

## Use structures of data Tables

---

### Summary:

This document describes the routines making it possible to use in FORTRAN the structures of data counts described in the document [D4.02.05] Structure of data `sd_table`.

## Contents

1	General information.....	4
1.1	What a table?.....	4
1.2	Some properties of the tables.....	5
2	List of the utilities for the tables.....	5
3	Routines of management of a table.....	6
3.1	Routine TBCRSD: To create a new table.....	6
3.2	Routine TBAJPA: To add parameters to the table.....	6
3.3	Routine TBEXIP: To test the existence and the type of a parameter.....	6
3.4	Routine TBFUTB: To amalgamate several only one tables counts.....	6
3.5	Routine DETRSD: To destroy a table.....	8
3.6	Routine COPISD: To duplicate a table.....	8
4	Routines of preparation of the values to be written in a table.....	8
4.1	Routine TBAJVA: To add a value associated with a parameter.....	8
4.2	Routine TBAJVC: To add a value associated with a complex parameter.....	9
4.3	Routine TBAJVI: To add a value associated with a whole parameter.....	9
4.4	Routine TBAJVK: To add a value associated with a parameter character string.....	9
4.5	Routine TBAJVR: To add a value associated with a real parameter.....	9
4.6	Examples of use.....	10
5	Routines of writing of values in a table.....	10
5.1	Routine TBAJLI: To add a line to the table.....	10
5.2	Routine TBNULI : Allows to recover a number of line in a table.....	11
5.3	Examples.....	11
6	Routines of reading of values in a table.....	12
6.1	Routine TBEXVE: Reading of the values of a column of a table.....	12
6.2	Routine TBEXFO: Create a function starting from a table.....	12
6.3	Routine TBLIVA: Reading of the value of a cell.....	13
7	Routines of filtering and sorting of a table.....	14
7.1	Mechanism of filtering of lines in a table.....	14
7.1.1	Typical case of the criteria of equality (or nonequality) for “floating” numbers.....	15
7.2	Routine TBEXTB: Filtering and extraction of a new table.....	17
7.3	Routine TBTRTB: Sorting of the table.....	18
8	Routine TBIMPR: Impression of a table.....	19
8.1	FORMAT: ‘EXCEL’.....	19
8.2	FORMAT: ‘TABLE’.....	19
8.3	FORMAT: ‘MOT_CLE’.....	20
8.4	FORMAT ‘EXCEL’ with pagination (here definite by the parameter ‘NODE’).....	20
9	Routine “TBEXLR”: Transformation of a table into a “SD LISTR8”.....	20

<a href="#">9.1 Goal of the routine.....</a>	<a href="#">20</a>
<a href="#">9.2 Interface of TBEXLR.....</a>	<a href="#">21</a>
<a href="#">9.3 Restrictions of use of TBEXLR.....</a>	<a href="#">21</a>
<a href="#">9.4 How a table it is transformed into LISTR8 ?.....</a>	<a href="#">23</a>
<a href="#">10 Simple example of creation and exploitation of a table.....</a>	<a href="#">25</a>

## 1 General information

---

### 1.1 What a table?

OneE counts is a structure of data of data-processing nature allowing to store a set of whole, real, complex values or character strings.

A table is comparable to a database EXCEL (version 5), i.e. one can see it as a list of columns (the term of "column" is here to make the bringing together with a list EXCEL ; that does not want to say that the impression of a table is always shown in column (the § 8 Impression of a table)) in opposite. Each column has a field name, that we call parameter, and contains of the similar data of type: I, R, C, K8, K16, K24 or K32.

Example 1: T1

NUME_ORDRE	INST	NODE	G
1	10.	N1	5.
1	10.	N2	6.
1	10.	N3	7.
1	10.	N4	8.
2	20.	N1	9.
2	20.	N2	9.
2	20.	N3	8.
2	20.	N4	8.
3	30.	N1	7.
3	30.	N2	6.
3	30.	N3	5.
3	30.	N4	4.

