

---

## Operator CREA\_TABLE

---

### 1 Drank

---

To create an `array`.

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

The lists used to generate the array 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 array is tabulated, which means that it is known only in certain points. It will have to be as a preliminary defined in the command file using command `DEFI_FONCTION`.

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

## 2 Syntax

```

Tb [*] = CREA_TABLE (
    ◆/FONCTION=_F      (
        ◆FONCTION=fonction
    [function]
        ◇PARA=          (npx, npy)          [l_K16]
        ),
    /LISTE      =_F      (
        ◆ /LISTE_I      =li          [l_I]
        / LISTE_R=lr          [l_R]
        / LISTE_K=lk          [l_K]
        ◇ TYPE_K=/          "K8",
    [DEFAULT]
        / "K16",
        / "K24",
        ◇ NUME_LIGN=/lind          [l_I]
        / (1,2,3,...)          [DEFAULT]
        ◆PARA=npx          [K16]
        ),
    /RESU      =_F      (
        ◆ /CHAM_GD      =changd ,          [cham_gd]
        / RESULTAT=resu ,          [result]
        ◆NOM_CHAM      = ncham,          [K16]
        ◇ /TOUT_ORDRE   = "OUI"          [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]
        ◇CRITERE       = "RELATIF",      [DEFAULT]
        / "ABSOLU",
        ◇PRECISION     = /1.E-6,          [DEFAULT]
        /prec,          [R]
        ◆ /TOUT_CMP     = ' OUI',          [TXM]
        /NOM_CMP        = ncmp ,          [TXM]
        ◆ /TOUT         = ' OUI',          [TXM]
        /GROUP_MA       = grma ,          [l_grma]
        /MAILLE         = ma ,            [l_ma]
        /GROUP_NO       = grno ,          [l_grno]
        /NOEUD          = noeu ,          [l_noeu]
        / LISTE_K=lk          [l_K]
        ),
    ◇TYPE_TABLE      = "ARRAY",          [DEFAULT]
        / "TABLE_FONCTION",
    ◇TITER=tit          ,          [kN]
)

```

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 Operands

---

### 3.1 Key word FONCTION

#### 3.1.1 Operand FONCTION

the array is created from a function. The first column of the array 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 array. This key word is optional: the names of the parameters are then identical to the `NOM_PARA` and `NOM_RESU` of the 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 Y-coordinates.

### 3.2 Key word LISTE

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

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

#### 3.2.1 Operand LISTE\_R

a column is created in the array from a list of realities.

#### 3.2.2 Operand LISTE\_I

a column is created in the array from a list of integers.

#### 3.2.3 Operand LISTE\_K

a column is created in the array from a list of character strings.

#### 3.2.4 Operand TYPE\_K

Length of the character strings for case `LISTE_K`.

#### 3.2.5 Operand NUME\_LIGN

When the lists do not have all the same cardinal or to build an array with holes, one uses this key word to specify which lines are filled.

#### 3.2.6 Operand PARA

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

### 3.3 Key word RESU

This key word should be used only one times. It makes it possible to create an array starting from the components of a field isolated (key word `CHAM_GD`) or fields from a data structure result (key word `RESULTAT`). 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 array. This field can come for example of the command 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 data structure results to treat. One can limit the sequence numbers with key keys INST , LIST\_INST ,... as well as the field names (" DEPL ", " SIEF\_ELGA ",...) with key word NOM\_CHAM .

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

## 3.3.3 Operands TOUT\_CMP/NOM\_CMP

These operands allow to choose the components which one wants to see in the array. Either one uses TOUT\_CMP=' OUI ' to choose all the components of the fields, or one uses NOM\_CMP to retain only some of them.

## 3.3.4 Operands TOUT/MAILLE/GROUP\_MA/NOEUD/GROUP\_NO

These operands make it possible to choose the entities of the mesh on which one will recover the values of the components field.

The operand TOUT = ' OUI ' makes it possible to consider all the mesh.

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

For a field of the type:

- NOEU : one can choose MESH and/or GROUP\_MA and/or NOEUD and/or GROUP\_NO .
- ELNO/ELGA : one can choose MESH and/or GROUP\_MA .

## 3.4 Operand TYPE\_TABLE

Determines the type of the produced array. The choices are:

- COUNT : an ordinary array
- TABLE\_FONCTION : an array which contains a column whose parameter is FONCTION or FONCTION\_C in which one finds names of concept function.

## 3.5 Operand TITER

Titres which will be given to the produced array.

## 4 Creation

### 4.1 examples of an array 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 printing of the array created is the following one:

X	TOUTRESU
0.000000E+00	5.000000E+00
4.000000E+00	1.750000E+01

### 4.2 Creation of an array 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 printing of the array created is the following one:

X	Y	K
0.000000E+00	5.000000E+00	6
4.000000E+00	1.750000E+01	7

### 4.3 Creation of an array 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= ("VALEUR",),  
            TYPE_K=' K24',  
            NUME_LIGNE=1,  
        ),  
    ),  
) ,
```

the printing of the array created is the following one:

NUME_ORDRE	VARCOMMENT
8	2.000000E+00VALEUR

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

## 4.4 Creation of an array from a field

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

the printing of the array created is the following one:

```
#TABLE_SDASTER
RESULTAT NOM_CHAM          NUME_ORDRE  NOEUD      COOR_X    COOR_Y    TEMP
TEMPO     TEMP              4 N1        1.0       0.0       85.0
TEMPO     TEMP              4 N2        2.0       0.0       85.0
TEMPO     TEMP              4 N3        2.0       4.0       85.0
TEMPO     TEMP              4 N4        1.0       4.0       85.0
```