

## Procedure CALC\_TABLE

---

### 1 Drank

---

To handle the data of `the arrays` in the manner of a spreadsheet.

The command makes it possible to carry out operations on the data of the arrays. The following operations are currently available:

- Concatenate/To combine two arrays,
- To apply a formula,
- To re-elect parameters,
- To filter the lines according to certain criteria,
- To extract certain columns from an array,
- To order the lines,
- To add lines or columns.

Product a data structure `counts`.

## Contents

---

1	But	1
2	Syntaxe	3
3	Opérandes	5
3.1	Operand TABLE	5
3.2	Operand ACTION	5
3.3	OPERATION = "COMB": concatenate/to combine two tables	5
3.3.1	Operand TABLE	5
3.3.2	Operand NOM_PARA	5
3.3.3	Operand RESTREINT	5
3.4	OPERATION = "OPER": To apply a formule	5
3.4.1	Operand FORMULE	6
3.4.2	Operand NOM_PARA	6
3.5	OPERATION = "RENOMME": To re-elect parameters of a table	6
3.5.1	Operand NOM_PARA	6
3.6	OPERATION = "FILTRE": To filter the lines according to some critères	6
3.7	OPERATION = "EXTR": To extract certain columns of a table	6
3.7.1	Operand NOM_PARA	6
3.8	OPERATION = "TRI": To order lines	6
3.8.1	Operand NOM_PARA	6
3.8.2	Operand ORDRE	6
3.9	OPERATION = "AJOUT_LIGNE": to add one line	6
3.9.1	Operand NOM_PARA	7
3.9.2	Operand VALE	7
3.10	OPERATION = "AJOUT_COLONNE": to add a column	7
3.10.1	Operand NOM_PARA	7
3.10.2	Operands VALE	7
3.11	Operand TITRE	7
3.12	Operand INFO	8
4	Exemples	9

## 2 Syntax

```

= CALC_TABLE (
    ◊reuse =tab , [array]
    ◆TABLE =matable , [array]

# Continuation of the actions to carry out (in the order)
◆ ACTION =_F (

    ◆ OPERATION = / "COMB",
                  / "OPER",
                  / "RENOMME",
                  / "FILTRE",
                  / "EXTR",
                  / "TRI",
                  / "AJOUT_LIGNE",
                  / "AJOUT_COLONNE",
                  / "SUPPRIME",

# 1. Combination of arrays ( OPERATION=' COMB' ):
    ◆ ARRAY = array, [array]
    ◊ NOM_PARA = will l_para,
[ l_Kn]
                                ◊RESTREINT=' NON',
[DEFAULT]
                                "OUI",

# 2. To apply a formula ( OPERATION=' OPER' ):
    ◆FORMULE= formulates, [formula]
    ◆NOM_PARA = para, [kN]

# 3. To re-elect parameters of an array ( OPERATION=' RENOMME' ):
    ◆NOM_PARA = [old name, new name], [ l_Kn]

# 4. To filter lines ( OPERATION=' FILTRE' ):
    ◆ 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]
      ◊ | accuracy = prec, [R8]
                                /1.0D-3,
[DEFAULT]
                                | CRITERE =/" RELATIF",
[DEFAULT]
                                /"ABSOLU",
                                /CRIT_COMP = /"REGEXP",
    ◆VALE_K = kval, [kN]
    /CRIT_COMP = /"MAXI",

```

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

```

                                /"MAXI_ABS",
                                /"MINI",
                                /"MINI_ABS",
                                /"VIDE",
                                /"NON_VIDE",

# 5. To extract certain columns ( OPERATION=' EXTR' ):
    ♦ NOM_PARA = will l_para,
[l_Kn]

# 6. To order the lines according to the values of a parameter ( OPERATION=' TRI' )
:
    ♦ NOM_PARA= will lpara,
[l_Kn]
    ◊ORDRE      = "CROISSANT" , [DEFAULT]
                  /"DECREASING",

# 7. To add line to an array ( OPERATION=' AJOUT_LIGNE' ) : ") :
    ♦ NOM_PARA= will lpara,
[l_Kn]
    ♦VALE       = lvale, [l_*]

# 8. To add constant columns to an array ( OPERATION=' AJOUT_COLONNE" ) :
    ♦ NOM_PARA= will lpara,
[l_Kn]
    ♦VALE       = lvale, [l_*]

# 9. To remove columns of an array ( OPERATION=' SUPPROME' ) :
    ♦ NOM_PARA= will lpara,
[l_Kn]

    ),
    ◊TITER =titer [l_Kn]
    ◊INFO  =/1 ,
[DEFAULT]
        /2 ,

)

```

## 3 Operands

---

### 3.1 Operand COUNTS

◆TABLE = matable

Name of the initial array on which one will carry out handling.

