Note of use of the contact in *Code_Aster*

Résumé:

This document describes the step to be followed for the taking into account of conditions of contact-friction in the nonlinear studies. Initially, one points out what means to take into account contact-friction in structural mechanics, then one traces the broad outlines of a problem of contact in *Code_Aster*: pairing and resolution.

The definition of the contact is carried out with command `DEFI_CONTACT` while the resolution is done with commands `STAT_NON_LINE` or `DYNA_NON_LINE`. One formulates recommendations for the parameterization of pairing and the choice of the methods of resolution in these operators.

Finally various methodologies are evoked (contact with a rigid surface, recover a contact pressure in postprocessing, large deformation and contact, rigid body motions locked by the contact,…). They make it possible to overcome the difficulties frequently encountered in the studies. In this section, the alternative modelizations of the phenomenon of contact-friction by elements of joints or discrete elements are also approached (through the constitutive law).

Contents

1 Introduction

1.1 Object of this Dire

document that two solid body put in contact does not interpenetrate but that on the contrary a reciprocal force is exerted one on the other and that this force disappears when the body is not touched any more, concerns the good sense. It is the briefest definition which one can make of the problem of “contact”: however to make observe these conditions in a computer code of structures like *Code_Aster* requires many forces.

To solve the problem of contact, it is finally to impose a boundary condition on certain degrees of freedom of displacement and to find an unknown factor additional, the reciprocal force being exerted between the two body.

The difficulty comes from the strong non-linearity induced by this “pseudo-condition in extreme cases”. Indeed the condition to be imposed on displacements (to prevent any interpenetration) depends it even on displacements (which will determine in which point surfaces come into contact).

Non-linearity due to the taking into account of contact is separate in *Code_Aster* in two points:

- non-linearity of contact (- friction): it rises from the conditions of contact (- friction) which are not univocal. One calls here upon an algorithm of tweaking under stresses or with a transformation of the problem in a not forced version.
- geometrical non-linearity: it rises from the great relative sliding likely to occur between surfaces in contact (ignorance *a priori* of effective final surfaces of contact). One calls here upon an algorithm of fixed point or Newton coupled to a geometrical search.

In *Code_Aster*, in the presence of contact, the user must *has minimum* to identify potential surfaces of contact. The technique of resolution rests then on two fundamental stages:

- Phase of pairing: it makes it possible to treat geometrical non-linearity as a succession of problems in small sliding (where the problem is geometrically linear). The technique to determine effective surfaces of contact and the advices of parameter setting of this phase are given to section 2.
- Phase of resolution: it makes it possible to solve the problem of tweaking under stresses involved in the non-linearity of contact and possibly of friction. The various algorithms of tweaking available are presented in section 3. One gives a routing to it to choose an algorithm adapted to his case of study.

It is essential to have understood that contact-friction is a non-linearity except for whole as well as non-linearities materials (nonlinear constitutive law) and kinematics (large displacements, large rotations). She thus asks at the same time to know the bases of the theory of the contact and to understand the processing of the aforementioned in *Code_Aster* in order to make the good choices of modelization (mesh and bets in data).
This document is there to assist the user in these choices.

1.2 A question of Afin de

vocabulary of facilitating the reading, one gives here some of the terms abundantly used in this document.
When one speaks about contact mechanics, one likes to introduce two characteristic quantities:

- often noted clearance g or d. It characterizes the distance signed between two contact surfaces;
- density of force of contact p. It is the reciprocal force exerted by a solid on the other when clearance is closed (null). It is carried by the norm at contact surfaces. One will also wrongly use the term of contact pressure.

These quantities intervene under the conditions of Hertz-Signorini-Moreau of respect of the contact (*cf* [R5.03.50]).

In the presence of friction, one introduces in addition:

- the direction of sliding $\vec{t}$
- the density of force of friction τ, range par. $\vec{t}$

Dans *Code_Aster*, one uses a criterion of friction of Coulomb, the conditions of friction are described in [R5.03.50].

1.3 Alternative modelizations of contact-friction

If the manner of treating the phenomenon of contact-friction described in introduction and in the essence of this document is most widespread, it is not only. *Code_Aster* thus proposes two alternative modelizations of the mechanical interactions:

- hydro-mechanical elements of joints (modelizations `*_JOINT_HYME`) for the representation of the opening of a crack under the pressure of a fluid and friction between the walls of closed crack
- discrete elements of shock (modelizations `*_DIS_T*`) for the representation of a specific contact by springs with possible taking into account of friction

Ces two other modelizations are based both on finite elements and thus on specific constitutive laws (`JOINT_MECA_FROT` for the elements of joints and `DIS_CHOC` for the discrete elements).

More precise details on these elements are provided to the §4.8et4.8 §4.94.9.

To finish, it will be noted that it is possible to modelize contact on edges of a crack represented with method X-FEM. One will refer to the note [U2.05.02] for more information.

2 Notion

2.1 pairing of areas and contact surfaces

is always to the user to define potential **surfaces** of contact: there does not exist in *automatic* Code_Aster of mechanism of detection of the possible interpenetrations in a structure.

The user thus provides in the command file a list of couples of contact surfaces. Each couple contains **a surface** known as “Master” and **a surface** known as “slave”. One calls “ **contact zone** ” such a couple.
The conditions of contact will be imposed area by area. To make respect the contact consists in **preventing the slave nodes from penetrating inside surfaces Masters** (on the other hand the reverse is possible).
On the example below (*cf* Illustration 2.1), the studied structure makes up of 3 solids, one defined 3 potential contact zones symbolized by the red ellipses. As their name indicates it these contact zones determine parts of structure where body is **likely** to come into contact. That means that one made there observe the conditions of contact-friction, the effective activation of contact depend *in fine* on the imposed loading.

There is no restriction on the number of contact zones. The areas must however be separate, i.e. the intersection of two distinct areas must be vide1Plus[1]. In addition, within an area, surfaces Masters and slaves of the same area must also have a null intersection: if it is not the case, computation is stopped. When a node is obligatorily common to surfaces Masters and slaves, because of a stress of mesh for example, to refer to the §2.3.42.3.4 for a solution. If a continuous formulation is used (*cf* 3.1.3), surfaces slaves must imperatively be two to two disjoined.
One should not hesitate to describe broad contact zones to avoid any interpenetration. It is the number of nodes of the surface slave which is determinant in the cost of computation. Surface Master can, it, being as large as it is wanted.

[1] precisely it is the intersection of surfaces slaves which must be empty

It is imperative that the nodes of contact surfaces (main and slaves) carry all of the degrees of freedom of displacement (DX, DY and possibly DZ), i.e. they belong to meshes of the model. An error message stops the user if it is not the case. One will refer to the §4.4*4.4* for the modelization of a contact with a rigid surface.

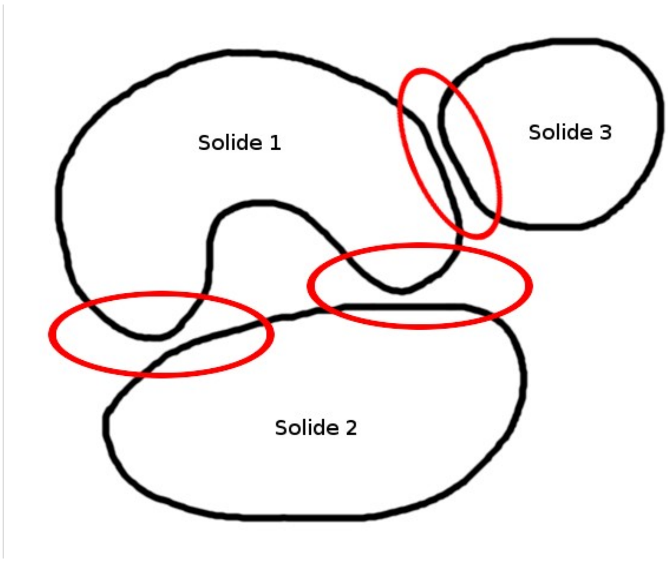


Illustration 2.1 : Definition of 3 Choix

2.2 contact zones of surfaces Masters and slaves

Comme one has just said it, each contact zone consists of a surface Master and a surface slave. In the actual position, one cannot make auto--contact in *Code_Aster* (except in the rare cases where one can predict the future contact zone and thus define a slave and a Master).

The need to differentiate two surfaces comes from the technique adopted in computation from clearance. This computation is carried out in a phase that one names **pairing**. Clearance is defined in any point of surface slave (for the discrete methods they is the nodes, for the continuous methods of the points of integration) as the minimal distance to surface Master. This dissymmetry implies a choice which can *a priori* prove to be difficult (how to decide?). The points which must prevail in this choice are given below.

One informs these surfaces in operator DEFI_CONTACT under the key word factor ZONE.

2.2.1 Case where a surface must be selected as main (GROUP_MA_MAIT)

Lorsqu' one of these conditions is joined together:

- one of two surfaces is **rigid** (A);
- one of two surfaces **recovers** the other (b);
- one of two surfaces has a rigidity connect **large** in front of the other ("apparent" in the sense that one does not speak about the Young moduli but about the stiffness in $N.m^{-1}$) (c);
- one of two surfaces is with a grid much **more coarsely** than the other (d);

then the aforementioned must be selected like surface Master.

2.2.2 Case where a surface must be selected as slave (GROUP_MA_ESCL)

Lorsqu' one of these conditions is joined together:

- one of two surfaces curved **east** (A);
- one of two surfaces is **smaller** than the other (b);
- one of two surfaces has a rigidity connect **small** in front of the other (c);

one of two surfaces is with a grid much **more finely** than the other (d);

then the aforementioned must be selected like surface slave.

