

---

Titre : Mise en œuvre de STAT\_NON\_LINE

Auteur(s) : I. VAUTIER

Département Mécanique et Modèles Numériques

Diffusion : Développeurs

Clé : D9.05.01

Date : 28/01/99

Page : 1/22

Indice : A

## Manuel de Descriptif Informatique Fascicule D9.05 Document D9.05.01

# Mise en œuvre de STAT\_NON\_LINE

---

### Résumé :

On décrit ici la mise en œuvre informatique de l'algorithme de résolution des problèmes quasi-statiques non linéaires. Le document [R5.03.01] décrivant en détail cet algorithme est supposé connu, on n'en rappellera que les principales étapes. On trouvera dans ce document un rappel des notations, l'organigramme simplifié de la routine `op0070`, permettant de discerner les principales articulations logiques de l'opérateur `STAT_NON_LINE` du *Code\_Aster*, son arbre d'appel, une description des objets informatiques et des principales routines, et quelques pièges à éviter lors du développement dans cet opérateur.

---

## Table des matières

---

Erreur! Aucune entrée de table des matières n'a été trouvée.

## 1 Notations et algorithme

On donne ici un résumé de l'algorithme exposé en détail dans le document [R5.03.01] pour pouvoir faire le lien avec les objets et les opérations informatiques.

### 1.1 Notations

On note  $\mathbf{u}$  le champ de déplacement et  $\boldsymbol{\lambda}$  les multiplicateurs de Lagrange associés aux conditions aux limites. Ce sont les inconnues du problème. Ces champs sont rangés dans un même vecteur informatique, mélangés au gré de la numérotation des inconnues. Par la suite, on notera ce vecteur de manière symbolique, sans présumer de l'ordre d'apparition des inconnues :

$$\begin{pmatrix} \mathbf{u} \\ \boldsymbol{\lambda} \end{pmatrix}$$

De même, les sollicitations appliquées à la structure (chargements mécaniques et conditions aux limites) sont rangées dans un vecteur chargement, dans un ordre qui nous importe peu. On notera symboliquement ce vecteur :

$$\begin{pmatrix} \mathbf{L}^{méca} \\ \mathbf{u}^d \end{pmatrix}$$

La longueur de ces vecteurs est rangée dans la plupart des routines dans la variable LONCH.

On doit résoudre pour une suite d'instants de calcul  $t_i$  (souvent fictifs, permettant de paramétrer le chargement), le système suivant :

$$\begin{cases} \mathbf{Q}^T \boldsymbol{\sigma}_i + \mathbf{B}^T \boldsymbol{\lambda}_i = \mathbf{L}_i^{méca} \\ \mathbf{B} \mathbf{u}_i = \mathbf{u}_i^d \end{cases}$$

avec les notations du document [R5.03.01].

**NB :**

*Dans le cas du pilotage, on a une inconnue supplémentaire  $\eta$  et une équation supplémentaire (cf. [§1.4]).*

L'algorithme est le suivant : pour chaque pas de temps, on effectue une prédiction d'Euler, puis on corrige à l'aide d'itérations de la méthode de Newton avec recherche linéaire.

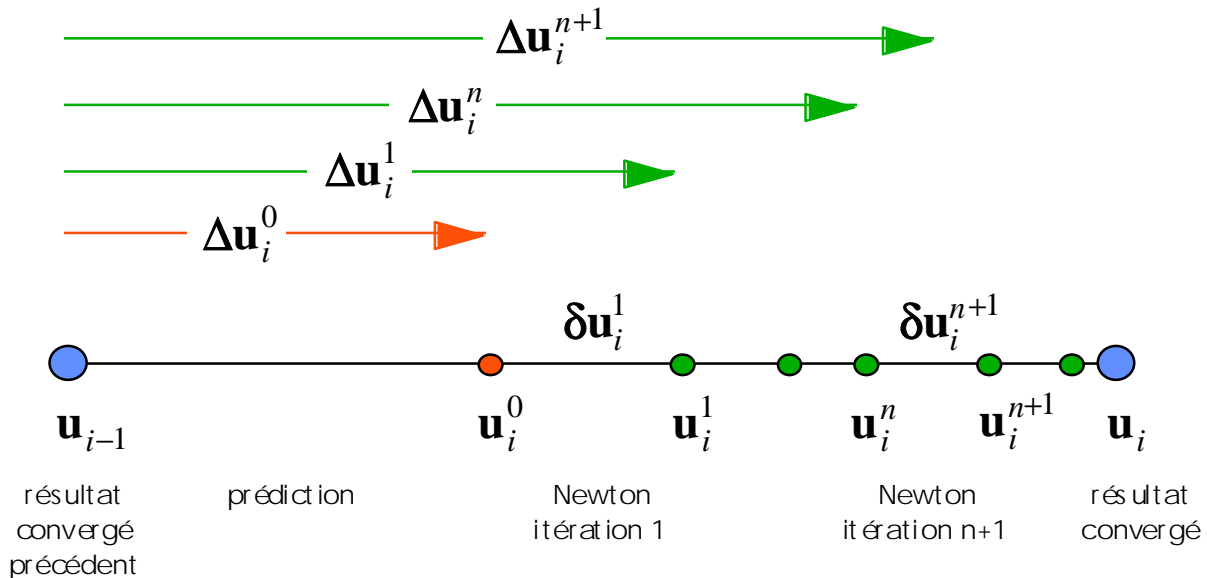
Le schéma ci-après donne les notations utilisées. On dénote par l'indice  $i$  (comme intant) le numéro du pas de temps, et par l'exposant  $n$  (comme newton) le numéro de l'itération de Newton. Les incréments sont notés  $\Delta$  lorsqu'ils sont mesurés par rapport à l'état obtenu au dernier pas de temps (équilibre), et  $\delta$  lorsqu'ils sont mesurés par rapport à l'état obtenu à la dernière itération de Newton.

On a ainsi :

$$\left\{ \begin{array}{l} \Delta \mathbf{u}_i = \mathbf{u}_i - \mathbf{u}_{i-1} \\ \Delta \lambda_i = \lambda_i - \lambda_{i-1} \\ \Delta \mathbf{L}_i^{méca} = \mathbf{L}_i^{méca} - \mathbf{L}_{i-1}^{méca} \\ \Delta \mathbf{u}_i^d = \mathbf{u}_i^d - \mathbf{u}_{i-1}^d \end{array} \right.$$

$$\Delta \mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \delta \mathbf{u}_i^{n+1} - \mathbf{u}_{i-1}$$

$$\Delta \mathbf{u}_i^{n+1} = \Delta \mathbf{u}_i^0 + \sum_{k=1}^{n+1} \delta \mathbf{u}_i^k$$



**NB :**

Lorsque l'on fait de la recherche linéaire, on écrit  $\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1}$  au lieu de  $\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \delta \mathbf{u}_i^{n+1}$ , le symbole  $\delta$  n'a donc plus tout à fait la même signification (c'est dans ce cas la direction de recherche).

## 1.2 Algorithme

On se borne à donner ici les systèmes d'équations composant l'algorithme (sans charges suiveuses et sans pilotage) : la logique de l'algorithme est détaillée dans le [§2]. La prise en compte des charges suiveuses est dans le [§1.3]. Les équations complètes (incluant le pilotage) sont dans le [§1.4].

**1.2.1 Phase de prédiction**

Elle consiste à résoudre un système linéaire représentant le problème tangent en vitesse (cf. [R5.03.01] et [R5.03.02]). Cela permet de mieux converger dans la méthode de Newton ultérieure lorsque l'on impose des déformations anélastiques données (exemple : thermique, métallurgie).

