

---

## Opérateur POST\_FATIGUE

---

### 1 But

---

Calculer, en un point, le dommage de fatigue d'une structure soumise à une histoire de chargement.

A la différence de CALC\_FATIGUE, POST\_FATIGUE n'opère pas sur un champ mais sur un « signal » préalablement extrait d'un calcul ou définit par ailleurs.

Les diverses méthodes disponibles [R7.04.01] sont :

- méthodes basées sur des essais uniaxiaux : méthodes de Wöhler, Manson-Coffin et Taheri. Ces méthodes ont pour point commun de déterminer une valeur de dommage à partir de l'évolution au cours du temps d'une composante scalaire caractérisant, pour le calcul du dommage, l'état de contraintes ou de déformations de la structure. Pour ce faire, il faut extraire, par une méthode de comptage de cycles, les cycles élémentaires de chargement subis par la structure, déterminer le dommage élémentaire associé à chaque cycle et déterminer le dommage total par une règle de cumul linéaire,
- méthode de Lemaître généralisée. Cette méthode permet de calculer le dommage (de Lemaître ou de Lemaître-Sermage) à partir de la donnée du tenseur des contraintes et de la déformation plastique cumulée,
- critères de Crossland et Dang Van-Papadopoulos. Ces critères s'appliquent à des chargements uniaxiaux ou multiaxiaux périodiques. Ils fournissent une valeur de critère indiquant s'il y a dommage ou non.

Produit un concept de type `table`.

## Table des matières

1 But.....	1
2 Syntaxe.....	5
3 Opérandes.....	9
3.1 Opérande CHARGEMENT.....	9
3.2 Mot clé HISTOIRE.....	9
3.2.1 Opérande SIGM.....	9
3.2.2 Opérande EPSI.....	9
3.2.3 Opérandes SIGM_XX / SIGM_YY / SIGM_ZZ / SIGM_XY / SIGM_XZ / SIGM_YZ.....	10
3.2.4 Opérande EPSP.....	11
3.2.5 Opérande TEMP.....	11
3.3 Opérande DELTA_OSCI.....	11
3.4 Mot clé COEF_MULT.....	13
3.4.1 Opérande KT.....	13
3.5 Opérande COMPTAGE.....	14
3.6 Opérande CORR_KE.....	14
3.7 Opérande DOMMAGE.....	14
3.7.1 Méthodes basées sur des essais uniaxiaux : méthode de Wöhler, méthode de Manson-Coffin, méthodes de Taheri.....	14
3.7.2 Méthodes de Lemaître et Lemaître-Sermage.....	19
3.8 Opérande CORR_SIGM_MOYE.....	20
3.9 Opérande TAHERI_NAPPE.....	20
3.10 Opérande TAHERI_FONC.....	20
3.11 Opérande MATER.....	22
3.12 Opérande CUMUL.....	22
3.13 Opérandes CRITERE.....	22
3.13.1 Critère de Crossland.....	22
3.13.2 Critère de Dang Van-Papadopoulos.....	24
3.13.3 Opérande COEF_CORR.....	24
3.14 Opérande INFO.....	25
3.15 Opérande TITRE.....	25
3.16 Table produite.....	25
4 Exemples.....	26
4.1 Calcul du dommage de Wöhler (avec correction de la contrainte moyenne).....	26
4.2 Calcul du dommage de Taheri.....	28
4.3 Calcul des critères de Crossland et Dang Van-Papadopoulos.....	29
4.4 Calcul du dommage de Lemaître.....	31
4.5 Calcul du dommage de Lemaître-Sermage.....	33

## 2 Syntaxe

```
tabl_post_fatig = POST_FATIGUE    (  
#   si chargement purement uniaxial (ou considéré comme uniaxial)  
  
    ♦   /   CHARGEMENT = 'UNIAXIAL' ,  
  
        ♦   HISTOIRE   = _F   (  
                                ♦   /   SIGM   = histsigm   /   [fonction]  
                                /   [formule]  
                                /   EPSI   = histepsi   /   [fonction]  
                                /   [formule]  
                                ),  
  
        ♦   COMPTAGE     =   /   'RAINFLOW' ,  
                                /   'RC CM' ,  
                                /   'NATUREL' ,  
  
        ◇   DELTA_OSCI    =   /   delta ,                               [R]  
                                /   0. ,                               [DEFAULT]  
  
        ◇   COEF_MULT = _F   (   ♦   KT   =   kt ),                               [R]  
  
        ◇   CORR_KE       =   'RCCM',  
  
        ◇   DOMMAGE       =   /   'WOHLER' ,  
                                /   'MANSON_COFFIN' ,  
                                /   'TAHERI_MANSON' ,  
                                /   'TAHERI_MIXTE' ,  
  
        ◇   MATER         =   mater ,  
  
        ◇   CORR_SIGM_MOYE =   /   'GOODMAN' ,  
                                /   'GERBER' ,  
  
        ◇   TAHERI_NAPPE   =   fnappe ,                               /[nappe]  
                                / [formule]  
        ◇   TAHERI_FONC    =   ffonc ,                               / [fonction]  
                                / [formule]  
  
        ◇   CUMUL          =   'LINEAIRE' ,  
  
#   Finsi
```

```
# si chargement périodique (pour fatigue à grands nombres de cycles et
pour des cycles périodiques)

♦ / CHARGEMENT = 'PERIODIQUE' ,

    ♦ HISTOIRE = _F (
        ♦ SIGM_XX = fxx ,           / [fonction]
                                           / [formule]
        ♦ SIGM_YY = fyy ,           / [fonction]
                                           / [formule]
        ♦ SIGM_ZZ = fzz ,           / [fonction]
                                           / [formule]
        ♦ SIGM_XY = fxy ,           / [fonction]
                                           / [formule]
        ◇ SIGM_XZ = fxz ,           / [fonction]
                                           / [formule]
        ◇ SIGM_YZ = fyz ,           / [fonction]
                                           / [formule]
    )

    ♦ CRITERE = / 'CROSSLAND' ,
                  / 'PAPADOPOULOS' ,

    ◇ DOMMAGE = 'WOHLER' ,

    ♦ MATER = mater ,                [mater]

    ◇ COEF_CORR = / corr ,           [R]

# Finsi

# si chargement quelconque (endommagement de Lemaitre ou de Lemaitre-
Sermage)

♦ / CHARGEMENT = 'QUELCONQUE' ,

    ♦ HISTOIRE = _F (
        ♦ SIGM_XX = fxx ,           / [fonction]
                                           / [formule]
        ♦ SIGM_YY = fyy ,           / [fonction]
                                           / [formule]
        ♦ SIGM_ZZ = fzz ,           / [fonction]
                                           / [formule]
        ♦ SIGM_XY = fxy ,           / [fonction]
                                           / [formule]
        ◇ SIGM_XZ = fxz ,           / [fonction]
                                           / [formule]
        ◇ SIGM_YZ = fyz ,           / [fonction]
                                           / [formule]
        ♦ EPSP = p ,                / [fonction]
                                           / [formule]
        ♦ TEMP = temp ,             / [fonction]
                                           / [formule]
    )

    ◇ DOMMAGE = 'LEMAITRE' ,

    ♦ MATER = mater ,

    ◇ CUMUL = 'LINEAIRE' ,

# Finsi

    ◇ INFO = / 1 ,                  [DEFAULT]
```