### 3.2 Operand ACTION

```
◆ACTION = ( _F (OPERATION=' xxx',...),  
            _F (OPERATION=' yyy',...),  
            ...)
```

The "actions" are carried out one after the other. The operation "yyy" takes the array in the state where the operation "xxx" left it.

It is definitely more powerful to repeat factor key word the ACTION than to do as many successive calls to command CALC\_TABLE.

### 3.3 OPERATION = "COMB" : concatenate/to combine two arrays

operation COMB allows concatenate, to combine two arrays between them, with seam on a common parameter list.

#### 3.3.1 Operand COUNTS

◆TABLE = array

Name of the array whose values must come to overload and/or enrich the initial array.

#### 3.3.2 Operand NOM\_PARA

◆NOM\_PARA = will l\_para

Name of the parameters whose values must be identical in the two arrays.

#### 3.3.3 Operand RESTREINT

◆RESTREINT = "OUI" or "NON"

Example of the combination of two arrays:

```
tab_resu=CALC_TABLE (TABLE=tab1,  
                    ACTION=_F (OPERATION=' COMB',  
                                TABLE=tab2,  
                                NOM_PARA= ("ABSC_CURV", "NOEUD"),  
                                RESTREINT=' NON'))
```

When ABSC\_CURV and NODE are identical between tab1 and tab2 , one inserts the values of tab2 on line of tab1 (for the other parameters common to the 2 arrays, it is thus the value of tab2 which crushes that of tab1 ).

If ABSC\_CURV and/or NODE differ between tab1 and tab2 , one adds line tab2 at the end of tab1 . NOM\_PARA acts like a primary word: if one does not find more once couple ( ABSC\_CURV , NOEUD ), one adds line.

The same operation with RESTREINT=' OUI ' produces the same lines when ABSC\_CURV and NODE are identical. On the other hand, no line is added when the parameters differ.

### 3.4 OPERATION = "OPER" : To apply a formula

Makes it possible to apply a formula whose variables are the parameters of the array and to insert result in a new column.

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

## 3.4.1 Operand FORMULATES

◆FORMULE = formula  
Name of the formula to apply

## 3.4.2 Operand NOM\_PARA

◆NOM\_PARA = para  
Name of the new column.

## 3.5 OPERATION = "RENOMME" : To re-elect parameters of an array

operation RENOMME makes it possible to re-elect one or more parameters of an array.

### 3.5.1 Operand NOM\_PARA

◆NOM\_PARA = will l\_para  
Name of the parameters: couples values (old name of the parameter, new name of the parameter)

## 3.6 OPERATION = "FILTRE" : Factor key word to filter the lines according to certain

criteria This makes it possible to filter the lines of the array. For the use of this key word to see command IMPR\_TABLE [U4.91.03].

## 3.7 OPERATION = "EXTR" : To extract certain columns from an array

operation EXTR makes it possible to extract certain columns from an array.

### 3.7.1 Operand NOM\_PARA

◆NOM\_PARA = will l\_para  
Name of the parameters which one wants to extract.

## 3.8 OPERATION = "TRI" : To order the lines

the TRI operation makes it possible to order the lines according to the values of the parameters.

### 3.8.1 Operand NOM\_PARA

◆NOM\_PARA = will l\_para  
Names of the parameters to which the sort relates.

### 3.8.2 Operand ORDRE

◆ORDRE = / "CROISSANT"  
/ "DECREASING"

This key word is used to specify if one must use an order ascending or decreasing. By default, one sorts by ascending order.

The relations of order used are:

- the natural order for the integers and the real,
- the alphabetical order for the texts and the names of concepts.

**Note:**

One cannot make use of a parameter complexes to classify the lines of an array.  
For the parameters of the *NODE* type (or *NETS*), the order is alphabetical because these parameters contain the name of the nodes (or of meshes).

If one specifies:

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

One will sort the lines of the array in the alphabetical order of the nodes. If there exist several lines corresponding to a given node, **the second** sort criterion (*INST*) will be used to classify these lines.

## 3.9 OPERATION = "AJOUT\_LIGNE" : to add line

operation *AJOUT\_LIGNE* makes it possible to add line to an existing array.