**Pas de temps courant (autre que le premier)**

$$\begin{cases} \mathbf{K}_{i-1} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_i^0 = \Delta \mathbf{L}_i^{méca} + \Delta \mathbf{L}_i^{ther} \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \Delta \mathbf{u}_i^d \end{cases} \quad \text{éq 1.2-1}$$

où  $\mathbf{K}_{i-1}$  est la matrice tangente calculée dans `merimo` via l'option de calcul `RIGI_MECA_TANG`.

**Premier pas de temps**

$$\begin{cases} \mathbf{K}_0 \Delta \mathbf{u}_1^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_1^0 = \mathbf{L}_1^{méca} - \mathbf{Q}^T \boldsymbol{\sigma}_0 + \Delta \mathbf{L}_1^{ther} \\ \mathbf{B} \Delta \mathbf{u}_1^0 = \mathbf{u}_1^d - \mathbf{B} \mathbf{u}_0 \end{cases} \quad \text{éq 1.2-2}$$

**Reprise avec conditions aux limites de type Dirichlet différentiel ('DIDI')**

$$\begin{cases} \mathbf{K}_0 \Delta \mathbf{u}_1^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_1^0 = \mathbf{L}_1^{méca} - \mathbf{Q}^T \boldsymbol{\sigma}_0 + \Delta \mathbf{L}_1^{ther} \\ \mathbf{B} \Delta \mathbf{u}_1^0 = \mathbf{u}_1^d \end{cases}$$

**1.2.2 Itération de Newton**

$$\begin{cases} \mathbf{K}_i^n \delta \mathbf{u}_i^{n+1} + \mathbf{B}^T \delta \boldsymbol{\lambda}_i^{n+1} = \mathbf{L}_i^{méca} - \mathbf{Q}^T \boldsymbol{\sigma}_i^n - \mathbf{B}^T \boldsymbol{\lambda}_i^n \\ \mathbf{B} \delta \mathbf{u}_i^{n+1} = 0 \end{cases} \quad \text{éq 1.2-3}$$

où  $\mathbf{K}_i^n$  est la matrice tangente calculée dans `merimo` via l'option de calcul `FULL_MECA`.

**1.2.3 Recherche linéaire**

$$g(\rho) = \left[ \delta \mathbf{u}_i^{n+1} \right]^T \left[ \mathbf{Q}^T \boldsymbol{\sigma}(\mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1}) + \mathbf{B}^T (\boldsymbol{\lambda}_i^n + \rho \delta \boldsymbol{\lambda}_i^{n+1}) - \mathbf{L}_i^{méca} \right] = 0$$

$$\rho^{p+1} = \rho^p - \frac{\rho^p - \rho^{p-1}}{g^p - g^{p-1}} g^p = \frac{\rho^{p-1} g^p - \rho^p g^{p-1}}{g^p - g^{p-1}}$$

On fait des itérations sur  $p$ , partant de  $\rho^0 = 0$  et  $\rho^1 = 1$ .

**1.2.4 Actualisation des inconnues**

$$\begin{cases} \Delta \mathbf{u}_i^{n+1} = \Delta \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \Delta \lambda_i^{n+1} = \Delta \lambda_i^n + \rho \delta \lambda_i^{n+1} \end{cases} \quad \text{idem pour } \mathbf{u} \text{ et } \lambda$$

**1.2.5 Tests de convergence**

$$\text{RESI\_GLOB\_RELA} : \frac{\left\| \mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \lambda_i^n - \mathbf{L}_i^{\text{méca}} \right\|_{\infty}}{\left\| \mathbf{L}_i^{\text{méca}} + \mathbf{L}_i^{\text{ther}} - \mathbf{B}^T \lambda_i^n \right\|_{\infty}} \leq \xi_1$$

$$\text{RESI\_GLOB\_MAXI} : \left\| \mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \lambda_i^n - \mathbf{L}_i^{\text{méca}} \right\|_{\infty} \leq \xi_2$$

$$\text{INCO\_GLOB\_RELA} : \frac{\left\| \delta \mathbf{u}_i^{n+1} \right\|_{\infty}}{\left\| \mathbf{u}_i^{n+1} \right\|_{\infty}} \leq \xi_3$$

ITER\_GLOB\_MAXI : si  $n > n_{\max}$  on arrête

**1.3 Chargements suiveurs**

Ce sont des chargements dont la valeur dépend de celle des inconnues (par exemple la pression en grands déplacements, cf. [R3.03.04]). Il faut donc les réévaluer, non seulement à chaque pas de temps, mais aussi à chaque itération de Newton (c'est-à-dire à chaque changement des inconnues). De plus, ils conduisent à un terme de rigidité supplémentaire.

On décompose le chargement mécanique en une partie fixe (non suiveuse) et une partie suiveuse :

$$\mathbf{L}_i^{\text{méca}} = \mathbf{L}_i^{\text{fixe}} + \mathbf{L}^{\text{suiv}}(\mathbf{u}_i, t_i)$$

Le système à résoudre est alors :

$$\begin{cases} \mathbf{Q}^T \boldsymbol{\sigma}_i + \mathbf{B}^T \lambda_i = \mathbf{L}_i^{\text{fixe}} + \mathbf{L}^{\text{suiv}}(\mathbf{u}_i, t_i) \\ \mathbf{B} \mathbf{u}_i = \mathbf{u}_i^d \end{cases}$$

**1.3.1 Phase de prédiction**

$$\begin{cases} \left( \mathbf{K}_{i-1} - \frac{\partial \mathbf{L}^{\text{suiv}}}{\partial \mathbf{u}} \bigg|_{\mathbf{u}_{i-1}} \right) \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \lambda_i^0 = \Delta \mathbf{L}_i^{\text{fixe}} + \Delta \mathbf{L}_i^{\text{suiv}} + \Delta \mathbf{L}_i^{\text{ther}} \\ \mathbf{B} \Delta \mathbf{u}_i^0 = \Delta \mathbf{u}_i^d \end{cases}$$

**1.3.2 Itération de Newton**

$$\begin{cases} \left( \mathbf{K}_i^n - \frac{\partial \mathbf{L}^{suiv}}{\partial \mathbf{u}} \bigg|_{\mathbf{u}_i^n} \right) \delta \mathbf{u}_i^{n+1} + \mathbf{B}^T \delta \boldsymbol{\lambda}_i^{n+1} = \mathbf{L}_i^{fixe} + \mathbf{L}^{suiv}(\mathbf{u}_i^n) - \mathbf{Q}^T \boldsymbol{\sigma}_i^n - \mathbf{B}^T \boldsymbol{\lambda}_i^n \\ \mathbf{B} \delta \mathbf{u}_i^{n+1} = 0 \end{cases}$$

**1.4 Chargements pilotés**

L'amplitude d'une partie du chargement à chaque pas de temps est gouverné par une équation supplémentaire imposant la valeur d'un degré de liberté. Le chargement (au sens large, incluant les conditions aux limites), se décompose alors en une partie dite "constante" (même si elle dépend du temps par exemple) d'amplitude prédéterminée (donnée par l'utilisateur) et d'une partie dite "pilotée" dont l'amplitude  $\eta_i$  est une inconnue supplémentaire du problème :

$$\begin{pmatrix} \mathbf{L}_i^{cste} + \eta_i \mathbf{L}^{pilo} \\ \mathbf{u}_i^{cste} + \eta_i \mathbf{u}^{pilo} \end{pmatrix}.$$

Le chargement dit "constant" inclut les éventuelles forces suiveuses :

$$\begin{cases} \mathbf{L}_i^{cste} = \mathbf{L}_i^{fixe} + \mathbf{L}_i^{suiv} \\ \mathbf{u}_i^{cste} = \mathbf{u}_i^{fixe} \end{cases}$$

De même la matrice de rigidité tangente, notée  $\mathbf{K}$  inclura l'éventuelle matrice de rigidité tangente due à des forces suiveuses.

