

Titre : Introduire un nouveau chargement de type "cinématique"
Auteur(s) : J. PELLET
Département Mécanique et Modèles Numériques
Diffusion : Développeurs

Clé : D5.03.02

Date : 05/05/97
Page : 1/12
Indice : A

Manuel de Descriptif Informatique
Fascicule D5.03 :
Document D5.03.02

Introduire un nouveau chargement de type "cinématique"

Résumé :

Ce document présente les deux routines utilitaires permettant d'introduire facilement de nouveaux types de conditions aux limites "cinématiques" (i.e des relations linéaires entre degrés de libertés inconnus).

Table des matières

Table des matières	2
1 Introduction	3
2 Qu'est-ce qu'une relation linéaire ?	3
3 Comment introduit-on des relations linéaires dans une modélisation ?	4
4 Introduire un nouveau mot clé de type "relation linéaire"	5
5 Les routines AFRELA et AFLRCH	6
5.1 La routine AFRELA	6
5.2 La routine AFLRCH	8
6 Principe de surcharge	8
7 Exemple : routine CALIAI	9

1 Introduction

Ce que l'on appelle "chargement" dans *Aster* (vocabulaire "mécanique") est ce que l'utilisateur définit dans les commandes `AFFE_CHAR_*`. On distingue en général les chargements en "force" [D5.03.01] et les chargements en "déplacements" (ou "cinématiques").

Ce document explique comment introduire de nouveaux chargements cinématiques.

2 Qu'est-ce qu'une relation linéaire ?

Cette expression désigne une **contrainte linéaire sur les degrés de liberté** du système à étudier :

- les ddl de la grandeur `TEMP_R` pour le phénomène thermique,
- les ddl des grandeurs `DEPL_R` ou `DEPL_C` pour le phénomène mécanique,
- les ddl de la grandeur `PRES_C` pour le phénomène acoustique.

Les coefficients de cette relation linéaire sont des constantes réelles (ou complexes), le second membre peut être réel, complexe ou de type "fonction" (K8).

Une relation linéaire peut s'écrire :

$$\alpha_1 \text{ddl1} + \alpha_2 \text{ddl2} + \dots + \alpha_n \text{ddl}n = \alpha_0$$

où

$$\alpha_i \in \mathbb{E} \text{ (ou } \mathbb{C}) \text{ (} i = 1, n \text{)}$$

$$\alpha_0 \in \mathbb{E} \text{ (ou } \mathbb{C}) \text{ (ou fonction)}$$

Les degrés de liberté ddl_i sont des degrés de liberté portés par un ou plusieurs nœuds différents.

Exemples de relations linéaires :

$\text{DX (N1)} = 0.$	blocage de la CMP "DX" du nœud "N1"
$\text{TEMP (N3)} = 100.$	température imposée à 100. pour le nœud "N3"
$\text{DY (N1)} - \text{DY (N2)} = 0.$	les nœuds "N1" et "N2" ont le même déplacement "DY"
$\cos \alpha. \text{DX (N1)} + \sin \alpha. \text{DY (N1)} = 0.$	le nœud "N1" est astreint à se déplacer sur la droite perpendiculaire au vecteur $(\cos \alpha, \sin \alpha)$ (en 2D).

3 Comment introduit-on des relations linéaires dans une modélisation ?

Les relations linéaires que l'on a définies dans le [§2] contraignent la solution que l'on cherche. Elles font partie de ce que l'on appelle en général les "conditions aux limites". Dans le *Code_Aster* elles sont une des composantes des **charges** (types `char_acou`, `char_ther`, `char_meca`).

Ces relations linéaires sont donc introduites par l'utilisateur via les commandes `AFFE_CHAR_MECA (_F)`, `AFFE_CHAR_THER (_F)`, `AFFE_CHAR_ACOU`, ou `AFFE_CHAR_CINE`.

