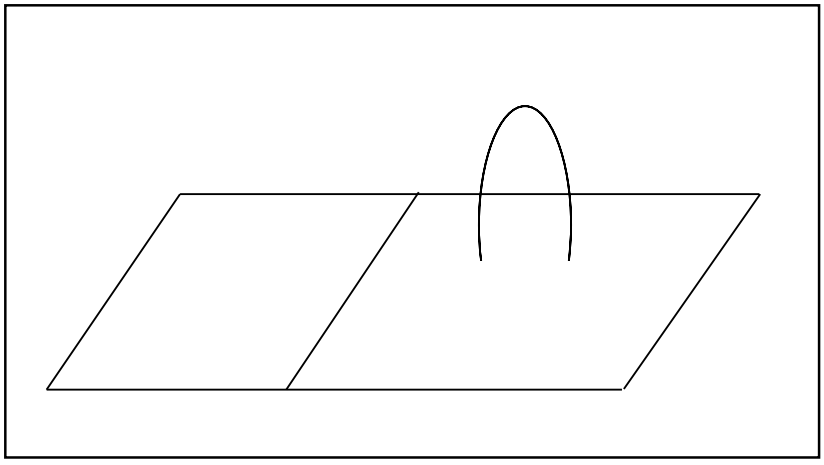
- A local viscosity coefficient in the normal direction to the surface is computed at each impacted node, depending upon this node's velocity or the computed elastic force at its location.

It is also possible to only define a constant stiffness factor, a constant friction coefficient or a constant viscosity coefficient. A time step is computed to ensure stability.

8.9.1 Type 14 interface: Hints

As the interface is defined as nodes impacting upon a surface, impact cannot be detected if the mesh is too coarse. In general, use a mesh which size is finer than the lowest semi-axis of the master surface.

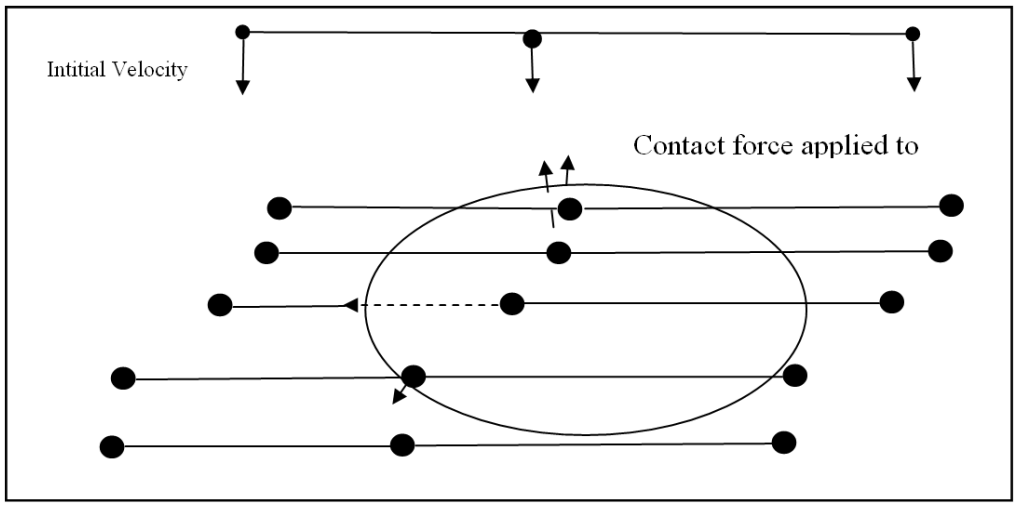
Figure 8.9.1 No impact is detected



The interface is designed to allow penetration of slave nodes. However, the contact algorithm does not ensure that a node will not cross the ellipsoid when sliding; nodes may slide along the ellipsoid until they fully cross the ellipsoid, resulting in that the structure itself fully crossing the surface and contact force is no longer applied to it (as shown in Figure 8.9.2, where perfect sliding is considered).

Increase interface stiffness or friction to avoid this problem.