Le système à résoudre s'écrit :

$$\begin{cases} \mathbf{Q}^T \boldsymbol{\sigma}_i + \mathbf{B}^T \boldsymbol{\lambda}_i = \mathbf{L}_i^{cste} + \eta_i \mathbf{L}^{pilo} \\ \mathbf{B} \mathbf{u}_i = \mathbf{u}_i^{cste} + \eta_i \mathbf{u}^{pilo} \\ \mathbf{D}(\mathbf{u}_i) = \beta_i \end{cases}$$

La troisième relation est scalaire et  $\beta_i$  vaut l'instant de calcul  $t_i$  dans l'état actuel du code. Pour plus de détails, on se reportera au document [R5.03.01].

**1.4.1 Phase de prédiction**

$$\begin{cases} \mathbf{K}_{i-1} \Delta \mathbf{u}_i^0 + \mathbf{B}^T \Delta \boldsymbol{\lambda}_i^0 - \Delta \eta_i^0 \mathbf{L}^{pilo} = \Delta \mathbf{L}_i^{cste} + \Delta \mathbf{L}_i^{ther} \\ \mathbf{B} \Delta \mathbf{u}_i^0 - \Delta \eta_i^0 \mathbf{u}^{pilo} = \Delta \mathbf{u}_i^{cste} \\ \mathbf{P}_{i-1} \Delta \mathbf{u}_i^0 = \beta_i - \mathbf{D}(\mathbf{u}_{i-1}) \end{cases}$$

$\mathbf{D}$  est l'opérateur de pilotage (celui qui impose la relation supplémentaire) et  $\mathbf{P}$  son opérateur tangent. Comme pour l'instant on impose la valeur d'un degré de liberté, l'opérateur  $\mathbf{D}$  est linéaire et  $\mathbf{P} = \mathbf{D}$ .

$$\left\{ \begin{array}{l} \Delta \eta_i^0 = \frac{\beta_i - \mathbf{D}(\mathbf{u}_{i-1}) - \mathbf{C}_{i-1} (\mathbf{A}_{i-1})^{-1} \begin{pmatrix} \Delta \mathbf{L}_i^{cste} + \Delta \mathbf{L}_i^{ther} \\ \Delta \mathbf{u}_i^{cste} \end{pmatrix}}{\mathbf{C}_{i-1} (\mathbf{A}_{i-1})^{-1} \begin{pmatrix} \mathbf{L}^{pilo} \\ \mathbf{u}^{pilo} \end{pmatrix}} \\ \begin{pmatrix} \Delta \mathbf{u}_i^0 \\ \Delta \lambda_i^0 \end{pmatrix} = (\mathbf{A}_{i-1})^{-1} \begin{pmatrix} \Delta \mathbf{L}_i^{cste} + \Delta \mathbf{L}_i^{ther} \\ \Delta \mathbf{u}_i^{cste} \end{pmatrix} + \Delta \eta_i^0 (\mathbf{A}_{i-1})^{-1} \begin{pmatrix} \mathbf{L}^{pilo} \\ \mathbf{u}^{pilo} \end{pmatrix} \end{array} \right.$$

éq 1.4-1

avec  $\mathbf{A}_{i-1} = \begin{pmatrix} \mathbf{K}_{i-1} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix}$  et  $\mathbf{C}_{i-1} = (\mathbf{P}_{i-1} \quad 0)$ .

## 1.4.2 Itération de Newton

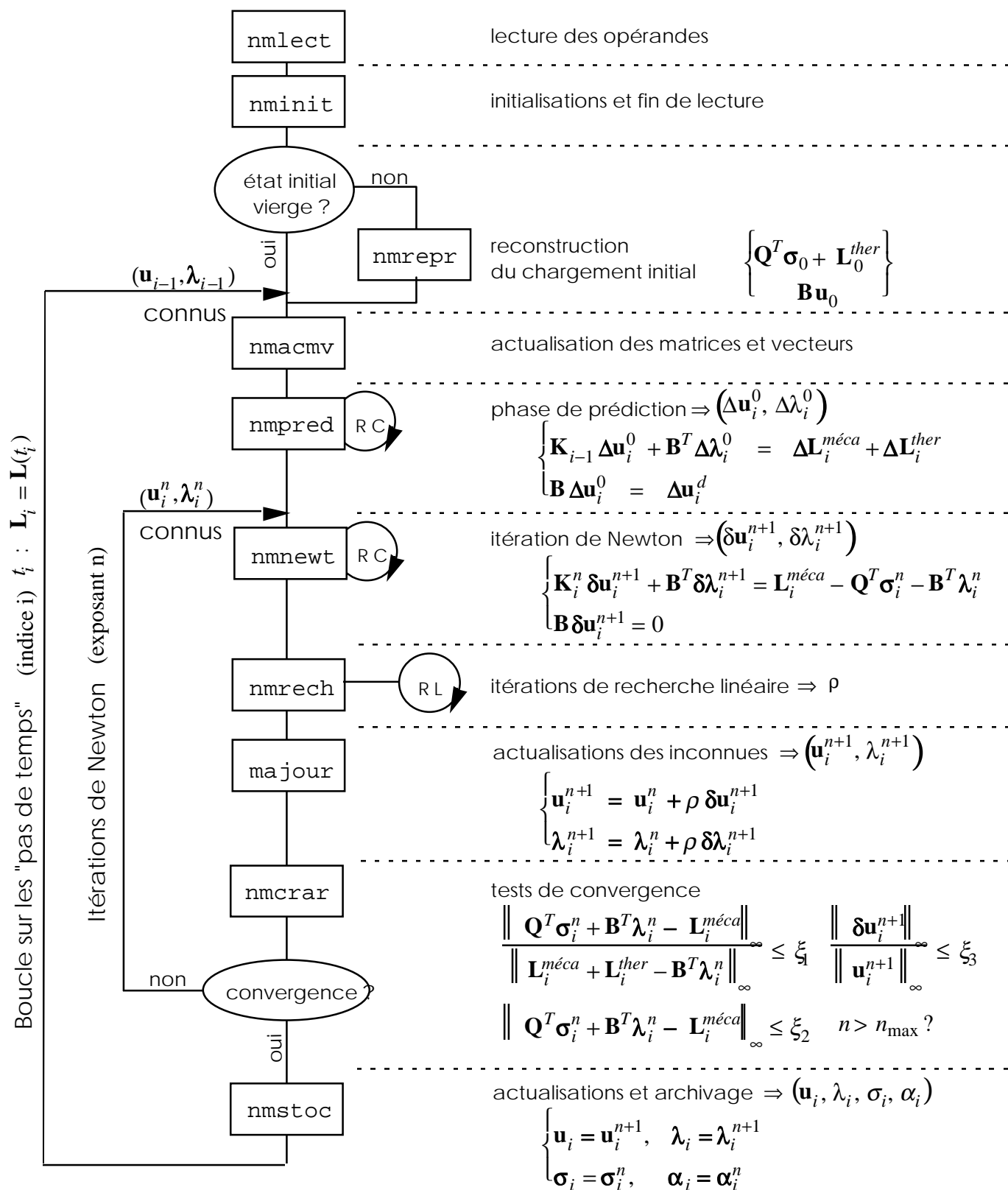
$$\left\{ \begin{array}{l} \mathbf{K}_i^n \delta \mathbf{u}_i^{n+1} + \mathbf{B}^T \delta \lambda_i^{n+1} - \delta \eta_i^{n+1} \mathbf{L}^{pilo} = \mathbf{L}_i^{cste} + \eta_i^n \mathbf{L}^{pilo} - (\mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \lambda_i^n) \\ \mathbf{B} \delta \mathbf{u}_i^{n+1} - \delta \eta_i^{n+1} \mathbf{u}^{pilo} = 0 \\ \mathbf{P}_i^n \delta \mathbf{u}_i^{n+1} = \beta_i - \mathbf{D}(\mathbf{u}_i^n) \end{array} \right.$$