Ces relations linéaires peuvent "traitées" de deux façons :

- on élimine une inconnue pour chaque relation linéaire : méthode d'élimination [D3.03.01],
- on "dualise" la relation en lui ajoutant 2 inconnues supplémentaires : les paramètres de Lagrange [R3.03.01].

Dans le *Code_Aster*, la méthode d'élimination est utilisée pour les relations issues de la commande `AFFE_CHAR_CINE`. On parlera dans ce cas de relations linéaires "cinématiques", bien que ce terme ne soit pas très judicieux. On se limite alors à des relations du type :

$$DDL = cste$$

Les autres relations issues des commandes `AFFE_CHAR_MECA`, `AFFE_CHAR_THER` et `AFFE_CHAR_ACOU` sont toujours dualisées.

Exemples de mots clés facteur engendrant des relations linéaires :

- | | |
|---------------------------------|------------------------------|
| • <code>AFFE_CHAR_CINE</code> | <code>MECA_IMPO</code> |
| • <code>AFFE_CHAR_MECA_F</code> | <code>LIAISON_OBLIQUE</code> |
| • <code>AFFE_CHAR_THER</code> | <code>TEMP_IMPO</code> |
| • <code>AFFE_CHAR_MECA</code> | <code>LIAISON_DDL</code> |

La commande `AFFE_CHAR_CINE` permet d'introduire facilement toutes les relations linéaires simples (`DDL = cste`) que l'on peut définir.

En revanche, bien qu'en théorie (grâce au mot clé `LIAISON_DDL`), on puisse introduire n'importe quelle relation linéaire, le nombre de coefficients à calculer peut devenir très grand. Penser par exemple aux relations linéaires qu'il faut écrire pour dire que 4 nœuds sont solidaires (reliés par un solide indéformable).

Les nombreux mots clés permettant à l'utilisateur de définir ces relations linéaires sont là pour lui faciliter le travail :

- | | |
|--------------------------------|--|
| • <code>LIAISON_OBLIQUE</code> | pour des appuis glissants dans un repère oblique |
| • <code>TEMP_IMPO</code> | pour imposer une température |
| • <code>LIAISON_GROUP</code> | pour relier des nœuds deux à deux |
| • ... | |
| • et <code>LIAISON_DDL</code> | pour les autres cas ... |

Ce grand nombre de mot clés (qui ne pourra que croître) nécessite de se donner des outils logiciels permettant :

- de ne pas dupliquer inutilement du code,
- de faciliter l'introduction de nouveaux mots clés dans les commandes `AFFE_CHAR_MECA`, `AFFE_CHAR_ACOU` et `AFFE_CHAR_THER`.

C'est de ces outils dont nous parlerons dans les paragraphes suivants.

4 Introduire un nouveau mot clé de type "relation linéaire"

Nous donnons dans ce paragraphe un canevas pour l'écriture d'une routine "réalisant" un mot clé de la commande AFPE_CHAR_MECA (ou _THER ou _ACOU). Ce mot clé facteur permettant à l'utilisateur de définir des relations linéaires.

Soient :

MFAC le mot clé facteur

CAMFAC le nom de la routine lui correspondant

Le but de la routine CAMFAC est de "scruter" les données de l'utilisateur derrière le mot clé MFAC, de traduire ces données en relations linéaires et de stocker ces relations dans la charge (ici de type char_meca) que l'utilisateur est en train de définir.

Pour cela, on dispose de deux routines utilitaires :

- AFRELA : pour affecter une relation linéaire à une SD de type LISTE_RELA (liste de relations linéaires)
- AFRLCH : pour "ajouter" une SD LISTE_RELA à une SD CHARGE

Ces routines imposent de passer par une SD intermédiaire (temporaire) de type LISTE_RELA. Cela alourdit un peu la programmation mais présente les avantages suivants :

- gains de performance, car la routine AFRLCH est coûteuse en CPU,
- une grande flexibilité pour réaliser le principe de surcharge (cf. [§?]).

