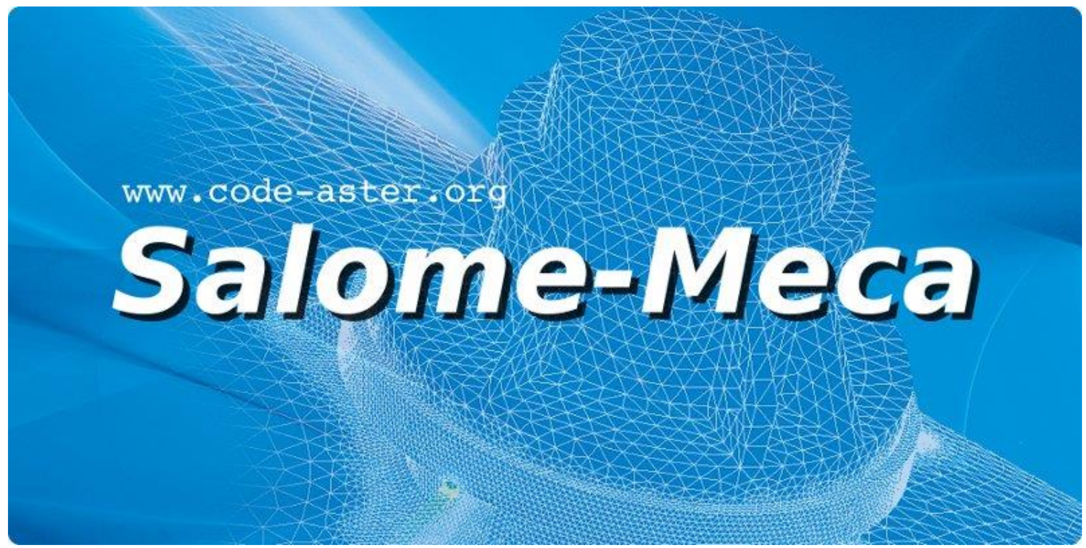
Contact and friction



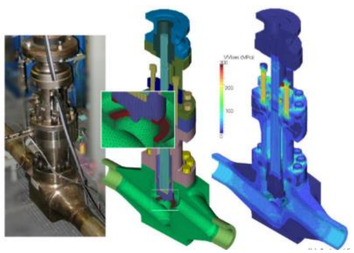
Code_Aster, Salome-Meca course material

Outline

- Introduction
- Frictional Contact problems : Definition
- Numerical treatment
  - Pairing operations
  - Formulations
  - Solution algorithmic for quasi-static
- General recommendations
  - Choice of Master or Slave
  - Choice of algorithm & formulation
  - STAT_NON_LINE/DYNA_NON_LINE
- Summary

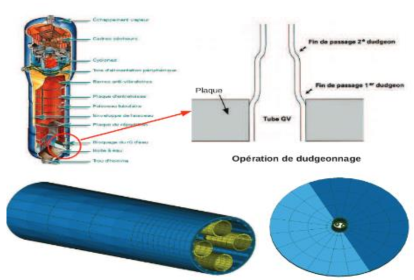
Introduction

- Frictional-contact problems are involved in many structural mechanical problems.
- code_aster offers differents manners to deal with it. DEFI_CONTACT is one of the possibilities.
- Contact problems remains one of the difficult subject in the field of computational contact mechanic.

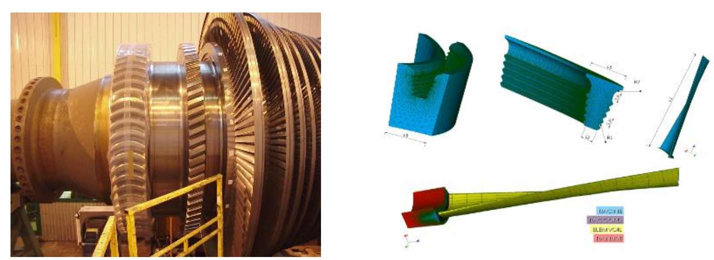


Tap problems

Wind turbine soil-structure problems



Fabrication process



Turbine blade problems

Introduction

Context of the presentation

- Continuum mechanics applied to deformable solid bodies with non linear materials
- The evolution of the structure can be in quasi-static or in dynamic
- Finite element framework is assumed

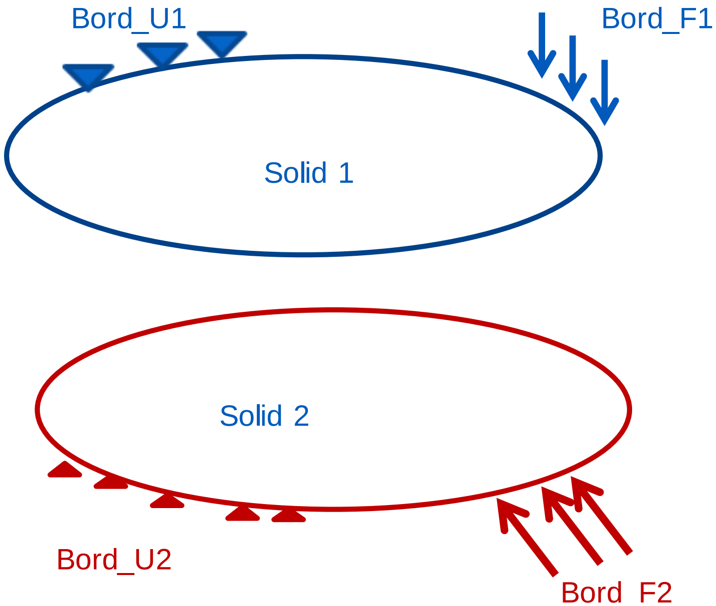
Goal of the presentation

- To show different manners to treat contact problems with *Code_Aster with DEFI_CONTACT and STAT(DYNA)_NON_LINE*
- To get over some difficulties induced by numerical treatment of frictional contact
- To show how to make robust and reliable computations of some problems