2.2.3 General case

Lors of the complex structure study, it arrives that the rules given to the §2.2.12.2.1 and §2.2.22.2.2 are difficult to apply. For example when a solid is almost rigid (with respect to other solid) and that it is curved, the rule (A) does not make it possible to decide: is it necessary to privilege the curved character or the rigid character?
In these situations "the art of the engineer" must prevail. In our example, if the two solids undergo weak sliding, the curved character of rigid solid will have only little influence and one will thus choose this main last like surface.

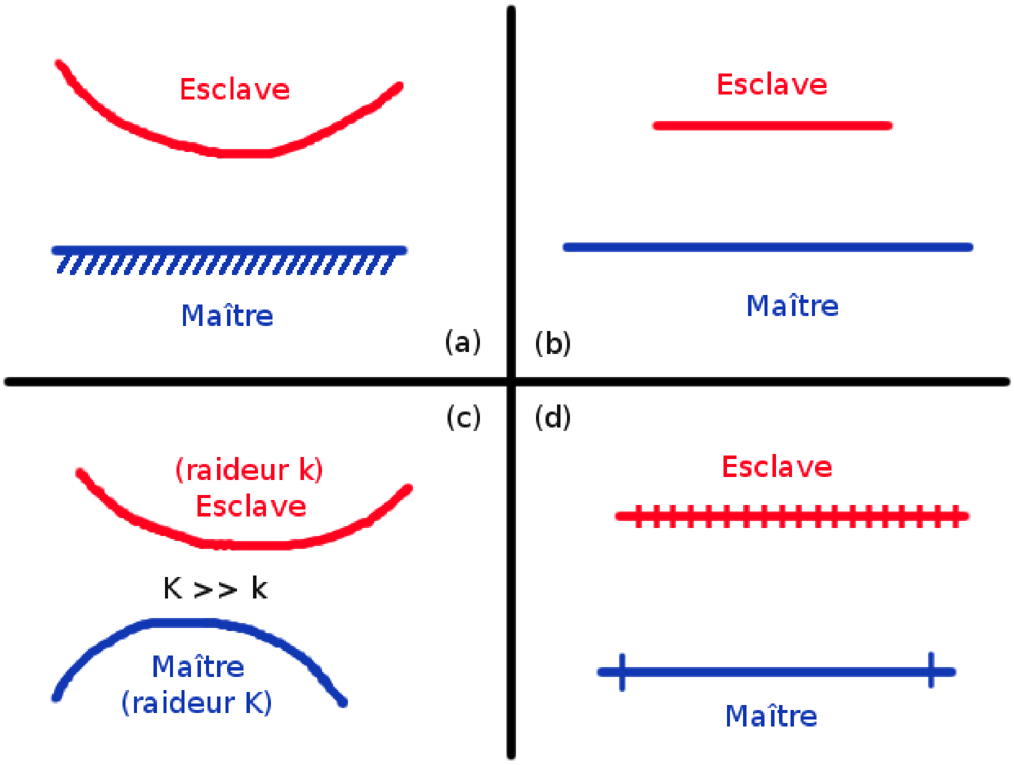


Illustration 2.2 : Choice of surfaces main and slaves according to various Orientation

2.2.4 situations of the norms

It is paramount always to direct **the norms** of contact surfaces so that they are **outbound**. One can do it using operator `MODI_MAILLAGE`. According to whether surface to be directed is a mesh of skin of a solid element, a shell or a beam, one will use respectively key words `ORIE_PEAU_2D` or `ORIE_PEAU_3D`, `ORIE_NORM_COQUE`, `ORIE_LIGNE`.
Note: in the case of `ORIE_LIGNE`, one directs the tangent, of kind to being able systematically to produce the norm by a cross product.

One checks by defect (key word `VERI_NORM` of `DEFI_CONTACT`) the good directional sense of the norms and one stops the user if need.

2.2.5 Smoothness and degree of mesh of curved surfaces

Lorsque contact surfaces are curved, it is necessary to guarantee the good continuity of the norm to the breakages. For that, one can is:
to net finely into linear and to use the option of smoothing (*cf* §2.3.22.3.2)
to net in quadratic

Dans le cas où one would choose the second possibility, **it is not necessary in discrete formulation** that contact surfaces consist of quadrangular meshes with 8 nodes (QUAD8) but rather of meshes with 9 nodes (QUAD9). One will thus transform the HEXA20 into HEXA27 and the PENTA15 in PENTA18 (with operator CREA_MAILLAGE). At present, the mixed meshes made up at the same time of HEXA20 and PENTA15 are not transformable by CREA_MAILLAGE.
In formulation continues, for curved edge meshes, the use of elements QUAD8 or TRIA6 can involve violations of the model of contact: this last is checked on average. One then observes clearances slightly positive or slightly negative in the presence of contact, which can disturb the results close to the contact zone or computations in recovery with initial state.

So that the quadratic mesh has an interest, it is necessary to have placed **the medium nodes on the geometry** and not to have used operator CREA_MAILLAGE/LINE_QUAD of *Code_Aster*.

When at the end of a computation one notices a strong rate of interpenetration of the master nodes inside surfaces slaves (what is possible contrary contrary), that generally means that the mesh of one or two surfaces is too coarse or that there is a too great difference of smoothness between the two meshes of surfaces. One can then either refine, or to reverse main and slave.

If a surface is rigid (and thus Master), a coarse mesh is sufficient except of course in the curved areas.

Finally in the typical case of **a contact cylinder-cylinder** or **sphere-sphere**, it is necessary to take care **sufficiently to net each surface** to avoid leaving too much vacuum between them. Indeed in *Code_Aster*, one does not make for time not repositioning of nodes nor of projections on splines passing by surface Master, a too coarse mesh will then cause **a strong oscillation of the contact pressure** (detection of the contact a node on two).

2.2.6 Sharp angles

Les algorithms of pairing function less better in the presence of sharp angles, this is why one will as much as possible avoid having some in the mesh of surfaces Masters and slaves. For example one will prefer modelize rather a fillet than a sharp angle.

If a sharp angle is essential, one will choose the surface which carries it like slave.

2.2.7 Quality of the mesh

the quality of the surface elements which constitute contact surface Master has a direct impact on the quality of pairing. Indeed distorted meshes, for example, can harm the accuracy of projections: the unicity of projection is not guaranteed any more.
For these reasons, it is recommended to check the quality of the produced meshes and if necessary to correct their defects. In *Code_Aster*, command MACR_INFO_MAIL makes it possible to display the distribution of the elements according to their quality.

2.3 Control Choix

2.3.1 pairing of the type of pairing

Dans *Code_Aster*, two types of pairing are available:
"master-slave" (by default): it is generic, it makes it possible to prevent the nodes of surface slave from penetrating the meshes of surface Master using orthogonal projections (of a node on a mesh).

"nodal": it makes it possible to prevent the slave nodes from penetrating the master nodes according to a direction (given by the normal slave). It is a pairing reserved for the compatible meshes of contact surfaces for computations in small sliding. It is not available in continuous formulation (*cf* §3.1.33.1.3).

One advises always to choose "master-slave**"** pairing, the value by default.

2.3.2 Smoothing of the norms

Comme its name indicates it this option makes it possible to smooth the norms. It is useful in the case of curved surfaces with a grid into linear. This process is based on the average of the norms to the nodes, then their interpolation starting from the shape functions and of the realised norms, it makes it possible to ensure **the continuity** of the norm the nodes.

The norm is not then any more the geometrical norm, one will thus take the precaution (advised in any case) to check the results visually well.

A checking of the facettisation of surfaces is carried out automatically at the end of the time step. It transmits an informational message when the aforementioned becomes too important and it is then advised to activate smoothing.

2.3.3 Choice of the norms

One always advises to leave the values by default: `NORMALE='MAIT″`, `VECT_MAIT='AUTO″`. I.e. pairing is carried out by an operation of minimization, slave nodes being projected according to the norm Master determined thanks to the mesh.

However there exist some rare situations where one can want to impose the choice of the norm: it is primarily the processing of the contact beam-beam (in 2D only) and of the case where surface Master is a mesh of the type `POI1`. One returns to the §3.1.6 of [U4.44.11] for more details.

2.3.4 Exclusion of slave nodes of pairing

key word `SANS_GROUP_NO`/`SANS_NOEUD` serves to exclude from pairing as the slave nodes. There can be several reasons with that:

surface main and slave have a nonempty intersection (crack tip, blocking of rigid body motions); the common nodes do not need to be treated by the contact, they must thus be excluded.

there already exists on the slave nodes considered of the linear relations (boundary conditions, blocking of rigid body motions); if those interfere with the direction of the contact (respectively of friction), one in general advises to privilege the boundary conditions and thus not to solve the contact on these nodes.

A fatal error is emitted when there exist nodes common to surfaces Masters and slaves and that the latter were not excluded.

2.4 To understand geometrical non-linearity

Comme one explained it, geometrical non-linearity rises owing to the fact that one must apply conditions of contact-friction to a geometrical configuration which one does not know. In this section, one makes a small digression in order to explain the step adopted to overcome this difficulty.

2.4.1 Assumption of small sliding

the phase of pairing is a phase preliminary to the formulation of the conditions of contact to solve. In practice that means:

for the discrete methods, the construction of a matrix A (for Pairing) as multiplied by the displacement increment δu since the paired configuration, it gives the increment of clearance (linearized).
for the continuous method, association enters a point of contact and its project the parametric space of the mesh Master paired. It is by bringing up to date the coordinates of the mesh Master with displacement δu that obtains it the new coordinates (linearized) of the project.

Just as the equilibrium conditions, **the conditions of contact are expressed on the deformed configuration** (or finale). This configuration is not known *a priori*.
The assumption of weak relative sliding of surfaces in contact is the analogue of the assumption of small disturbances (for the writing of the relations of equilibrium).

It consists in saying that the final configuration of surfaces in contact is not very different from the initial setup, which thus makes it possible once and for all to carry out pairing at the beginning of computation on the initial setup. Then to use the conditions established on this configuration for all computation.