Le canevas de la routine CAMFAC est donc le suivant :

```
SUBROUTINE CAMFAC (ch)
CHARACTER * (*) ch
C in jxvar ch: SD CHAR_MECA à enrichir
C but: enrichir la charge ch des relations linéaires définies
      sous le mot clé facteur MFAC
```

boucle sur les relations linéaires

- acquisition des coefficients de la relation linéaire : α_i
(routines GETVXX),
- ajout de la relation linéaire à la SD LISTE_RELA
Call AFRELA (α_i , '&&CAMFAC.LISTE_RELA')

fin boucle

- ajout de la SD LIST_RELA à la CHARGE : ch
Call AFLRCH ('&&CAMFAC.LIST_RELA', ch)

END

Remarques :

- la SD LISTE_RELA (temporaire) est propre à la routine CAMFAC, son nom respecte la convention des noms d'objets de travail : `&&nom_routine`,
- le principe de surcharge (cf. [U2.01.00 §3.7]) ne concerne donc que les occurrences du mot clé MFAC,
- cette SD est détruite lors de l'appel à AFLRCH.

5 Les routines AFRELA et AFLRCH

5.1 La routine AFRELA

```
      SUBROUTINE AFRELA (COEFR, COEFC, DDL, NOEUD , NDIM, DIRECT,
+      NBTERM, BETAR, BETAC, BETAF, TYPCOE, TYPVAL, LISREL)
C-----
C BUT : AFFECTATION D'UNE  RELATION ENTRE DDLS A UNE SD LISTE_RELA
C      (SI L'OBJET LISREL N'EXISTE PAS, IL EST CREE)
C-----
C COEFR(NBTERM) - IN - R      - : TABLEAU DES COEFFICIENTS DE LA RELATION
C                               LES COEFFICIENTS SONT REELS
C-----
C COEFC(NBTERM) - IN - C      - : TABLEAU DES COEFFICIENTS DE LA RELATION
C                               LES COEFFICIENTS SONT COMPLEXES
C-----
C DDL(NBTERM)   - IN - K8     - : TABLEAU DES DDL DE LA RELATION
C-----
C NOEUD(NBTERM) - IN - K8     - : TABLEAU DES NOEUDS DE LA RELATION
C-----
C NDIM(NBTERM)  - IN - I      - : DIMENSION DU PROBLEME (0, 2 OU 3)
C                               SI = 0 PAS DE CHANGEMENT DE REPERE
C                               LA RELATION EST DONNEE DANS LA BASE
C                               GLOBALE
C-----
C DIRECT(3,NBTERM)- IN - R    - : TABLEAU DES VECTEURS RELATIFS A CHAQUE
C                               TERME DEFINISSANT LA DIRECTION DE LA
C                               COMPOSANTE QUE L'ON VEUT CONTRAINDRE
C-----
C NBTERM        - IN - I      - : NOMBRE DE TERMES DE LA RELATION
C-----
C BETAR         - IN - R      - : VALEUR REELLE DU SECOND MEMBRE
C-----
C BETAC         - IN - C      - : VALEUR COMPLEXE DU SECOND MEMBRE
C-----
C BETAF         - IN - K8     - : VALEUR FONCTION DU SECOND MEMBRE
C-----
C TYPCOE        - IN - K4     - : TYPE DES COEFFICIENTS DE LA RELATION :
C                               = 'REEL' OU 'COMP'
C-----
C TYPVAL        - IN - K4     - : TYPE DU SECOND MEMBRE
C                               = 'REEL' OU 'COMP' OU 'FONC'
C-----
C LISREL        - IN - K19    - : NOM DE LA SD LISTE_RELA
C                               - JXVAR      -
C-----
```

Deux cas de figure sont à considérer :

- a) les ddl à relier sont donnés dans le repère absolu : DX, DY, ...
- b) certains ddl à relier sont donnés dans un repère local.

Cas A (tout dans le repère absolu) :

NBTERM est le nombre de ddl reliés par la relation.
NDIM est un vecteur rempli de 0
DIRECT est inutile.

