

**Manuel d'Utilisation**  
**Fascicule U4.3- : Fonction**  
**Document : U4.33.03**

## Procédure *CALC\_TABLE*

---

### 1 But

---

Manipuler les données des *tables* à la manière d'un tableur.

La commande permet d'effectuer des opérations sur les données des *tables*. Les opérations suivantes sont actuellement disponibles :

- Concaténer / Combiner deux *tables*,
- Appliquer une formule,
- Renommer des paramètres,
- Filtrer les lignes selon certains critères,
- Extraire certaines colonnes d'une *table*,
- Ordonner les lignes,

Produit une structure de données *table*.

## Table des matières

1 But .....	1
2 Syntaxe .....	3
3 Opérandes .....	4
3.1 Opérande <i>TABLE</i> .....	4
3.2 Opérande <i>COMB</i> : concaténer/combiner deux tables .....	4
3.2.1 Opérande <i>TABLE</i> .....	4
3.2.2 Opérande <i>NOM_PARA</i> .....	4
3.3 Opérande <i>OPER</i> : Appliquer une formule .....	4
3.3.1 Opérande <i>FORMULE</i> .....	4
3.3.2 Opérande <i>NOM_PARA</i> .....	4
3.4 Opérande <i>RENOMME</i> : Renommer des paramètres d'une table .....	5
3.4.1 Opérande <i>NOM_PARA</i> .....	5
3.5 Opérande <i>FILTRE</i> : Filtrer les lignes selon certains critères .....	5
3.5.1 Opérande <i>NOM_PARA</i> .....	5
3.5.2 Opérande <i>CRIT_COMP</i> .....	5
3.5.3 Opérandes <i>VALE</i> / <i>VALE_I</i> / <i>VALE_C</i> / <i>VALE_K</i> .....	6
3.5.4 Opérandes <i>CRITERE</i> / <i>PRECISION</i> .....	6
3.5.5 Exemple d'utilisation du mot clé <i>FILTRE</i> .....	6
3.6 Opérande <i>EXTR</i> : Extraire certaines colonnes d'une table .....	6
3.6.1 Opérande <i>NOM_PARA</i> .....	6
3.7 Opérande <i>TRI</i> : Ordonner les lignes .....	6
3.7.1 Opérande <i>NOM_PARA</i> .....	7
3.7.2 Opérande <i>ORDRE</i> .....	7
3.8 Opérande <i>INFO</i> .....	7
4 Exemples .....	8

## 2 Syntaxe

```

tab = CALC_TABLE (
    ◇ reuse = tab, [tabl_*]
    ◆ TABLE = matable, [tabl_*]
    ◇ TITRE = titre [l_Kn]

# 1. Concaténer/combiner deux tables :
    ◇ COMB =_F( ◆ TABLE = table, [tabl_*]
                ◇ NOM_PARA = l_para, [l_Kn]
                ),

# 2. Appliquer une formule :
    ◇ OPER =_F( ◆ FORMULE = formule, [formule]
                ◇ NOM_PARA = para, [Kn]
                ),

# 3. Renommer des paramètres d'une table :
    ◇ RENOMME =_F( ◆ NOM_PARA = l_para, [l_Kn]
                   ),

# 4. Filtrer des lignes :
    ◇ FILTRE =_F( ◆ NOM_PARA = para, [Kn]
                  ◇ / CRIT_COMP = / 'EQ', [DEFAULT]
                  / 'NE',
                  / 'LT',
                  / 'GT',
                  / 'LE',
                  / 'GE',
                  ◆ / VALE_I = ival, [I]
                  / VALE_K = kval, [Kn]
                  / / VALE = rval, [R]
                  / VALE_C = cval, [C]
                  ◇ | PRECISION = / prec, [R8]
                  / 1.0D-3, [DEFAULT]
                  | CRITERE = / 'RELATIF', [DEFAULT]
                  / 'ABSOLU',
                  / CRIT_COMP = / 'MAXI',
                  / 'ABS_MAXI',
                  / 'MINI',
                  / 'ABS_MINI',
                  / 'VIDE',
                  / 'NON_VIDE',
                  ),

# 5. Extraire certaines colonnes :
    ◇ EXTR =_F( ◆ NOM_PARA = l_para, [l_Kn]
                ),

# 6. Ordonner les lignes selon les valeurs d'un paramètre :
    ◇ TRI =_F( ◆ NOM_PARA = lpara, [l_Kn]
               ◇ ORDRE = / 'CROISSANT', [DEFAULT]
               / 'DECROISSANT',
               ),

    ◇ INFO = / 1, [DEFAULT]
              / 2,
              )

```

## 3 Opérandes

### 3.1 Opérande `TABLE`

- ◆ `TABLE` = `matable`  
Nom de la table initiale sur laquelle on va effectuer des manipulations

### 3.2 Opérande `COMB` : concaténer/combiner deux tables

/ `COMB`

Le mot-clé `COMB` permet de concaténer, de combiner deux tables entre elles, avec couture sur une liste de paramètres communs.