Example 2: T2

ACTION	NUME_ORDRE	INST	NODE	DX	DY	MESH	SIXX
ENTITLE 1	1	10.	N1	3.	5.		
ENTITLE 1	1	10.	N2	6.	7.		
ENTITLE 1	1	10.	N3	8.	9.		
ENTITLE 1	2	20.	N1	11.	12.		
ENTITLE 1	2	20.	N2	15.	13.		
ENTITLE 1	2	20.	N3	19.	18.		
ENTITLE 2	2	20.				MA1	-12.
ENTITLE 2	2	20.				MA2	-14.

## 1.2 Some properties of the tables

- A table has a limited number of columns (or parameters). These parameters are chosen by the developers of the orders creating of the tables. The name of a parameter is a chain of with more the 16 characters.
- On the other hand, the number of lines of a table is often "dynamic": it depends in general on the choices of the user: nodes of examination, moments of calculations,...
- The values contained in a column of a table are very in the same way standard FORTRAN: realities, complexes, entireties or texts. One can store in form "text" in a table of the names of SD Aster or names of objects JEVEUX; for example of the names of functions.
- A table is known as "full" when all its lines contain values for all the parameters of the table. The table T1 above is full. A table which is not full is known as "hollow" (counts t2 above).
- The lines of a table are naturally ordered by their order of insertion in the table (routine TBAJLI).
- The columns of a table are naturally ordered by the order of declaration of their parameters (routine TBAJPA).
- A table has at least a line. On a line, there can be blank cells: the associated parameter is not affected (table t2 above).

## 2 List of the utilities for the tables

Function	Name
To create a new table	TBCRSD
To declare the parameters	TBAJPA
To recover the existence and the type of a parameter	TBEXIP
To amalgamate several tables in only one	TBFUTB
To destroy a table	DETRSD
To duplicate a table	COPISD
To build a table while adding to it of the lines to one	TBAJLI
To recover the number of a line in a table	TBNULI
To print a table on listing	TBIMPR
To filter the lines of a table by imposing criteria on one or more parameters to create a new table of smaller size	TBEXTB
To sort the lines of a table according to certain selected parameters. The result of the sorting is a table whose lines were reordered.	TBTRTB
To collect in a vector the values corresponding to a given parameter	TBEXVE
To create a function starting from 2 columns of a table	TBEXFO
To see the value associated with a parameter given for a line with a table	TBLIVA
To recover all the digital values of a table in one SD LISTR8	TBEXLR

## 3 Routines of management of a table

### 3.1 Routine TBCRSD: To create a new table

TBCRSD (nomtab, bases)

nomtab	in	K19	Name of the new table to be created If it already exists, it is destroyed + emission <A>
base	in	K1	Base creation of the table counts ( 'G' , 'V' , ...)

### 3.2 Routine TBAJPA: To add parameters to the table

TBAJPA (nomtab, nbpar, nompar, typpar)

nomtab	in	K19	Name of the table where one wants to add parameters
nbpar	in	I	Many parameters to be added
nompar	in	V (K16)	List of the names of the parameters to be added
typpar	in	V (K8)	List of the types of the parameters: 'R', 'I', 'C', 'K8', 'K16', 'K24', 'K32'

### 3.3 Routine TBEXIP: To test the existence and the type of a parameter

TBEXIP (nomtab, para, exist, typpar)

nomtab	in	K19	Name of the table to be examined
para	in	K16	parameter to be tested
exist	out	L	.TRUE. : the parameter already exists in the table nomtab
typpar	out	K8	type of the parameters if there already exists in the table: 'R', 'I', 'C', 'K8', 'K16', 'K24', 'K32'