Titre : Introduire un nouveau chargement de type "cinématique"
Auteur(s) : J. PELLET

Date : 18/09/2003
Clé : D5.03.02 Page : 7/12

Exemple 1 :

on veut imposer : $3 \cdot DX(N1) + 2 \cdot DY(N2) - 4 \cdot DRZ(N1) = 'F'$ (fonction)

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

Cas B (repère local) :

Pour chaque noeud impliqué dans la relation, on peut donner un repère local dans lequel la relation est plus simple (la normale à une surface par exemple).

Exemple 2 :

soit n , un vecteur unitaire de composantes (n_x , n_y , n_z).

On veut que le déplacement suivant n au nœud $N3$ soit nul.

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

Remarques :

- *NBTERM n'est pas le nombre de termes de la relation finale (ici : 3).*
- *Quand on emploie (pour un "terme") la possibilité d'un repère local NDIM $\neq 0$ le nom du DDL doit être conventionnellement 'DEPL' ou 'ROTA'*

Exemple 3 :

RC

soient : $n1$: un vecteur unitaire de composantes ($n1_x$, $n1_y$, $n1_z$) et
 $n2$: un vecteur unitaire de composantes ($n2_x$, $n2_y$, $n2_z$)

les données suivantes :

```
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.
```

decrivent la relation à 7 termes :

```
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 La routine AFLRCH

```
      SUBROUTINE AFLRCH (LISREL, CHARGE)
C -----
C  AJOUT D'UNE LISTE_RELA DANS UNE CHARGE
C
C  LES RELATIONS IDENTIQUES AU SEIN DE LISTE_RELA SONT
C  ELIMINEES. LE PRINCIPE DE SURCHARGE EST APPLIQUE :
C  C'EST LE DERNIER SECOND MEMBRE QUI EST CONSERVE.
C -----
C  LISREL  IN/JXVAR - K19    - : NOM DE LA SD LISTE_RELA
C                               LA LISTE_RELA EST DETRUITE
C                               A LA FIN DE LA ROUTINE
C -----
C  CHARGE  IN/JXVAR  - K8    - : NOM DE LA SD CHARGE
C                               LA CHARGE EST ENRICHEE
C -----
```

6 Principe de surcharge

Il peut arriver que l'utilisateur définisse plusieurs fois une même relation linéaire (à un coefficient multiplicateur près).

Exemple :

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

Ici, les 2 premières équations sont identiques. La troisième est contradictoire avec les précédentes (à cause du second membre).

Si deux équations d'un système linéaire à résoudre ont le même 1er membre, on ne peut inverser la matrice, car les équations ne sont pas indépendantes. Il faut donc éliminer toutes les équations qui sont multiples les unes des autres.

On veut pouvoir appliquer le principe de "surcharge" [U2.01.00 §3.7] : c'est donc le dernier second membre qui est conservé.

Cette élimination des relations "redondantes" est faite au moment où on ajoute la LISTE_RELA à la CHARGE (routine AFLRCH). On élimine les doublons de la LISTE_RELA, on imprime les relations éliminées, puis on ajoute les relations conservées à la CHARGE.

Si on garde le schéma conseillé ici [§4] : une seule LISTE_RELA par mot clé facteur, le principe de surcharge est donc naturellement appliqué pour chaque mot clé. Les dernières occurrences priment sur les premières.

Si l'on voulait (on ne le veut pas aujourd'hui !) une surcharge entre différents mots clés (par exemple : DDL_IMPO prime sur FACE_IMPO), il suffirait que ces 2 mots clés soient associés à la même LISTE_RELA :

```
CALL FACIMPO (CH,LISREL)
CALL DDLIMPO (CH,LISREL)
CALL AFLRCH  (LISREL,CH)
```


7 Exemple : routine CALIAI

Cette routine traite le mot clé LIAISON_DDL des commandes :

- AFPE_CHAR_MECA(_F)
- AFPE_CHAR_THER(_F)