Outline

- Introduction
- Frictional Contact problems : Definition
- Numerical treatment
  - Pairing operations
  - Formulations
  - Solution algorithmic for quasi-static
- General recommendations
  - Choice of Master or Slave
  - Choice of algorithm & formulation
  - STAT_NON_LINE /DYNA_LINE
- Summary

Unilateral Contact – Definition (1)



Contact conditions

- **Geometrical** condition : « non-penetrability » of the matter (Signorini)

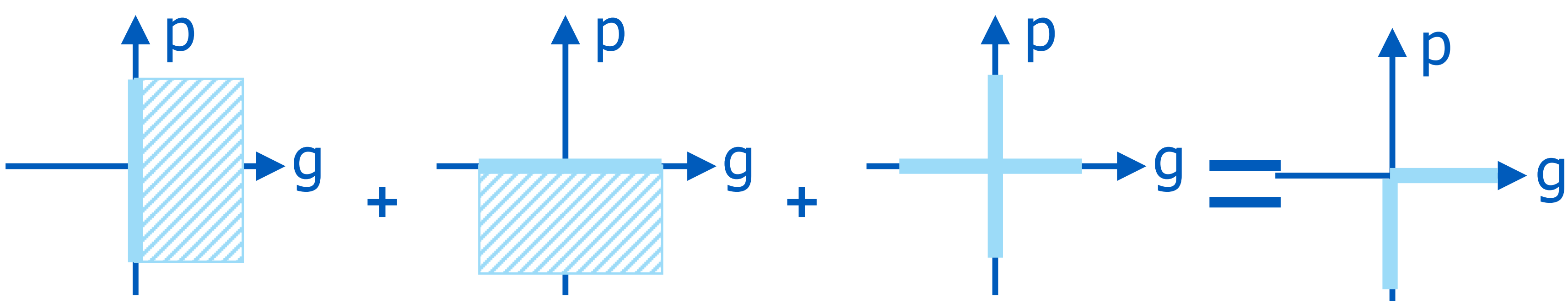
  g≥0 g>0 no contact
  g=0 contact (continuity of matter)

- **Mechanical** condition : « no tension » (Hertz)

  p≤0 p=0 no contact
  p<0 contact (compression)

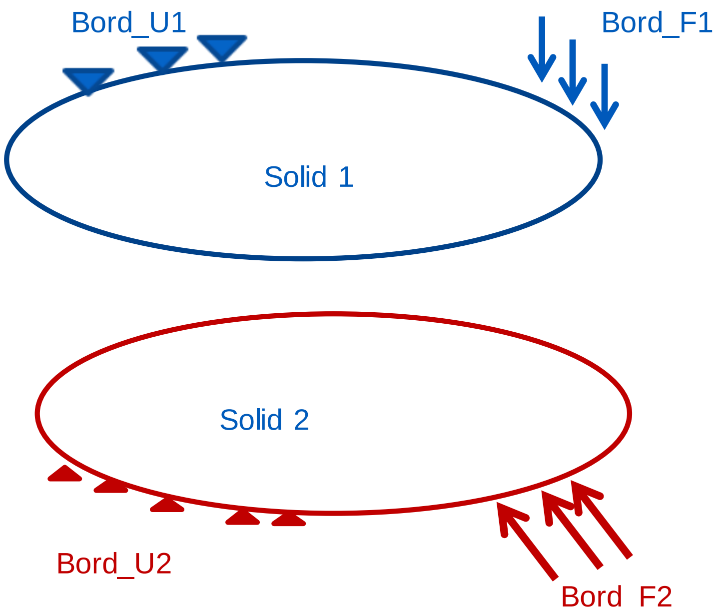
- **Energetic** condition : complementarity or exclusion (Moreau)

  p.g=0 p=0 separation
  g=0 contact



No penetration *No tension* *Complementarity*

Non-associated Friction law – Definition (2)



Friction conditions

- **Geometrical** condition :sliping or sticking

  $g \geq 0$ $g>0$ no contact

  $g=0$ contact (continuity of matter)

  Vt =0 → Stick

  else → Slip

- **Mechanical** condition : « no tension » (Hertz)

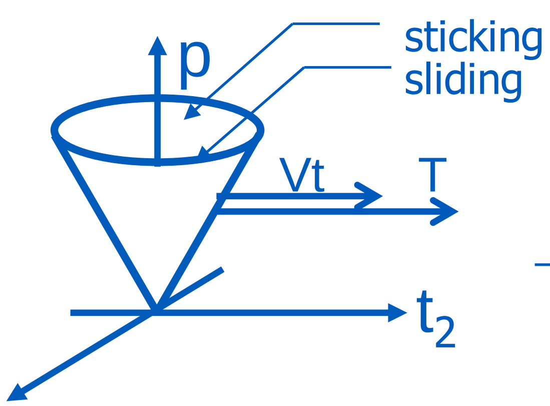
  $p \leq 0$ $p=0$ no contact

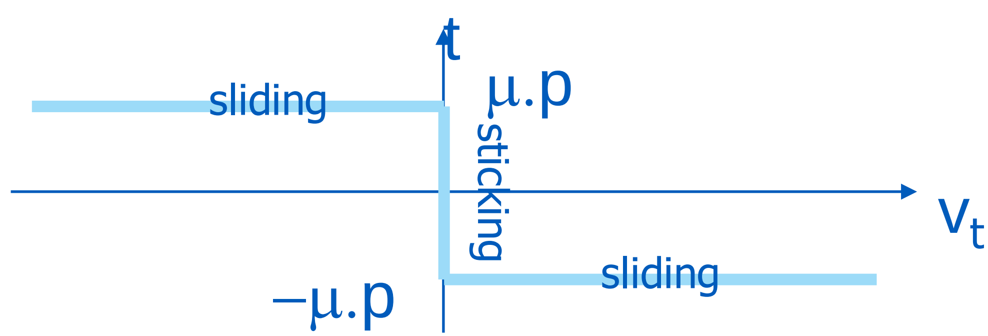
  $p<0$ contact (compression)

  if (Vt=0, Stick) then $| T | < \mu * p$

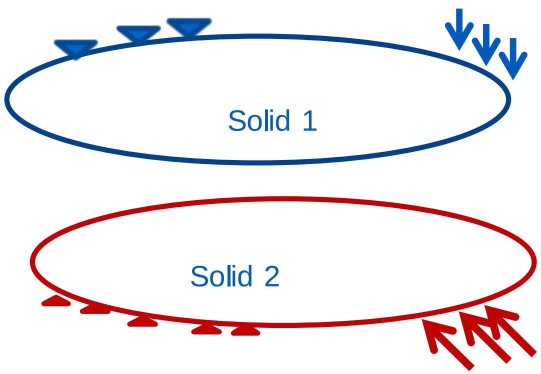
  else (Vt = alpha * T and $| T | = \mu * p$)