$$\left\{ \begin{array}{l} \delta \eta_i^{n+1} = \frac{\beta_i - \mathbf{D}(\mathbf{u}_i^n) - \mathbf{C}_i^n (\mathbf{A}_i^n)^{-1} \begin{pmatrix} \mathbf{L}_i^{cste} \\ 0 \end{pmatrix} - \mathbf{C}_i^n (\mathbf{A}_i^n)^{-1} \begin{pmatrix} \eta_i^n \mathbf{L}^{pilo} - (\mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \lambda_i^n) \\ 0 \end{pmatrix}}{\mathbf{C}_i^n (\mathbf{A}_i^n)^{-1} \begin{pmatrix} \mathbf{L}^{pilo} \\ \mathbf{u}^{pilo} \end{pmatrix}} \\ \begin{pmatrix} \delta \mathbf{u}_i^{n+1} \\ \delta \lambda_i^{n+1} \end{pmatrix} = (\mathbf{A}_i^n)^{-1} \begin{pmatrix} \mathbf{L}_i^{cste} \\ 0 \end{pmatrix} + (\mathbf{A}_i^n)^{-1} \begin{pmatrix} \eta_i^n \mathbf{L}^{pilo} - (\mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \lambda_i^n) \\ 0 \end{pmatrix} + \delta \eta_i^{n+1} (\mathbf{A}_i^n)^{-1} \begin{pmatrix} \mathbf{L}^{pilo} \\ \mathbf{u}^{pilo} \end{pmatrix} \end{array} \right.$$

éq 1.4-2



## 2 Organigramme simplifié de STAT\_NON\_LINE



Les boucles avec le symbole RC indiquent l'expression de la relation de comportement (à travers la routine `merimo`, cf. [§5.10]).

La boucle avec le symbole RL indique les itérations de recherche linéaire (cf. [§5.5]).

On a donné dans l'organigramme des équations simplifiées qui ne tiennent pas compte des chargements suiveurs ni des chargements pilotés.

### 3 Arbre d'appel de l'op0070

On donne ici les principales routines appelées par l'op0070 ainsi que leur fonction. On ne parle pas des appels aux routines superviseur, aux routines de création d'objets JEVEUX, d'assemblage de vecteurs et de matrices, d'impression de messages. Une description plus détaillée des routines principales est donné dans le [§5].

op0070	nmlect	début de la lecture du fichier de commandes (cf. mots-clés)
	nmdome	modèle, matériau, chargement (MODELE, CHAM_MATER, CARA_ELEM, EXCIT)
	nmdorc	relation de comportement (COMP_ELAS, COMP_INCR)
	nmdomt	paramètres de la méthode de résolution (NEWTON)
	cresol	paramètres du solveur (SOLVEUR)
	nmdocn	critères de convergence (CONVERGENCE)
	nmdoop	options de calcul (OPTION)
	nminit	initialisations et fin de la lecture du fichier de commandes
	numero	création de la numérotation - profil de la matrice
	nmdoet	saisie de l'état initial (ETAT_INIT)
	nminst	calcul de l'instant initial
	nmldoin	liste des instants de calcul
	nminin	instant initial
	nmpogd	initialisations pour poutres en grandes rotations
	nmdopi	saisie des données de pilotage (PILOTAGE)
	medime	matrices de rigidité élémentaires (contribuant à la matrice <b>B</b> ) associées aux conditions aux limites
	nmdete	récupération du champ de température initial
	nmrepr	prise en compte d'un état initial non nul
	vefnme	calcul des forces nodales initiales ( $\mathbf{Q}^T \boldsymbol{\sigma}_0$ )
	vebume	calcul des déplacements imposés initialement ( $\mathbf{B} \mathbf{u}_0$ )
	vectme	calcul du chargement thermique initial ( $\mathbf{L}_0^{ther}$ )
	vaczme	modification due à la métallurgie ( $\rightarrow \mathbf{L}_0^{ther}$ )
	veanme	modification due à des déformations anélastiques données ( $\rightarrow \mathbf{L}_0^{ther}$ )
	nmitin	récupération de l'instant de calcul courant $t_i$

nmacmv	actualisation des matrices et vecteurs pour le nouveau t
nmrema	réassemblage de la matrice tangente ? (booléen REASMA)
merimo	calcul de la matrice tangente et des forces nodales
mecgme	calcul de la matrice tangente des chargements suiveurs
preres	factorisation ou calcul du préconditionneur
vedime	vecteur des déplacements imposés non pilotés
vechme	vecteur des charges mécaniques non suiveuses non pilotées
vedpme	vecteur des déplacements imposés pilotés
vefpme	vecteur des charges mécaniques non suiveuses pilotées
vecgme	vecteur des charges mécaniques suiveuses
vectme	vecteur du chargement thermique
vaczme	modification du chargement thermique due à la métallurgie
veanme	modification du chargement thermique due aux déformations anélastiques données
nmpred	phase préliminaire de prédiction
vatzme	incrément de charge thermique dû à la métallurgie
nmreso	résolution de système linéaire
nmnico	traitement du contact unilatéral
nmnewt	traitement d'une itération de Newton
vecgme	vecteur des charges mécaniques suiveuses
nmrema	réassemblage de la matrice tangente ? (booléen REASMA)
merimo	calcul de la matrice tangente et des forces nodales
mecgme	calcul de la matrice tangente des chargements suiveurs
preres	factorisation ou calcul du préconditionneur
nmreso	résolution de système linéaire
nmrech	recherche linéaire
majour	actualisation des inconnues
merimo	calcul des forces nodales
nmcrrl	test d'arrêt de la recherche linéaire
majour	actualisation des inconnues
nmcrar	test d'arrêt des itérations de Newton
nmstoc	actualisations et archivage des résultats
ndarch	archivage des résultats du pas de temps