/ 2,  
◇ TITRE = titre [1\_Kn]  
)

## 3 Opérandes

### 3.1 Opérande CHARGEMENT

Ce mot clé permet à l'utilisateur de préciser le type de chargement traité. Le chargement peut être 'UNIAXIAL', 'PERIODIQUE' ou 'QUELCONQUE'. A chaque chargement correspond sa (ou ses) méthode(s) d'évaluation du dommage par fatigue.

**Remarque :** Lorsque le chargement est périodique, il suffit de donner l'histoire du chargement sur une période. Dans le cas où le chargement est quelconque, il faut fournir l'ensemble de l'histoire du chargement.

### 3.2 Mot clé HISTOIRE

Ce mot clé regroupe toute la phase de définition de l'histoire de chargement.

Suivant la méthode de calcul du dommage, l'histoire de chargement peut être l'évolution :

- d'une valeur de contrainte ou de déformation uniaxiale au cours du temps,

**Remarque :**

*Cela ne signifie pas que le chargement ne peut pas être multiaxial, mais seulement que pour le calcul du dommage, le chargement est caractérisé par l'évolution d'une composante scalaire, au cours du temps (von Mises signé, invariant d'ordre 2 signé, ...). C'est l'évolution de cette composante scalaire que l'utilisateur doit fournir à la commande POST\_FATIGUE.*

- du tenseur de contraintes au cours du temps,
- du tenseur de contraintes, de la déformation plastique cumulée et de la température au cours du temps.

#### 3.2.1 Opérande SIGM

◇ SIGM = histsigm,

Nom de la fonction ou de la formule décrivant l'histoire du chargement en contraintes en un point. C'est une fonction ou une formule du paramètre INST, qui donne l'évolution au cours du temps d'une composante scalaire caractérisant l'état de contraintes de la structure.

Cet opérande est obligatoire pour le calcul du dommage par une méthode de WOHLER.

#### 3.2.2 Opérande EPSI

◇ EPSI = histepsi,

Nom de la fonction ou de la formule décrivant l'histoire du chargement en déformations en un point. C'est une fonction ou une formule du paramètre INST, qui donne l'évolution au cours du temps d'une composante scalaire caractérisant l'état de déformations de la structure.

Cet opérande est obligatoire pour le calcul du dommage par les méthodes de MANSON\_COFFIN ou TAHERI\_MANSON ou TAHERI\_MIXTE.

#### 3.2.3 Opérandes SIGM\_XX / SIGM\_YY / SIGM\_ZZ / SIGM\_XY / SIGM\_XZ / SIGM\_YZ

Noms des fonctions ou des formules décrivant l'histoire de chaque composante du tenseur des contraintes au cours du temps. Chaque fonction ou formule dépend du paramètre INST. Toutes les fonctions ou formules doivent être définies pour les mêmes instants.

En 2D sont obligatoires les opérandes SIGM\_XX, SIGM\_YY, SIGM\_ZZ et SIGM\_XY.

En 3D sont obligatoires les opérandes SIGM\_XX, SIGM\_YY, SIGM\_ZZ, SIGM\_XY, SIGM\_XZ et SIGM\_YZ.

## 3.2.4 Opérande EPSP

◇ EPSP = p,

Nom de la fonction décrivant l'histoire de la déformation plastique cumulée au cours du temps, uniquement pour le calcul du dommage de LEMAITRE.

Cette fonction ou formule dépend du paramètre INST et doit être définie pour les mêmes instants que les fonctions ou formules décrivant l'histoire des composantes du tenseur des contraintes.

L'opérande EPSP doit être utilisé conjointement aux opérandes SIGM\_XX, ...

## 3.2.5 Opérande TEMP

◇ TEMP = temp,

Nom de la fonction ou de la formule décrivant l'histoire de la température au cours du temps, uniquement pour le calcul du dommage de LEMAITRE. Elle sert dans ce cas à déterminer la valeur des caractéristiques mécaniques (module d'Young  $E$ , coefficient de Poisson  $\nu$  et paramètre matériau  $S$ ) aux instants de calcul du dommage.

Cette fonction ou formule dépend du paramètre INST et doit être définie pour les mêmes instants que les fonctions ou formules décrivant l'histoire des composantes du tenseur des contraintes.

L'opérande TEMP doit être utilisé conjointement aux opérandes EPSP, SIGM\_XX,...

## 3.3 Opérande DELTA\_OSCI

◇ DELTA\_OSCI = delta,

Filtrage de l'histoire du chargement. Dans tous les cas, si la fonction reste constante ou décroissante sur plus de deux points consécutifs on supprime les points intermédiaires pour ne garder que les deux points extrêmes. Puis, on supprime de l'histoire de chargement les points pour lesquels la variation de la valeur de la contrainte est inférieure à la valeur delta. Par défaut delta est égal à zéro, ce qui revient à garder toutes les oscillations du chargement, même celles de faible amplitude.

**Exemple :** Considérons l'histoire de chargement suivante :

N° point	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Instant	0.	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.
Chargement	4.	7.	2.	10.	9.6	9.8	5.	9.	3.	4.	2.	2.4	2.2	12.	5.
N° point	16	17	18	19	20	21	22	23	24	25	26	27	28	29	
Instant	15.	16.	17.	18.	19.	20.	21.	22.	23.	24.	25.	26.	27.	28.	
Chargement	11.	1.	4.	3.	10.	6.	8.	12.	4.	8.	1.	9.	4.	6.	

L'extraction des pics de cette histoire de chargement, avec une valeur de delta de 0,9 conduit à détruire toutes les oscillations d'amplitude inférieure à 0,9. Ce qui conduit à l'histoire de chargement suivante :

N° point	1	2	3	4	7	8	9	10	11	14	15	16	17	18	19
Instant	0.	1.	2.	3.	6.	7.	8.	9.	10.	13.	14.	15.	16.	17.	18.
Chargement	4.	7.	2.	10.	5.	9.	3.	4.	2.	12.	5.	11.	1.	4.	3.
N° point	20	21	23	24	25	26	27	28	29						
Instant	19.	20.	22.	23.	24.	25.	26.	27.	28.						
Chargement	10.	6.	12.	4.	8.	1.	9.	4.	6.						

