To introduce new kinematical boundary conditions

Abstract:

This document presents the two utility routines making it possible to rather easily introduce new types of "kinematical" boundary conditions (i.e of the linear relations between unknown degrees of freedom). Contents

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

1

what one calls "loading" in Aster (vocabulary of the "mechanics") is what the user defines in commands AFFE _CHAR_*. One distinguishes the loadings in general of "forces" [D5.03.01] and the loadings in "displacements" (or "kinematics"). This

document explains how to introduce new kinematical loadings. What

2 a linear relation? This

statement indicates a linear stress on the degrees of freedom of the system to be studied:

- degrees of freedom of quantity TEMP_R for the thermal phenomenon,
- the degrees of freedom of quantities DEPL_R or DEPL _C for the mechanical phenomenon,
- the degrees of freedom of the quantity NEAR _C for the acoustic phenomenon.

The coefficients of this linear relation are real constants (or complexes), the second member can be real, complex or of standard " function " (K8) .

A linear relation can be written: where

$$\alpha_1 ddl_1 + \alpha_2 ddl_2 + ... + \alpha_n ddl_n = \alpha_0$$

(ou)

$\alpha_i \in \mathbb{R}$ $\mathbb{C}$ () ($i=1,n$ or
$\alpha_0 \in \mathbb{R}$ formulates $\mathbb{C}$ or function)

degrees of freedom are ddl_i degrees of freedom carried by one or more different nodes. Linear

examples of relations: blocking

$DX(N1)=0.$	of the component of DX the node temperature $N1$
$TEMP(N3)=100.$	imposed on for $100.$ the node $N3$
$DY(N1)-DY(N2)=0.$	the nodes and $N1$ have $N2$ same displacement DY
$\cos\alpha\, DX(N1)+\sin\alpha\, DY(N1)=0.$	the node east $N1$ compels to move on the line perpendicular to the vector (, $\cos\alpha$) $\sin\alpha$ (in 2D). How

3 does one introduce linear relations into a modelization?

The linear relations which one defined in the paragraph §2: force

- the solution which one seeks, they
- belong to what one in general calls the "boundary conditions", in
- Code_Aster *they* are of the components loads (standard tank _acou, tank _ther, tank _meca). These

linear relations are thus introduced by the user via commands AFFE_CHAR_MECA (_F), AFFE_CHAR_THER (_F), AFFE _CHAR_ACOU, or AFFE_CHAR_CINE . These

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

linear relations can be "dealt" with two ways: one

- eliminates an unknown for each linear relation: method of elimination [D3.03.01], one
- "dualise" the relation by adding 2 additional unknowns to him: parameters of Lagrange [R3.03.01]. In

Code_Aster , the method of elimination is used for relations resulting of the command AFFE_CHAR_CINE . One will speak in this case about linear relations "kinematical", although this term is not very judicious. One limits oneself then to relations of the type: DDL

= cste

the other relations resulting from commands AFFE_CHAR_MECA , AFFE_CHAR_THER and AFFE _CHAR_ACOU are always dualisées. Examples

of keywords factor generating of the linear relations: Commands

Key	**Keys Factor processing**	**AFFE_CHAR_CINE**
MECA_IMPO	elimination	AFFE_CHAR_MECA_F
LIAISON_OBLIQUE	dualisation	AFFE_CHAR_THER
TEMP_IMPO	dualisation	AFFE_CHAR_MECA
LIAISON_DDL	dualisation	AFFE_CHAR_MECA
LIAISON_SOLIDE	dualisation	

command AFFE_CHAR_CINE makes it possible to easily introduce all the linear relations simple (DDL = cste) which one can define. On the other hand

, although in theory (thanks to key word LIAISON_DDL), one can introduce any linear relation, the number of coefficients to calculating can become very large. To think for example of the linear relations which it should be written to say that 4 nodes are interdependent (connected by an indeformable solid).

The many keywords making it possible to the user to define these linear relations are there to facilitate work to him: LIAISON_OBLIQUE

for	bearings slipping into an oblique coordinate system TEMP_IMPO
to impose	a temperature LIAISON_GROUP
	to connect nodes two to two…
and	
LIAISON_DDL for	the other cases… This

a large number of key key (which will be able to only grow) requires to give itself software tools allowing:

- not to duplicate a code unnecessarily,
- to facilitate the introduction of new key keys into commands AFFE_CHAR_MECA , AFFE _CHAR_ACOU and AFFE_CHAR_THER . It

is of these tools about which we will speak in the following paragraphs. To introduce

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

4 a new keyword of type "linear relation" We

of the command give in this paragraph a groundwork for the writing of a routine "carrying out" a key word AFFE_CHAR_MECA (or _THER or _ACOU), this key word factor allowing the user to define linear relations. Are

: MFAC

- the key word factor CAMFAC
- the name of the routine corresponding to him

the goal of routine CAMFAC is "to scan" the data of the user behind key word MFAC , to translate these data into linear relations and to store these relations in the load (here of standard tank _meca) that the user is defining. For

that, one has two utility routines: AFRELA

	to assign a linear relation to a SD of the type LISTE _RELA (list of linear relations) AFRLCH
	"to add" a SD LISTE _RELA to a SD CHARGE These

routines force to pass by a SD intermediary (temporary) of type LISTE _RELA. That weighs down a little the programming but has the following advantages: performance profits

- , because routine AFRLCH is expensive in CPU ,
- a great compliance to carry out the principle of overload (See the § 6).