Such a problem is then linear geometrically: only the non-linearity of contact-friction remains, it is treated with adapted algorithms (*cf* section 3).

2.4.2 General case

Pour process of the problems of great relative sliding of surfaces in contact, two possibilities exists: the use of a fixed loop of point to be reduced to the cases of small sliding or for the formulation continues the simultaneous resolution within the algorithm of Newton.

2.4.2.1 Loops of point fixes (ALGORITHME_RESO_GEOM='POINT_FIXE")

the adopted step is very similar to the resolution of a nonlinear problem by the method of Newton. **One transforms a geometrical nonlinear problem into a succession of geometrical linear problems**. For that one will solve a succession of problems on the assumption of small sliding.
I.e. one carries out a pairing (on balanced initial setup) and a resolution of Newton (with resolution of the contact as one will explain it in section 3). This gives us a new configuration; if this configuration is “close” to the initial setup then one converged (it was thus the final configuration), if not one loops: one remakes a pairing then a resolution… and so on until finding the configuration final (*cf* 2.3).

The difficulty is in the characterization of the convergence of this process of fixed point. What two “close” configurations? In *Code_Aster*, they are two configurations whose “mechanical” vector displacement to pass from the one to the other (i.e. the displacement increment obtained by Newton restricted with degrees of freedom DX, DY, DZ) has a small infinite norm in front of the infinite norm of the vector preceding displacement.

That implies that one thus makes always at least two iterations of geometry with this criterion (in order to give a vector initial displacement). One returns in paragraph 3.7 of [R5.03.50] for the exact statement of the infinite norm.

2.4.2.2 Algorithm of Newton generalized (ALGORITHME_RESO_GEOM='NEWTON")

the formulation continues offer the possibility of treating geometrical non-linearity directly within the algorithm of Newton. For that a pairing is carried out with each iteration and the geometrical terms of the tangent matrix are also reactualized.
The geometrical convergence criterion becomes thus an additional criterion of the algorithm of Newton: the displacement increment must tend towards zero.

2.4.3 Convergence of the loop of Linéarisation

2.4.3.1 geometry of the norm

pairing provides two information:
clearance on the configuration paired,
the coordinates of the point of potential contact on surface Master.

That one uses a discrete method of contact or continues, one must derive (linearize) the principle of virtual power, in particular the terms of contact.

The fixed assumption of point led to also make the assumption of a weak variation of the norm during iterations of Newton (of a time step). This assumption is coherent with the assumption of small sliding. The operator "clearance" is thus linearized more easily. There remains in particular constant during an iteration of geometry. This has also another implication: geometrical convergence towards the final configuration in mode not fixes can sometimes be very slow.
Contrary, the resolution by a method of Newton generalized, possible only in continuous formulation, presents a convergence much faster but can prove less ruggedized. It is the tuning by default for the formulation continues: in the event of not-convergence, one will be able to pass by again in mode not fixes by key word `ALGO_RESO_GEOM`.

2.4.3.2 Parameter setting of the loop of geometry

One saw higher than the convergence of the loop of geometry is done on a geometrical criterion: the difference of the vectors displacements between two successive geometrical configurations is small into relative. For the typical case of the algorithm of Newton generalized in continuous formulation, the criterion applies directly to the displacement increment of Newton.

For the discrete methods as for the continuous methods, small by default corresponds to lower than 1% of displacement since the beginning of the time step (0.0001 % for the case generalized Newton).

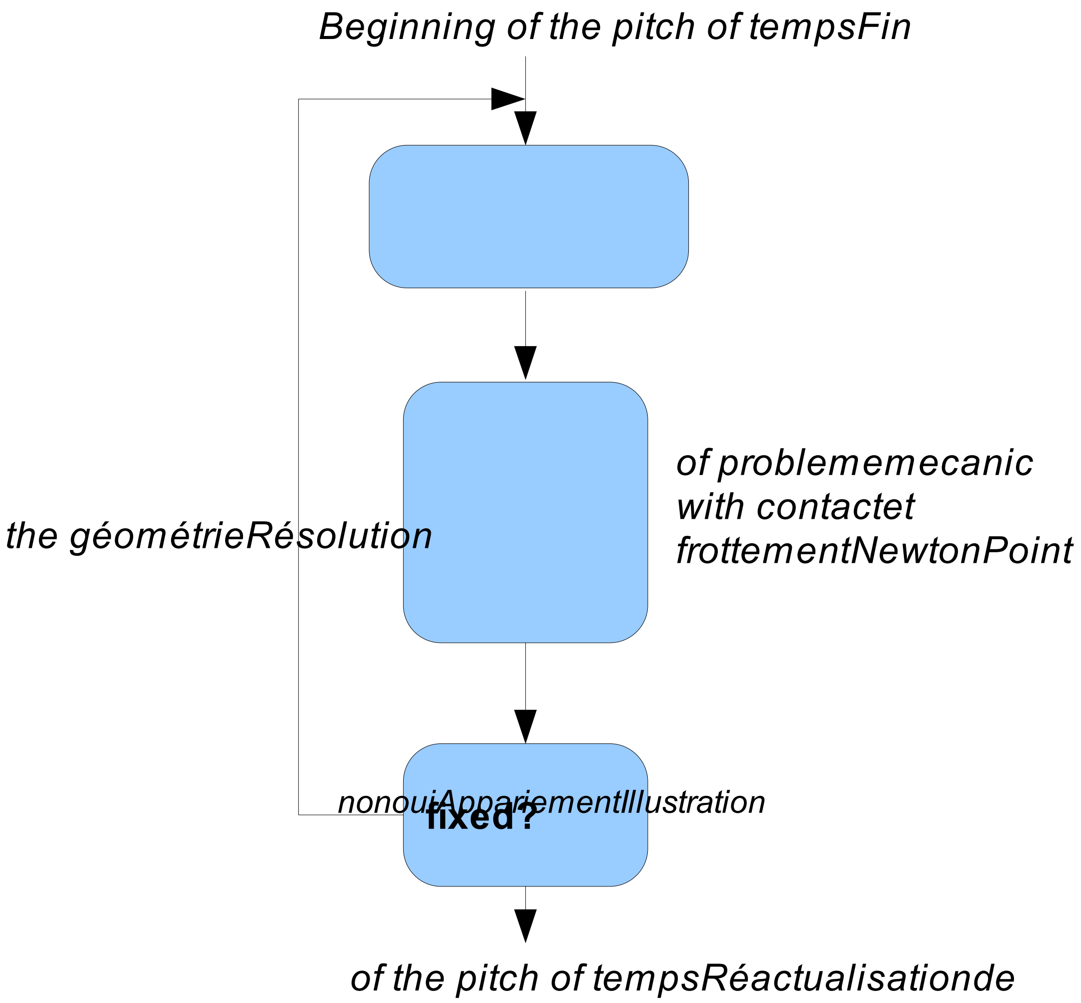
When following a computation, one observes an interpenetration of the slave nodes in surface Master, only the explication2ll[2] is a not-checking of the geometrical criterion above.
One then should not hesitate to harden the criterion. For that one uses key word `RESI_GEOM=0.005` of kind to lower the threshold around 0.5% for example.

Sometimes geometrical convergence is simply slow, in this case it is enough to increase the maximum number of tolerated iterations: `ITER_GEOM_MAXI=20` for example.

In certain situations, one in vain exploits the value of the criterion or the iteration count, computation does not converge: it cycles. Several possibilities are offered then to the user:

- the most current explanation is a bad discretization of contact surfaces (i.e. a too coarse mesh, a difference of smoothness between two surfaces or a bad choice of surfaces main and slaves). One returns then to the §2.22.2
- when contact surfaces are curved and with a grid relatively coarsely, the explanation can come from a too great discontinuity of the norm (facettisation). It is noted whereas **the activation of smoothing very often facilitates convergence** (*cf* §2.3.22.3.2). That should not however prevent the user from re-examining his mesh.
- if the case is really pathological, oneself should be forced the number of reactualizations by using `REAC_GEOM='CONTROLE"` and `NB_ITER_GEOM=n`. After n iterations of geometry, computation will pass to the time step according to whatever the value of the geometrical criterion but it will emit an alarm when the criterion is not checked with less than 1% (for $n\geqslant 2$).

[2] exists about it actually another: the use of a nonnodal diagram of integration for the formulation continues


Beginning of the pitch of tempsFin
of problememecanic
with contactet
frottementNewtonPoint
the géométrieRésolution
nonouiAppariementIllustration
fixed?
of the pitch of tempsRéactualisationde

2.3: Processing of the geometrical non-linearity of the contact by a loop of point fixes

3 Résolution

3.1 Schéma general of the algorithm of Définition

3.1.1 resolution

what one calls “resolution of the contact”, it is the operation consisting in solving the system formed by the juxtaposition of the conventional equations of the mechanics and the equations of contact-friction (the geometrical aspect being treated by pairing, it remains at this stage only the non-linearity of threshold of friction and the non-linearity of statute of the contact).

It should be noted that the two formulations available in the code differ notably on this point. Without going into the details, one briefly explains these differences for the continuation.

If the formulations discrete and continuous amount well solving the same physical problem, as their name indicates it they do not formulate it numerically same manner:

in discrete methods, the conditions of contact-friction are applied to the system discretized by the finite element method. One thus modifies **only** the resolution of the linear system obtained by Newton $Ku=f$: one then obtains a linear system under stresses (these stresses are linear inequalities). The discrete methods thus call upon algorithms of tweaking to solve it.

in continuous method, one writes a variational formulation for the equations of contact-friction, they are thus discretized as for the principle of virtual work. The step adopted to solve the nonlinear system obtained is a decoupling of non-linearities or a linearization of Newton associated with Lagrangian increased: the system thus becomes linear and nonconstrained, there is thus no algorithm specific to apply as of discrete method but rather a choice of judicious parameters (in the check out of the loops and the Lagrangian one increased).