On a supprimé :

- le point 5 car  $\Delta \sigma = |\sigma(5) - \sigma(4)| < 0,9$  ,
- le point 6 car  $\Delta \sigma = |\sigma(6) - \sigma(4)| < 0,9$  ,
- le point 12 car  $\Delta \sigma = |\sigma(12) - \sigma(11)| < 0,9$  ,
- le point 13 car  $\Delta \sigma = |\sigma(13) - \sigma(11)| < 0,9$  .

De même on supprime le point 22 car l'histoire de chargement est croissante entre les points 21, 22 et 23 et donc on ne garde que les points extrêmes.

## 3.4 Mot clé COEF\_MULT

◇ COEF\_MULT = \_F

Ce mot clé facteur regroupe les coefficients d'amplification de l'histoire de chargement. Pour l'instant, un seul coefficient multiplicateur de l'histoire de chargement est disponible : le coefficient de concentration de contraintes  $K_T$ .

Des valeurs du coefficient de concentration de contraintes sont disponibles dans le RCC\_M.

### 3.4.1 Opérande $K_T$

◇  $K_T = k_t$

$k_t$  est le coefficient de concentration de contraintes qui dépend de la géométrie de la pièce, de la géométrie d'un éventuel défaut et du type de chargement. Ce coefficient est utilisé pour appliquer à l'histoire de chargement "filtrée" une homothétie de rapport  $k_t$ .



## 3.5 Opérande COMPTAGE

♦ COMPTAGE =

Pour pouvoir calculer le dommage subi par une structure, il faut préalablement extraire les cycles élémentaires de l'histoire de chargement. Pour cela de nombreuses méthodes sont disponibles. Dans *Code\_Aster*, trois méthodes ont été programmées.

~ 'RAINFLOW' ,

Méthode de comptage des étendues en cascade ou méthode de RAINFLOW (recommandation AFNOR A03-406 de novembre 1993) pour l'extraction des cycles élémentaires de l'histoire de chargement [R7.04.01].

/ 'RCCM' ,

Méthode du RCC-M [R7.04.01].

/ 'NATUREL' ,

Méthode dite naturelle qui consiste à générer les cycles dans l'ordre de leur application [R7.04.01].

## 3.6 Opérande CORR\_KE

◇ CORR\_KE = 'RCCM' ,

Cet opérande permet de tenir compte d'un coefficient de concentration élasto-plastique,  $K_e$  qui est défini par le RCC-M comme étant le rapport entre l'amplitude de déformation réelle et l'amplitude de déformation déterminée par une analyse élastique.

$$\begin{cases} K_e = 1 & \text{si } D_s < 3S_m \\ K_e = 1 + (1-n) \left( \frac{\Delta \sigma}{3 \cdot S_m} - 1 \right) / (n(m-1)) & \text{si } 3S_m < D_s < 3mS_m \\ K_e = 1/n & \text{si } 3mS_m < D_s \end{cases}$$

où  $S_m$  est la contrainte maximale admissible et  $n$  et  $m$  deux constantes dépendant du matériau.

Les valeurs  $S_m$ ,  $n$  et  $m$  sont fournies dans l'opérateur `DEFI_MATERIAU` [U4.43.01] sous le mot clé `facteur FATIGUE` et les opérandes `SM_KE_RCCM`, `N_KE_RCCM` et `M_KE_RCCM`.

## 3.7 Opérande DOMMAGE

Pour calculer le dommage subi par une structure en un point, diverses méthodes sont disponibles [R7.04.01].

### 3.7.1 Méthodes basées sur des essais uniaxiaux : méthode de Wöhler, méthode de Manson-Coffin, méthodes de Taheri

Ces méthodes ont pour point commun de déterminer une valeur de dommage à partir de l'évolution au cours du temps d'une **composante scalaire** caractérisant l'état de contrainte ou de déformation de la structure.

Cela ne signifie pas que l'état de contraintes ne peut pas être multiaxial, mais seulement que pour le calcul du dommage on a choisi une composante uniaxiale caractérisant l'état de contrainte ou de déformation (contrainte de von Mises signée, invariant d'ordre 2 signé du tenseur des déformations, ...).

Les méthodes de Manson-Coffin et de Taheri utilisent les déformations engendrées par le chargement.

La méthode de Wöhler utilise les contraintes engendrées par le chargement.

◇ DOMMAGE = 'WOHLER' ,

Pour une histoire de contraintes associée à un chargement uniaxial, le nombre de cycles à la rupture est déterminé à l'aide de la courbe de Wöhler du matériau

$$\left( N_{rupt} = \text{WOHLER} \left( \frac{\Delta \sigma}{2} \right) \right) .$$

La courbe de Wöhler du matériau doit être introduite dans l'opérateur DEFI\_MATERIAU [U4.43.01] sous une des trois formes mathématiques [R7.04.01] :

- fonction discrétisée point par point (mot clé facteur FATIGUE, opérande WOHLER),
- forme analytique de Basquin (mot clé facteur FATIGUE, opérandes A\_BASQUIN et BETA\_BASQUIN),
- forme « zone courante » (mot clé facteur FATIGUE, opérandes E\_REFE, A0, A1, A2, A3 et SL et mot clé facteur ELAS opérande E).

## Remarque sur les courbes de fatigue :

*Pour les petites amplitudes de contraintes, le problème du prolongement de la courbe de fatigue peut se poser : par exemple, pour les courbes de fatigue du RCC-M au-delà de  $10^6$  cycles, la contrainte correspondante, 180 MPa est considérée comme limite d'endurance, c'est-à-dire que toute contrainte inférieure à 180 MPa doit produire un facteur d'usage nul ou un nombre de cycles admissible infini.*

*La méthode adoptée ici correspond à cette notion de limite d'endurance : si l'amplitude de contrainte est inférieure à la première abscisse de la courbe de fatigue, alors on prend un facteur d'usage nul c'est-à-dire un nombre de cycles admissible infini.*

◇ DOMMAGE = 'MANSON\_COFFIN' ,

Pour une histoire de chargement uniaxiale de type déformations, le nombre de cycles à la rupture est déterminé à l'aide de la courbe de Manson-Coffin du matériau

$$\left( N_{rupt} = \text{MANSON\_COFFIN} \left( \frac{\Delta \varepsilon}{2} \right) \right) .$$

La courbe de Manson-Coffin du matériau doit être introduite dans l'opérateur DEFI\_MATERIAU [U4.43.01] (mot clé facteur FATIGUE, opérande MANSON\_COFFIN).