Figure 8.9.2 Sliding until structure fully crosses surface

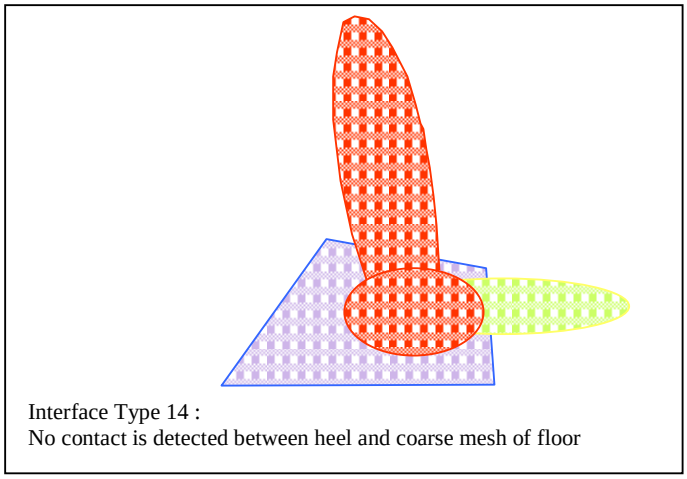


8.10 Type 15 - Ellipsoidal surface to segment contact

Type 15 interface between surfaces made up of 4-node or 3-node segments and hyper-ellipsoids is a penalty contact interface without damping.

It applies to type 14 interface, especially when the mesh is coarser than the ellipsoid size. Remember that in such a case, type 14 interface is able to compute low quality contact forces even if it fails to find contact, as shown below.

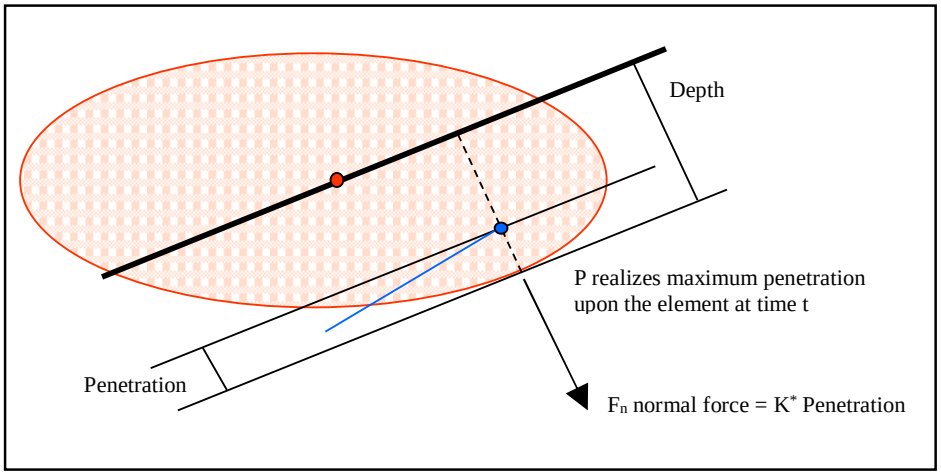
Figure 8.10.1 No contact is detected



Interface stiffness is a non-linear increasing function of penetration, computed in order to avoid penetrations up to half the ellipsoid:

$$K = K_0 \left[\frac{Depth}{Depth - Penetration} \right]^2$$ where K_0 is an input stiffness factor.

Figure 8.10.2 Penetration is detected



A Kinematic Time Step is computed so that the element does not cross the line Lt within one time step.

A friction coefficient m is input.

Interface takes into account sliding/rolling effects.

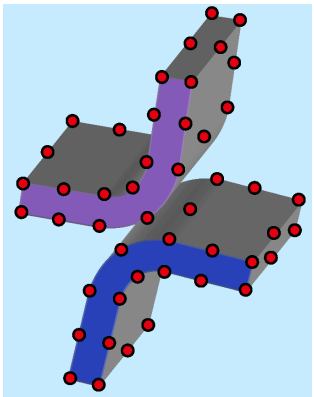
Coulomb Friction condition is expressed as:

$F_t \leq mF_n$ for each penetrated element

8.11 Type 16- Node to Curved Surface Contact

Interface type 16 will enable to define contact conditions between a group of nodes (slaves) and a curve surface of quadratic elements (master part) as shown in Figure 8.11.1 for a symmetric contact. The master part may be made of 16-node thick shells or 20 node-bricks. The Lagrange Multiplier Method (LMM) is used to apply the contact conditions. By the way that the LMM is used, no gap is necessary to be applied. Some applications of this interface are sliding contacts without gaps as in gear box modelling.

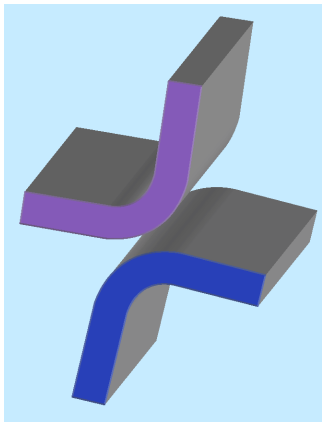
Figure 8.11.1 Node to curved surface contact in interface type 16



8.12 Type 17- General Surface to Surface Contact

The interface is used in the modelling of surface-to-surface contact. It is a generalized form of type 16 interface in which the contact on the two quadratic surfaces are directly resolved without needs of gap as the Lagrange Multiplier Method is used (Figure 8.12.1). The contact is supposed to be sliding or tied.