### 3.4 Routine TBFUTB: To amalgamate several only one tables counts

TBFUTB (tabout, basout, ntab, ltabin, para, typpar, VI, vr, vc, vk)

tabout	in	K19	Name of the table which one wants to create
basout	in	K1	Base creation of the table tabout ( 'G' , 'V' , ...)
ntab	in	I	Many tables which one wants to amalgamate
ltabin	in	K19	Names of the tables which one wants to amalgamate
para	in	K16	New parameter (optional) which will make it possible to distinguish the origin from each line of the new table if para=' ' the following arguments are not used.
typpar	in	K8	Type of the new parameter (optional)
VI	in	V (I)	List of the values for the new parameter 'I' (optional)
vr	in	V (R)	List of the values for the new parameter 'R' (optional)
vc	in	V (C)	List of the values for the new parameter 'C' (optional)

vk	in	V (K*)	List of the values for the new parameter 'K' (optional)
----	----	--------	---

This routine can be practical to create a hollow table, the developer can realize each under - table separately and to use routine TBFUTB to amalgamate them into only one counts.

Example:

One wants to amalgamate the 2 tables: T1 and T2

Wi th	B	C	D
x1	x2		x3
	x4		x5
x6	x7	x8	

Wi th	B	E
y1	y2	y3
y4		y5

If one writes:

```
ltabin (1) =T1  
ltabin (2) =T2  
CAL TBFUTB (T3, 'V', 2, ltabin, '', kbid, IBID, rbid, cbid, kbid)
```

The table is obtained: T3

Wi th	B	C	D	E
x1	x2		x3	
	x4		x5	
x6	x7	x8		
y1	y2			y3
y4				y5

If one writes:

```
ltabin (1) =T1  
ltabin (2) =T2  
VK (1) = ' ACTION1'  
VK (2) = ' ACTION2'  
CAL TBFUTB (T3, 'V', 2, ltabin, '\', 'K8', IBID, rbid, cbid, VK)
```

The table is obtained: T3

NR	Wi th	B	C	D	E
ACTION1	x1	x2		x3	
ACTION1		x4		x5	

ACTION1	x6	x7	x8		
ACTION2	y1	y2			y3
ACTION2	y4				y5

### Remarks on the order of the lines and the columns of the new table:

The lines of the new table are ordered by putting the lines of the tables end to end that one amalgamates.

To order the parameters the following rules are adopted:

- the new parameter (optional) is numbered in first,  
- parameters of the 1st table of *ltabin* are then added in the order that they have in *ltabin* (1)

- parameters of the 2nd table of *ltabin* are then added (except those already present in *ltabin* (1) ) in the order which they have in *ltabin* (2)

- ...

## 3.5 Routine DETRSD: To destroy a table

```
CAL DETRSD ('TABLE_SDASTER', nomtab)
```

## 3.6 Routine COPISD: To duplicate a table

```
CAL COPISD ('TABLE_SDASTER', 'V', tabin, tabout)
```

# 4 Routines of preparation of the values to be written in a table

## 4.1 Routine TBAJVA: To add a value associated with a parameter

```
TBAJVA (table, will nbpara, nompar, VI, livi, vr, livr, vc, livc, vk, livk)
```

table	in	K19	Name of the table where one wants to add values
will nbpara	in	I	Many parameters for the line
nompar	in	K16	Name of the parameter to be written
VI	in	I	Value 'I' for the parameter given
livi	in	V (I)	List of the values for the parameters 'I'
vr	in	R	Value 'R' for the parameter given
livr	in	V (R)	List of the values for the parameters 'R'
vc	in	C	Value 'C' for the parameter given
livc	in	V (C)	List of the values for the parameters 'C'
vk	in	K*	Value 'K' for the parameter given
livk	in	V (K*)	List of the values for the parameters 'K'



This routine is useful when one does not know the order of the parameters in the table. Thus, one associates a value with a parameter which is written in a list (with the good position). To write the values, in the table, it is necessary to give the lists of the values built to TBAJLI.