◇ DOMMAGE = 'TAHERI\_MANSON' ,

Cette méthode de calcul du dommage ne s'applique qu'à des chargements en déformations.

Soient  $n$  cycles élémentaires (extraits par une méthode de comptage) de demi-amplitude

$$\frac{\Delta \varepsilon_1}{2}, \dots, \frac{\Delta \varepsilon_n}{2} .$$

La valeur du dommage élémentaire du premier cycle est déterminée par interpolation sur la courbe de Manson-Coffin du matériau.

Le calcul du dommage élémentaire des cycles suivants est effectué par l'algorithme décrit ci-dessous :

- Si  $\frac{\Delta \varepsilon_{i+1}}{2} \geq \frac{\Delta \varepsilon_i}{2}$

le calcul du dommage élémentaire du cycle  $(i+1)$  est déterminé par interpolation sur la courbe de Manson-Coffin du matériau,

- Si  $\frac{\Delta \varepsilon_{i+1}}{2} < \frac{\Delta \varepsilon_i}{2}$

on détermine :

$$\frac{\Delta \sigma_{i+1}}{2} = \text{Fnappe} \left( \frac{\Delta \varepsilon_{i+1}}{2}, \text{Max}_{j < i} \left( \frac{\Delta \varepsilon_j}{2} \right) \right)$$

$$\frac{\Delta \varepsilon_{i+1}^*}{2} = \text{Ffonc} \left( \frac{\Delta \sigma_{i+1}}{2} \right)$$

où :

Fnappe est une nappe introduite sous l'opérande TAHERI\_NAPPE,

Ffonc est une fonction introduite sous l'opérande TAHERI\_FONC.

La valeur du dommage du cycle  $(i+1)$  est obtenue par interpolation de  $\frac{\Delta \varepsilon_{i+1}^*}{2}$  sur la courbe de Manson-Coffin du matériau.

$N_{rupt_{i+1}}$  est le nombre de cycles à la rupture du cycle  $(i+1)$

$$N_{rupt_{i+1}} = \text{MANSON\_COFFIN} \left( \frac{\Delta \varepsilon_{i+1}^*}{2} \right)$$

et  $Dom_{i+1}$  est le dommage du cycle  $(i+1) = \frac{1}{N_{rupt_{i+1}}}$ .

La courbe de Manson-Coffin du matériau doit être introduite dans l'opérateur DEFI\_MATERIAU [U4.43.01] (mot clé facteur FATIGUE, opérande MANSON\_COFFIN).

◇ DOMMAGE = 'TAHERI\_MIXTE' ,

Cette méthode de calcul du dommage ne s'applique qu'à des chargements en déformations.

Soient  $n$  cycles élémentaires (extraits par une méthode de comptage) de demi-amplitude

$$\frac{\Delta \varepsilon_1}{2}, \dots, \frac{\Delta \varepsilon_n}{2}.$$

La valeur du dommage élémentaire du premier cycle est déterminée par interpolation sur la courbe de Manson-Coffin du matériau.

Le calcul du dommage élémentaire des cycles suivants est effectué par l'algorithme décrit ci-dessous :

- Si  $\frac{\Delta \varepsilon_{i+1}}{2} \geq \frac{\Delta \varepsilon_i}{2}$

le calcul du dommage élémentaire du cycle  $(i+1)$  est déterminé par interpolation sur la courbe de Manson-Coffin.

- Si  $\frac{\Delta \varepsilon_{i+1}}{2} < \frac{\Delta \varepsilon_i}{2}$

on détermine :

$$\frac{\Delta \sigma_{i+1}}{2} = \text{Fnappe} \left( \frac{\Delta \varepsilon_{i+1}}{2}, \text{Max}_{j < i} \left( \frac{\Delta \varepsilon_j}{2} \right) \right)$$

où Fnappe est une nappe introduite sous l'opérande de TAHERI\_NAPPE.

La valeur du dommage du cycle  $(i+1)$  est obtenue par interpolation de  $\frac{\Delta \sigma_{i+1}}{2}$  sur la courbe de Wöhler du matériau.

$N_{rupt_{i+1}}$  est le nombre de cycles à la rupture du cycle  $(i+1)$

$$N_{rupt_{i+1}} = \text{WOHLER} \left( \frac{\Delta \sigma_{i+1}}{2} \right)$$

et  $Dom_{i+1}$  est le dommage du cycle  $(i+1) = \frac{1}{N_{rupt_{i+1}}}$ .

Cette méthode nécessite les données de la courbe de Wöhler et de la courbe de Manson-Coffin du matériau qui doivent être introduites dans l'opérateur `DEFI_MATERIAU` [U4.43.01] (mot clé facteur `FATIGUE`).

## 3.7.2 Méthodes de Lemaître et Lemaître-Sermage

Ces deux méthodes permettent de calculer le dommage  $D(t)$  à partir de la donnée du tenseur des contraintes  $\sigma(t)$  et de la déformation plastique cumulée  $p(t)$ .

Elles s'appliquent donc à des chargements quelconques et ne s'utilisent qu'en post-traitement d'une loi plastique ou viscoplastique ayant  $p$  comme variable.

L'évolution de  $D$  est définie par :

$$\begin{cases} \dot{D} = \frac{1}{(1-D)^{2s}} \left( \frac{1}{3ES} \cdot (1+\nu) \sigma_{eq}^2 + \frac{3}{2ES} (1-2\nu) \cdot \sigma_H^2 \right)^s \dot{p} & \text{si } p > p_d \\ D = 0 & \text{sinon} \end{cases}$$

où  $E$  : module d'Young,  $\nu$  : coefficient de Poisson,  $S$  et  $s$  : paramètres matériau,  $\sigma_{eq}$  : contrainte équivalente de von Mises,  $\sigma_H$  : pression hydrostatique,  $p$  : déformation plastique cumulée et  $p_d$  : seuil d'endommagement.

◇ `DOMMAGE = 'LEMAITRE'` ,

Permet de calculer le dommage de Lemaître ou de Lemaître-Sermage  $D(t)$  à partir de la donnée du tenseur des contraintes  $\sigma(t)$  et de la déformation plastique cumulée  $p(t)$ .  
Noter que le dommage de Lemaître est obtenu en assignant la valeur 1.0 à l'exposant  $s$  ( $s=1$ ).

## 3.8 Opérande CORR\_SIGM\_MOYE

◇ CORR\_SIGM\_MOYE = / 'GOODMAN' ,  
/ 'GERBER' ,

Cet opérande n'est utilisé que dans le cas du calcul du dommage par la méthode de WOHLER.

