

Procedure CALC_TABLE

1 Goal

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

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

- Concaténer/To combine two tables,
- To apply a formula,
- To re-elect parameters,
- To filter the lines according to certain criteria,
- To extract certain columns from a table,
- To order the lines,
- To add lines or columns,
- To add information on the table itself.

Product a structure of data `table`.

Contents

1 Goal.....	1
2 Syntax.....	3
3 Operands.....	5
3.1 Operand TABLE.....	5
3.2 Operand ACTION.....	5
3.3 OPERATION = 'COMB' : concaténer/to combine two tables.....	5
3.3.1 Operand TABLE.....	5
3.3.2 Operand NOM_PARA.....	5
3.3.3 Operand FORMAT_R.....	5
3.3.4 Operand RESTRICTED.....	5
3.4 OPERATION = 'OPER' : To apply a formula.....	6
3.4.1 Operand FORMULA.....	6
3.4.2 Operand NOM_PARA.....	6
3.4.3 Operand NOM_COLONNE.....	6
3.5 OPERATION = 'RE-ELECTS' : To re-elect parameters of a table.....	6
3.5.1 Operand NOM_PARA.....	6
3.6 OPERATION = 'FILTER' : To filter the lines according to certain criteria.....	6
3.7 OPERATION = 'EXTR' : To extract certain columns from a table.....	6
3.7.1 Operand NOM_PARA.....	6
3.8 OPERATION = 'TRI' : To order the lines.....	6
3.8.1 Operand NOM_PARA.....	6
3.8.2 Operand ORDER.....	7
3.9 OPERATION = 'AJOUT_LIGNE' : to add a line.....	7
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.....	8
3.10.2 Operand VALE.....	8
3.10.3 Operand VALE_COLUMN.....	8
3.11 OPERATION = 'REMOVES' : to remove columns.....	8
3.11.1 Operand NOM_PARA.....	8
3.12 OPERATION = ' STATISTICS '	8
3.13 Operand TITLE.....	8
3.14 Operand INFORMATION.....	8
4 Examples.....	9

2 Syntax

```

= CALC_TABLE (
  ◊ reuse = , [table]
  ◆ TABLE = matable, [table]

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

    ◆ OPERATION = / 'COMB',
                  / 'OPER',
                  / 'RE-ELECTS',
                  / 'FILTER',
                  / 'EXTR',
                  / 'TRI',
                  / 'AJOUT_LIGNE',
                  / 'AJOUT_COLONNE',
                  / 'REMOVES',
                  / 'STATISTICS',

    # 1. Combination of tables ( OPERATION=' COMB' ) :
      ◆ TABLE = table, [table]
      ◊ NOM_PARA = will l_para,
      [l_Kn]
      ◊ RESTRICTED = 'NOT',
      [DEFECT]
                  'YES',
      ◊ FORMAT_R = fmt, [KN]

    # 2. To apply a formula ( OPERATION=' OPER' ) :
      ◆ FORMULA = formula, [formula]
      ◆ NOM_PARA = para, [KN]
      ◊ NOM_COLONNE = collars, [l_Kn]

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

    # 4. To filter lines ( OPERATION=' FILTRE' ) :
      ◆ NOM_PARA = para, [KN]
      ◊ / CRIT_COMP = / 'EQ', [DEFECT]
                  /'',
                  / 'LT',
                  / 'WP',
                  / 'IT',
                  / 'GE',
      ◆ / VALE_I = ival, [I]
        / VALE_K = kval, [KN]
        / / VALE = rval, [R]
          / VALE_C = cval, [C]
          ◊ | PRECISION = / prec, [R8]
              / 1.0D-3, [DEFECT]
              | CRITERION =/'RELATIF', [DEFECT]
              / 'ABSOLUTE',
        / CRIT_COMP = / 'REGEXP',
      ◆ VALE_K = kval, [KN]
        / CRIT_COMP = / 'MAXIMUM',
                  / 'MAXI_ABS',

```

```

                                / 'MINI' ,
                                / 'MINI_ABS' ,
                                / 'VACUUM' ,
                                / 'NON_VIDE' ,

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

# 6. To order the lines according to the values of a parameter (OPERATION=' TRI')
:
    ♦ NOM_PARA = will lpara, [l_Kn]
    ♦ ORDER = / 'GROWING' , [DEFECT]
              / 'DECREASING' ,

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

# 8. To add constant columns to a table (OPERATION=' AJOUT_COLONNE') :
    ♦ NOM_PARA = will lpara, [l_Kn]
    ♦ / VALE = lvale, [l_*]
      / VALE_COLUMN = lval_collar, [l_*]

# 9 . To remove columns of a table (OPERATION=' SUPPR I ME') :
    ♦ NOM_PARA = will lpara, [l_Kn]

# 10 . To add statistics in the table (OPERATION=' STATISTICS ` )
) ,
  ♦ TITLE = title [l_Kn]
  ♦ INFORMATION =/ 1,
[DEFECT]
                / 2,

)

```

3 Operands

3.1 Operand TABLE

- ◆ TABLE = matable

Name of the initial table 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 table in the state where the operation 'xxx' left.

It is definitely more performing to repeat the keyword factor ACTION to do as many successive calls to the order CALC_TABLE.

3.3 OPERATION = 'COMB' : concaténer/to combine two tables

The operation COMB concaténer allows, to combine two tables between them, with seam on a common parameter list.

3.3.1 Operand TABLE

- ◆ TABLE = table

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

3.3.2 Operand NOM_PARA

- ◆ NOM_PARA = will l_para

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

3.3.3 Operand FORMAT_R

It is about a format which will be used before comparing two real numbers in order to determine if they are identical or not. Let us note that in the table, one recovers the nonround value (second table).