## 4 Contenu des principales variables manipulées

### 4.1 Chargements

Le chargement mécanique total se décompose de la façon suivante :

$$\begin{pmatrix} \mathbf{L}_i^{méca} \\ \mathbf{u}_i^d \end{pmatrix} = \begin{pmatrix} \mathbf{L}_i^{fixe} \\ \mathbf{u}_i^{fixe} \end{pmatrix} + \begin{pmatrix} \mathbf{L}_i^{suiv} \\ 0 \end{pmatrix} + \eta_i \begin{pmatrix} \mathbf{L}_i^{pilo} \\ \mathbf{u}_i^{pilo} \end{pmatrix}$$

variable	adresse JEVEUX	contenu	description	routine créatrice	vecteurs élémentaires
CNCHMP	JCHMP	$\begin{pmatrix} \mathbf{L}_i^{fixe} \\ 0 \end{pmatrix}$	chargements de Neumann non suiveurs non pilotés	vechme	VECHMP
CNFPPI	JFPPI	$\begin{pmatrix} \mathbf{L}_i^{pilo} \\ 0 \end{pmatrix}$	chargements de Neumann non suiveurs pilotés	vefpme	VEFPPI
CNCGMP	JCGMP	$\begin{pmatrix} \mathbf{L}_i^{suiv} \\ 0 \end{pmatrix}$	chargements de Neumann suiveurs non pilotés	vecgme	VECGMP
CNDIRP	JDIRP	$\begin{pmatrix} 0 \\ \mathbf{u}_i^{fixe} \end{pmatrix}$	chargements de Dirichlet non pilotés	vedime	VEDIRP
CNDPIP	JDPIP	$\begin{pmatrix} 0 \\ \mathbf{u}_i^{pilo} \end{pmatrix}$	chargements de Dirichlet pilotés	vedpme	VEDPIP
CNCHTP	JCHTP	$\begin{pmatrix} \mathbf{L}_i^{ther} \\ 0 \end{pmatrix}$	chargement thermique + métallurgie + déf. anélastiques	vectme vaczme veanme	VECHTP
CHFIXE		$\begin{pmatrix} \mathbf{L}_i^{fixe} + \eta_i \mathbf{L}_i^{pilo} \\ 0 \end{pmatrix}$	total des chargements de Neumann non suiveurs		
CHCSTE	JCHCS	$\begin{pmatrix} \mathbf{L}_i^{fixe} + \mathbf{L}_i^{suiv} + \mathbf{L}_i^{ther} \\ \mathbf{u}_i^{fixe} \end{pmatrix}$	total des chargements non pilotés		
CHPILO	JCHPI	$\begin{pmatrix} \mathbf{L}_i^{pilo} \\ \mathbf{u}_i^{pilo} \end{pmatrix}$	total des chargements pilotés		

### 4.2 Inconnues, réactions d'appui et forces nodales

On rappelle les notations en incréments :

$$\begin{cases} \Delta \mathbf{u}_i^{n+1} = \mathbf{u}_i^{n+1} - \mathbf{u}_{i-1} \\ \rho \delta \mathbf{u}_i^{n+1} = \mathbf{u}_i^{n+1} - \mathbf{u}_i^n \end{cases}$$

**Attention :**

Le contenu des objets (en particulier les inconnues) est variable au cours des routines : on donne dans le [§5] le détail des objets selon les routines traversées. Dans le tableau suivant on donne la signification **prévue** pour ces objets.

Par exemple, l'objet DDEPLA est destiné à recueillir l'incrément des inconnues entre deux itérations de Newton. Pourtant, dans nmpred on y stocke provisoirement le second membre du système linéaire à résoudre, puis la solution de ce système, qui est l'incrément entre l'équilibre précédent et la prédiction.

objet	adresse JEVEUX	contenu	description
DEPMOI	JDEPM	$\begin{pmatrix} \mathbf{u}_{i-1} \\ \lambda_{i-1} \end{pmatrix}$	déplacement et Lagrange à l'équilibre précédent
DEPLU	JDEPP	$\begin{pmatrix} \mathbf{u}_i^n \\ \lambda_i^n \end{pmatrix}$	déplacement et Lagrange courants
DEPDEL	JDEPDE	$\begin{pmatrix} \Delta \mathbf{u}_i^n \\ \Delta \lambda_i^n \end{pmatrix}$	incrément des inconnues depuis l'équilibre précédent
DEPDET	JDEPDT	$\begin{pmatrix} \Delta \mathbf{u}_i^n + \rho^p \delta \mathbf{u}_i^{n+1} \\ \Delta \lambda_i^n + \rho^p \delta \lambda_i^{n+1} \end{pmatrix}$	vecteur auxiliaire pour la recherche linéaire
DDEPLA	JDDEPL	$\begin{pmatrix} \delta \mathbf{u}_i^n \\ \delta \lambda_i^n \end{pmatrix}$	incrément des inconnues depuis l'itéré de Newton précédent
CNRESI	JCNRE	$\begin{pmatrix} \mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \lambda_i^n \\ 0 \end{pmatrix}$	forces nodales (au sens large, incluant les réactions d'appui)
CNDIRI	JDIRI	$\begin{pmatrix} \mathbf{B}^T \lambda_i^n \\ 0 \end{pmatrix}$	réactions d'appui
SIGMOI		$\boldsymbol{\sigma}_{i-1}$	contraintes à l'équilibre précédent
SIGPLU		$\boldsymbol{\sigma}_i^n$	contraintes courantes
VARMOI		$\alpha_{i-1}$	variables internes à l'équilibre précédent
VARMOJ		$\alpha_{i-1}^{n-1}$	variables internes à l'itéré précédent
VARPLU		$\alpha_{i-1}^n$	variables internes courantes
TEMMOI		$T_{i-1}$	champ de température à l'équilibre précédent
TEMPLU		$T_i$	champ de température courant

## NB :

On fait une distinction entre les valeurs de l'équilibre précédent (c'est-à-dire en  $t_{i-1}$ ) et les valeurs de l'itéré précédent, c'est-à-dire celles que l'on obtient au cours des itérations de Newton et qui ne sont a priori pas en équilibre.

## 5 Détail des routines principales

### 5.1 Routine `nmrepr`

Cette routine permet de traiter prendre en compte un état initial non vierge (cas des reprises notamment). On suppose connus (lus sous le mot-clé `ETAT_INIT` dans la routine `nmdoet`) le champ de déplacements  $\mathbf{u}_0$  et de contraintes  $\boldsymbol{\sigma}_0$ . Dans ce cas le booléen `NEWCAL` vaut `.FALSE`.

Principales routines appelées :

- `vefnme` : calcul de  $\begin{pmatrix} \mathbf{Q}^T \boldsymbol{\sigma}_0 \\ 0 \end{pmatrix}$ , forces nodales initiales ; le calcul sera effectué sur la géométrie déformée initiale (prise en compte de  $\mathbf{u}_0$ ) si on a indiqué `GEOMETRIE: 'DEFORMEE'` sous le mot-clé `ETAT_INIT` ;
- `vebume` : calcul de  $\begin{pmatrix} 0 \\ \mathbf{B}\mathbf{u}_0 \end{pmatrix}$ , déplacements imposés initiaux ;
- `vectme`, `vaczme`, `veanme` : calcul de  $\begin{pmatrix} \mathbf{L}_0^{ther} \\ 0 \end{pmatrix}$ , chargement initial dû à la thermique, à la métallurgie et aux déformations anélastiques imposées.

On stocke le chargement initial (reconstruit) dans `CHCSTE` :

$$\text{CHCSTE} = \begin{pmatrix} \mathbf{L}_0^{ther} + \mathbf{Q}^T \boldsymbol{\sigma}_0 \\ \mathbf{B}\mathbf{u}_0 \end{pmatrix}$$

Si `NEWCAL=.true.`, l'état initial est vierge et `CHCSTE` vaut 0.

### 5.2 Routine `nmacmv`

Elle actualise les matrices et vecteurs au début d'un nouveau de pas de temps (dans l'opérateur `THER_MECA_NOLI`, cette routine est également appelée au sein d'un pas de temps pour chaque nouvelle itération de couplage thermo-mécanique).