#### 3.2.1 Opérande `TABLE`

- ◆ `TABLE` = `table`  
Nom de la table dont les valeurs doivent venir surcharger et/ou enrichir la table initiale.

#### 3.2.2 Opérande `NOM_PARA`

- ◇ `NOM_PARA` = `l_para`  
Nom des paramètres dont les valeurs doivent être identiques dans les deux tables.

**Remarques pour la combinaison :**

```
Ex: tab_resu=CALC_TABLE(TABLE=tab1,  
                        COMB=_F(TABLE=tab2,  
                                NOM_PARA=('ABSC_CURV', 'NOEUD') ) )
```

Quand `ABSC_CURV` et `NEUD` sont identiques entre `tab1` et `tab2`, on insère les valeurs de `tab2` sur la ligne de `tab1` (pour les autres paramètres communs aux 2 tables, c'est donc la valeur de `tab2` qui écrase celle de `tab1`). Sinon on ajoute la ligne de `tab2` à la fin de `tab1`. `NOM_PARA` agit comme une clé primaire : si on ne trouve pas plus d'une fois le couple (`ABSC_CURV`, `NEUD`), on ajoute la ligne.

Cet opérateur n'est pas répétable car l'ordre des occurrences du mot-clé facteur pourrait modifier le résultat final.

### 3.3 Opérande `OPER` : Appliquer une formule

/ `OPER`

Permet d'appliquer une formule dont les variables sont les paramètres de la table et d'insérer le résultat dans une nouvelle colonne.

**Remarque :**

| Cet opérateur est répétable.

#### 3.3.1 Opérande `FORMULE`

- ◆ `FORMULE` = `formule`  
Nom de la formule à appliquer

#### 3.3.2 Opérande `NOM_PARA`

- ◇ `NOM_PARA` = `para`  
Nom du nouveau paramètre.

### 3.4 Opérande RENOMME : Renommer des paramètres d'une table

/ RENOMME

Le mot-clé **RENOMME** permet de renommer un ou plusieurs paramètres d'une table.

#### 3.4.1 Opérande NOM\_PARA

◆ NOM\_PARA = l\_para

Nom des paramètres : couple de valeurs (ancien nom du paramètre, nouveau nom du paramètre)

**Remarque :**

| Cet opérateur est répétable.

### 3.5 Opérande FILTRE : Filtrer les lignes selon certains critères

/ FILTRE

Le mot clé facteur **FILTRE** permet de ne retenir dans la table que les lignes vérifiant certains critères imposés par l'utilisateur. Les **occurrences** du mot clé s'ajoutent les unes aux autres comme des **filtres successifs**. Pour chaque occurrence de ce mot clé, on précise le nom du paramètre pour lequel on impose une condition, le type de condition (égalité, non-égalité, plus petit, ..) ainsi que la valeur associée à la condition.

**Remarque :**

| Cet opérateur est répétable.

#### 3.5.1 Opérande NOM\_PARA

◇ NOM\_PARA = para

para est le nom du paramètre sur lequel porte la contrainte de filtrage.

#### 3.5.2 Opérande CRIT\_COMP

◇ CRIT\_COMP = crit

crit est le type de la contrainte de filtrage.