- **Non associativity** : the sliding speed is non colinear to the friction cone in the space of contact forces.





Problem formulation – Definition (3)



$$
\begin{cases}
Sd*\mathbf{M}.\ddot{\mathbf{u}} + \mathbf{K}.\mathbf{u} + F_c(u_n, \lambda_n) + F_t(u_n, \lambda_n, v_t, \lambda_t) = \mathbf{F} \\
\mathbf{u}_n \le d_0, \boldsymbol{\lambda}_n \ge 0, (\mathbf{u}_n - d_0)\,.\boldsymbol{\lambda}_c = 0 \\
\dfrac{\mathbf{A}_t.v_t}{\|\mathbf{A}_t.v_t\|} = \dfrac{\lambda_t}{\|\lambda_t\|} \quad \text{and} \quad \|\lambda_t\| = \mu\lambda_c \quad \text{and} \quad \mathbf{A}.\mathbf{u}_n - d_0 = 0, \\
v_t = 0 \quad \text{and} \quad \|\lambda_t\| \prec \mu\lambda_c \\
Sd = 0 \rightarrow \text{quasi-static}, Sd = 1 \rightarrow \text{dynamic} \\
F_c(u_n, \lambda_n) \rightarrow \text{contribution of contact force} \\
F_t(u_n, \lambda_n, v_t, \lambda_t) \rightarrow \text{contribution of friction force}
\end{cases}
$$

A discrete formulation

Mechanical problem

- **Equilibrium problems**
  - Quasi-static (STAT_NON_LINE)
  - Dynamic (DYNA_NON_LINE)
- **Boundary conditions : Dirichlet, Neuman,…**
  - Direct imposition with AFFE_CHAR_CINE
  - Lagrange reinforcement with imposition with AFFE_CHAR_MECA
- **Frictional-contact conditions**
  - Various method in DEFI_CONTACT which will be explained later
- **Difficulties of the numerical treatment**
  - Non-smooth mechanics : severe discontinuities due to frictional-contact status
  - Contact forces depend on the pairing
  - Friction depend on the contact pressure
  - High coefficient of friction induce loss of unicity such that algorithms can fail to converge
  - In dynamics, optimal dissipative schemes must be used
  - Non linear materials such as Large transformations or plasticity increase the cost of the computation

You say DEFI_CONTACT ?

```
CONT=DEFI_CONTACT(MODELE=MO,
                  FORMULATION='DISCRETE‘, # ‘CONTINUE’
                       ZONE=_F(
                      GROUP_MA_MAIT='CONT2‘, GROUP_MA_ESCL='CONT1‘,   ALGO_CONT='CONTRAINTE',),);
```

Name of concept

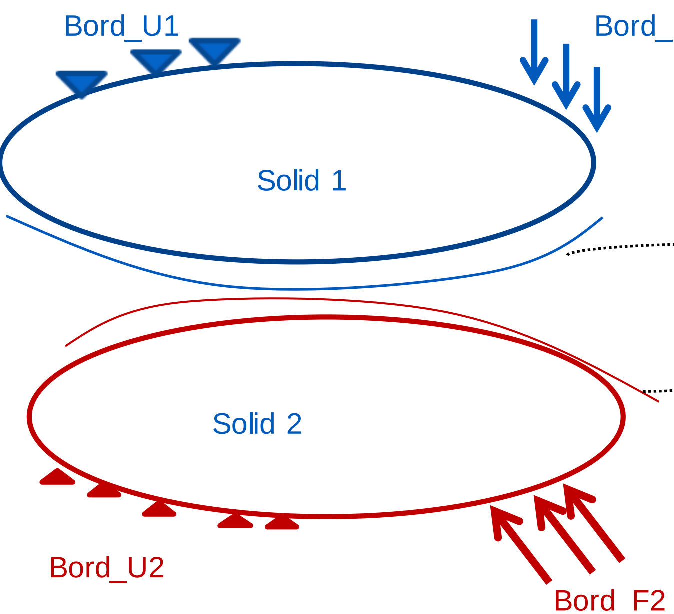
Contact Operator

Formulation choice

Master surface

Slave surface

Solution method



The contact surfaces must be choosen properly

Outline

- Introduction
- Frictional Contact problems : Definition
- Numerical treatment
  - Pairing operations
  - Formulations
  - Solution algorithmic for quasi-static
- General recommendations
  - Choice of Master or Slave
  - Choice of algorithm & formulation
  - STAT_NON_LINE / DYNA_NON_LINE
- Summary