### 3.9.1 Operand *NOM\_PARA*

◆*NOM\_PARA* = will *lpara*

Names of the parameters (names of the columns of the array) of line added. One can define very well values only for certain columns of the array.

If a name of parameter does not exist in the array, it is added. Its type is given from the provided value.

### 3.9.2 Operand *VALE*

◆*VALE* = *lvale*

List of the values for each parameter of *NOM\_PARA*. The type of the values must be compatible with the types of the columns of the array.

*VALE* can contain heterogeneous values among integer, reality, character string. The type must be in conformity with the type of the parameter of the array.

The first value corresponds to the first parameter given in *NOM\_PARA*, the second value with the second parameter, etc

The lists *NOM\_PARA* and *VALES* thus have the same cardinal.

## 3.10 OPERATION = "AJOUT\_COLONNE" : to add a column

operation *AJOUT\_COLONNE* makes it possible to add one or more columns to an existing array.  
Constant columns are added: the value is identical on all the lines.

### 3.10.1 Operand *NOM\_PARA*

◆*NOM\_PARA* = will *lpara*

Names of the parameters, names of the columns added to the array. None of these parameters must be already present in the array.

### 3.10.2 Operands *VALE*

◆*VALE* = *lvale*

List of the values of each column.

*VALE* can contain heterogeneous values among integer, reality, character string. The type associated with the parameter is given from this value.

The first value corresponds to the first parameter given in `NOM_PARA`, the second value with the second parameter, etc  
The lists `NOM_PARA` and `VALES` thus have the same cardinal.

## 3.11 OPERATION = "SUPPRIME" : to remove columns

operation `SUPPRIME` makes it possible to remove one or more columns of an existing array.

### 3.11.1 Operand `NOM_PARA`

◆`NOM_PARA` = will `lpara`

Names of the parameters, names of the columns to be removed in the array.

## 3.12 Operand `TITER`

◇`TITER` = `tit`

Title of the produced array. When this one is not provided, the title of the array as starter, according to the operations, is supplemented.

## 3.13 Operand `INFO`

◇`INFO` = `inf`

Prints in the file "message" of additional information if `inf=2`. Nothing occurs if `inf=1`.



## 4 Examples

Counts "TB1 N"			
"	Y	Z	NOEUD
0.2.0.4. 5			N01
1.4.0		17.5	N03
2	17.5	9.0	N06

Counts "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

```
#--- COMBINATION
TB3=CALC_TABLE (TABLE=TB1,
                ACTION =_F (OPERATION=' COMB',
                            ARRAY = TB2, NOM_PARA=' NOEUD'))
```

The contents of array TB3 are:

N	Y	Z	NOEUD	X1
	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

```
#--- FORMULATE
DNOR=FORMULE (NOM_PARA = ("X", "Z"),
              VALE = "sqrt (X*X+Z*Z)")
```

```
TB3=CALC_TABLE (ARRAY = TB3,
                reuse = TB3,
                ACTION =_F (OPERATION=' OPER',
                            FORMULE=DNOR, NOM_PARA=' NOR_COOR'))
```

The contents of array TB3 are:

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

```
#--- TO RE-ELECT
TB3=CALC_TABLE (ARRAY = TB3,
                reuse = TB3,
                ACTION=_F (OPERATION=' RENOMME',
                           NOM_PARA= ("NOR_COOR", "NORM_XZ")))
```

The contents of array TB3 are:

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 (ARRAY = TB3,
                ACTION = _F (OPERATION=' FILTRE',
                             NOM_PARA=' NORM_XZ', CRIT_COMP=' LE', VALE=30.))
```

The contents of array TB4 are:

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 (ARRAY = TB3, reuse =TB3,
                ACTION = _F (OPERATION=' EXTR',
                             NOM_PARA= ("NOEUD", "X", "Z", "NORM_XZ")))
```

The contents of array TB3 are:

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 (ARRAY = TB3, reuse=TB3,
                ACTION = _F (OPERATION=' TRI',
                             NOM_PARA=' NORM_XZ', ORDRE=' DECROISSANT'))
```

The contents of array TB3 are:

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	-

# Code Aster

Version  
default

Titre : Procédure CALC\_TABLE  
Responsable : Mathieu COURTOIS

Date : 15/05/2013 Page : 11/11  
Clé : U4.33.03 Révision : 11019

N03	-	1.75000E+01	-
-----	---	-------------	---