The call to this routine is encapsulated by routines TBAJVC, TBAJVI, TBAJVK, TBAJVR.

## 4.2 Routine TBAJVC: To add a value associated with a complex parameter

TBAJVC (table, will nbpara, nompar, vc, livc)

table	in	K19	Name of the table where one wants to add values
will nbpara	in	I	Many parameters for the line
nompar	in	K16	Name of the parameter to be written
vc	in	C	Value 'C' for the parameter given
livc	in	V (C)	List of the values for the parameters 'C'

## 4.3 Routine TBAJVI: To add a value associated with a whole parameter

TBAJVI (table, will nbpara, nompar, VI, livi)

table	in	K19	Name of the table where one wants to add values
will nbpara	in	I	Many parameters for the line
nompar	in	K16	Name of the parameter to be written
VI	in	I	Value 'I' for the parameter given
livi	in	V (I)	List of the values for the parameters 'I'

## 4.4 Routine TBAJVK: To add a value associated with a parameter character string

TBAJVK (table, will nbpara, nompar, vk, livk)

table	in	K19	Name of the table where one wants to add values
will nbpara	in	I	Many parameters for the line
nompar	in	K16	Name of the parameter to be written
vk	in	K*	Value 'K' for the parameter given
livk	in	V (K*)	List of the values for the parameters 'K'

## 4.5 Routine TBAJVR: To add a value associated with a real parameter

TBAJVR (table, will nbpara, nompar, vr, livr)

table	in	K19	Name of the table where one wants to add values
will nbpara	in	I	Many parameters for the line
nompar	in	K16	Name of the parameter to be written
vr	in	R	Value 'R' for the parameter given
livr	in	V (R)	List of the values for the parameters 'R'

## 4.6 Examples of use

One wishes to add in a table the values associated with the parameters NUME\_ORDRE, INST and G.

```
CAL TBAJVI (RESULT, NBPRUP, 'NUMÉRIQUE_ORDRE', IORD, LIVI)
CAL TBAJVR (RESULT, NBPRUP, 'INST', TIME, LIVR)
CAL TBAJVR (RESULT, NBPRUP, 'G', G, LIVR)

CAL TBAJLI (RESULT, NBPRUP, NOPRUP, LIVI, LIVR, LIVC, LIVK, 0)
```

## 5 Routines of writing of values in a table

### 5.1 Routine TBAJLI: To add a line to the table

**TBAJLI (nomtab, nbpar, nompar, VI, vr, vc, vk, digital)**

nomtab	in jxvar	K19	Name of the table where one wants to add a line
nbpar	in	I	Many parameters for the line
nompar	in	V (K16)	List of the names of the parameters of the line
VI	in	V (I)	List of the values for the parameters 'I'
vr	in	V (R)	List of the values for the parameters 'R'
vc	in	V (C)	List of the values for the parameters 'C'
vk	in	V (K*)	List of the values for the parameters 'K'
digital	in	I	/ 0 : one adds the line at the end of the table / I : I am replaced <sup>ème</sup> line of the table

Example:

That is to say a table containing the "whole" parameters of the type 'I1', 'I2', 'I3', the "real" parameters 'R1' and 'R2' and parameters "character strings" 'K1' and 'K2'

Let us suppose that one wants to add a line to it containing: I2=i2, R1=r1, K2=k2, K1=k1.