Numerical treatment – General procedure

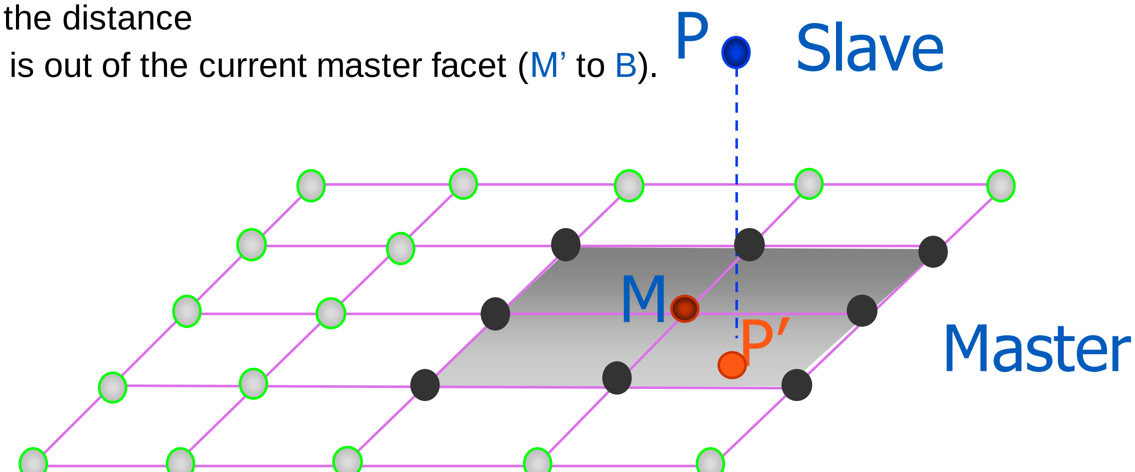
- Step 1 : preliminary treatments which consist to build the contact element according to the the user parametrization in DEFI_CONTACT.
  - Node-To-Node contact element (FORMULATION=DISCRETE only)
  - Node-To-Segment contact element (FORMULATION=DISCRETE or CONTINUE)
  - Segment-To-Segment contact element (FORMULATION=CONTINUE with Local Average Contact method)

- Step 2 : Computational contact mechanic treatment in STAT_NON_LINE.
  - Step 2.1 : Pairing technique : this step is important for the computation of the gap at each Newton iteration
  - Step 2.2 : Contact initializations
  - Step 2.3 : Formulations : Contact matrix and contact vectors calculation
  - Step 2.4 : Specific treatments of mechanical quantities according to adaptation method

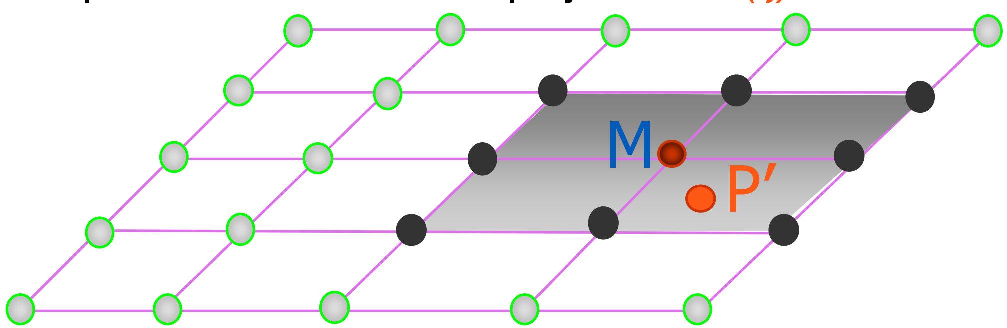
- Step 3 : Extract contact mechanics quantities
  - CALC_PRESSION : to compute contact pressure using the stress fields
  - CONT_NOEU : default field generated by code_asterwhich contains contact mechanics quantities
    - Avalaible with all type of resolution except Local Average Contact method
  - CONT_ELEM : default field generated by code_asterwhich contains contact mechanics quantities
    - Avalaible with only resolution withLocal Average Contact method

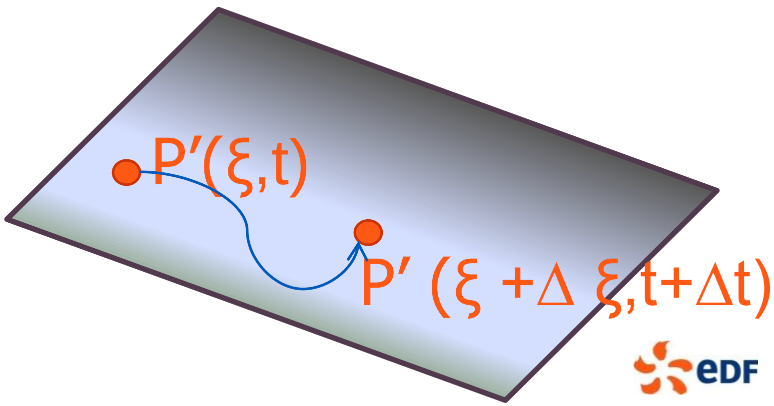
Numerical treatment – Pairing (1)

- Node-to-facet pairing : `APPARIEMENT` / `DEFI_CONTACT`
  - Search the closest master node M to the slave node P (distance)
  - Search for all master facets containing M, the closest one for wich PP’ is minimum
  - Selection of the master element that minimizes the distance to the slave node (PM)
    - Compute the projection M and the distance
    - Correction of the projection if it is out of the current master facet (M’ to B).



  - Within the deformation process, it is necessary to compute by Newton Method the parametrization of the projection P’(ξ)





Outline

- Introduction
- Frictional Contact problems : Definition
- Numerical treatment
  - Pairing operations
  - Formulations
  - Solution algorithmic for quasi-static
- General recommendations
  - Choice of Master or Slave
  - Choice of algorithm & formulation
  - Avoid frictional status cycling
  - Influence of the time integration schemes in dynamics
  - General recommendations
- Summary

Numerical treatment – Formulations (1)

In General

- Frictional contact problems are called constrained opitmization problems
  - Problem consists to find u(x) = Argmin F(u(x)) with u(x) in C(x)
  - The constraints C(x) can be removed using Lagrange multiplier or penalty function
  - Solution algorithms depend strongly to the formulation

Two families of formulations in *Code_Aster*

- `FORMULATION='DISCRETE'`
  - Suits to mathematical programming approaches in the litterature (Like flexibility method)
  - Use **Lagrange multiplier** or **penalization** to reinforce constraints **on contact nodes**
  - Frictional-contact status depend on the gap
  - Theoretical documentation : R3.05.50