Figure 8.12.1 Quadratic surface to quadratic surface contact



8.13 Some Common Problems

The following sections contain examples of some common problems in the contact interfaces and solutions to overcome them.

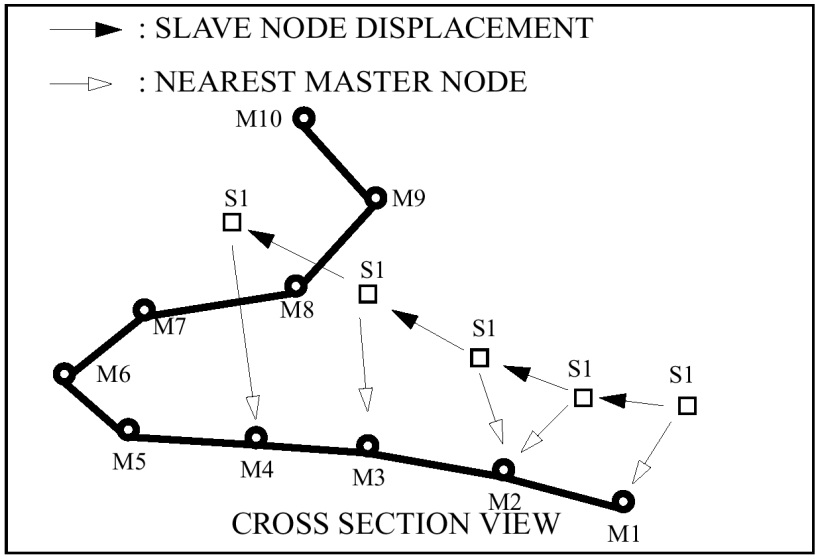
8.13.1 Incorrect Nearest Master Node Found

If the interface surface is not simply convex, the simplified master node search may find an incorrect nearest master node.

This problem occurs with interface types 3, 6 and 5 (master side only).

The solution to this problem is to use type 7 interface.

Figure 8.13.1 Incorrect Master Node Found



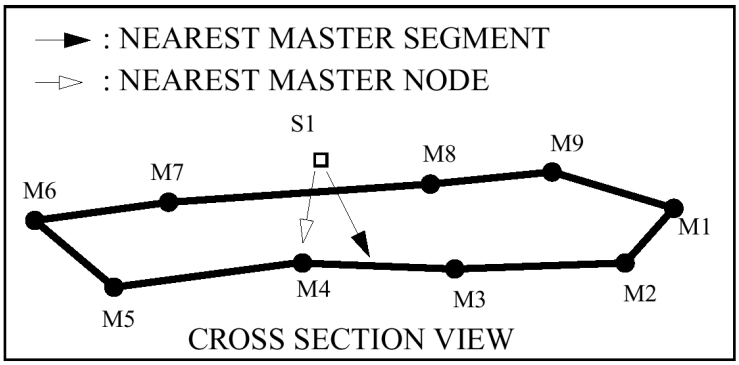
8.13.2 Incorrect Nearest Master Segment Found - B1

In some cases the nearest master node is not connected to the nearest segment.

This problem can occur with interface types 3, 6 and 5 (master side only).

The solution is to either use type 7 interface or to refine the mesh.

Figure 8.13.2 Wrong Nearest Master Segment 1



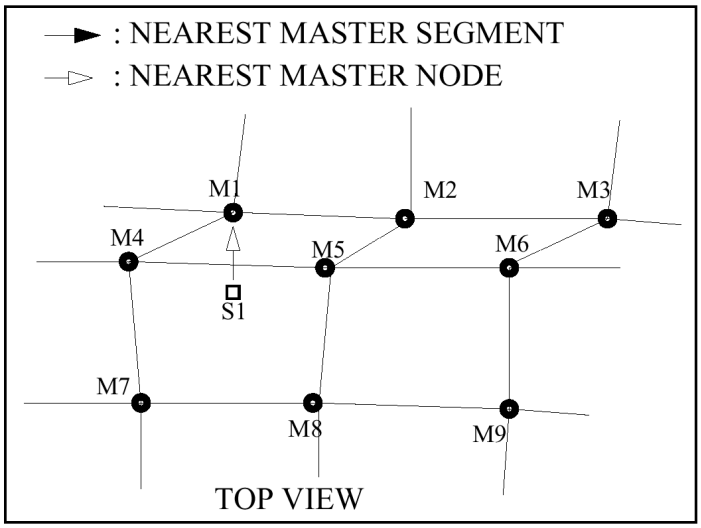
8.13.3 Incorrect Nearest Master Segment Found - B2

In some cases the nearest master node is not connected to the nearest segment.