One will be able to call routine TBAJLI with the arguments:

```
nbpar = 4
nompar = ('I2', 'R1', 'K2', 'K1')
VI = (i2)
vr = (r1)
```

```
vk = (k2, k1)
digital = 0
```

## 5.2 Routine TBNULI : Allows to recover a number of line in a table

**TBNULI (tabin, npacri, lipacr, VI, vr, vc, vk, lprec, lcrit, digital)**

tabin	in	K19	Name of the table which one wants to recover a line
npacri	in	I	Many parameters implied in the selection criteria of the line (dimension of lipacr)
lipacr	in	V (K16)	List of the names of the parameters criteria
VI	in	V (I)	Values of the criteria for the parameters 'I'
vr	in	V (R)	Values of the criteria for the parameters 'R'
vc	in	V (C)	Values of the criteria for the parameters 'C'
vk	in	V (K)	Values of the criteria for the parameters 'K'
lprec	in	V (R)	List of the precise details (for the criteria of equality of the floating parameters [cf §7.1.1 Typical case of the criteria of equality...].)
lcrit	in	V (K8)	List of the secondary criteria of equality for the floating parameters: 'EQUAL', 'RELA', 'ABSO' [cf §7.1.1 Typical case of the criteria of equality...].
digital	out	I	= 0: There is no line corresponding to the criteria. = I: The line I is the only one which corresponds to the selection criteria < 0: There are several lines corresponding to the criteria

One searches a line in the table `tabin` by imposing conditions on its parameters.

The mechanism of selection of a line in a table (arguments `lipacr`, `lcrit`, `lprec`, `VI`, `vr`, `vc`, `vk`) is explained to the §7 Routines of filtering and sorting of a table.

When this line is found (and single), one returns his number what makes it possible to modify this line using routine `TBAJLI`.

## 5.3 Examples

That is to say the table: T3

Wi th	B	C
x1	x2	
x6	x7	x8

One adds a line to T3 :

```
CAL TBAJLI (T3,1, 'B', IBID, z1, cbid, kbid, 0)
```

The table then is obtained:

With	B	C
x1	x2	
x6	x7	x8
	z1	

One recovers the number of the line such as A=x1:

```
CAL TBNULI (T3,1, 'with', IBID, x1, cbid, kbid, 1.d-6, 'RELA',  
ilig)
```

The line ilig is modified:

```
CAL TBAJLI (T3,1, 'It, IBID, z2, cbid, kbid, ilig)
```

The table then is obtained:

With	B	C
		z2
x6	x7	x8
	z1	

## 6 Routines of reading of values in a table

### 6.1 Routine TBEXVE: Reading of the values of a column of a table

**TBEXVE** (nomtab, para, nomobj, basobj, nbval, typval)

nomtab	In jxin	K19	Name of the table in which one wants to extract a column
para	in	K16	Parameter indicating the column to be extracted
nomobj	In jxout	K24	Name of the object JEVEUX containing the values read in the table
basobj	in	K1	Base 'G', 'V' on which one created the vector nomobj
nbval	out	I	Many extracted values
typval	out	K4	Type JEVEUX extracted values: I/R/C/K8, K16,...

### 6.2 Routine TBEXFO: Create a function starting from a table

**TBEXFO** (nomtab, parax, paray, nomfo, interp, prolgd, basfon)

nomtab	In jxin	K19	Name of the table in which one wants to extract a function.
parax	in	K16	Parameter X-coordinate indicating the column to be extracted
paray	in	K16	Parameter ordered indicating the column to be extracted

nomfo	In jxout	K24	Name of the function to be created
interp	in	K1	
prolgd	in	K1	
basfon	in	K1	Base 'G', 'V' on which one creates the function