- `FORMULATION='CONTINUE'`
  - Suits to continuum mechanical approaches
  - Use **augmented Lagrange multiplier** or **penalization** to reinforce constraints on **slave contact surface**
  - **Need numerical integration techniques to compute contact forces**
  - Frictional-contact status depend a combination of gap and force
  - Theoretical documentation : R3.05.52

Numerical treatment – Formulation (3)

	Perf	Reliability	DOFs	Robustness
DISCRETE/ 'CONTRAINTE'	+++ Schur	+	Hundreds	++ no_friction
DISCRETE/ 'GCP'	+++	+	Limitless	+ no_friction
DISCRETE/ 'PENALISATION'	+++	+	Thousands	++ with_friction
'CONTINUE'/ 'STANDARD'/ 'PENALISATION'	++	++	Limitless	+++ (with friction)
'CONTINUE'/ 'LAC'	+	++++ Suits for incompatible mesh	Limitless	++ (without friction)

Outline

- Introduction
- Frictional Contact problems : Definition
- Numerical treatment
  - Pairing operations
  - Formulations
  - Solution algorithmic for quasi-static
- General recommendations
  - Choice of Master or Slave
  - Choice of algorithm & formulation
  - STAT_NON_LINE / DYNA_NON_LINE
- Summary

Numerical treatment – Solution Algorithms(1)

- **FORMULATION='DISCRETE'**

ZONE=_F(
…
ALGO_CONT="CONTRAINTE"/"PENALISATION"/"GCP",
..
ALGO_FROT="PENALISATION",
)

- **FORMULATION='CONTINUE'**

ALGO_RESO_GEOM='POINT_FIXE' or 'NEWTON'
ALGO_RESO_FROT='POINT_FIXE' or 'NEWTON'
ALGO_RESO_CONT='POINT_FIXE' or 'NEWTON'
ZONE=_F(
…
ALGO_CONT="STANDARD"/"PENALISATION''/''LAC'',
..
ALGO_FROT ="STANDARD"/"PENALISATION",
)

General remarks :

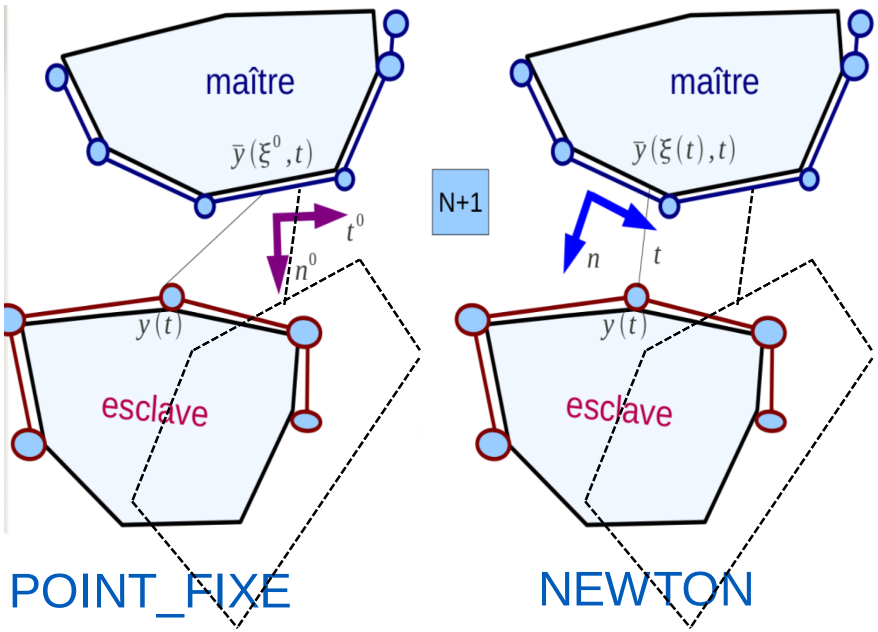
- **Friction is complex to solve due to the dependence on contact and additional cycling problems (slip back/slip forward or stick/slip)**
- **Friction coefficient is supposed to be constant (no dynamic friction coefficient)**
- **For Large slip problem and non linear material this family of method seems to be less robust**
- **Friction matrices induce non-symmetric problems**
- **More recommendations in the next sections**
- **For incompatible mesh use LAC method**

Numerical treatment – Solution Algorithms(2)

What is ALGO_RESO_GEOM : 'POINT_FIXE' Vs 'NEWTON'

- FORMULATION='CONTINUE'

ALGO_RESO_GEOM='POINT_FIXE' or 'NEWTON'
Only avalaible for 'STANDARD' and 'PENALISATION' methods not for 'LAC' method.



POINT_FIXE NEWTON

- Help for small sliding problems
- If small variation of normals use ALGO_RES_GEOM='POINT_FIXE' + REAC_GEOM='CONTROLE'

Numerical treatment – Solution Algorithms(1)

Formulations	Status	Geometry	Friction	Robustness
'CONTINUE' : 'STANDARD' / 'PENALISATION'	POINT_FIXE or NEWTON Use GLISSIERE if necessary	POINT_FIXE or NEWTON Use LISSAGE if necessary	POINT_FIXE or NEWTON	+ Small perturbation, + Large sliding with plastic behaviors - Mesh compatibility in contact zones
'CONTINUE' / 'LAC'	NEWTON	NEWTON Use LISSAGE if necessary	Does not exists	+ Small perturbation, + Mesh incompatibility in contact zones - Large sliding with plastic behaviors → TYPE_JACOBIEN could help.
'DISCRETE'	Use GLISSIERE if necessary	Use LISSAGE if necessary		+ Only hundreds contact dofs

Outline

- Introduction
- Frictional Contact problems : Definition
- Numerical treatment
  - Pairing operations
  - Formulations
  - Solution algorithmic for quasi-static
- General recommendations
  - Choice of Master or Slave
  - Choice of algorithm & formulation
  - STAT_NON_LINE/DYNA_NON_LINE
- Summary