The groundwork of routine CAMFAC is thus the following: SUBROUTINE

```
        CAMFAC (CH) CHARACTER
        * (*) CH C
in  jxvar  CH: SD       CHAR_MECA with enriching goal
C  : to enrich        the CH load by the linear relations definite C
under               the key word factor MFAC buckles

on the linear relations •

acquisition of the coefficients of the linear relation: (routines
GETVXX), •
addition of the linear relation to SD LISTE_RELA Cal
AFRELA (… , "&&CAMFAC.LISTE_RELA") fine

buckles •

addition of SD LIST_RELA with the CHARGE: CH Cal
AFLRCH ("&&CAMFAC.LISTE_RELA", CH) END

Note:
```

> *SD* LISTE _RELA (temporary *) is specific to routine CAMFAC , its name respects the convention of the names of objects of work: ""* ,
> *the principle of overload (cf [U2.01.00 §3.7]) thus relates to only the occurrences of key word MFAC , This*
> *SD is destroyed at the time of the call to AFLRCH .*

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

5 Routines AFRELA and AFLRCH

5.1 routine AFRELA SUBROUTINE

```
      AFRELA (COEFR, COEFC, DDL, NOEUD, NDIM, DIRECT, +
     NBTERM          , BETAR, BETAC, BETAF, TYPCOE, TYPVAL, LISREL)
C----------------------------------------------------------------------
C
GOAL: ASSIGNMENT OF RELATION  BETWEEN DDLS A A SD LISTE_RELA C
(IF       OBJET LISREL DOES NOT EXIST, IT EAST CREATES) C
C----------------------------------------------------------------------
C
COEFR (NBTERM) - IN - R -   : TABLEAU OF THE COEFFICIENTS OF RELATION C
                             THE COEFFICIENTS ARE REAL
C----------------------------------------------------------------------
C
COEFC (NBTERM) - IN - C -   : TABLEAU OF THE COEFFICIENTS OF RELATION C
                             THE COEFFICIENTS ARE COMPLEX
C----------------------------------------------------------------------
C
DDL (NBTERM) -   IN - K8 -  : TABLEAU OF THE DDL OF RELATION
C----------------------------------------------------------------------
C
NOEUD (NBTERM) - IN - K8 -  : TABLEAU OF THE NODES OF RELATION
C-----------------------------------------------------------------------
C
NDIM (NBTERM) -  IN - I -   : DIMENSION OF THE PROBLEM (0, 2 OR 3) C
IF                              = 0 PAS OF CHANGE OF COORDINATE C
                             RELATION ARE GIVEN IN THE BASE C
TOTAL
C----------------------------------------------------------------------
C
DIRECT (3, NBTERM) - IN - R -: TABLEAU OF THE RELATIVE VECTORS A EACH C
TERM                              DEFINING THE DIRECTION OF C
COMPONENT                              WHICH ONE WANTS TO FORCE
C----------------------------------------------------------------------
C
NBTERM -        IN - I -    : TERM MANY OF RELATION
C----------------------------------------------------------------------
C
BETAR -         IN - R -    : VALEUR REAL OF THE SECOND MEMBER
C----------------------------------------------------------------------
C
BETAC -         IN - C -    : VALEUR COMPLEXE OF THE SECOND MEMBER
C----------------------------------------------------------------------
C
BETAF -         IN - K8 -   : VALEUR FONCTION OF THE SECOND MEMBER
C----------------------------------------------------------------------
C
TYPCOE -        IN - K4 -   : TYPE OF THE COEFFICIENTS OF RELATION: C
=                               "REEL" OR "COMP"
C----------------------------------------------------------------------
C
TYPVAL -        IN - K4 -   : TYPE OF THE SECOND MEMBER C
=                               "REEL" OR "COMP" OR "FONC"
C----------------------------------------------------------------------
C
LISREL -        IN - K19 -: NOM OF SD LISTE_RELA C
```

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

```
-                JXVAR -
C---------------------------------------------------------------------
Two
```

cases are to be considered:

- the degrees of freedom to be connected are given in the absolute coordinate system: DX , DY , … certain
- degrees of freedom to be connected are given in a local coordinate system. Case

A (all in the absolute coordinate system): NBTERM

- `is` the number of degrees of freedom connected by the relation. NDIM
- `is` a vector filled with 0 DIRECT
- `is` useless. Example