Si la pièce n'est pas soumise à des contraintes alternées pures ou symétriques, c'est-à-dire si la contrainte moyenne du cycle n'est pas nulle, on peut pondérer la courbe de Wöhler pour calculer le nombre de cycles effectifs à la rupture à l'aide du diagramme de Haigh [R7.04.01].

A partir d'un cycle  $(S_{alt}, \sigma_m)$  identifié dans le signal, on calcule la valeur de la contrainte alternée corrigée  $S'_{alt}$ .

Si l'on utilise la droite de Goodman

$$S'_{alt} = \frac{S_{alt}}{1 - \frac{\sigma_m}{S_u}}$$

Si l'on utilise la parabole de Gerber

$$S'_{alt} = \frac{S_{alt}}{1 - \left(\frac{\sigma_m}{S_u}\right)^2}$$

La valeur de la limite à la rupture du matériau  $S_u$  doit être introduite dans l'opérateur DEFI\_MATERIAU [U4.43.01] (mot clé facteur RCCM, opérande Su).

## 3.9 Opérande TAHERI\_NAPPE

◇ TAHERI\_NAPPE = fnappe,

Cet opérande permet de spécifier le nom d'une nappe.

$F_{nappe} = \left( \frac{\Delta \varepsilon}{2}, \varepsilon_{max} \right)$  nécessaire au calcul du dommage par les méthodes TAHERI\_MANSON et TAHERI\_MIXTE.

La nappe doit avoir pour paramètres X et EPSI. Le paramètre X correspond à la déformation maximale atteinte lors d'un éventuel pré-écrouissage.

La nappe introduite sous l'opérande TAHERI\_NAPPE est la courbe d'écrouissage cyclique avec pré-écrouissage du matériau.

La courbe d'écrouissage cyclique sans pré-écrouissage, donnée sous le mot-clé TAHERI\_FONC, doit être obligatoirement une des courbes définissant la nappe. Cette courbe doit être donnée pour X = 0.

## 3.10 Opérande TAHERI\_FONC

◇ TAHERI\_FONC = ffonc,

Cet opérande permet de spécifier le nom d'une fonction  $F_{fonc} = \left( \frac{\Delta \sigma}{2} \right)$  nécessaire au calcul du dommage par la méthode TAHERI\_MANSON.

Le paramètre de cette fonction doit être SIGM.

Cette fonction est la courbe d'écrouissage cyclique du matériau.

## 3.11 Opérande MATER

◆ MATER = mater,

Permet de spécifier le nom du matériau `mater` créé par `DEFI_MATERIAU` [U4.43.01].

Le matériau `mater` doit contenir les valeurs de toutes les données matériaux nécessaires au calcul du dommage.

## 3.12 Opérande CUMUL

◇ `CUMUL = 'LINEAIRE' ,`

Les méthodes de `WOHLER`, `MANSON_COFFIN` et `TAHERI` calculent une valeur de dommage pour chaque cycle élémentaire extrait du chargement uniaxial introduit par l'utilisateur.

L'opérande `CUMUL` permet de demander le calcul du dommage total subi par la structure en un point.

La seule règle disponible est la règle de Miner, qui consiste à sommer tous les dommages élémentaires  $D = \sum_i D_i$ .

## 3.13 Opérands CRITERE

◇ `CRITERE = / 'CROSSLAND' ,`  
`/ 'PAPADOPOULOS' ,`

Les critères de Crossland et Dang Van-Papadopoulos s'appliquent sur des chargements uniaxiaux ou multiaxiaux **périodiques**.

L'utilisateur introduit les valeurs de chaque composante du tenseur des contraintes en divers instants  $(t_0, \dots, t_N)$ , et on suppose que  $[t_0, t_N]$  est une période du chargement.

Les chargements doivent être en contraintes.

Le but de ces critères n'est pas de déterminer une valeur de dommage, mais une valeur de critère  $R_{crit}$  telle que :

$$\begin{cases} R_{crit} \leq 0 & \text{pas de dommage} \\ R_{crit} > 0 & \text{dommage possible} \end{cases}$$

On peut cependant déterminer une valeur de dommage par extension [§15].

### 3.13.1 Critère de Crossland

Le critère s'écrit:

$$R_{crit} = \tau_a + a \cdot P_{max} - b$$

où

$$\tau_a = \frac{1}{2} \max_{0 \leq t_0 \leq T} \max_{0 \leq t_1 \leq T} \|\tilde{S}(t_1) - \tilde{S}(t_0)\| \quad \text{est l'amplitude de cisson}$$

avec  $\tilde{S}$  déviateur du tenseur des contraintes  $\sigma$

$$P_{max} = \max_{0 \leq t \leq T} \left( \frac{1}{3} \text{trace } \sigma \right) \quad \text{est la pression hydrostatique maximale}$$

$$a = \frac{\left( \tau_0 - \frac{d_0}{\sqrt{3}} \right)}{\frac{d_0}{3}} \quad \text{et} \quad b = \tau_0$$

avec  $\tau_0$  la limite d'endurance en cisaillement pur alterné

et  $d_0$  la limite d'endurance en traction-compression pure alternée

### 3.13.2 Critère de Dang Van-Papadopoulos

Le critère s'écrit:

$$R_{crit} = k^* + a \cdot P_{max} - b$$

où

$k^* = R$   
 $R$  rayon de la plus petite sphère circonscrite au trajet de chargement dans l'espace des déviateurs de contraintes  $\tilde{S}$

$$R = \text{Max}_{0 \leq t \leq T} \sqrt{\frac{1}{2} \cdot (\tilde{S}(t) - C^*) : (\tilde{S}(t) - C^*)}$$

$C^* = \text{Min Max} \sqrt{(\tilde{S}(t) - C) : (\tilde{S}(t) - C)}$  est le centre de l'hypersphère

$P_{\max} = \text{Max}_{0 \leq t \leq T} \left( \frac{1}{3} \text{trace } \sigma \right)$  est la pression hydrostatique maximale

$$a = \frac{\left( \tau_0 - \frac{d_0}{\sqrt{3}} \right)}{\frac{d_0}{3}} \quad \text{et} \quad b = \tau_0$$

avec  $\tau_0$  la limite d'endurance en cisaillement pur alterné

et  $d_0$  la limite d'endurance en traction-compression pure alternée

### 3.13.3 Opérateur COEF\_CORR

◇ COEF\_CORR = corr ,

Les critères de Crossland et Dang Van-Papadopoulos permettent pour des chargements périodiques de calculer une valeur  $R_{\text{crit}}$  qui indique s'il y a dommage ou non pour le nombre de cycles associé aux limites d'endurances  $\tau_0$  et  $d_0$ .