General recommandations on DEFI_CONTACT

```
CONT=DEFI_CONTACT(MODELE=MO,
                    FORMULATION='DISCRETE‘, # ‘CONTINUE’
                          ZONE=_F(
                         GROUP_MA_MAIT='CONT2‘, GROUP_MA_ESCL='CONT1‘,   ALGO_CONT='CONTRAINTE',),);
```

Name of concept

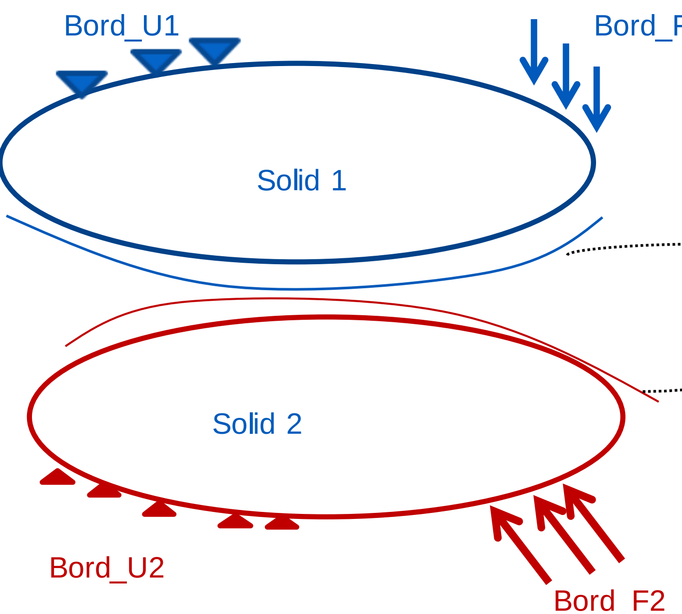
Contact Operator

Formulation choice

Master surface

Slave surface

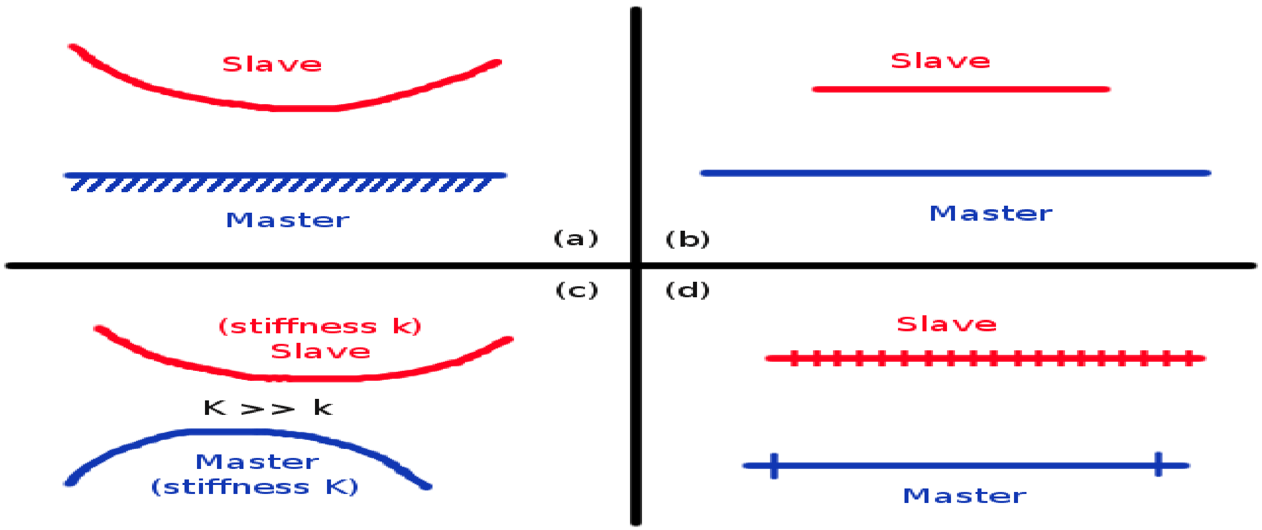
Solution method



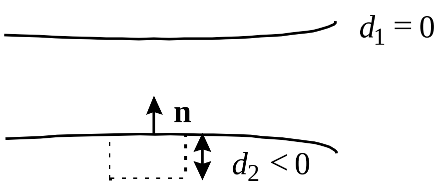
The contact surfaces must be choosen properly

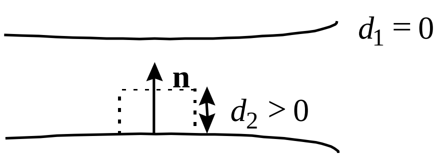
General recommendations – master/slave

- Master slave should be the most rigid surface and slave should be the most curved one



- Use LISSAGE='OUI' for non-smooth surfaces
- For structural element such as beam/shells use **DIST_POUTRE** or **DIST_COQUE**
- It is possible to add extra gap **DIST_MAIT**, **DIST_ESCL**, **DIST_POUTRE** or **DIST_COQUE**





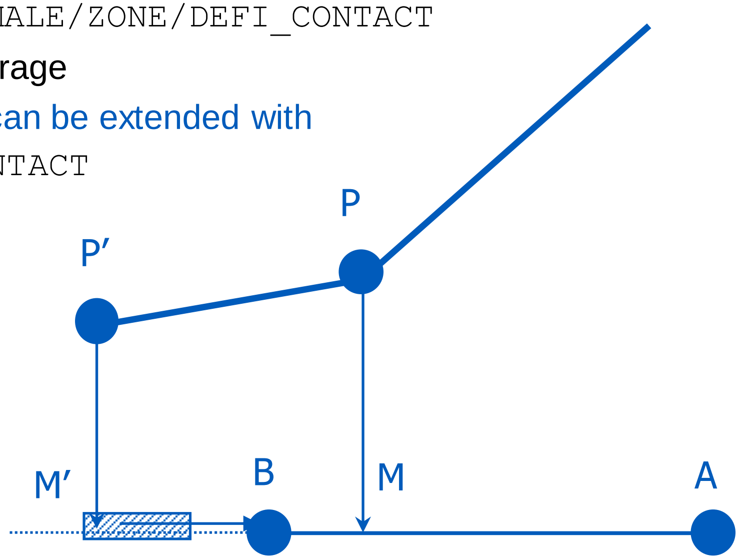
$$(\delta \mathbf{u}_P - \delta \mathbf{u}_M).\mathbf{N} \le d_{-} - (d_1 + d_2)$$