```

SUBROUTINE CALIAI(FONREE, CHARGE)
IMPLICIT REAL*8 (A-H,O-Z)
CHARACTER*4      FONREE
CHARACTER*8      CHARGE
C -----
C MODIF MODELISA  DATE 23/01/95  AUTEUR VABHHTS J.PELLET
C
C   TRAITER LE MOT CLE LIAISON_DDL DE AFPE_CHAR_XXX
C   ET ENRICHIR LA CHARGE (CHARGE) AVEC LES RELATIONS LINEAIRES
C
C IN      : FONREE : 'REEL' OU 'FONC'
C IN/JXVAR : CHARGE : NOM D'UNE SD CHARGE
C -----
C   ----- COMMUNS NORMALISES  JEVEUX -----
C   INTEGER          ZI
C   COMMON / IVARJE / ZI(1)
C   REAL*8           ZR
C   COMMON / RVARJE / ZR(1)
C   COMPLEX*16        ZC
C   COMMON / CVARJE / ZC(1)
C   LOGICAL           ZL
C   COMMON / LVARJE / ZL(1)
C   CHARACTER*8       ZK8
C   CHARACTER*16      ZK16
C   CHARACTER*24      ZK24
C   CHARACTER*32      ZK32
C   CHARACTER*80      ZK80
C   COMMON / KVARJE / ZK8(1), ZK16(1), ZK24(1), ZK32(1), ZK80(1)
C   CHARACTER*32      JEXNOM, JEXNUM
C -----
C ----- FIN COMMUNS NORMALISES  JEVEUX -----
C
C   COMPLEX*16        BETAC
C   CHARACTER*7       TYPCHA
C   CHARACTER*8       BETAF
C   CHARACTER*8       K8BID, MOTCLE, MOGROU, MOD, NOMA, NOMNOE
C   CHARACTER*16      MOTFAC
C   CHARACTER*19      LISREL
C   CHARACTER*24      TRAV, GROUMA, NOEUMA
C   CHARACTER*19      LIGRMO
C -----
C
C   MOTFAC = 'LIAISON_DDL'
C   MOTCLE = 'NOEUD'
C   MOGROU = 'GROUP_NO'
C
C   LISREL = '&&CALIAI.RLLISTE'
C   CALL GETFAC(MOTFAC,NLIAI)
C   IF (NLIAI.EQ.0) RETURN
C
C   BETAC = (1.0D0,0.0D0)
C
C   CALL DISMOI('F','TYPE_CHARGE',CHARGE,'CHARGE',IBID,
+             TYPCHA,IER)
C   CALL DISMOI('F','NOM_MODELE',CHARGE,'CHARGE',IBID,MOD,IER)
C   CALL DISMOI('F','NOM_MAILLA',CHARGE,'CHARGE',IBID,NOMA,IER)
C
C   NOEUMA = NOMA//'.NOMNOE'
C   GROUMA = NOMA//'.GROUPENO'

```

Titre : Introduire un nouveau chargement de type "cinématique"
Auteur(s) : J. PELLET

Date : 18/09/2003
Clé : D5.03.02 Page : 10/12

```
C      -- CALCUL DE NDIM1 : NBRE DE TERMES MAXI D'UNE LISTE
C      DE GROUP_NO OU DE NOEUD
C      -----
NDIM1   = 0
DO 10 I=1,NLIAI
    CALL GETVID (MOTFAC,MOGROU,I,1,0,K8BID,NENT)
    NDIM1 = MAX(NDIM1,-NENT)
    CALL GETVID (MOTFAC,MOTCLE,I,1,0,K8BID,NENT)
    NDIM1 = MAX(NDIM1,-NENT)
10 CONTINUE

TRAV = '&&CALIAI.'//MOTFAC
CALL WKVECT(TRAV,'V V K8',NDIM1,JJJ)

