Operator CREA_TABLE

1 Goal

To create one `table`.

This order makes it possible to create a table starting from a function or lists of real numbers or a real field. The table created has two parameters at least and as many lines as the user wishes it.

The lists used to generate the table must have the same number of terms, or it is necessary to indicate the numbers of the lines which one wishes to fill.

The function making it possible to fill the cells of the table is tabulée, which means that it is known only in certain points. It will have to be definite first in the command file using the order `DEFI_FONCTION`.

The concept created by this operator is of type `table`.

2 Syntax

```
Tb [*]  =  CREA_TABLE (

            ♦  /   FUNCTION  =   _F (
                        ♦  FUNCTION   =   function
    [function]
                        ◊  PARA       =   (npx, npy)                      [l_K16]
                                          ),

                /   LIST   =   _F (
                     ♦ /   LISTE_I    =   Li                          [l_I]
                         / LISTE_R    =   Lr                          [l_R]
                         / LISTE_K    =   lk                          [l_K]
                              ◊  TYPE_K =   /   ‘K8 ‘,                [DEFECT]
                                            /   ‘K16‘,
                                            /   ‘K24’,
                    ◊  NUME_LIGN     =   /   lind,                    [l_I]
                                         /   (1,2,3,…)            [DEFECT]
                    ♦  PARA          =   npx                          [K16]
                                    ),

                /   RESU   =   _F (
                     ♦ /   CHAM_GD    =   chamgd,                     [cham_gd]
                       /   RESULT =   resu,                       [result]
                           ♦  NOM_CHAM = ncham,                       [K16]
                           ◊  /   TOUT_ORDRE = ‘YES’                  [TXM]
                              /   NUME_ORDRE = numord                 [l_I]
                              /   LIST_ORDRE = lnumord                [listis]
                              /   INST = inst                         [l_R]
                              /   LIST_INST = linst,                  [listr8]
                              /   MODE = mode,                        [l_I]
                              /   LIST_MODE = lmode,                  [listis]
                              /   FREQ = freq,                        [l_R]
                              /   LIST_FREQ = lfreq,                  [listr8]
                           ◊  CRITERION =  /  ‘RELATIVE’,
    [DEFECT]
                                        /   ‘ABSOLUTE’,
                           ◊  PRECISION =  / 1.E-6,                   [DEFECT]
                                           / prec,                    [R]
                     ♦ /   TOUT_CMP   =   ‘YES’,                      [TXM]
                       /   NOM_CMP    =   ncmp,                       [TXM]
                       /   NOM_VARI   =   nvari,                      [TXM]
                     ♦ /   ALL    =   ‘YES’,                      [TXM]
                       /   GROUP_MA   =   grma,                       [l_grma]
                       /   GROUP_NO   =   grno,                       [l_grno]
                         / LISTE_K    =   lk                          [l_K]
                                    ),


            ◊  TYPE_TABLE = /   ‘TABLE’,                              [DEFECT]
                            /   ‘TABLE_FONCTION’,

            ◊  TITLE  =  tit,                                         [KN]

)
```

3 Operands

3.1 Keyword FUNCTION

3.1.1 Operand FUNCTION

The table is created starting from a function. The first column of the table contains the X-coordinates of the function and the second the values of the function to the corresponding X-coordinates.

3.1.2 Operand PARA

Names of the parameters of the table. This keyword is optional: the names of the parameters are then identical to NOM_PARA and NOM_RESU function. If it is indicated, its cardinal must be equal to 2: name of parameter for the column associated with the X-coordinates, name of parameter for the column associated with the ordinates.

3.2 Keyword LIST

This keyword must be provided at least 2 times: each occurrence makes it possible to define a column of the table.

The lists do not have necessarily the same cardinal, one uses the keyword NUME_LIGN to indicate which lines must be filled.

3.2.1 Operand LISTE_R

A column is created in the table starting from a list of realities.

3.2.2 Operand LISTE_I

A column is created in the table starting from a list of entireties.

3.2.3 Operand LISTE_K

A column is created in the table starting from a list of character strings.

3.2.4 Operand TYPE_K

Length of the character strings for the case LISTE_K.

3.2.5 Operand NUME_LIGN

When the lists do not have all the same cardinal or to build a table with holes, one uses this keyword to specify which lines are filled.

3.2.6 Operand PARA

Name of the parameter of the table associated with the provided list.

3.3 Keyword RESU

This keyword should be used only once. It makes it possible to create a table starting from the components of a field isolated (keyword CHAM_GD) or of the fields of a structure of data result (keyword RESULT). The treated fields must have actual values.