Ces critères ne donnent pas de valeur du dommage, ce qui peut cependant être intéressant.

Pour ce faire, on propose d'utiliser la valeur du critère et la courbe de Wöhler du matériau, en définissant une contrainte équivalente:

$$\sigma^* = (R_{\text{crit}} + b) \times \text{corr}$$

La plupart des courbes de Wöhler sont obtenues avec des essais de traction-compression pure alternée. Or le critère de Dang-Van-Papadopoulos est un critère en cisaillement. Par conséquent, il est nécessaire de « corriger » la contrainte équivalente  $\sigma^*$  avant de l'appliquer sur une courbe de Wöhler obtenue avec des essais de traction-compression ; c'est le rôle de l'opérateur COEF\_CORR.

La valeur du dommage est obtenue en appliquant  $\sigma^*$  sur la courbe de Wöhler du matériau.

Pour qu'il y ait cohérence entre le critère et la courbe de Wöhler, il faut que :

$$\begin{cases} \sigma^* \leq \tau_0 & \text{pas de dommage} \\ \sigma^* > \tau_0 & \text{dommage} \end{cases}$$

pour une courbe de Wöhler définie en cisaillement et que :

$$\begin{cases} \sigma^* \leq d_0 & \text{pas de dommage} \\ \sigma^* > d_0 & \text{dommage} \end{cases}$$

pour une courbe de Wöhler définie en traction-compression.

L'utilisateur peut donc spécifier une valeur `corr`, en tenant compte du type de courbe de Wöhler dont il dispose. La valeur prise par défaut pour `COEF_CORR` est  $\frac{d_0}{\tau_0}$ , en cohérence avec des courbes de Wöhler en traction-compression.

#### Remarque :

Dans le cas où  $R_{crit} < 0$ , si le prolongement à gauche de la courbe de Wöhler est linéaire (dans `DEFI_FONCTION (... PROL_GAUCHE = 'LINEAIRE' ...)`), l'utilisateur obtiendra un dommage différent de zéro. Pour obtenir un dommage nul quand  $R_{crit} < 0$ , il faut que le prolongement à gauche soit égal à 'EXCLU' ou 'CONSTANT'.

## 3.14 Opérande INFO

◇ INFO = / 1,

Impression :

- des cycles élémentaires déterminés par la méthode de comptage choisie par l'utilisateur,
- des dommages élémentaires associés à chaque cycle pour les méthodes WOHLER, MANSON\_COFFIN et TAHERI,
- des dommages de LEMAITRE en chaque point de calcul,
- du dommage total (si l'utilisateur a demandé son calcul).

◇ INFO = / 2,

Impression :

- de l'histoire de chargement introduite par l'utilisateur sous les opérandes SIGM et EPSI,
- des pics extraits de l'histoire de chargement (introduit sous les opérandes SIGM et EPSI),
- des cycles élémentaires déterminés par la méthode de comptage choisie par l'utilisateur,
- des dommages élémentaires associés à chaque cycle pour les méthodes WOHLER, MANSON\_COFFIN et TAHERI,
- des dommages de LEMAITRE en chaque point de calcul,
- du dommage total (si l'utilisateur a demandé son calcul).

Les impressions sont faites dans le fichier message.

## 3.15 Opérande TITRE

◇ TITRE = titre

Titre associé à la table.

## 3.16 Table produite

L'opérateur POST\_FATIGUE crée une table qui est différente suivant les calculs de post-traitement effectués :

- **Chargement uniaxial** (méthodes Wöhler, Manson-Coffin et Taheri).

La table comprend cinq paramètres :

NB_CYCL	:	nombre de cycles élémentaires extraits par la méthode de comptage,
VALE_MIN	:	valeurs des contraintes ou déformations minimales de chaque cycle élémentaire,
VALE_MAX	:	valeurs des contraintes ou déformations maximales de chaque cycle élémentaire,
DOMMAGE	:	valeurs du dommage pour chaque cycle élémentaire,
DOMM_CUMU	:	valeur du dommage total après cumul sur tous les cycles élémentaires.

- **Chargement périodique** (critères de Crossland et Dang Van-Papadopoulos).

La table comprend cinq paramètres :

CRITERE	:	valeur du critère
PRES_HYDRO_MAX	:	valeur de la pression hydrostatique maximale,



AMPLI\_CISSION : valeur de l'amplitude de cission  $\tau_a$  (si calculée),  
RAYON\_SPHERE : valeur du rayon de la sphère circonscrite au chargement  
k\* (si calculée),  
DOMMAGE : valeur du dommage de Wöhler (si demandé par l'utilisateur).

- **Chargement quelconque** (dommage de Lemaître et Lemaître-Sermage).

La table comprend deux paramètres :

DOMMAGE : valeur du dommage en chaque point de discrétisation  
du chargement,  
D\_CUMULE : valeur du dommage cumulé (si demandé par l'utilisateur).

La commande `IMPR_TABLE` [U4.91.03] permet d'imprimer la table produite.

## 4 Exemples

### 4.1 Calcul du dommage de Wöhler (avec correction de la contrainte moyenne)

```
# Définition du chargement
taun = DEFI_FONCTION (
    NOM_PARA = 'INST' ,
    VALE      = ( 0. ,      50. ,
                  1. ,      600. ,
                  2. ,      50. ,
                  3. ,      - 500. ,
                  4. ,      50. ,))

# Définition de la courbe de Wöhler
wohl = DEFI_FONCTION (
    NOM_PARA      = 'SIGM',
    INTERPOL      = 'LOG' ,
    PROL_DROIT    = 'LINEAIRE' ,
    PROL_GAUCHE   = 'LINEAIRE' ,
    VALE          = (138. , 1000000. ,
                    :      :
                    .      .
                    2900., 10.,))

# Définition du Matériau
mat = DEFI_MATERIAU (
    FATIGUE = _F (WOHLER = wohl)
    RCCM    = _F (SU      = 850. ))

# Calcul de dommage
TABLE = POST_FATIGUE (
    CHARGEMENT = 'UNIAXIAL' ,
    HISTOIRE   = _F ( SIGM = taun),
    COMPTAGE   = 'RCCM' ,
    MATER      = mat ,
    DOMMAGE    = 'WOHLER' ,
    CUMUL      = 'LINEAIRE' ,
    INFO       = 2          )
```

Cet exemple est issu du test SZLZ100 (voir [V9.01.100]).