Principales routines appelées :

- `merimo` : calcul de la matrice tangente si `OPTION= 'RIGI_MECA_TANG'`, sinon rien.
- `mecgme` : calcul de la matrice tangente associée aux chargements suiveurs.

**NB :**

Les matrices élémentaires issues de *merimo* (objet *MERIGI*) et *mecgme* (objet *MESUIV*) sont assemblées avec les matrices relatives aux Lagrange (calculées une fois pour toutes par *medime* au début de l'op0070 et stockées dans l'objet *MEDIRI*) dans une grande matrice (objet *MATASS*), que l'on peut symboliquement noter :

$$\mathbf{A}_{i-1} = \begin{pmatrix} \mathbf{K}_{i-1} - \frac{\partial \mathbf{L}^{suiv}}{\partial \mathbf{u}} \Big|_{\mathbf{u}_{i-1}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix}$$

- *preres* : factorisation de la matrice tangente (si solveur LDLT ou MULT\_FRONT) ou calcul du préconditionneur (si solveur GCPC),
- *vedime*, *vechme*, *vedpme*, *vefpme*, *vecgme*, *vectme*, *vaczme*, *veanme* : actualisations des vecteurs représentant les différentes contributions au chargement (cf. tableau du [§ 4.1]).

## 5.3 Routine *nmpred*

Cette routine effectue le calcul du prédicteur  $\begin{pmatrix} \Delta \mathbf{u}_i^0 \\ \Delta \lambda_i^0 \end{pmatrix}$ , en résolvant dans le cas général (avec pilotage) le système [eq 1.4-1].

On affecte à *DEPPLU* les valeurs obtenues à l'équilibre précédent, soit  $\begin{pmatrix} \mathbf{u}_{i-1} \\ \lambda_{i-1} \end{pmatrix}$ .

On stocke dans *DDEPLA* l'incrément de chargement non piloté (mécanique + "thermique") :

*DDEPLA* = *CNCHMP* + *CNCGMP* + *CNDIRP* + *CNCHTP* - *CHCSTE* (courant-précédent)

$$\text{soit } \mathbf{DDEPLA} = \begin{pmatrix} \Delta \mathbf{L}_i^{fixe} + \Delta \mathbf{L}_i^{suiv} + \Delta \mathbf{L}_i^{ther} \\ \Delta \mathbf{u}_i^{fixe} \end{pmatrix}$$

La routine *vatzme* complète ce second membre par la prise en compte de l'incrément de charge dû à la plasticité de transformation (métallurgie).

**NB :**

$$\left| \begin{array}{l} \text{Au premier pas de temps, } \mathbf{CHCSTE} \text{ contient } \begin{pmatrix} \mathbf{Q}^T \boldsymbol{\sigma}_0 + \mathbf{L}_0^{ther} \\ \mathbf{B} \mathbf{u}_0 \end{pmatrix} \text{ donc } \mathbf{DDEPLA} \text{ vaudra :} \\ \mathbf{DDEPLA} = \begin{pmatrix} \mathbf{L}_1^{fixe} + \mathbf{L}_1^{suiv} - \mathbf{Q}^T \boldsymbol{\sigma}_0 + \Delta \mathbf{L}_1^{ther} \\ \mathbf{u}_1^{fixe} - \mathbf{B} \mathbf{u}_0 \end{pmatrix} \text{ (cf. [eq 1.2-2]).} \end{array} \right.$$

On actualise le chargement non piloté :

*CHCSTE* = *CNCHMP* + *CNCGMP* + *CNDIRP* + *CNCHTP* (courant)

$$\text{soit } \mathbf{CHCSTE} = \begin{pmatrix} \mathbf{L}_i^{fixe} + \mathbf{L}_i^{méca} + \mathbf{L}_i^{ther} \\ \mathbf{u}_i^{fixe} \end{pmatrix}$$

On actualise le chargement piloté :

$$\text{CHPILO} = \text{CNFPIP} + \text{CNDPIP} \quad (\text{courant})$$

$$\text{soit } \text{CHPILO} = \begin{pmatrix} \mathbf{L}_i^{\text{pilo}} \\ \mathbf{u}_i^{\text{pilo}} \end{pmatrix}$$

On résout le système linéaire [éq 1.4-1] en appelant `nmreso` (cf. [§5.9]) avec en arguments `CHPILO` et `DDEPLA` (second membre `CHRESU` en entrée, solution en sortie).

**NB :**

| *DEPDEL est remis à zéro dans la routine `initia`.*

Après `nmpred` dans l'op0070, on actualise `DEPPLU` et `DEPDEL` (routine `majour`) :

$$\text{DEPPLU} = \text{DEPPLU} + \text{DDEPLA} = \begin{pmatrix} \mathbf{u}_{i-1} + \Delta \mathbf{u}_i^0 \\ \boldsymbol{\lambda}_{i-1} + \Delta \boldsymbol{\lambda}_i^0 \end{pmatrix}$$

$$\text{DEPDEL} = \text{DEPDEL} + \text{DDEPLA} = \begin{pmatrix} 0 + \Delta \mathbf{u}_i^0 \\ 0 + \Delta \boldsymbol{\lambda}_i^0 \end{pmatrix}$$

## 5.4 Routine `nmnewt`

On actualise le second membre associé aux chargements suiveurs `CNCGMP`.

Appel de `merimo` : on calcule les forces nodales (si `OPTION='RAPH_MECA'`) ou les forces nodales et la matrice tangente (si `OPTION='FULL_MECA'`). S'il y a réassemblage (`REASMA=.TRUE.`), on calcule la matrice tangente associée aux chargements suiveurs, puis on réassemble et on factorise (`preres`) la matrice tangente globale notée  $\mathbf{A}_i^n$ . Tout ceci est identique aux opérations effectuées dans `nmacmv` (seule différence : l'option de calcul de la matrice tangente est '`FULL_MECA`' et non '`RIGI_MECA_TANG`', cf [R5.03.01] et [R5.03.02]).

On assemble les forces nodales et les réactions d'appui relatives à l'itéré précédent, respectivement dans `CNRESI` et `CNDIRI`.

On stocke dans `DDEPLA` le second membre du système linéaire à résoudre :

$$\begin{aligned} \text{DDEPLA} &= \text{CNCHMP} + \text{CNCGMP} + \text{ETA} * \text{CNFPIP} - \text{CNRESI} \\ \text{DDEPLA} &= \begin{pmatrix} \mathbf{L}_i^{\text{fixe}} + \mathbf{L}_i^{\text{suiv}} + \eta_i^n \mathbf{L}_i^{\text{pilo}} - (\mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \boldsymbol{\lambda}_i^n) \\ 0 \end{pmatrix} \end{aligned}$$

On résout le système linéaire [éq 1.4-2] en appelant `nmreso` (cf. [§5.9]) avec en arguments `CHPILO` (inchangé) et `DDEPLA` (second membre en entrée, solution en sortie).



## 5.5 Routine nmrech

Cette routine calcule le scalaire  $\rho$  (RHO) qui sera le pas d'avancement dans la direction donnée par DDEPLA.

**NB :**

| Pour l'instant, la recherche linéaire n'est appelée que si  $ETA = 0$  (pas de pilotage).

On envoie séparément les vecteurs : CNCHMP, CNCGMP, CNFPIP.

La recherche linéaire consiste à annuler le produit :

$$g = DDEPLA * [CNRESI - CNCHMP - CNCGMP - ETA * CNFPIP]$$

en utilisant une méthode de sécante.