3.3.1 Operand CHAM_GD

This operand makes it possible to collect the name of the field which will be read to create the table. This field can come for example from the order CREA_CHAMP.

3.3.2 Operands RESULTAT/NOM_CHAM/TOUT_ORDRE/NUMÉRIQUE_ORDRE/LISTE_ORDRE /INST/LIST_INST/MODE/LIST_MODE/FREQ/LIST_FREQ/CRITERE/PRECISION

These operands make it possible to choose the fields of the structure of data results to treat. One can limit the sequence numbers with the keywords INST , LIST_INST ,... as well as the field names (' DEPL ',' SIEF_ELGA ',...) with the keyword NOM_CHAM .

For more information concerning these keyword, please consult Doc. [U4.71.00]

3.3.3 Operands TOUT_CMP/NOM_CMP/NOM_VARI

These operands make it possible to choose the components which one wants to see in the table.
Either one uses TOUT_CMP=' OUI' to choose all the components of the fields, one is uses NOM_CMP or NOM_VARI to retain only some of them.

```
◊   NOM_CMP    =      nocmp,                  [KN]
```

Name of the component that one wants to see in the table.

```
◊   NOM_ VARI   =      novari ,               [KN]
```

For the fields of the internal variables (VARI_*), one can give the name of the internal variable that one wants to see in the table (see [U4.51.11] for the rules of naming of the internal variables).

3.3.4 Operands TOUT/GROUP_MA/GROUP_NO

These operands make it possible to choose the entities of the grid on which one will recover the values of the components of the field.
The operand ALL = ' OUI' allows to consider all the grid.

To restrict part of grid, should be used the remaining operands.

For a field of the type:

- NOEU : one can choose GROUP_MA and/or GROUP_NO .
- ELNO/ELGA : one can choose GROUP_MA.

3.4 Operand TYPE_TABLE

Determine the type of the produced table. The choices are:

- TABLE : an ordinary table
- TABLE_FONCTION : a table which contains a column whose parameter is FUNCTION or FONCTION_C in which one finds names of concept function.

3.5 Operand TITLE

Title which will be given to the produced table.

4 Examples

4.1 Creation of a table starting from a function

```
FONCTION=DEFI_FONCTION (NOM_PARA=' X',
                  VALE= (0.0, 5.0, 4.0, 17.5)) ;

T_FCT=CRÉA_TABLE (FONCTION=_F (FONCTION=FONCTION));

IMPR_TABLE (TABLE=T_FONCTION);
```

The impression of the table created is the following one:

```
     X            TOUTRESU
0.00000E+00      5.00000E+00
4.00000E+00      1.75000E+01
```

4.2 Creation of a table starting from three lists, heterogeneous types

```
T_LST=CRÉA_TABLE (LISTE= (
                           _F (LISTE_R= (0.0, 4.0), PARA=' X'),
                           _F (LISTE_R = (5.0, 17.5), PARA=' Y'),);
                           _F (LISTE_I = (6.7), PARA=' K')

IMPR_TABLE (TABLE=T_LST);
```

The impression of the table created is the following one:

```
     X                Y           K
0.00000E+00      5.00000E+00      6
4.00000E+00      1.75000E+01      7
```

4.3 Creation of a table with holes

```
TAB=CRÉA_TABLE (
   LISTE= (
      _F (   PARA=' NUMÉRIQUE_ORDRE',
         LISTE_I= (8,15,156,67),
      ),
      _F (   PARA=' VAR',
         LISTE_R= (2. , 15. , 18),
         NUME_LIGN= (1,3,4),
      ),
      _F (   PARA=' COMMENT',
         LISTE_K= ('VALUE',),
          TYPE_K=' K24',
         NUME_LIGNE=1,
      ),
   ),
```

The impression of the table created is the following one:

```
NUME_ORDRE  VAR            HOW
         8   2.00000E+00   VALUE
```

```
 15   -             -
156   1.50000E+01   -
 67   1.80000E+01   -
```

4.4 Creation of a table starting from a field

```
TB01=CRÉA_TABLE (RESU=_F (RESULTAT=TEMP0,
                          TOUT=' OUI',
                          NUME_ORDRE=4,
                          TOUT_CMP=' OUI',
                          NOM_CHAM=' TEMP',),)
```

The impression of the table created is the following one:

```
#TABLE_SDASTER
RESULT NOM_CHAM         NUME_ORDRE   NODE    COOR_X    COOR_Y    TEMP
TEMP0    TEMP                        4 N1       1.0       0.0       85.0
TEMP0    TEMP                        4 N2       2.0       0.0       85.0
TEMP0    TEMP                        4 N3       2.0       4.0       85.0
TEMP0    TEMP                        4 N4       1.0       4.0       85.0
```