## 4.2 Calcul du dommage de Taheri

```
# Définition du chargement
taun = DEFI_FONCTION (
    NOM_PARA      = 'INST' ,
    PROL_DROITE   = 'LINEAIRE' ,
    PROL_GAUCHE   = 'LINEAIRE' ,
    VALE          = ( 0. ,      0. ,
                      1. ,      3.5,
                      2. ,      3. ,
                      3. ,      3.5,
                      4. ,      3. ,
                      5. ,      3.5,
                      6. ,      1. ,
                      7. ,      2.5,
                      8. ,      0. ,
                      9. ,      0.5, ))

# Définition de la fonction Ffunc : courbe d'écrouissage cyclique du
# matériau
f_eps = DEFI_FONCTION (
    NOM_PARA      = 'SIGM' ,
    PROL_DROITE   = 'LINEAIRE' ,
    PROL_GAUCHE   = 'LINEAIRE' ,
    VALE          = ( 0. ,      0.,
                      1000.,    10.,)
    TITRE         = 'Fonction de Taheri' )

# Définition de la nappe Fnappe : courbe d'écrouissage cyclique avec pré-
# contrainte du matériau
f_epsmx = DEFI_NAPPE (
    NOM_PARA      = 'X' ,
    PROL_DROITE   = 'LINEAIRE' ,
    PROL_GAUCHE   = 'LINEAIRE' ,
    PARA          = (0.5 ,      1.,),
    NOM_PARA_FONC = 'EPSI' ,
    DEFI_FONCTION = (
        _F( PROL_DROITE = 'LINEAIRE' ,
            PROL_GAUCHE = 'LINEAIRE' ,
            VALE        = (0. ,      25.,
                          10. ,    525.,)),
        _F ( PROL_DROITE = 'LINEAIRE' ,
            PROL_GAUCHE = 'LINEAIRE' ,
            VALE        = ( 0. ,      50.,
                          10. ,    550.,      )),
    TITRE = 'NAPPE DE TAHERI' )

# Définition de la courbe de Wöhler
f_wohl = DEFI_FONCTION (
    NOM_PARA      = 'SIGM' ,
    PROL_DROITE   = 'LINEAIRE' ,
    PROL_GAUCHE   = 'LINEAIRE' ,
    VALE          = (0. ,      200000.,
                      200. ,    0.,),
    TITRE = 'FONCTION DE WOHLER' )

# Définition de la courbe de Manson-Coffin
f_mans = DEFI_FONCTION (
    NOM_PARA      = 'EPSI' ,
```

```
        PROL_DROITE = 'LINEAIRE' ,
        PROL_GAUCHE = 'LINEAIRE' ,
        VALE        = (0. ,      200000.,
                        2. ,      0.),
        TITRE = 'FONCTION DE MANSO-COFFIN' )

# Définition du matériau

mat0 = DEFI_MATERIAU (
        FATIGUE = _F (WOHLER      = f_who1,
                      MANSO-COFFIN = f_mans))

# Calcul du dommage
tabl1 = POST_FATIGUE (
        CHARGEMENT      = 'UNIAXIAL' ,
        HISTOIRE         = _F (EPSI = taun),
        COMPTAGE         = 'RAINFLOW' ,
        DOMMAGE          = 'TAHERI_MANSO' ,
        TAHERI_FONC      = f_eps,
        TAHERI_NAPPE     = f_epsmx,
        MATER            = mat0,
        CUMUL            = 'LINEAIRE' ,
        INFO             = 2
)

tabl2 = POST_FATIGUE (
        CHARGEMENT      = 'UNIAXIAL' ,
        HISTOIRE         = _F (EPSI = taun),
        COMPTAGE         = 'RAINFLOW' ,
        DOMMAGE          = 'TAHERI_MIXTE' ,
        TAHERI_NAPPE     = f_epsmx,
        MATER            = mat0,
        CUMUL            = 'LINEAIRE' ,
        INFO             = 2
)
```

Cet exemple est issu du test SZLZ108 (voir [V9.01.108]).

## 4.3 Calcul des critères de Crossland et Dang Van-Papadopoulos

```
# Définition du chargement
taun1 = DEFI_FONCTION (
        NOM_PARA = 'INST' ,
        VALE     = (1. ,   411.,
                    2. ,    0.,
                    3. ,  -411.,))

taun2 = DEFI_FONCTION (
        NOM_PARA = 'INST' ,
        VALE     = (1. ,   205.,
                    2. ,    0.,
                    3. ,  -205.,))

taun3 = DEFI_FONCTION (
        NOM_PARA = 'INST' ,
        VALE     = (1. ,    0.,
                    2. ,    0.,
                    3. ,    0.,))

# Définition du matériau
mat0 = DEFI_MATERIAU (
```

```
FATIGUE = _F (WOHLER = whol,  
              D0    = 540.97,  
              TAU0  = 352., ))  
  
# Calcul du critère de Crossland  
table1 = POST_FATIGUE (  
    CHARGEMENT = 'PERIODIQUE' ,  
    HISTOIRE    = _F ( SIGM_XX = taun1 ,    SIGM_XY = taun2,  
                      SIGM_YY = taun3 ,    SIGM_XZ = taun3,  
                      SIGM_ZZ = taun3 ,    SIGM_YZ = taun3 ),  
  
    CRITERE = 'CROSSLAND' ,  
    MATER   = mat0,  
    INFO    = 2  
)  
  
# Calcul du critère de Dang Van-Papadopoulos  
table2 = POST_FATIGUE (  
    CHARGEMENT = 'PERIODIQUE' ,  
    HISTOIRE    = _F ( SIGM_XX = taun1,  
                      SIGM_YY = taun3,  
                      SIGM_ZZ = taun3,  
                      SIGM_XY = taun2,  
                      SIGM_XZ = taun3,  
                      SIGM_YZ = taun3 ),  
  
    CRITERE = 'PAPADOPOULOS' ,  
    MATER   = mat0,  
    INFO    = 2  
)
```

Cet exemple est issu du test SZLZ107 (voir [V9.01.107]).