3.1.2 Discrete formulation

Pour to illustrate the definition of the preceding paragraph, one gives 3.1 in the case of the general outline of the algorithm a discrete formulation. We can make the following remarks on this diagram:

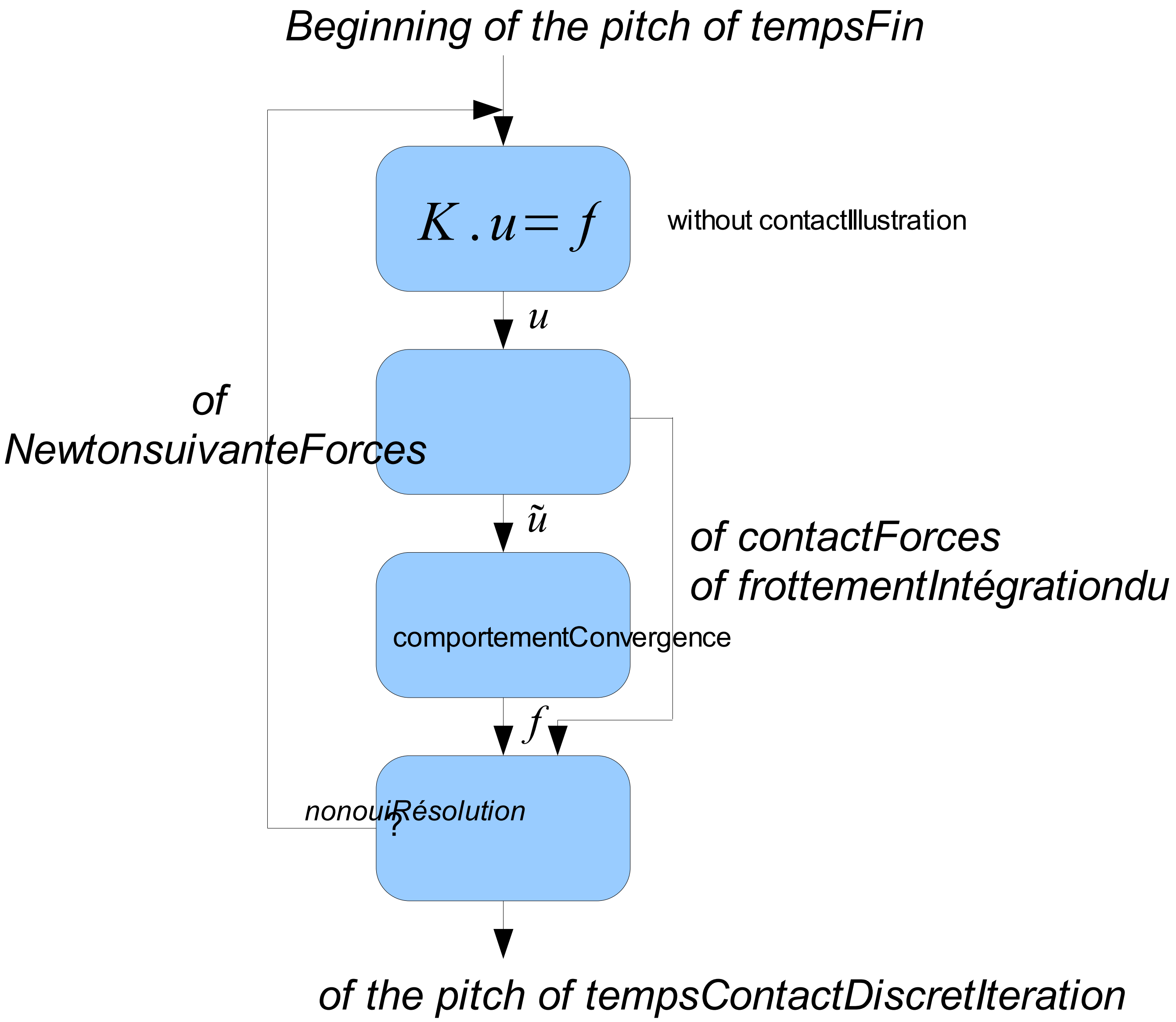
- it represents one time step by supposing that one places oneself in small sliding (one thus does not reveal the offsite loop (*cf* 2.3) treating geometrical non-linearity as described with the §2.42.4);
- in this diagram, the 3 conventional stages of an iteration of Newton appear: assembly and resolution of the linear system, integration of the constitutive law, analyze convergence;
- the characteristic of the discrete formulation of the contact consists of **the addition** of an additional stage between the resolution of the linear system (without contact) and the integration of the constitutive law. **One can see this stage like a postprocessing of the solution of the system without contact.**

The purpose of the additional stage that carries out it limps “discrete contact” is construction then the resolution of the system increased by the conditions of contact and friction. Two approaches exist to formulate the discrete conditions of contact-friction:

- writing of Lagrangian and dualisation of the conditions of contact-friction, one then increases artificially the size of the total system to solve and one uses an algorithm of tweaking to satisfy the stresses inequalities. This approach is treated with the §3.2.13.2.1 and §3.3.23.3.2
- penalization (or regularization) of the conditions of contact-friction, one preserves the same size for the total system but one enriches the matrix, it does not have there a specific algorithm, it is the algorithm of Newton which ensures convergence. On the other hand the contact is solved only roughly and the user must provide parameters to control the algorithm. This approach is treated with the §3.2.23.2.2 and §3.3.33.3.3.

What produces it limps “discrete contact” in output is a field of displacement checking the conditions of contact-friction as well as reactions of contact-friction. These reactions are used in the checking of the equilibrium.

The discrete formulation is thus based on the resolution of a mechanical problem without contact what has an important consequence: **one cannot simply treat the case of a structure where the contact as friction take part directly in blockings of rigid body motions** (*cf* §4.24.2).


Beginning of the pitch of tempsFin
$K.u=f$
without contactIllustration
u
of
NewtonsuivanteForces
$\tilde{u}$
of contactForces
of frottementIntégrationdu
comportementConvergence
f
nonouiRésolution
?
of the pitch of tempsContactDiscretIteration

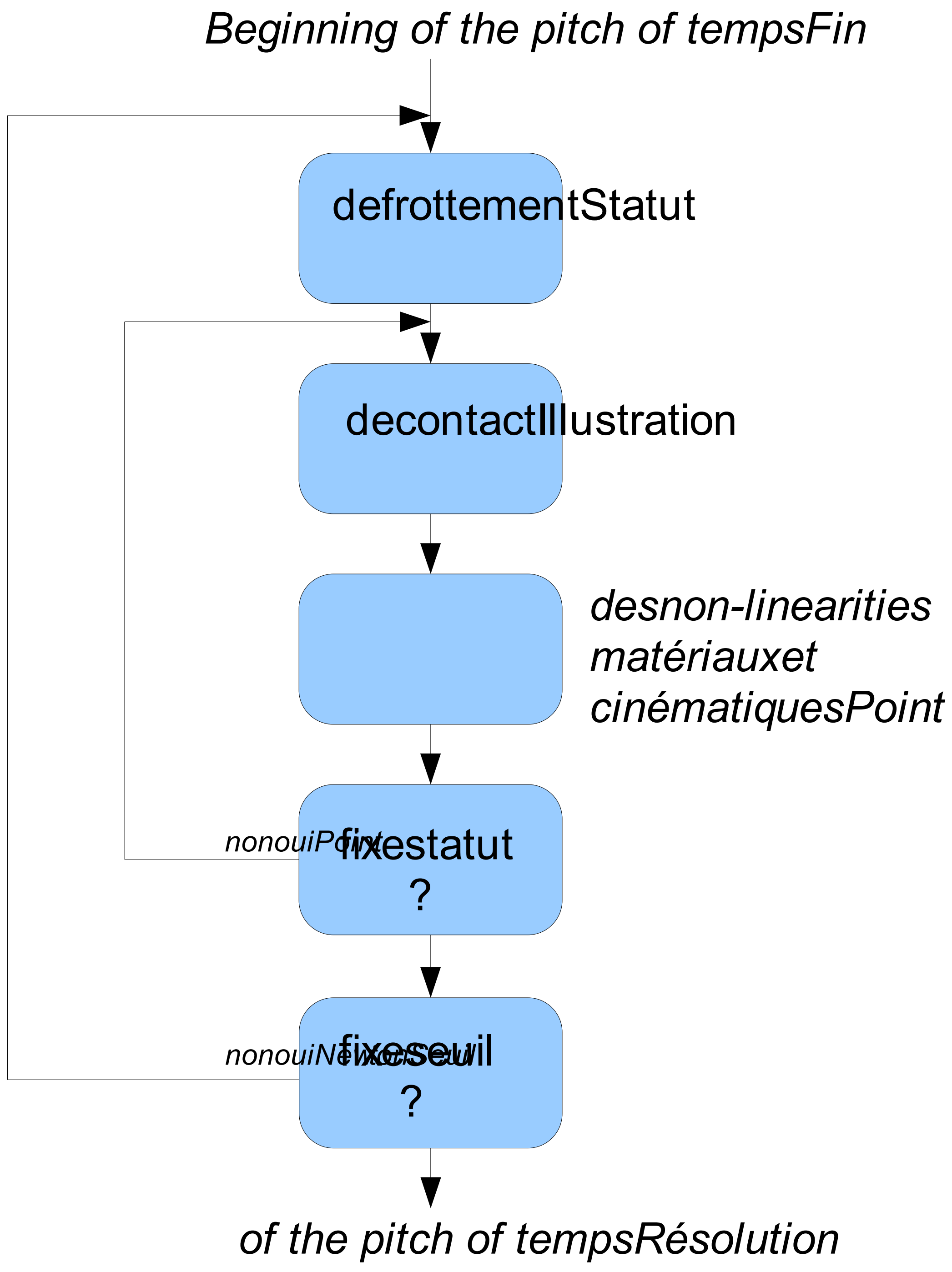
3.1 : General algorithm of a time step in discrete formulation (small sliding)

3.1.3 Formulation continues

3.2 gives the general algorithm of resolution of contact-friction with a continuous formulation, the aforementioned differs notably from the diagram in discrete formulation. Whereas with the latter contact-friction is solved by under-iterations (in “Contact Discret limps”), the formulation continues lean on a decoupling of non-linearities:

- the non-linearity of friction (the threshold of Coulomb depends on the contact pressure which is itself an unknown factor) is treated by a fixed point on the value of the multiplier of contact or an algorithm of Newton generalized
- the non-linearity of contact lean on an algorithm of the statutes (with toggle per packages) or an algorithm of Newton generalized

Lorsque all non-linearities are uncoupled, it remains in the algorithm of Newton only conventional non-linearities materials and kinematics.
Lagrangian increased makes it possible to transform the system constrained by the inequalities of contact-friction into a nonconstrained system. Each iteration of Newton in continuous formulation does not cost more than in a computation without contact of size equivalent contrary to the discrete formulation. Nevertheless the overlap of the loops or the processing by the algorithm of generalized Newton implies one plus a large number of iterations (of Newton).
In continuous formulation, there exist additional degrees of freedom in the modelization, consequence of the variational writing of the conditions of contact, as explained to the §4.1.14.1.1.