EQ	"égalité" pour les entiers, les textes, les réels ou les complexes. Pour les nombres flottants (réels ou complexes), cette égalité est évaluée avec une certaine tolérance donnée par les mots clés PRECISION et CRITERE.
NE	"non-égalité" (Cf. EQ)
LT	"plus petit que" Relations d'ordre : - naturelle pour les entiers et les réels - alphabétique pour les textes - <b>invalide</b> pour les <b>complexes</b>
GT	"plus grand que" (Cf. LT)
LE	"plus petit ou égal à" (Cf. LT)
GE	"plus grand ou égal à" (Cf. LT)
VIDE	cellule vide
NON_VIDE	cellule non vide
MAXI	la ligne retenue sera celle qui aura la valeur maximale
ABS_MAXI	la ligne retenue sera celle qui aura la valeur absolue maximale
MINI	la ligne retenue sera celle qui aura la valeur minimale
ABS_MINI	la ligne retenue sera celle qui aura la valeur absolue minimale

### 3.5.3 Opérandes VALE / VALE\_I / VALE\_C / VALE\_K

Ces différents mots clés sont utilisés en fonction du type de la colonne associée au paramètre sur lequel porte la contrainte : entier, réel, complexe ou texte.

On donne en argument la valeur associée à la contrainte ; par exemple 12 si on ne s'intéresse qu'au numéro d'ordre 12.

Cet opérande est inutile quand on utilise les types de contrainte : 'VIDE', 'NON\_VIDE', 'MAXI', 'MINI', ...

### 3.5.4 Opérandes CRITERE / PRECISION

Lorsque le paramètre contraint est de type flottant (réel ou complexe) et que le type de contrainte est l'égalité (ou la non-égalité), cette égalité est évaluée avec une certaine tolérance.

```
PRECISION = eps,          eps est la tolérance
CRITERE=   / 'RELATIF' ,   |x-xref| < eps*|xref|
           / 'ABSOLU'  ,   |x-xref| < eps
```

### 3.5.5 Exemple d'utilisation du mot clé FILTER

Si l'on précise :

```
FILTER= ( _F(NOM_PARA= 'NOEUD', VALE_K = 'N7',),
          _F(NOM_PARA= 'INST' , CRIT_COMP= 'GT' , VALE= 3.0,),
          _F(NOM_PARA= 'INST' , CRIT_COMP= 'LT' , VALE= 13.0,),
        ),
```

On sélectionne ainsi les lignes de la table telles que le paramètre NOEUD vaut 'N7' et telles que le paramètre INST soit compris entre 3. et 13.

## 3.6 Opérande EXTR : Extraire certaines colonnes d'une table

/ EXTR

Le mot-clé EXTR permet d'extraire certaines colonnes d'une table.

### 3.6.1 Opérande NOM\_PARA

- ♦ NOM\_PARA = l\_para  
Nom des paramètres que l'on veut extraire.

Remarque :

| Cet opérateur n'est pas répétable.

## 3.7 Opérande TRI : Ordonner les lignes

/ TRI

Le mot-clé TRI permet d'ordonner les lignes selon les valeurs des paramètres.

Remarque :

| Cet opérateur n'est pas répétable car l'ordre des occurrences du mot-clé facteur pourrait modifier le résultat final.

## 3.7.1 Opérande **NOM\_PARA**

- ♦ `NOM_PARA = l_para`  
Noms des paramètres sur lesquelles porte le tri.

## 3.7.2 Opérande **ORDRE**

- ♦ `ORDRE = / 'CROISSANT'`  
`/ 'DECROISSANT'`

Ce mot clé sert à préciser si on doit utiliser un ordre croissant ou décroissant. Par défaut, on trie par ordre croissant.

Les relations d'ordre utilisées sont :

- l'ordre naturel pour les entiers et les réels,
- l'ordre alphabétique pour les textes et les noms de concepts.

### Remarques :

*On ne peut pas se servir d'un paramètre complexe pour classer les lignes d'une table.  
Pour les paramètres de type *NOEUD* (ou *MAILLE*), l'ordre est alphabétique car ces paramètres contiennent le nom des nœuds (ou des mailles).*

Si l'on précise :

```
TRI=_F (NOM_PARA= ('NOEUD', 'INST'), ORDRE= 'CROISSANT'),
```

On triera les lignes de la table dans l'ordre alphabétique des nœuds. S'il existe plusieurs lignes correspondant à un nœud donné, le **second** critère de tri (*INST*) sera utilisé pour classer ces lignes.

## 3.8 Opérande **INFO**

- ♦ `INFO = inf`

Imprime dans le fichier "message" des informations complémentaires si `inf=2`. Rien ne se passe si `inf=1`.