3.3.4 Operand RESTRICTED

- ◆ RESTRICTED = 'YES' or 'NOT'

Example of the combination of two tables:

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

When ABSC_CURV and NODE are identical enters tab1 and tab2 , one inserts the values of tab2 on the line of tab1 (for the other parameters common to the 2 tables, it is thus the value of tab2 who crushes that of tab1).

If ABSC_CURV and/or NODE differ between tab1 and tab2 , one adds the line of tab2 at the end of tab1 . NOM_PARA acts like a primary key: if one does not find more once couple (ABSC_CURV , NODE), the line is added.

The same operation with `RESTREINT=' OUI '` product 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

Allows to apply a formula whose variables are the parameters of the table and to insert the result in a new column.

3.4.1 Operand FORMULA

Name of the formula to be applied

3.4.2 Operand NOM_PARA

Name of the new column.

3.4.3 Operand NOM_COLONNE

This keyword makes it possible to use a formula in the parameters are not parameters of the table.

For example, there is a formula which depends on `INST`. If one wants to evaluate this formula by using the column `INST_FIN` table, it is enough to specify `NOM_COLONNE=' INST_FIN '`.

3.5 OPERATION = 'RE-ELECTS' : To re-elect parameters of a table

The operation `RE-ELECT` allows to re-elect one or more parameters of a table.

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 = 'FILTER' : To filter the lines according to certain criteria

This keyword factor makes it possible to filter the lines of the table. For the use of this keyword to see the order `IMPR_TABLE` [U4.91.03].

3.7 OPERATION = 'EXTR' : To extract certain columns from a table

The operation `EXTR` allows to extract certain columns from a table.

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 sorting relates.

3.8.2 Operand ORDER

◆ ORDER = / 'GROWING'
/ 'DECREASING'

This keyword 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 entireties and realities,
- 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 a table.

For the parameters of the type NREUD (or MESH), the order is alphabetical because these parameters contain the name of the nodes (or of the meshes).

If one specifies:

```
TRI=_F (NOM_PARA= ('NODE', 'INST'), ORDRE= 'GROWING'),
```

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

3.9 OPERATION = 'AJOUT_LIGNE' : to add a line

The operation AJOUT_LIGNE allows to add a line to an existing table.

3.9.1 Operand NOM_PARA

◆ NOM_PARA = will lpara

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

If a name of parameter does not exist in the table, he is added. Its type is given starting 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 table.

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

The first value corresponds to the first parameter given in NOM_PARA, the second value with the second parameter, etc.

Lists NOM_PARA and VALE thus have the same cardinal.

3.10 OPERATION = 'AJOUT_COLONNE' : to add a column

The operation AJOUT_COLONNE allows to add one or more columns constants (identical value for all the lines, by using the keyword VALE), or, only one column with distinct values (by using the keyword VALE_COLONNE).

3.10.1 Operand NOM_PARA

- ◆ NOM_PARA = will lpara
Names of the parameters, names of the columns added to the table. None of these parameters must be already present in the table.
One should provide only one parameter if one uses VALE_COLONNE.

3.10.2 Operand VALE

- ◆ VALE = lvale
List of the values of each column.
VALE can contain heterogeneous values among entirety, reality, character string. The type associated with the parameter is given starting from this value.
The first value corresponds to the first parameter given in NOM_PARA, the second value with the second parameter, etc.
Lists NOM_PARA and VALE thus have the same cardinal.

3.10.3 Operand VALE_COLUMN

- ◆ VALE_COLUMN = lval_collar
List of the values lines of column added.
The type of the provided values can to be entirety, reality, chain characters. All the values must be in the same way standard.
The values are added on the lines such as they are ordered at the time of the addition. It can there have added-values more or less that lines in the table. Either the column will be longer than the initial table, or it will contain empty values.

3.11 OPERATION = 'REMOVES' : to remove columns

The operation REMOVE allows to remove one or more columns of an existing table.

3.11.1 Operand NOM_PARA

- ◆ NOM_PARA = will lpara
Names of the parameters, names of the columns to be removed in the table.

3.12 OPERATION = 'STATISTICS'

The operation STATISTICS allows to add information on the table itself.
The added columns are named STAT_NOM and STAT_VALE. They contain respectively the name of the parameter and its value.
The added elementary statistics are: the number of lines (named NB_LIGNES), the number of columns (NB_COLONNES) and the number of blank cells (NB_VIDE).

3.13 Operand TITLE

- ◇ TITLE = tit
Title of the produced table. When this one is not provided, the title of the table as starter, according to the operations, is supplemented.

3.14 Operand INFORMATION

- ◇ INFORMATION = inf
Print in the file "message" of the additional information if inf=2. Nothing occurs if inf=1.

4 Examples

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

Table 'TB2'			
NR	X	Z	NODE
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',
                            TABLE = TB2, NOM_PARA=' NOEUD'))
```

Contents of the table TB3 is:

NR	Y	Z	NODE	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

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

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

Contents of the table TB3 is:

NR	Y	Z	NODE	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 (TABLE = TB3,
                reuse = TB3,
                ACTION=_F (OPERATION=' RENOMME',
```

NOM_PARA= ('NOR_COOR', 'NORM_XZ'))

Contents of the table TB3 is:

NR	Y	Z	NODE	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

#--- FILTER

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

Contents of the table TB4 is:

NR	Y	Z	NODE	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,  
                ACTION = _F (OPERATION=' EXTR',  
                             NOM_PARA= ('NODE', 'X', 'Z', 'NORM_XZ')))
```

Contents of the table TB3 is:

NODE	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

#--- SORTING

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

Contents of the table TB3 is:

NODE	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	-