3.2 : General algorithm of a time step in formulation continues with point fixes (small sliding)

3.2 Résolution of a problem with contact only

3.2.1 Dualisation in discrete formulation (`FORMULATION='DISCRETE'`)

3.2.1.1 Principle

the dualisation of the discrete system consists of the introduction of Lagrangien (cf [R5.03.50]). The system to be solved takes the following form when it is tiny room on active connections:

$$\begin{cases} \boldsymbol{C}.\delta\tilde{\boldsymbol{u}}+\boldsymbol{A}_{\mathrm{c}}^{T}.\boldsymbol{\mu}_{i}=\boldsymbol{F}_{i} \\ \boldsymbol{A}_{\mathrm{c}}.\delta\tilde{\boldsymbol{u}}=d_{i-1} \end{cases} \quad (1)$$

Sachant that the resolution of the system without contact was already carried out, one knows the solution of the following system :

$$\boldsymbol{C}.\delta\boldsymbol{u}=\boldsymbol{F}_{i} \quad (2)$$

the technique of resolution is based then on the use of the complement of Schur of the system (1) to transform the system:

$$\boldsymbol{S}_{\mathrm{schur}}=-\boldsymbol{A}_{c}.\boldsymbol{C}^{-1}.(\boldsymbol{A}_{c})^{T} \quad (3)$$

the problem thus transformed has the size amongst slave nodes and it is full. Two algorithms with the choice are available for dealing with this new problem:

- a method of active stresses (`ALGORITHME_CONT='FORCED'`) lean on explicit **construction** and the factorization of the complement of Schur
- a method of conjugate gradient project (`ALGORITHME_CONT='GCP'`) lean on the iterative **resolution** of the system formed by the complement of Schur of the system

It should be noted that the dualisation requires the use of a direct linear solver: in *Code_Aster*, that means `"MULT_FRONT"` or `"MUMPS"`.
Each of the 2 algorithms quoted above indeed carries out under-iterations during which it is necessary to solve the linear system (2) with $\boldsymbol{C}$ the stiffness matrix of the total system without contact (what is much faster if $\boldsymbol{C}$ is already factorized).

3.2.1.2 " `FORCED" method`

lean on a factorization (thus a direct solver) to solve the system associated with the complement with Schur, the " `FORCED" method` **does not require any parameter setting**. In addition its convergence3On[3] is shown, which explains why it is the method by default in the presence of contact.

Nevertheless the use of a direct solver presents a major drawback: **this algorithm is not adapted as soon as the number of slave nodes exceeds a few hundreds (500)**. Indeed the factorization of a full matrix very quickly becomes crippling.

The construction of the complement of Schur can be accelerated by using parameter `NB_RESOL` (*cf* [U4.44.11], default value 10) to the detriment of the consumed memory (the larger the total number of degrees of freedom is, the more the increase of this parameter is expensive). In order to optimize a computation with the method of the active stresses, it is advised to make a computation on a time step in order to find a compromise time/memory (*cf* [U1.03.03] for the reading of information on the consumed memory).

[3] uses a direct solver well to build the complement of Schur but the method of the active stresses consists in activating or deactivate one by one connections of contact until satisfying the total system, it is thus an iterative algorithm.

3.2.1.3 Method "GCP"

Lorsque which one cannot use any more the method of contact by default because it is too expensive, an alternative is the use of method "GCP". As one mentioned above this method consists of the application of an iterative solver (conjugate gradient project) to solve the dual problem.
The main advantage of such a method is not to be more restricted in the face of problem (several thousands of slave nodes are perfectly atteignables). The counterpart, specific to any iterative solver, is a compulsory parameter setting for the user.
This method is usable in parallel computation, it is besides the only discrete method with really benefitting from it.

Like any iterative solver, method "GCP" uses a convergence criterion: it is about a criterion on the value of clearance. Given by key word RESI_ABSO, it controls the tolerated maximum interpenetration. It is compulsory and is expressed in the same unit as that used for the mesh. One advises to initially use a criterion equal to 10^{-3} time the average interpenetration when the contact is not taken into account (cf §4.64.6).

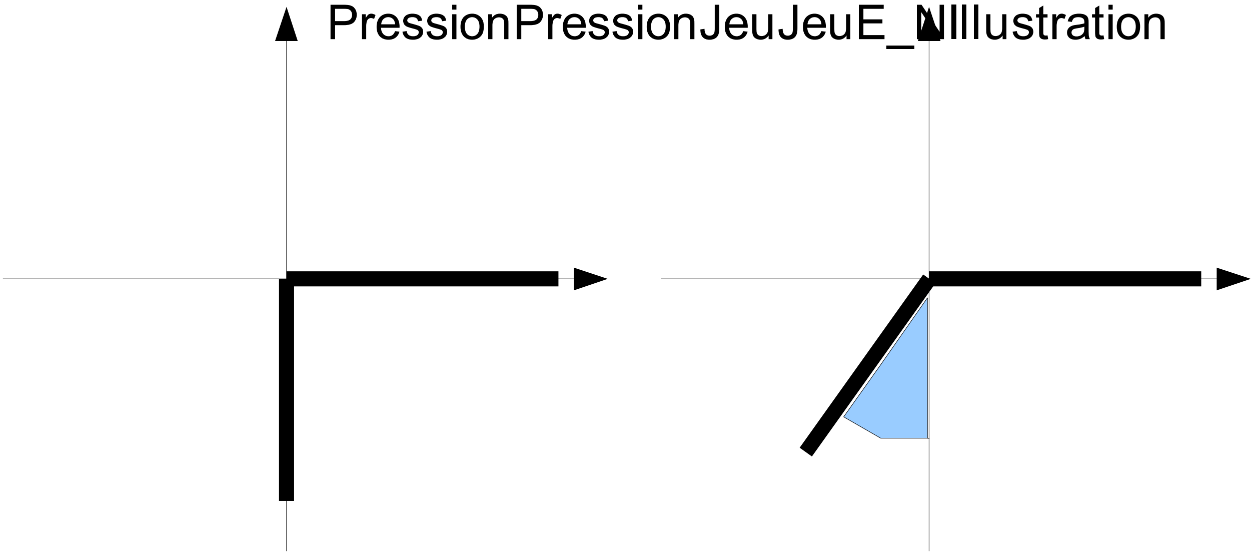
If one notes difficulties of convergence of the algorithm of the conjugate gradient project, there exist 2 parameters which, one advises to exploit (in an additive way, i.e. one then the other):

- use an not-acceptable linear search (RECHERCHE_LINEAIRE='NON_ADMISSIBLE')
- use a pre-conditioner of Dirichlet (PRE_COND='DIRICHLET')

The pre-conditioner has the advantage of being optimal and thus decreases appreciably the iteration count necessary to convergence. Moreover when one is close to the solution, it makes it possible to make decrease the residue very quickly and thus to reach very weak criteria of interpenetrations.
Its disadvantage is high costs which can often prevent a saving of time of computation in spite of the reduction amongst iterations.
For this reason, it is possible to ask its activation only when the residue sufficiently decreased: the pre-conditioner then makes it possible ideally to converge in some iterations. The difficulty lies in the quantification of “sufficiently decreased” or in other words vicinity of the solution. One controls this release by the key word COEF_RESI which is the coefficient (lower than 1) by which it is necessary to have multiplied the initial residue (initial maximum interpenetration thus) before applying the pre-conditioner.

3.2.2 Penalization in discrete formulation: algorithm "PENALISATION"

the penalization consists in regularizing the problem of contact: instead of seeking to solve exactly the conditions on clearance and the pressure, one introduces a univocal approximate relation which implies that **one will always observe an interpenetration when the contact is established**.



3.3 : Condition of contact (on the left) and regularization (on the right)

Comme installs it 3.3 one adds a parameter `E_N` to regularize the condition of contact: the larger it is, the more one tends towards the exact condition, the more it is small, the more one tolerates interpenetration.
In discrete formulation, the concept of contact pressure does not exist because one reasons on the nodes of the mesh finite element: one thus works with nodal forces (*cf* §4.14.1). Coefficient `E_N` known as of penalization thus has the dimension of a stiffness ($N.m^{-1}$).

One generally makes the analogy between the coefficient of penalization and the stiffness of unilateral springs which one would place between surface Master and slave where interpenetration is observed.

One generally chooses `E_N` by successive tests:

- first of all one will start by taking a value equalizes with 10 times the largest Young modulus of structure multiplied by a length characteristic of the aforementioned;
- if computation gives a result (satisfying or not), one will each time increase then the value by multiplying it by 10 until obtaining a stable result in terms of displacements and especially in terms of stresses.