C      -- CALCUL DE NDIM2 ET VERIFICATION DES NOEUDS ET GROUP_NO
C      NDIM2 EST LE NOMBRE MAXI DE NOEUDS IMPLIQUES DANS UNE
C      RELATION LINEAIRE
C      -----
NDIM2   = NDIM1
DO 20 IOCC = 1, NLIAI
    CALL GETVID (MOTFAC,MOGROU,IOCC,1,NDIM1,ZK8(JJJ),NGR)
    NBGT = 0
    DO 30 IGR = 1, NGR
        CALL JEEIXIN (JEXNOM(GROUMA,ZK8(JJJ+IGR-1)),IRET)
        IF (IRET.EQ. 0) THEN
            CALL UTMESS('F',MOTFAC,'LE GROUPE '//ZK8(JJJ+IGR-1)//
+                'NE FAIT PAS PARTIE DU MAILLAGE : '//NOMA )
        ELSE
            CALL JELIRA (JEXNOM(GROUMA,ZK8(JJJ+IGR-1)), 'LONMAX',
+                N1,' ')
            NBGT = NBGT + N1
        ENDIF
30 CONTINUE
    NDIM2 = MAX (NDIM2,NBGT)

    CALL GETVID (MOTFAC,MOTCLE,IOCC,1,NDIM1,ZK8(JJJ),NNO)
    DO 40 INO = 1, NNO
        CALL JENONU (JEXNOM(NOEUMA,ZK8(JJJ+INO-1)),IRET)
        IF (IRET.EQ. 0) THEN
            CALL UTMESS('F',MOTFAC,MOTCLE//' '//ZK8(JJJ+INO-1)//
+                'NE FAIT PAS PARTIE DU MAILLAGE : '//NOMA )
        ENDIF
40 CONTINUE
20 CONTINUE

C      -- ALLOCATION DE TABLEAUX DE TRAVAIL
C      -----
CALL WKVECT ('&&CALIAI.LISTE1','V V K8',NDIM1,JLIST1)
CALL WKVECT ('&&CALIAI.LISTE2','V V K8',NDIM2,JLIST2)
CALL WKVECT ('&&CALIAI.DDL ','V V K8',NDIM2 ,JDDL )
CALL WKVECT ('&&CALIAI.COEMUR','V V R' ,NDIM2 ,JCMUR )
CALL WKVECT ('&&CALIAI.COEMUC','V V C' ,NDIM2 ,JCMUC )
CALL WKVECT ('&&CALIAI.DIRECT','V V R' ,3*NDIM2 ,JDIREC )
CALL WKVECT ('&&CALIAI.DIMENSION','V V I' ,NDIM2 ,JDIME )

C
C
C      BOUCLE SUR LES RELATIONS LINEAIRES
C      -----
DO 50 I = 1, NLIAI
    CALL GETVR8 (MOTFAC,'COEF_MULT',I,1,NDIM1,ZR(JCMUR),N2)
    CALL GETVTX (MOTFAC,'DDL',I,1,NDIM1,ZK8(JDDL),N1)
C
```

Titre : Introduire un nouveau chargement de type "cinématique"
Auteur(s) : J. PELLET

Date : 18/09/2003
Clé : D5.03.02 Page : 11/12

```
C      EXCEPTION :SI LE MOT-CLE DDL N'EXISTE PAS DANS AFPE_CHAR_THER,
C      ON CONSIDERE QUE LES RELATIONS LINEAIRES PORTENT
C      SUR LE DDL 'TEMP'
      IF (N1.EQ.0.AND.TYPCHA(1:4).EQ.'THER') THEN
          N1 = N2
          DO 60 K=1,N1
              ZK8(JDDL-1+K) = 'TEMP'
60      CONTINUE
      ENDIF