## 4.4 Calcul du dommage de Lemaître

```
# Définition du chargement  
taun1 = DEFI_FONCTION (  
    NOM_PARA = 'INST' ,  
    VALE     = (    43.11 ,    300.,  
                  100.   ,    300.,  
                  1000.  ,    300.,  
                  10000. ,    300.,  
                  20000. ,    300.,  
                  21000. ,    300.,  
                  22000. ,    300.,  
                  22200. ,    300.,  
                  22400. ,    300., ))  
  
taun2 = DEFI_FONCTION (  
    NOM_PARA = 'INST' ,  
    VALE     = (    43.11 ,    0.,  
                  100.   ,    0.,  
                  1000.  ,    0.,  
                  10000. ,    0.,  
                  20000. ,    0.,  
                  21000. ,    0.,  
                  22000. ,    0.,  
                  22200. ,    0.,  
                  22400. ,    0., ))  
  
t_epsp = DEFI_FONCTION (  
    NOM_PARA = 'INST' ,
```

```

                VALE      = (      43.11 ,                0.019996,
                                100. ,                0.046384,
                                1000. ,                0.46384 ,
                                10000. ,                4.6384 ,
                                20000. ,                9.2768 ,
                                21000. ,                9.74064 ,
                                22000. ,                10.20448 ,
                                22200. ,                10.297248,
                                22400. ,                10.390016, ))

t_temp = DEFI_FONCTION (
    NOM-PARA = 'INST' ,
    VALE      = (      43.11 ,                20.,
                                100. ,                20.,
                                1000. ,                20.,
                                10000. ,                20.,
                                20000. ,                20.,
                                21000. ,                20.,
                                22000. ,                20.,
                                22200. ,                20.,
                                22400. ,                20., ))

t_e = DEFI_FONCTION (
    NOM_PARA = 'TEMP' ,
    VALE      = ( 20. ,                2.E+5, ),
    PROL_DROITE = 'CONSTANT' ,
    PROL_GAUCHE = 'CONSTANT' ,)

t_nu = DEFI_FONCTION (
    NOM_PARA = 'TEMP' ,
    VALE      = ( 20. ,                0., ),
    PROL_DROITE = 'CONSTANT' ,
    PROL_GAUCHE = 'CONSTANT' )

t_s = DEFI-FONCTION (
    NOM_PARA = 'TEMP' ,
    VALE      = ( 20. ,                7., ),
    PROL_DROITE = 'CONSTANT' ,
    PROL_GAUCHE = 'CONSTANT' ,)

epsp_s = DEFI-FONCTION (
    NOM_PARA = 'TEMP' ,
    VALE      = ( 20. ,                0.02, ),
    PROL_DROITE = 'CONSTANT' ,
    PROL_GAUCHE = 'CONSTANT' ,)

mat1 = DEFI_MATERIAU (
    ELAS_FO = _F      ( E                = t_e,
                        NU                = t_nu ),
    DOMMA_LEMAITRE = _F ( S                = t_s,
                        EPSP_SEUIL        = epsp_s,
                        EXP_S              = 1.0 )
    )

TAB_1 = POST_FATIGUE (
    CHARGEMENT = 'QUELCONQUE' ,
    HISTOIRE = _F ( SIGM_XX = taun1 ,    SIGM_XY = taun2,
                    SIGM_YY = taun2 ,    SIGM_XZ = taun2,
                    SIGM_ZZ = taun2 ,    SIGM_YZ = taun2,
                    EPSP     = t_epsp,
                    TEMP     = t_temp ),

    MATER      = mat1,

```

```
DOMMAGE = 'LEMAITRE' ,
INFO     = 2                                )
```

Cet exemple est issu du test SZLZ109 (voir [V9.01.109]).

## 4.5 Calcul du dommage de Lemaître-Sermage

```
TAUN1=DEFI_FONCTION( NOM_PARA='INST',
                     VALE=( 43.11,          300.,
                           100.,           300.,
                           1000.,          300.,
                           10000.,         300.,
                           20000.,         300.,
                           21000.,         300.,
                           22000.,         300.,
                           22200.,         300.,
                           22400.,         300., ) )
```

```
TAUN2=DEFI_FONCTION( NOM_PARA='INST',
                     VALE=( 43.11,          0.,
                           100.,           0.,
                           1000.,          0.,
                           10000.,         0.,
                           20000.,         0.,
                           21000.,         0.,
                           22000.,         0.,
                           22200.,         0.,
                           22400.,         0., ) )
```

```
T_EPSP=DEFI_FONCTION( NOM_PARA='INST',
                     VALE=( 43.11,          0.019996,
                           100.,           0.046384,
                           1000.,          0.46384,
                           10000.,         4.6384,
                           20000.,         9.2768,
                           21000.,         9.74064,
                           22000.,         10.20448,
                           22200.,         10.297248,
                           22400.,         10.390016, ) )
```

```
T_TEMP=DEFI_FONCTION( NOM_PARA='INST',
                     VALE=( 43.11,          20.,
                           100.,           20.,
                           1000.,          20.,
                           10000.,         20.,
                           20000.,         20.,
                           21000.,         20.,
                           22000.,         20.,
                           22200.,         20.,
                           22400.,         20., ) )
```

```
T_E=DEFI_FONCTION(NOM_PARA='TEMP',
                  PROL_DROITE='CONSTANT',
                  PROL_GAUCHE='CONSTANT',
                  VALE=( 20.,          2.E+5, ) )
```

```
T_NU=DEFI_FONCTION(NOM_PARA='TEMP',
                   PROL_DROITE='CONSTANT',
                   PROL_GAUCHE='CONSTANT',
```

```
VALE=( 20.,      0.,      )      )

T_S=DEFI_FONCTION(NOM_PARA='TEMP',
                  PROL_DROITE='CONSTANT',
                  PROL_GAUCHE='CONSTANT',
                  VALE=( 20.,      7.0,      )      )

EPSP_S=DEFI_FONCTION(NOM_PARA='TEMP',
                    PROL_DROITE='CONSTANT',
                    PROL_GAUCHE='CONSTANT',
                    VALE=( 20.,      0.02,      )      )

MAT2=DEFI_MATERIAU(ELAS_FO=_F(E = T_E,
                             NU = T_NU, ),
                  DOMMA_LEMAITRE=_F(S = T_S,
                                     EPSP_SEUIL = EPSP_S,
                                     EXP_S = 1.003, ), );

TAB_2=POST_FATIGUE(CHARGEMENT='QUELCONQUE',
                  HISTOIRE=_F(SIGM_XX = TAUN1,
                             SIGM_YY = TAUN2,
                             SIGM_ZZ = TAUN2,
                             SIGM_XY = TAUN2,
                             SIGM_XZ = TAUN2,
                             SIGM_YZ = TAUN2,
                             EPSP = T_EPSP,
                             TEMP = T_TEMP, ),
                  MATER=MAT2,
                  DOMMAGE='LEMAITRE',
                  CUMUL='LINEAIRE', );
```

Cet exemple est issu du test SZLZ109 (voir [V9.01.109]).

On peut trouver d'autres exemples dans les tests :

SZLZ101 ([V9.01.101]) : Calcul du dommage / méthode Rainflow.  
SZLZ102 ([V9.01.102]) : Fatigue avec différentes méthodes comptage.  
SZLZ103 ([V9.01.103]) : Fatigue comptage par méthode Rainflow norme AFNOR.