The advantage of the method of penalization is **not to increase the size of the system contrary to the dualisation, but also not to restrict the choice of the linear solver**. The counterpart is a sensitivity to the coefficient of penalization which systematically implies a parametric study before launching out in long computations (*cf* [U1.04.00] and [U2.08.07] for launching of distributed parametric computations).

To help to gauge the coefficient of penalization, there exists an automatic adaptation mechanism lean on command `DEFI_LIST_INST` [U4.34.03]. One will find an example of implementation in benchmark SDNV103I [V5.03.103].

3.2.3 Formulation "`CONTINUE`"

Pour the problem of contact alone, the continuous method has the advantage like the method (discrete) of the active stresses of not requiring any adjustment by the user.

As it **is not dependent on a direct solver**, it is possible to use an iterative solver (like "`PCG`" or "`PETSC`") associated with pre-conditioners "`LDLT_SP`" or "`LDLT_INC`" to gain enormously over the CPU time. In this case, it is advised to activate the method of Newton-Krylov (*cf* key word `METHODE` of `STAT_NON_LINE` [U4.51.03]) which makes it possible to adapt the convergence criterion of the iterative solver automatically.

The main advantage of the continuous method is to propose via degree of freedom `LAGS_C` (in field `DEPL`) **the access to the contact pressure on surface slave**.
One however draws the attention to the fact that this quantity is in fact only **one density of force of contact per unit of area expressed on the reference configuration**. In particular, in large deformation, one cannot any more qualify it pressure because it does not have any more a physical meaning.

In formulation continues two algorithms exist:

- method of point fixes on the statutes of contact: the state of the statutes of contact is evaluated in an offsite loop with the loop of Newton
- generalized method of Newton: the statutes of contact are evaluated with each iteration of Newton (it is the defect)

Pour to choose the algorithm, it is necessary to use total key word `ALGO_RESO_CONT`.
The method of the point fixes (`ALGORITHME_RESO_CONT='POINT_FIXE'`) is most ruggedized but also most expensive since the nonlinear problem (plasticity for example) is solved with each change of the statutes of contact.
Method of Newton generalized (`ALGORITHME_RESO_CONT='NEWTON'`) is more powerful but poses sometimes problems of convergence, particularly in dynamics. In this case one will return towards a method of point fixed.

3.3 Resolution of a problem with friction

3.3.1 Processing of the non-linearity of threshold

Dans *Code_Aster*, the only model of friction available is that of Coulomb (cf [R5.03.50]). An additional non-linearity must be treated in the presence of friction: it is the non-linearity of threshold.
The threshold of friction depends indeed on the contact pressure which is itself unknown.
The model of Coulomb utilizes a coefficient μ , called party coefficient of Coulomb. During the phase known as of bond, a point in contact does not move (it has a null velocity and there exists a tangential reaction). During the phase of sliding, the point has a non-zero velocity and is subjected to a tangential reaction equalizes with μ time the normal reaction.

In general, **if the coefficient of kinetic friction is very low, it is advised to neglect frictions**. In addition, it is advised in the studies to initially treat **only the contact**, this in order to introduce non-linearities the ones after the others.

The discrete methods that they work by penalization or dualisation lean on algorithms dedicated in the presence of friction (distinct from those used for the contact) while the continuous method uses two different algorithms:

- method of point fixes on the thresholds of friction: the threshold is brought up to date in an offsite loop with the loop of Newton (and with the loop on the statutes of contact);
- method of Newton generalized: the non-linearity of friction is treated in the process of Newton, by explicit derivative of all the nonlinear terms.

3.3.2 Discrete formulation: dualisation of friction (algorithm "`LAGRANGIEN`")

Ce type d' algorithm is well adapted to the processing of models 2D, it does not require **any** parameter setting and converges relatively well.

For problems 3D, convergence appears more difficult in particular as soon as the coefficient of kinetic friction becomes larger than 0.1. If it is observed that convergence is very slow but that the residue decrease, it is possible to accelerate computation by informing coefficient `COEF_MATR_FROT` with a value of 0.5 (*cf* [U4.44.11] for the meaning of this parameter).

3.3.3 Discrete formulation: penalization of friction (algorithm "PENALISATION")

Pour problems 3D or of big size, it is advised to deal with the problem of friction by penalization. That requires, as for the penalization of the contact, the input of a parameter of penalization (E_T). More difficult to choose than his equivalent E_N, a small parametric study will be always privileged.

To make the analogy with the case of the penalization of the contact it will be noticed that the phase of bond strictly speaking disappears (as soon as the contact is activated there is interpenetration, in friction there is always sliding).

Convergence can also be accelerated by the use of key word COEF_MATR_FROT.

3.3.4 Formulation "CONTINUE"

It is the method of choice when one must deal with problem of contact-friction: it is most ruggedized moreover it tolerates the great coefficients of kinetic friction well (larger than $0{,}3$).

It is possible to choose among two algorithms of resolution for friction with key word ALGO_RESO_FROT.
The method of the point fixes (ALGORITHME_RESO_FROT='POINT_FIXE') is ruggedized but expensive since the nonlinear problem (plasticity for example) is solved with each change of the threshold of friction and with each change of the statutes of contact.
Method of Newton generalized (ALGORITHME_RESO_FROT='NEWTON', by default choice) is very powerful and offers a good level of robustness.
The large advantage of this algorithm is its least dependence with the value of the coefficient of kinetic friction, since there is no loop on the thresholds. One produces a not-symmetric matrix tangent, which represents a light overcost during factorization and limit the range of the iterative solvers usable.
It is preferable to use the generalized method of Newton since the coefficient of kinetic friction is not negligible. The savings of time computation are very important (up to 80% of gain compared to the fixed point).
The two algorithms give the same results and there is no risk of false results.

When however difficulties of convergence appear, in particular in the presence of important sliding, the user will be able to parameterize the coefficient COEF_FROT (which has the dimension of the reverse of a distance). This parameter takes a value of 100 by defaults: one will respectively test a value 1000 times smaller and 1000 times larger.

3.4 Summary for the choice of the methods of resolution

3.4.1 Pour contact-friction

Pour the problems with low number of degrees of freedom in contact (lower than 500 degrees of freedom), one will privilege a discrete formulation with algorithm of the active stresses ("STRESS"). If friction must be activated, one will turn towards a formulation "CONTINUE".

For the problems with a large number of degrees of freedom in contact (higher than 500 degrees of freedom), the iterative algorithm of resolution by active stresses "GCP" is most suitable. If however one must take into account friction, one will be able to turn once again towards formulation "CONTINUE".

For the problems of big size (independently amongst degrees of freedom in contact), the resolution of the linear system consumes most of the CPU time, the choice of the linear solver is thus paramount. Method "CONTINUE" (and in a less measurement method by "PENALISATION") is well adapted in the sense that it leaves to the user the choice of the linear solver and that it is well paralleled.

3.4.2 For the linear system

If a discrete formulation is used (except penalization), only the direct linear solvers are accessible. One will thus choose solver "MULT_FRONT" except if one carries out a parallel computation in which case one selects "MUMPS". Allied method "GCP " with linear solver "MUMPS" benefits from a good level of parallelization in the contact algorithm.

If a continuous formulation is used, it is advised, as soon as the total problem exceeds 100,000 degrees of freedom, to use an iterative solver associated with pre-conditioner "LDLT_SP" and the method of Newton-Krylov (cf § 3.2.3). If computation implements friction or is parallel, iterative solver "PETSC" is the best choice.

4 Dans

methodologies this part, one answers the questions frequently put at the time of the studies with contact-friction. The techniques installations in this part are often pressed on other operators that `DEFI_CONTACT`, one will briefly describe the key words to be used but the user will be able advantageously to refer to documentations of use of these commands.

4.1 To recover the contact pressure

In postprocessing of a computation of contact, one generally wishes to reach the forces of contact-friction. More precisely, one wishes to know the normal stress and tangential on edge of solids in contact.
The formulation continues contact directly gives access to the pressure of contact-friction, while the discrete formulations require to approximate it by the stresses on edge.
An example of implementation for the two formulations exists in benchmark SSNP154 [V6.03.154].

4.1.1 Continuous formulation

In continuous formulation, field `DEPL` contains one or more additional unknown factors:

- `LAGS_C` represents the surface density of force of contact expressed on the reference configuration.
- `LAGS_F1` and `LAGS_F2` represent the coordinates of a directing vector in the tangent plane. This vector of norm lower or equal to 1 indicates the direction of sliding or bond when which one takes into account friction.

These quantities are defined in any point of surface slave of contact. One can thus easily reach the contact pressure. It will be noted however that in large displacements, initial configuration and finale being confused, degree of freedom `LAGS_C` does not have any more the meaning of a pressure.

To reach the surface density of force of friction (in phase of bond like sliding), should be carried out an additional computation: the norm of the directing vector in the tangent plane indeed gives the amplitude compared to the threshold of friction.
If one notes λ the contact pressure then the density of force of friction τ is written:

$$\tau=\mu.\lambda.\sqrt{\left(LAGS_{F1}^{2}+LAGS_{F2}^{2}\right)}$$

In penalized formulation (`ALGORITHME_CONT='PENALISATION'`), the degrees of freedom of pressure continue to exist, one can thus apply what precedes.

4.1.2 Discrete formulation

In discrete formulation, no degree of freedom is added with the main unknown factors. The problem of contact being formulated on the discrete system, the possible Lagrange multipliers used do not even have the dimension of a pressure but that of nodal forces.
This absence obliges with compute the tensor of the stresses of Cauchy on edge of surfaces in contact. The contact pressure is written indeed:

$$\lambda=(\boldsymbol{\sigma}.\underline{n}).\underline{n}$$

where $\underline{n}$ is the norm at contact surface and $\boldsymbol{\sigma}$ the tensor of the stresses of Cauchy.