On fait une boucle sur un indice  $p$ , en partant de  $\rho^0 = 0$  et  $\rho^1 = 1$ . Connaissant  $\rho^{p-1}$  (stocké dans RHO0),  $g^{p-1}$  (stocké dans F0) et  $\rho^p$  (stocké dans RHO) :

- calcul pour  $\rho^p$  de  $DEPDEL + RHO * DDEPLA$  stocké dans DEPDET (routine majour) :
- $\Delta \mathbf{u}_i^n + \rho^p \delta \mathbf{u}_i^{n+1}$
- calcul des forces nodales CNRESI (option 'RAPH\_MECA') correspondant à DEPDET (routine merimo)
- calcul de  $DDEPLA * [CNRESI - CNCHMP - CNCGMP - ETA * CNFPIP]$ , soit :

$$g(\rho^p) = [\delta \mathbf{u}_i^{n+1}]^T [Q^T \sigma(\mathbf{u}_i^n + \rho^p \delta \mathbf{u}_i^{n+1}) + \mathbf{B}^T (\lambda_i^n + \rho^p \delta \lambda_i^{n+1}) - \mathbf{L}_i^{méca}]$$

stocké dans F1

- vérification du critère d'arrêt de la recherche linéaire (routine nmcr1) : valeur de  $CNRESI - CNCHMP - CNCGMP - ETA * CNFPIP$  inférieure à  $RESI\_LINE\_RELA$ , ou  $p$  supérieur à  $ITER\_LINE\_MAXI$
- calcul du nouveau RHO (RHOT variable auxiliaire de transfert) :

$$\rho^{p+1} = \frac{-\rho^p g^{p-1} + \rho^{p-1} g^p}{g^p - g^{p-1}} = \frac{-RHO * F0 + RHO0 * F1}{F1 - F0}$$

Après nmrech dans l'op0070, on actualise DEPPLU et DEPDEL (routine majour) :

$$DEPPLU = DEPPLU + RHO * DDEPLA = \begin{pmatrix} \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \lambda_i^n + \rho \delta \lambda_i^{n+1} \end{pmatrix}$$

$$DEPDEL = DEPDEL + RHO * DDEPLA = \begin{pmatrix} \Delta \mathbf{u}_i^n + \rho \delta \mathbf{u}_i^{n+1} \\ \Delta \lambda_i^n + \rho \delta \lambda_i^{n+1} \end{pmatrix}$$

## 5.6 Routine nmcrar

Cette routine teste les différents critères d'arrêt des itérations de Newton.

On lui envoie séparément les vecteurs représentatifs du chargement :

```
CHFIXE = CNCHMP + ETA*CHFPIP
CNCGMP
CNCHTP
CNDIRI
```

On calcule les quantités précisées au [§ 1.2.5] et on compare aux valeurs limites fixées par l'utilisateur. On imprime les valeurs des différents critères et le nombre d'itérations de la recherche linéaire. Les critères vérifiés sont affublés d'une \* à côté de leur valeur. La convergence est décrétée si tous les critères spécifiés sont vérifiés. L'op0070 est donc informée que CONVER=.TRUE.

**NB :**

*Avant de regarder la convergence, on regarde si ITERAT (n) est strictement supérieur à la valeur donnée sous ITER\_GLOB\_MAXI, et si c'est le cas on s'arrête. Cela veut dire dans ce cas que l'on a effectué une itération de Newton inutile (puisque l'on ne teste pas la convergence) : à corriger dans une prochaine version.*

## 5.7 Routine nmstoc

Cette routine effectue les actualisations nécessaires à la fin du pas de temps (basculement des valeurs XXXPLU et XXXMOI : déplacements, contraintes, variables internes, températures, ETA) ainsi que les archivages (routine ndarch) dans la structure de données résultats evol\_noli.

La solution stockée est :

$$\begin{cases} \mathbf{u}_i = \mathbf{u}_i^{n+1}, & \lambda_i = \lambda_i^{n+1} \\ \boldsymbol{\sigma}_i = \boldsymbol{\sigma}_i^n, & \boldsymbol{\alpha}_i = \boldsymbol{\alpha}_i^n \end{cases}$$

Il faut actualiser le chargement (qui sera pris comme chargement du pas de temps précédent dans le prochain nmpred). On a besoin de recalculer CHCSTE puisque les forces suiveuses ont changé depuis nmpred (dans nmnewt et nmrech) :

```
CHCSTE = CNCHMP + CNCGMP (nouveau) + CNDIRP + CNCHTP
```

$$\text{CHCSTE} = \begin{pmatrix} \mathbf{L}_i^{\text{fixe}} + \mathbf{L}_i^{\text{suiv}} + \mathbf{L}_i^{\text{ther}} \\ \mathbf{u}_i^{\text{fixe}} \end{pmatrix}$$

Le vecteur CHPILO ne sert pas à calculer des incréments et est construit quand on en a besoin, on n'a donc pas à le mettre à jour. Idem pour le vecteur CHFIXE qui ne sert que dans le critère d'arrêt.

**NB :**

*On pourrait stocker CHFIXE et CHPILO dans le même objet informatique puisque l'on les recalcule à chaque fois qu'on en a besoin. Mais il est préférable pour la clarté de l'op0070 d'avoir ces deux objets.*

## 5.8 Routine merimo

Elle exprime la relation de comportement :

- calcul des contraintes  $\sigma_i^n$  (SIGPLU) et des variables internes  $\alpha_i^n$  (VARPLU) à partir des valeurs  $\sigma_{i-1}$  (SIGMOI) et  $\alpha_{i-1}$  (VARMOI) à l'équilibre précédent (en  $t_{i-1}$ ) et de l'incrément de déplacement  $\Delta \mathbf{u}_i^n = \mathbf{u}_i^n - \mathbf{u}_{i-1}$  depuis cet équilibre (DEPDEL) ;
- calcul des "forces nodales" (forces internes + réactions d'appui) :  $\mathbf{Q}^T \sigma_i^n + \mathbf{B}^T \lambda_i^n$  (VERESI) et des réactions d'appui  $\mathbf{B}^T \lambda_i^n$  (VEDIRI), que l'on assemble ensuite ;
- calcul éventuel de la matrice de raideur tangente :  $\mathbf{K}_i^n$  (MERIGI).

Pour mémoire on donne ici l'enchaînement des routines qui, partant de `merimo`, traite la relation de comportement `VMIS_ISOT_TRAC` pour les éléments isoparamétriques 2D :

- `merimo` : remplissage des champs d'entrée et de sortie pour `calcul`, appel de `calcul` pour calculer les forces nodales, appel de `calcul` pour calculer les réactions d'appui
- `calcul` : boucle sur les éléments
- `te0100` : pour chaque élément, aiguillage des options de calcul (`FULL_MECA`, ...) et appel de `nmp12d`
- `nmp12d` : boucle sur les points de Gauss, calcul des éléments géométriques, des déformations, appel de `nmcomp`, remplissage de la matrice élémentaire de rigidité et du vecteur élémentaire des forces nodales
- `nmcomp` : aiguillage des relations de comportement
- `nmisot` : calcul des contraintes, variables internes, et dérivée des contraintes par rapport aux déformations pour la relation de comportement `VMIS_ISOT_TRAC`