General recommendations – pairing

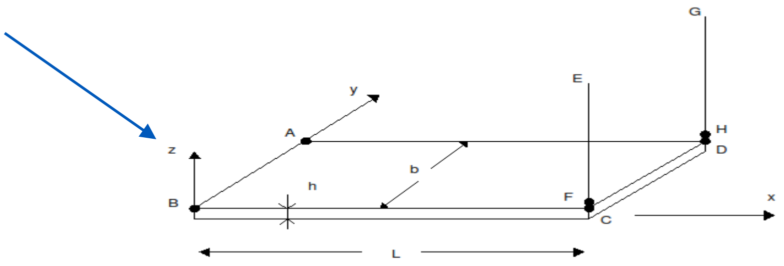
- Advanced options for the user :
  - The user can activate the initial gap PM.N with DIST_MAIT/ZONE/DEFI_CONTACT
  - The user can define Normal N. NORMALE/ZONE/DEFI_CONTACT

  It may be the master, slave or the average
  - For specific cases the master facet can be extended with

    TOLE_PROJ_EXT/ZONE/DEFI_CONTACT

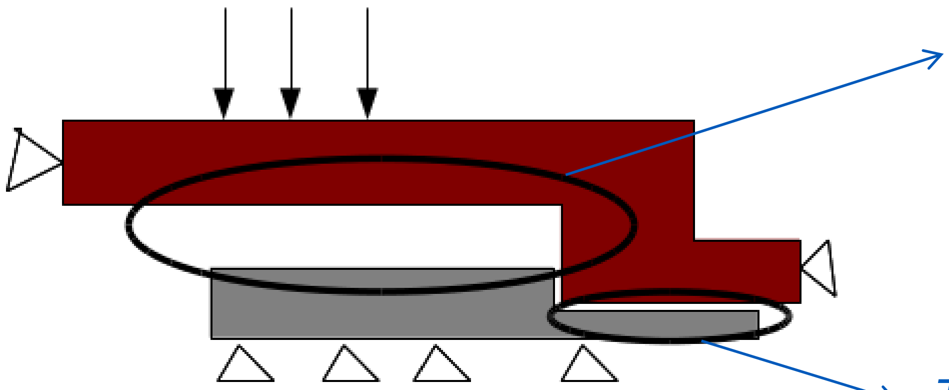


- Specific case of pairing techniques : Node-to-Node pairing,
  - APPARIEMENT = 'NODAL' only for FORMULATION = 'DISCRETE'
  - It suits for beam contact problems for example



General recommendations – Optimization of the resolution (1)

- Contact without resolution (**`RESOLUTION='NON'`**)
  - Simply check for interpenetration without imposing contact
  - Really cheap (only pairing)
  - Useful for checking the interpenetration of the lips of a crack
  - Tolerance to detect interpenetration **`TOLE_INTERP`**
  - Optional stop when interpenetration detected **`STOP_INTERP`**

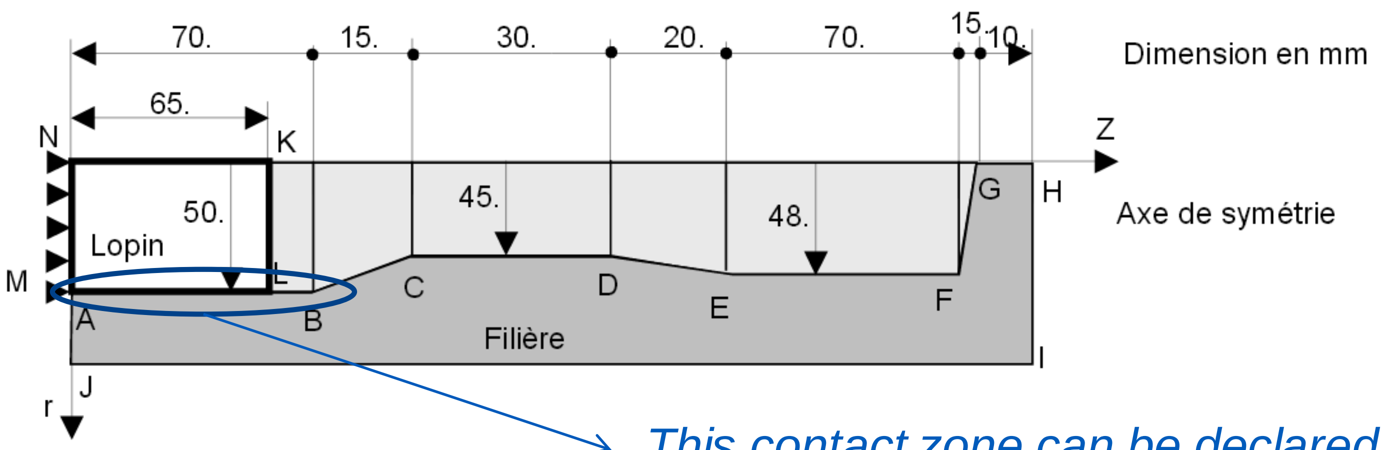


This potential contact zone should be declared with RESOLUTION='NON'

This potential contact zone should be declared with RESOLUTION='OUI'

General recommendations – Optimization of the resolution (2)

- Bi-lateral contact
  - Option **`GLISSIERE='OUI'`** with the active set or the continuum-based methods
  - Contact is kept active as soon as the interpenetration is detected
  - The two surfaces can slide over each other
  - **Generate tension when the surfaces should normally separate**