For compute the stresses of Cauchy on edge, they should be interpolated starting from the stresses at the Gauss points. For then obtaining the pressure, it is necessary compute the norms on edge (of the geometry deformed in large displacements) then to create a field starting from the interpolated stresses and norms.
Framed below the watch how one can proceed for a computation 2D. That supposes to have already calculated in result (`RESU`) the field of the stresses of Cauchy to nodes (`SIEF_NOEU`).

```
sigm=CREA_CHAMP (TYPE_CHAM='NOEU_SIEF_R',
                OPERATION='EXTR',
                RESULTAT=RESU,
                NOM_CHAM='SIEF_NOEU',
                INST=1.0,
                );

depl=CREA_CHAMP (TYPE_CHAM='NOEU_DEPL_R',
                OPERATION='EXTR',
                RESULTAT=RESU,
                NOM_CHAM='DEPL',
                INST=1.0,
                );

# normal on the configuration initialeNormaleI=CREA_CHAMP
(TYPE_CHAM='NOEU_GEOM_R',
                   OPERATION='NORM',
                   MODELE=MO,
                   GROUP_MA= (“Slavic”, “Master degree”,),
                   );

Pressure = FORMULA (VALE='SIXX*X*X+SIYY*Y*Y+2*SIXY*X*Y',
                   NOM_PARA= (“SIXX”, “SIYY”, “SIXY”, “X”, “Y”,),);

Pres=CREA_CHAMP (TYPE_CHAM='NOEU_NEUT_F',
                OPERATION='AFFE',
                MAILLAGE=MESH,
                AFFE=_F (GROUP_MA= (“Slavic”, “Master degree”,),
                        NOM_CMP='X1',
                        VALE_F=Pression,),);

pI=CREA_CHAMP (TYPE_CHAM='NOEU_NEUT_R',
             OPERATION='EVAL',
             CHAM_F=Pres,
             CHAM_PARA= (NormaleI, sigm,),);
```

For a computation in large displacements, the norm must be computed on the deformed configuration. For that it is necessary to relocate the mesh with command MODI_MAILLAGE.
In the above example, one uses a formula for compute explicitly the contact pressure. In the particular case where the edge on which one extracts the pressure is parallel to the axes of the reference, the pressure is directly equal to one of the diagonal components of the tensor of the stresses of Cauchy (SIXX, SIYY or SIZZ).

4.2 Rigid body motions locked by the contact

This paragraph applies only to the studies in statics. In dynamics rigid body motions do not exist.

It arrives in the studies that the contact makes it possible to lock rigid body motions of certain solids (and make so that those become deformed). The initial not-catch in account of this phenomenon will thus involve the singularity of the stiffness matrix (and thus impossibility of solving).

The discrete formulations are not adapted to an initial taking into account of the contact, the realization of studies with solids only held by the contact thus will require in this case an enrichment of the modelization.
The formulation continues makes it possible to take account of an initial contact naturally and for this reason thus is well adapted under investigation mechanisms.

For studies in three dimensions, there exist 6 possible rigid body motions: 3 translations, 3 rotations. For studies in two dimensions (modelizations D_PLAN, C_PLAN), there exist 3 rigid body motions: 2 translations and a rotation. The axisymmetric modelization (AXIS) is particular: there exists one rigid body motion, the translation along the axis Oy (cylindrical axis of symmetry).

When one notes the existence of rigid body motions in his modelization, **one will always start by checking that there do not exist symmetries in structure and its loading**. The conditions of symmetry indeed make it possible to remove most of rigid body motions.

An example of blocking of rigid body motions in continuous formulation (by CONTACT_INIT) and in discrete formulation (by springs) is available in benchmark SSNA122 [V6.01.122].

4.2.1 Continuous formulation

In continuous formulation, the taking into account of an initial contact is assured area by area with key word CONTACT_INIT. By defaults at the beginning of a computation all connections with null clearance (or interpenetrated) are activated (CONTACT_INIT='INTERPENETRE'). The tolerance, to determine if a clearance null or is interpenetrated, is built-in-house in the program with $10^{-6}\times a_{min}$ where a_{min} the smallest non-zero edge of the mesh represents.

It is possible of deactivate this activation automatic (CONTACT_INIT='NON'). When one makes nonlinear computations with recovery (i.e. with key word ETAT_INIT of STAT_NON_LINE), it is essential to use the default value ("INTERPENETRE") in order to ensure a recovery starting from the true state of contact (and not of a virgin state).

Finally if one wants to paste initially all contact surfaces independently of initial clearance, one can select CONTACT_INIT='OUI' (that can be useful if the meshes are not perfectly in contact).

In all the cases where an initial contact is declared, of the forces will be generated: **it is not about a simple geometrical repositioning** aiming at pasting the meshes.

The activation of an initial contact locks rigid body motions according to the normal direction on the surface. If one wants to take into account an initial adherent state in order to lock the tangent direction, one will be able to specify an initial threshold of non-zero contact *via* SEUIL_INIT. This parameter informs the initial value of the contact pressure (homogeneous with one density of surface force).

It should be noted that the use of an initial contact in continuous formulation as makes it possible to be freed from not-convergence when as a structure is subjected only to displacements. For example, when 2 solids initially in contact are in a hurry one against the other by displacements (it is thus about a rigid body motion).

4.2.2 Discrete formulation

In discrete formulation, one is obliged to manually lock rigid body motions of solid accused by springs of low stiffness. By "weak" one understands sufficiently small not to generate that negligible nodal forces in front of the nodal forces settings concerned in computation.
The goal of springs east making so that computation without contact is able to turn in linear mechanics (i.e. in operator MECA_STATIQUE or in STAT_NON_LINE once withdrawn the conditions of contact).

There exist 2 approaches for the addition of springs:

- to add a come out from low stiffness in any point structure to add it
- springs in quite selected points to lock rigid body motions of structure

the first approach has the advantage of the generics but can disturb sometimes too much the solution (whatever the stiffness of springs). Indeed such an approach amounts adding on all the diagonal terms of the matrix a positive term which makes it invertible.
The second approach only adds springs where it is necessary. When there exist points of structure which will be brought to have a weak displacement (thus to generate only one weak nodal force in spring), this approach is adapted.

To apply a spring in *Code_Aster*, it is necessary to create meshes of the type POI1 starting from nodes. For that one uses operator CREA_MAILLAGE/CREA_POI1. To use the first approach one will choose to create this mesh group on all structure (TOUT='OUI'), while for the second approach, one will indicate the desired nodes group. The mesh group lately created will be used to affect a modelization of the type "DIS_T" or "2D DIS T" in AFFE_MODELE.

The definition of the characteristics of spring is carried out in operator AFFE_CARA_ELEM. By defaults, the stiffness entered the total reference. If for example one wishes to lock a rigid body motion in a direction parallel with the axes of the total reference, one will define a non-zero stiffness only according to this direction. Below an example of definition of a stiffness for a computation 2D according to direction DY.

```
RESSORT=AFFE_CARA_ELEM (MODELE=model,
                       DISCRET_2D=_F (CARA='K_T_D_N',
                                      GROUP_MA='SPRING',
                                      VALE= (0. , 1.0e-1,),),);
```

Whenever the direction to be locked is not parallel to the axes, two alternatives are possible:

- define a stiffness according to all the directions
- to define the stiffness in a local coordinate system. It is then necessary to lay down the directional sense of this reference (key word DIRECTIONAL SENSE of AFFE_CARA_ELEM) or to use springs being pressed either on meshes POI1 but SEG2.

For an example of use of springs, one will consult the benchmark ZZZZ237 and his documentation [V1.01.237].

4.3 Large deformation, large displacements and contact

the taking into account of conditions of contact-friction is completely uncoupled from the taking into account of large displacements or large deformation. More generally any non-linearity which is of a nature material or geometrical is *a priori* compatible with the use of the contact.

In practice, one often notes difficulties of convergence in studies mixing three non-linearities. One gives in the continuation of this section the step to be adopted in this case.

Examples of computation mixing three non-linearities are available in benchmarks SSNP155 [V6.03.155] and SSNP157 [V6.03.157].

4.3.1 To uncouple Lorsque

non-linearities that such a computation fails, the first step is to retrogress: by uncoupling non-linearities and while trying to apply the good practices into nonlinear (*cf* [U4.51.03]).
That means:

- carry out an elastic design in small disturbances with the activated contact. If this computation fails, apply the advices delivered in this document (directional sense of the norms, main choices of surfaces and slaves, choice of the algorithm of resolution,…)
- carry out a computation with a nonlinear constitutive law but without contact. If the aforementioned fails, then the problem comes from the integration of the behavior. One will refer then to documentations [U2.04.02] and [U2.04.03].
- if necessary carry out a computation in large displacements but without contact and non-linearity material. If this computation does not function, try to use another model among those of large displacements available in *Code_Aster* (‶SIMO_MIEHE″, ‶GDEF_LOG″, ‶PETIT_REAC″).

4.3.2 To parameterize the algorithm of Newton well

If complete computation (mixing all non-linearities) does not converge in spite of the application of the preceding advices then one can try to exploit the parameters of the algorithm of Newton. That leaves the following official report:
When one couples contact and non-linearity material for example, it is possible (by the “abrupt” correction of the contact) to start in the constitutive law of the mechanisms (left the elastic domain, discharges) which should not be active in the final solution and which are likely to degrade the tangent matrix (until making it noninvertible). That makes then any convergence impossible.

One thus proposes to use the following tuning in the algorithm of Newton (operator `STAT_NON_LINE` or `DYNA_NON_LINE`):

- reactualization of the tangent matrix to each iteration (`REAC_ITER=1`)
- use of an elastic prediction (`PREDICTION='ELASTIQUE'`)
- in large deformation (`RELATION='SIMO_MIEHE'`), the tangent matrix is asymmetric, it is thus necessary well to take care to inform `SYME='NON'` in the key word `SOLVER`.