## 4 Exemples

Table 'TB1'			
N	Y	Z	NOEUD
0	2.0	4.5	N01
1	4.0	17.5	N03
2	17.5	9.0	N06

Table 'TB2'			
N	X	Z	NOEUD
1	2.0	2.5	N01
3	4.0	5.5	N031
4	17.5	20.5	N062
6	5.0	8.0	N013

#--- COMBINAISON

```
TB3=CALC_TABLE(TABLE=TB1,
                COMB =_F(TABLE = TB2, NOM_PARA='NOEUD'))
```

Le contenu de la table TB3 est :

N	Y	Z	NOEUD	X
1	2.00000E+00	2.50000E+00	N01	2.00000E+00
1	4.00000E+00	1.75000E+01	N03	-
2	1.75000E+01	9.00000E+00	N06	-
3	-	5.50000E+00	N031	4.00000E+00
4	-	2.05000E+01	N062	1.75000E+01
6	-	8.00000E+00	N013	5.00000E+00

#--- FORMULE

```
DNOR=FORMULE(NOM_PARA = ('X','Z'),
              VALE = 'SQRT(X*X+Z*Z)')
```

```
TB3=CALC_TABLE(TABLE = TB3,
                reuse = TB3,
                OPER =_F(FORMULE=DNOR , NOM_PARA='NOR_COOR'))
```

Le contenu de la table TB3 est :

N	Y	Z	NOEUD	X	NOR_COOR
1	2.00000E+00	2.50000E+00	N01	2.00000E+00	3.20156E+00
1	4.00000E+00	1.75000E+01	N03	-	-
2	1.75000E+01	9.00000E+00	N06	-	-
3	-	5.50000E+00	N031	4.00000E+00	6.80074E+00
4	-	2.05000E+01	N062	1.75000E+01	2.69537E+01
6	-	8.00000E+00	N013	5.00000E+00	9.43398E+00



#--- RENOMMER

```
TB3=CALC_TABLE(TABLE = TB3,
               reuse = TB3,
               RENOMME=_F(NOM_PARA=( 'NOR_COOR' , 'NORM_XZ' ) ) )
```

Le contenu de la table TB3 est :

N	Y	Z	NOEUD	X	NORM_XZ
1	2.00000E+00	2.50000E+00	N01	2.00000E+00	3.20156E+00
1	4.00000E+00	1.75000E+01	N03	-	-
2	1.75000E+01	9.00000E+00	N06	-	-
3	-	5.50000E+00	N031	4.00000E+00	6.80074E+00
4	-	2.05000E+01	N062	1.75000E+01	2.69537E+01
6	-	8.00000E+00	N013	5.00000E+00	9.43398E+00

#--- FILTRE

```
TB4=CALC_TABLE(TABLE = TB3,
               FILTRE =_F(NOM_PARA='NORM_XZ' , CRIT_COMP='LE' , VALE=30. ) )
```

Le contenu de la table TB4 est :

N	Y	Z	NOEUD	X	NORM_XZ
1	2.00000E+00	2.50000E+00	N01	2.00000E+00	3.20156E+00
3	-	5.50000E+00	N031	4.00000E+00	6.80074E+00
4	-	2.05000E+01	N062	1.75000E+01	2.69537E+01
6	-	8.00000E+00	N013	5.00000E+00	9.43398E+00

#--- EXTRACTION

```
TB3=CALC_TABLE(TABLE = TB3,reuse=TB3,
               EXTR =_F(NOM_PARA=( 'NOEUD' , 'X' , 'Z' , 'NORM_XZ' ) ) )
```

Le contenu de la table TB3 est :

NOEUD	X	Z	NORM_XZ
N01	2.00000E+00	2.50000E+00	3.20156E+00
N03	-	1.75000E+01	-
N06	-	9.00000E+00	-
N031	4.00000E+00	5.50000E+00	6.80074E+00
N062	1.75000E+01	2.05000E+01	2.69537E+01
N013	5.00000E+00	8.00000E+00	9.43398E+00

#--- TRI

```
TB3=CALC_TABLE(TABLE = TB3,reuse=TB3,
               TRI =_F(NOM_PARA='NORM_XZ' , ORDRE='DECROISSANT' ) )
```

Le contenu de la table TB3 est :

NOEUD	X	Z	NORM_XZ
N062	1.75000E+01	2.05000E+01	2.69537E+01
N013	5.00000E+00	8.00000E+00	9.43398E+00
N031	4.00000E+00	5.50000E+00	6.80074E+00
N01	2.00000E+00	2.50000E+00	3.20156E+00
N06	-	9.00000E+00	-
N03	-	1.75000E+01	-

Page laissée intentionnellement blanche.