## 5.9 Routine nmreso

On rappelle que :

$$\begin{cases} \mathbf{L}_i^{cste} = \mathbf{L}_i^{fixe} + \mathbf{L}_i^{suiv} \\ \mathbf{u}_i^{cste} = \mathbf{u}_i^{fixe} \end{cases}$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{K} - \frac{\partial \mathbf{L}^{suiv}}{\partial \mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix}$$

$$\mathbf{C} = (\mathbf{P} \quad 0)$$

En entrée de nmreso provenant de nmpred :

$$\text{CHPILO} = \begin{pmatrix} \mathbf{L}_i^{pilo} \\ \mathbf{u}_i^{pilo} \end{pmatrix}$$

$$\text{CHRESU} = \begin{pmatrix} \Delta \mathbf{L}_i^{cste} + \Delta \mathbf{L}_i^{ther} \\ \Delta \mathbf{u}_i^{cste} \end{pmatrix}$$

$$\text{MATASS} = \mathbf{A} = \mathbf{A}_{i-1}$$

$$\text{PILOTE} \rightarrow \mathbf{C} = \mathbf{C}_{i-1}$$

$$\text{DEPPLU} = \mathbf{u}_{i-1}$$

En entrée de nmreso provenant de nmnewt :

$$\text{CHPILO} = \begin{pmatrix} \mathbf{L}_i^{pilo} \\ \mathbf{u}_i^{pilo} \end{pmatrix}$$

$$\text{CHRESU} = \begin{pmatrix} \mathbf{L}_i^{cste} + \eta_i^n \mathbf{L}_i^{pilo} - (\mathbf{Q}^T \boldsymbol{\sigma}_i^n + \mathbf{B}^T \boldsymbol{\lambda}_i^n) \\ 0 \end{pmatrix}$$

$$\text{MATASS} = \mathbf{A} = \mathbf{A}_i^n$$

$$\text{PILOTE} \rightarrow \mathbf{C} = \mathbf{C}_i^n$$

$$\text{DEPPLU} = \mathbf{u}_i^n$$

Dans nmreso : on résout [éq 1.4-1] ou [éq 1.4-2] :

$$\text{CHSOL1} = \mathbf{A}^{-1} \cdot \text{CHPILO}$$

$$\text{CHSOL2} = \mathbf{A}^{-1} \cdot \text{CHRESU}$$

$$\text{DU} = \mathbf{D}(\mathbf{u})$$

$$\text{RD} = \mathbf{C} \cdot \text{CHSOL1}$$

$$\text{RN} = \mathbf{C} \cdot \text{CHSOL2}$$

$$\text{DETA} = \delta \eta_i^{n+1} = \frac{\text{BETA} - \text{DU} - \text{RN}}{\text{RD}}$$

$$\text{CHRESU} = \begin{pmatrix} \Delta \mathbf{u}_i^0 \\ \Delta \boldsymbol{\lambda}_i^0 \end{pmatrix} \text{ ou } \begin{pmatrix} \delta \mathbf{u}_i^{n+1} \\ \delta \boldsymbol{\lambda}_i^{n+1} \end{pmatrix} = \text{DETA} \cdot \text{CHSOL1} + \text{CHSOL2}$$

$$\text{ETA} = \eta_i^{n+1} = \eta_i^n + \delta \eta_i^{n+1} = \text{ETA} + \text{DETA}$$

## 6 Pièges et astuces

Dans ce chapitre on donne en désordre un certain nombre d'informations importantes (voire essentielles) pour qui veut développer dans l'op0070.

- VERESI, RESIDU, CNRESI, ... : ce sont les forces nodales, c'est-à-dire les forces internes + les réactions d'appui. Contrairement à ce que pourrait laisser penser le nom de la variable, **ce n'est pas un résidu** : ça ne tend donc pas vers 0 lorsqu'on converge (ça tend vers le chargement extérieur) !!!
- le calcul des matrices élémentaires associées aux Lagrange est fait une fois pour toutes au début de l'op0070 (appel de medime) : en effet, même si la valeur des Dirichlet imposés change au cours du temps, ça modifie le second membre  $\mathbf{u}^d$  et non la matrice  $\mathbf{B}$ .
- le chargement thermique n'intervient que par son  $\Delta$  (sauf pour le critère d'arrêt) : néanmoins on calcule toujours sa valeur à l'instant précédent et à l'instant courant et on en prend la différence (pour tenir compte de la variation éventuelle du coefficient de dilatation  $\alpha$  en fonction de la température).
- l'objet CNCHTP (chargement thermique) est appelé un peu différemment des autres chargements. En effet, pour les chargements mécaniques (CNCHMP par exemple), l'objet est une liste de vecteurs assemblés de type VACHMP. Puisqu'une seule charge peut être thermique, l'objet CNCHTP est en fait l'équivalent d'un seul VACHMP : on prend son adresse JTP, et les valeurs sont à l'adresse de l'objet ZK24(JTP), soit JCHTP.
- pour limiter le nombre d'arguments des routines, on a regroupé certains objets dans un tableau donnant leurs noms :
  - VALMOI contient au maximum DEPMOI, SIGMOI, TEMMOI, VARMOI, et VARMOJ
  - VALPLU contient au maximum DEPPLU, SIGPLU, TEMPLU, et VARPLU
  - SECMBR contient au maximum CNCHMP, CNFPIP, CNCGMP, CNDIRP, CNDPIP, CNCHTP, CNRESI, et CNDIRI
  - POUGD contient des vecteurs relatifs aux calculs de poutres en grandes rotationsLe "remplissage" et "vidage" de ces tableaux de chaînes de caractères est fait par l'intermédiaire des routines agglom et desaggl.
- le traitement du contact consiste à appeler une routine nmmico après nmpred et nmnewt, ce qui corrige le champ de déplacement, et à rajouter les forces de contact obtenues (stockées à l'adresse JATMU) aux forces nodales CNRESI et aux réactions d'appui CNDIRI.
- pour une raison pas bien comprise de l'auteur de ce document, l'appel à la routine nmreso invalide l'adresse de l'objet DDEPLA : il faut donc la resaisir par un appel à jeveuo après nmpred et nmnewt.
- dans les routines nmxxx tout n'est pas forcément passé en argument : par exemple, MEDIRI dans nmacmv est connu par un DATA et utilisé ainsi.
- attention au maniement des DATA : la substitution des noms n'est faite que lors de la compilation. Il est préférable lorsque l'on veut imposer des noms fixes pour des objets calculés à plusieurs reprises, de les déclarer comme instruction en début de routine (cf. nmacmv).

- si un vecteur élémentaire est stocké dans une variable de nom `&&VECHME...`, le vecteur assemblé est stocké dans une variable de nom `&&VACH00...`. Les seuls caractères discriminants pour les noms de vecteurs sont donc les 1<sup>er</sup>, 3<sup>ème</sup> et 4<sup>ème</sup>. Ainsi deux vecteurs `&&VECHME` et `&&VECHTP` conduiraient à une ambiguïté et donc des bugs. C'est pour cela que l'on a utilisé les noms `&&VEMCHA...` (pour `VECHMP`), `&&VEMSUI...` (pour `VECGMP`), `&&VEMTPP...` (pour `VECHTP`), etc. (cf. `nmacmv` par exemple). Depuis la création de l'opérateur `THER_MECA_NOLI`, il faut également faire attention à éviter des ambiguïtés entre les vecteurs thermiques (flux, températures imposées...) et les vecteurs mécaniques.
- les routines `initia` et `majour` se rapportent partiellement aux poutres en grandes rotations.
- le test de convergence est fait après la première itération de Newton (on en fait donc toujours au moins une) : ainsi pour un calcul élastique où on a la solution à l'issue de la phase de prédiction, on fait une itération de Newton pour rien...