When computation has still evil to converge, it is necessary to return to the modelization:

- does my computation cause problems of incompressibility? In this case, consult documentations [U2.04.01] [U2.04.02] and try to use adapted finite elements (under-integrated, with mixed formulation).
- the behavior that I use have a coherent tangent matrix? If it is not the case, one can try in last recourse to use a matrix "`ELASTIQUE`" and to increase the iteration count of Newton.

4.3.3 Resolution of a quasi-static problem in slow dynamics

As a last resort, for the quasi-static problems, to carry out a dynamic computation in long time can bring a solution. The mass matrix causes to stabilize structure, it should however be made sure that the inertia forces remain weak in front of the internal forces of the system.

One advises for this kind of modelization to assign to structure his true density (it is compulsory in any case in the presence of loading of gravity) and to carry out computation by using great time steps.
An example of implementation is available in benchmark SSNP155 [V6.03.155].

4.4 Sometimes rigid surface and

contact One wishes to modelize in the studies of the rigid solids which come into contact with deformable solids. In this section, it is explained how to optimize such studies.

In order not to weigh down the modelization the rigid solids will not be entirely modelized: **only their edge will carry degrees of freedom**. In order to facilitate the directional sense of the norms of this rigid solid the mesh will comprise complete solid however.

After having directed the norms, one will thus assign in `AFFE_MODELE` only elements of edge to the skin of rigid solid: as the edge elements do not carry rigidity, an alarm is emitted to prevent noninvertible risk of stiffness matrix. **This alarm is normal in this case** and can be been unaware of.

To prevent that the stiffness matrix is singular, **it is necessary to force the displacement of all the degrees of freedom carried by rigid edge**. That is done with the commands:

- `AFFE_CHAR_CINE/MECA_IMPO` from which the advantage is to eliminate unknown factors
- `AFFE_CHAR_MECA/DDL_IMPO` which adds additional unknown factors to the problem.

One thus advises to eliminate unknown factors (`AFFE_CHAR_CINE`).
Rigid surface will be declared like **surface Master** in `DEFI_CONTACT` as explained to the §2.2.12.2.1.

One will be able to refer to benchmark SSNV506 [V6.04.506] for an example of contact with rigid surface.

4.5 Redundancy between conditions of contact-friction and boundary conditions (symmetry)

En présence de symmetry in studied structure, it is current that the conditions of friction enter in conflict with the boundary conditions of symmetry. Figure 4.1 shows the example of 2 cubes in contact-friction, the hatched part represents the sides of the cubes subjected to a condition of symmetry (`DX=0`).
In this example, the edge of the higher cube **in thick feature** belongs to surface slave and also carries the condition of symmetry. This condition enters in conflict with the condition of friction written the

tangent plane (here the plane xOz). In practice computation will once stop the established contact because the tangent matrix will be singular.
Mechanically it is seen that the condition of symmetry implies that the bond or the sliding will occur only according to direction `DZ` (green tangent vector). To eliminate the redundancy it is thus necessary to exclude the direction from friction following `DX` (red tangent vector).

For that one will use key word `SANS_GROUP_NO_FR` to indicate the list of nodes of the edge slave then one will inform (in the total reference) `DIRE_EXCL_FROT= (1,0,0)` that is to say direction `DX` to be excluded.

Benchmark ZZZZ292 implements functionality `SANS_GROUP_NO_FR`.

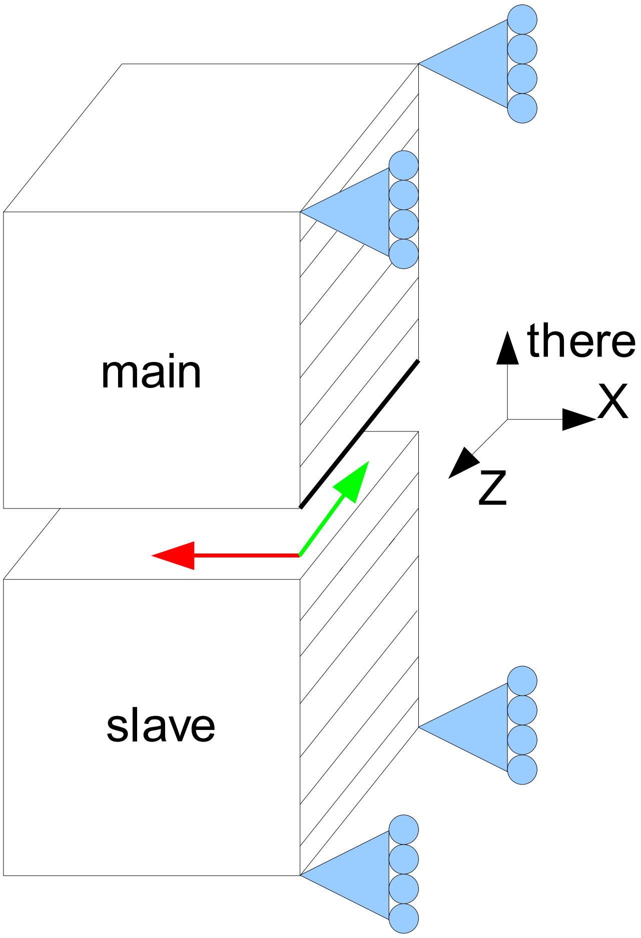


Illustration 4.1: Elimination of directions of Mesurer

4.6 friction the interpenetration without solving the contact

the resolution of a problem of contact sometimes which can be expensive it can be advantageous to override the imposition of the conditions of contact by a simple checking of the interpenetration. It is all the more interesting whenever one simply wishes to check that solids will not come into contact.
For each contact zone defined in operator `DEFI_CONTACT`, it is possible to choose if one wishes to make there respect contact (`RESOLUTION='OUI'`) or not (`RESOLUTION='NON'`).
It is advised always to begin a study of contact with a computation with `RESOLUTION='NON'` and then post-treat the interpenetrations (for example to make sure that the contact zones were well defined).

The interest of such an approach is not to weigh down a computation: when that a computation carried out without resolution **on integrality of the contact zones** shows that there is no interpenetration then one can be unaware of the modelization of the contact.
Attention however: so at least one of the contact zones "is solved" and the another "unsolved" then existence of an interpenetration does not prejudge a solution of a complete computation with contact (because of possible interactions between contact zones).

Finally this technique can be also used to measure the rate of interpenetration on the level them contact zones to gauge a criterion like the coefficient of penalization or the maximum interpenetration tolerated in the method of resolution "GCP" .

4.7 To display the results of a computation of Lorsque

contact one displays the results of a computation of contact-friction in a software of postprocessing, it is necessary to take guard with several things:

- for the display of the deformed shapes, **a factor of amplification different from 1** can result in visualizing nonreal interpenetrations
- for computations 2D in formulation "CONTINUE", one will pay attention during the display of deformed shapes to the software of postprocessing which regards the first 3 components of a field as the components according to X, Y and Z of displacement. In 2D, the 3rd component corresponds to LAGS_C and must thus be been unaware of
- during the visualization of the field of postprocessing of contact (VALE_CONT) and more particularly of the component CONT which indicates the state of the contact, one will sometimes automatically pay attention to the interpolation of the fields to the nodes realized. Indeed this component takes values 0 (not contact), 1 (adherent contact) or 2 (which drags contact). The adherent state is not possible that in the presence of friction: if one visualizes such a value for a computation of contact without friction it is that there is interpolation of the field.

4.8 Specific contact with discrete elements (springs)

Les discrete elements (or springs) 2D_DIS_T* or DIS_T* associated with constitutive law DIS_CHOC [R5.03.17] makes it possible to give an account of a specific contact in a built-in direction. They are well adapted to the modelization of shocks and for this reason are often used in dynamics on modal base [U4.53.21] and in explicit dynamics [U4.53.01].
Springs can be based indifferently on a specific mesh or a segment. In all the cases, it is necessary to correctly direct each element with command AFFE_CARA_ELEM [U4.42.01].
The contact as friction are solved by penalization (cf §3.2.23.2.2). The stiffness of penalization, the coefficient of kinetic friction as well as initial clearances is specified in material DIS_CONTACT (command DEFI_MATERIAU, [U4.43.01]).

This kind of element is not usable in large displacements because the direction of contact fixed and is given by the initial directional sense of the discrete element.

Benchmarks SSNL130A and SDND100C implement springs of contact.

4.9 Elements of joints (hydro) mechanical with contact and friction

Les elements of joints (hydro) mechanical PLAN_JOINT (_HYME) and 3D_JOINT (_HYME) make it possible to modelize the opening of a crack under the pressure of a fluid and friction on edges of crack closed with model JOINT_MECA_FROT [R7.01.25]. It is possible to couple the opening of crack and the propagation of the fluid with the modelizations *_HYME.
The formulation of contact-friction is penalized and the related parameters are indicated under key word JOINT_MECA_FROT of command DEFI_MATERIAU [U4.43.01].

Benchmarks SSNP142C and SSNP142D provide an example of application of such elements on the modelization of a stopping.

5 Bibliography

[1] Mr. ABBAS. “Discrete Formulation of contact-friction”. Documentation of reference of *Code_Aster*. [R5.03.50].

[2] MR. ABBAS. “Elements of contact derived from a continuous hybrid formulation”. Documentation of reference of *Code_Aster*. [R5.03.52].

[3] T. SOZA. “Operator `DEFI_CONTACT` ”. Documentation of use of *Code_Aster*. [U4.44.11].

[4] A. KONTER. “Advanced Finite Element Benchmarks Contact”. NAFEMS, 2006.

[5] A. KONTER. “How to undertake has contact and friction analysis”. NAFEMS, 2000.