This contact zone can be declared as GLISSIERE='OUI'

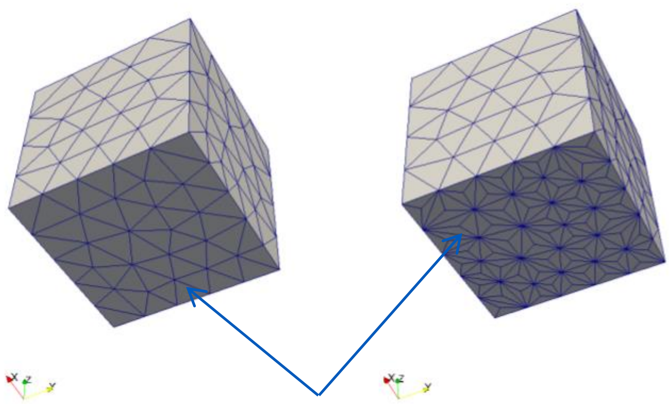
General recommandations – Optimization of the resolution (3)

- Compatibility with Dirichlet boundary conditions
  - Adding Lagrange multipliers on the slave nodes: possibility of zero pivots in the case of redundancy
  - For the discrete formulations with Lagrange multipliers, an algorithm for automatic elimination is active (message "**`ZERO PIVOT OTE`**" in the .mess file)
    - The automatic algorithm mainly detects the redundancy along the reference axis
  - The redundant nodes can be excluded by **`SANS_NOEUD / SANS_GROUP_NO`**

This contact node should be excluded to the contact zone SANS_NOEUD or SANS_GROUP_NO

General recommandations – USE OF LAC METHOD

- Pre-treatment of the mesh incompatibility with DECOUPE_LAC
  - This step help the code to prepare a suit treatment for mesh incompatibility in contact zone : use DECOUPE_LAC

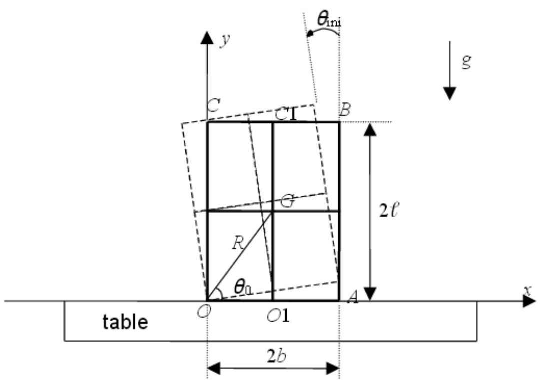


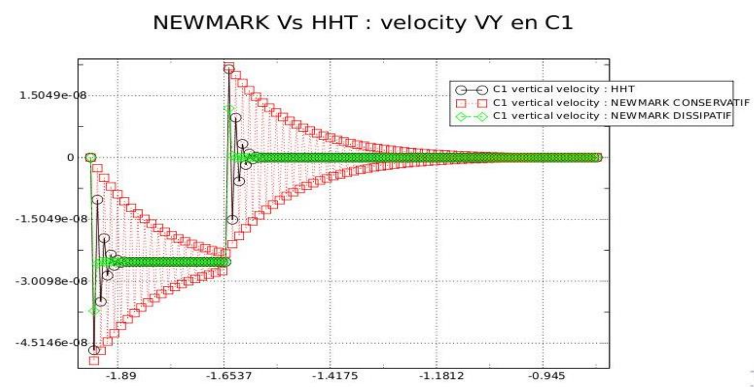
The initial slave contact zone is modified with DECOUPE_LAC

  - With FORMULATION='CONTINUE' use ALGO_CONT='LAC'

General recommandations – *_NON_LINE (1)

- Contact is solved in **STAT_NON_LINE/DYNA_NON_LINE**
  - Contact definition is given with the keyword **CONTACT**
  - Line search may be used in conjunction with the discrete contact formulation but its efficiency in this case is not proven
  - In dynamics choose : HHT schemes





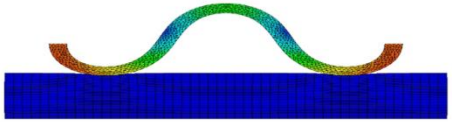
  - In friction, use ADAPTAPTION=‘ADAPT_COEF’ in DEFI_CONTACT
  - If incompatible mesh in contact zone, use INTEGRATION=‘SIMPSON’/NEWTON-COTES in DEFI_CONTACT
  - The option **INFO=2** of **STAT_NON_LINE** provides a lot of information when contact is present and should therefore be avoided.

General recommandations – *_NON_LINE (2)

- Read the manual !
  - `DEFI_CONTACT` User command **[U4.44.11]**
  - « instructions for the use of contact with *Code_Aster* » **[U2.04.04]**
- Outward normal vectors ?
- Is **`STAT_NON_LINE`** OK without using contact? (mainly for the discrete formulation)
- Are there redundant Dirichlet boundary conditions with contact ?
- Is the model OK in linear elasticity?
- If the Quasi-static problem can't be solved, someone can use slow dynamic framework to come over difficulties.

  Ex : ring on block

contact



friction

General recommandations – Post-processing

- Post-processing of contact
  - **CALC_CHAMP** : compute the nodal forces (**FORC_NODA**) and the nodal reactions (**REAC_NODA**)
  - Contact forces: **RESULTANTE** of the **FORC_NODA** with **POST_RELEVE_T**. Caution if a node is submitted to kinematic constraints
  - The data structure **CONT_NOEU or CONT_ELEM** provided by **STAT_NON_LINE** contains information on the contact status (gaps, contact status, forces, …)
  - In the continuum-based method, the **LAGS_C** dof is the contact pressure if there is small transformations otherwise use **CALC_PRESSION**

End of presentation

Is something missing or unclear in this document?
Or feeling happy to have read such a clear tutorial?

Please, we welcome any feedbacks about Code_Aster training materials.
Do not hesitate to share with us your comments on the Code_Aster forum dedicated thread.