1: one

wants to impose: (function $3.\times DX(N1)+2.\times DY(N2)-4.\mathrm{DRZ}(N1)=F$) NBTERM

```
=  3 TYPCOE
=  "REEL" TYPVAL
=  "FONC" COEFR
=   (3. ,  2. , -4. ) NDIM
=    (0,   0,  0  )  DDL
=      ("DX", "DY", "DRZ") NOEUD
=   ("N1", "N2", "N1") BETAF
=   "F" Case
```

B (local coordinate system): For

each node implied in the relation, one can give a local coordinate system in which the relation is simpler (the norm on a surface for example). Example

2: maybe

, n an unit vector of components. (nx, ny, nz) It
is wanted that following displacement with n the node is $N3$ null. NBTERM

```
=  1 TYPCOE
=  "REEL" TYPVAL
=  "REEL" COEFR
=   (1.) NDIM
=     (3) DIRECT
=  (nx, ny, nz) DDL
=      ("DEPL") NOEUD
=   ("N1") BETAR
=   0. Note:
```

NBTERM

> *is not the number of terms of the final relation here (: 3). When one employs (for a "term") the possibility of a local coordinate system NDIM `/= 0` the name of the degree of freedom must be conventionally " DEPL `" or "` ROTA `" Example`*

3: RC
are

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

n1	: an unit vector of components (n1 X, n1 there, n1 Z) N2
	: an unit vector of components (N2 X, N2 there, N2 Z)

following data: NBTERM

```
=  3 TYPCOE
=  "REEL" TYPVAL
=  "REEL" COEFR
=   (4. , 2. , - 3.) NDIM
=    (3,0,3) DIRECT
=  (n1x, n1y, n1z, rbid, rbid, rbid, n2x, n2y, n2z) DDL
=     ("DEPL", "DX", "ROTA") NOEUD
=   ("N1", "N3", "N2") BETAR
=   5. describe
```

the relation in the 7 terms: 4.

```
* (n1x*DX (N1) +n1y*DY (N1) +n1z*DZ (N1)) +
2.  *DX (N3) +
-3.* (n2x*DX (N2) +n2y*DY (N2) +n2z*DZ (N2)) =   5.
```

5.2 Routine AFLRCH SUBROUTINE

```
  AFLRCH (LISREL, CHARGE) C
------------------------------------------------------- C
AJOUT  OF A LISTE_RELA IN A CHARGE C
C
 THE IDENTICAL RELATIONS WITHIN LISTE_RELA ARE C
ELIMINEES  . LE PRINCIPLE OF SURCHARGE IS BRACKET: C
IT  IS LE LAST SECOND MEMBER WHO IS PRESERVE. C
------------------------------------------------------- C
LISREL  IN  /JXVAR - K19 -   : NOM OF SD LISTE_RELA C
                              THE LISTE_RELA IS DESTROYED A.C.
                              THE FIN OF THE ROUTINE C
------------------------------------------------------- C
CHARGE  IN  /JXVAR -  K8 -   : NOM OF SD CHARGE C
                              THE CHARGE IS ENRICHED C
------------------------------------------------------- Principle
```

6 of overload It

can arrive that L" user defines several times the same linear relation (except for a multiplying coefficient). Example:

3.

```
DX (N1) -1.DY (N2) = 4. 6.
DX (N1) -2.DY (N2) = 8. 3.
DX (N1) -1.DY (N2) = 5. Here
```

, the first 2 equations are identical. Third is contradictory with the preceding ones (because of second member). If

two equations of a linear system to solve have the same 1ier member , one cannot reverse the matrix, because the equations are not independent. It is thus necessary to eliminate all the equations which are multiple from/to each other. One

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

wants to be able to apply the principle of “overload” [U2.01.00 §3.7]: it is thus the last second member who is preserved. This

elimination of the “redundant” relations is made at the time or one adds LISTE _RELA to the CHARGE (routine AFLRCH). One eliminates duplicates from LISTE _RELA, one prints the eliminated relations, then one adds the relations preserved at the CHARGE . If

one keeps the diagram advised here with the § 4: only one LISTE _RELA by key word factor, the principle of overload is thus naturally applied for each key word. The last occurrences take precedence over the first. If

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)

one wanted (it today is not wanted!) an overload between various keywords (for example: DDL_IMPO takes precedence over FACE_IMPO), it would be enough that these 2 key keys are associated has same LISTE _RELA: CAL

```
FACIMPO (CH, LISREL) CAL
DDLIMPO (CH, LISREL) CAL
AFLRCH (LISREL  , CH) Example
```

7 :

To illustrate the use of the two routines presented higher, one will be able to consult the source of the routine caliai.f . This

routine treats key word LIAISON_DDL of the commands : AFFE_CHAR_MECA

- (_F) AFFE_CHAR_THER
- (_F)

Warning : The translation process used on this website is a "Machine Translation". It may be imprecise and inaccurate in whole or in part and is provided as a convenience.

Licensed under the terms of the GNU FDL (http://www.gnu.org/copyleft/fdl.html)