      IF (N1.NE.N2) THEN
          CALL UTDEBM('F','CALIAI','LE NOMBRE DE DDLS FIGURANT DANS'
&      //' LA LIAISON N'EST PAS EGAL AU NOMBRE DE COEF_MULT :')
          CALL UTIMPI('S',' ',1,N1)
          CALL UTIMPI('S',' ',1,N2)
          CALL UTFINM()
      ENDIF

C      -- RECUPERATION DU 2ND MEMBRE :
C      -----
      IF (FONREE.EQ.'REEL') THEN
          CALL GETVR8 (MOTFAC, 'COEF_IMPO', I, 1, 1,BETA, NB)
      ELSE IF (FONREE.EQ.'FONC') THEN
          CALL GETVID (MOTFAC, 'COEF_IMPO', I, 1, 1,BETAF, NB)
      ELSE
          CALL UTMESS('F','CALIAI','CAS NON PREVU')
      ENDIF

C
C      CALL GETVID (MOTFAC,'GROUP_NO',I,1,0,ZK8(JLIST1),NG)
      IF (NG .NE.0) THEN

C
C
C      -- CAS DE GROUP_NO :
C      -----
      NG = -NG
      CALL GETVID (MOTFAC,'GROUP_NO',I,1,NG,ZK8(JLIST1),N)
      INDNOE = 0
      DO 80 J = 1, NG
          CALL JEVEUO (JEXNOM(GROUMA,ZK8(JLIST1-1+J)), 'L',JGR0)
          CALL JELIRA (JEXNOM(GROUMA,ZK8(JLIST1-1+J)), 'LONMAX',
+              N,' ')
          DO 90 K = 1, N
              IN = ZI(JGR0-1+K)
              INDNOE = INDNOE + 1
              CALL JENUNO(JEXNUM(NOMA//'.NOMNOE',IN),NOMNOE)
              ZK8(JLIST2+INDNOE-1) = NOMNOE
90      CONTINUE
80      CONTINUE

C
C      AFFECTATION DE LA RELATION A LA LISTE_RELA :
C      -----
      CALL AFRELA(ZR(JCMUR),ZC(JCMUC),ZK8(JDDL),ZK8(JLIST2),
+      ZI(JDIME),ZR(JDIREC),INDNOE,BETA,BETAC,BETAF,
+      FONREE,FONREE,LISREL)

C
C      ELSE

C
C
C      CAS DE NOEUD :
C      -----
      CALL GETVID (MOTFAC, 'NOEUD', I, 1, 0, ZK8(JLIST1),NBNO)
      IF (NBNO .NE. 0) THEN
          NBNO=-NBNO
          CALL GETVID (MOTFAC,'NOEUD',I,1,NBNO,ZK8(JLIST1),N)
      ENDIF

C
```

Titre : Introduire un nouveau chargement de type "cinématique"

Date : 18/09/2003

Auteur(s) : J. PELLET

Clé : D5.03.02

Page : 12/12

```
C      AFFECTATION DE LA RELATION A LA LISTE_RELA :
C      -----
C      CALL AFRELA(ZR(JCMUR),ZC(JCMUC),ZK8(JDDL),ZK8(JLIST1),
+          ZI(JDIME),ZR(JDIREC),NBNO,BETA,BETAC,BETAF,
+          FONREE,FONREE,LISREL)
C      ENDIF
C
C 50  CONTINUE
C
C      -- AJOUT DE LA LISTE_RELA A LA CHARGE :
C      -----
C      CALL AFLRCH(LISREL,CHARGE)
C
C      -- MENAGE :
C      -----
C      CALL JEDETR (TRAV)
C      CALL JEDETR ('&&CALIAI.LISTE1')
C      CALL JEDETR ('&&CALIAI.LISTE2')
C      CALL JEDETR ('&&CALIAI.DDL ')
C      CALL JEDETR ('&&CALIAI.COEMUR')
C      CALL JEDETR ('&&CALIAI.COEMUC')
C      CALL JEDETR ('&&CALIAI.DIRECT')
C      CALL JEDETR ('&&CALIAI.DIMENSION')
C
C      END
```