This problem can occur with interface types 3, 5 and 6 (master side only).

The solution is to either use type 7 interface or change the mesh (for initial mesh problem).

Figure 8.13.3 Incorrect Nearest Master Segment 2



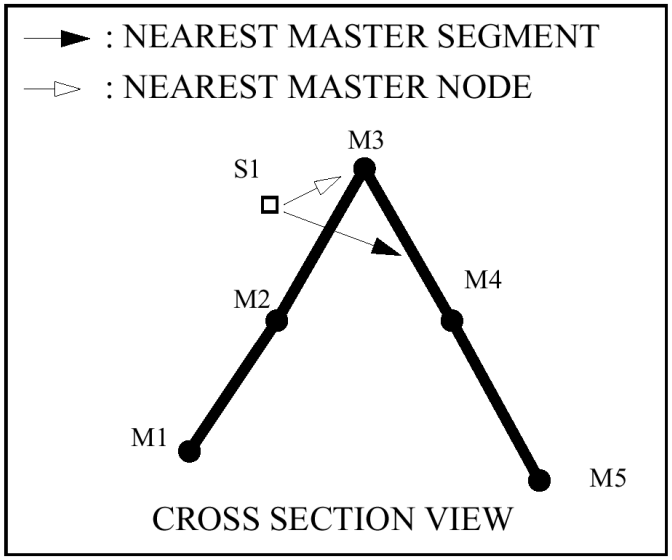
8.13.4 Incorrect Nearest Master Segment Found - B3

If the angle between segments is less than 90 degrees, the incorrect nearest segment may sometimes be found, as in Figure 8.13.4.

This problem can occur with interface types 3, 5 and 6.

The solution is to use a type 7 interface or to refine the mesh. With a finer mesh, the shape is smoother.

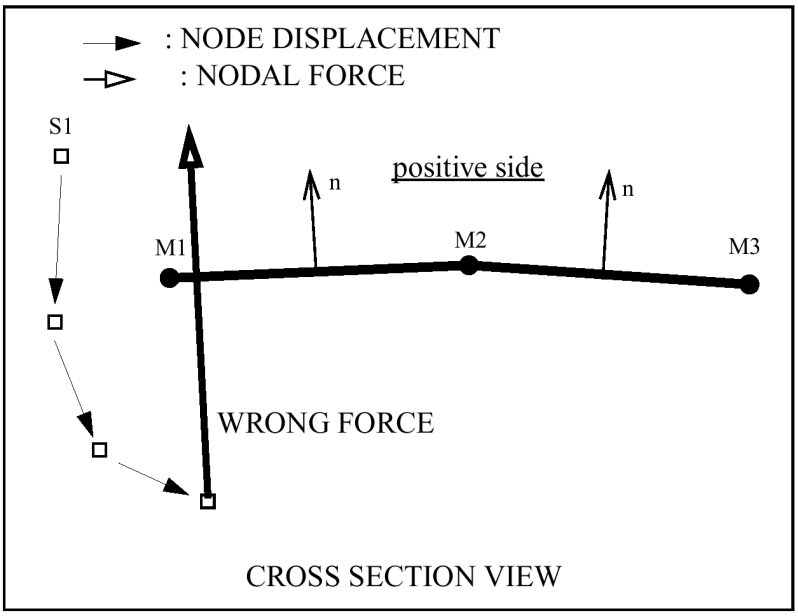
Figure 8.13.4 Master Segment Angle to Acute



8.13.5 Incorrect Impact Side - C1

A node can only impact on the positive side of a segment for interface types 3, 6, and 5 (master side). The solution is to use a type 7 interface.

Figure 8.13.5 Wrong Normal Direction



8.13.6 No Master Node Impact - D1

With type 5 interface, only slave nodes impact master segments; master nodes cannot impact slave segments.

This can be solved by either inverting the slave and master sides, or by changing the type of interface. Interface types 3 and 7 will solve this problem adequately.

Figure 8.13.6 Master Node Penetration