## 6.3 Routine TBLIVA: Reading of the value of a cell

**TBLIVA** (nomtab, npacri, lipacr, VI, vr, vc, vk, lcrit, lprec, para, ctype, vali, valr, valc, valk, lst)

nomtab	In jxin	K19	Name of the table in which one wants to extract the value from a cell
npacri	in	I	Many parameters implied in the selection criteria of the line (dimension of lipacr)
lipacr	in	V (K16)	List of the parameters criteria
VI	in	V (I)	Values of the criteria for the parameters 'I'
vr	in	V (R)	Values of the criteria for the parameters 'R'
vc	in	V (C)	Values of the criteria for the parameters 'C'
vk	in	V (K*)	Values of the criteria for the parameters 'K'
lcrit	in	V (K8)	List of the secondary criteria of equality for the floating parameters: 'EQUAL', 'RELA', 'ABSO' to see §7.1.1 Typical case of the criteria of equality
lprec	in	V (R)	List of the precise details (for the criteria of equality of the floating parameters to see with the §7.1.1 Typical case of the criteria of equality...)
para	in	K16	Parameter associated with the column with the sought value
ctype	out	K8	Type of the found value
vali	out	I	Found value if parameter 'I'
valr	out	R	Found value if parameter 'R'
valc	out	C	Found value if parameter 'C'
valk	out	K*	Found value if parameter 'K'
lst	out	I	Code return: 0 : OK 1 : para do not exist in the table for the found line 2 : no the line found corresponding to the criteria 3 : several lines found corresponding to the criteria

This routine makes it possible to read the value associated with a parameter given for a line selected in a table. One selects the line by imposing values on certain parameters. The mechanism of selection of a line in a table (arguments lipacr, lcrit, lprec, VI, vr, vc, vk) is explained to the §7.1.1 Typical case of the criteria of equality...

Example:

That is to say the table: T3

Wit h	B	C
7	4	'Z'
12	0	'A1'
	4	'A2'

```
lipacr (1) = ' A', vr (1) =6.999, lcrit (1) = ' RELA', lprec (1) =0.01  
lipacr (2) = ' B', VI (1) = 4  
CAL TBLIVA (T3,2, lipacr, VI, vr, vc, vk, lcrit, lprec, 'It, ctype, vali,  
& valr, valc, valk, lst)
```

at exit one a:

```
valk=' Z'  
ier=0  
ctype=' K8'
```

## 7 Routines of filtering and sorting of a table

### 7.1 Mechanism of filtering of lines in a table

In routines TBLIVA, TBNULI and TBEXTB, it is necessary “to filter” an existing table to retain only one of them (or several) lines. It is this mechanism which we explain here.

To filter a table, the user forces criteria on certain parameters. He will say for example to retain only the lines of the table for which NUME\_ORDRE=1. He can use several selection criteria of the lines and the same parameter can appear several times in the list of the criteria.

The possible “types” of selection criterion are:

EQ	“equality” for the entireties, the texts, realities or the complexes. For the floating numbers (real or complex), this criterion is supplemented by one “secondary” criterion explained below cf §7.1.1 Typical case of the criteria of equality...).
	“not-equality” (cf. EQ)
LT	“smaller than” Relations of order: - natural for the entireties and realities - alphabetical for the texts - invalid for the complexes
WP	“larger than” (cf. LT)
	“smaller or equal to” (cf. LT)
GE	“larger or equal to” (cf. LT)
VACUUM	blank cell
NON_VIDE	Nonempty cell

Note:

*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.*

Copyright 2021 EDF R&D - Licensed under the terms of the GNU FDL (<http://www.gnu.org/copyleft/fdl.html>)

In the routines *TBLIVA* and *TBNULI*, the type of selection criteria is (for the moment) always the equality ( 'EQ' ). The list of the criteria given in these 2 routines ( *lcrpt* ) of the “secondary” criteria (cf § is thus the list 7.1.1 Typical case of the criteria of equality...).

Arguments: VI, vr, vc, vk of these routines contain the values associated with these criteria (according to the type associated with each parameter on which carry the criteria).

We will explain the use of all these arguments on an example (without floating numbers):

That is to say a table containing of the parameters I1, I2, I3 of “whole” type and K1, K2 of standard “character”.

One wants to extract from them the lines which satisfy the following criteria:

- the value of K1 is different from 'ACTION1'
- the value of I2 lies between 12 and 21
- the value of I3 is worth 999
- the value of I1 is “nonempty”
- the value of K2 is “higher” (within the meaning of the alphabetical order) than 'III'

One will give like arguments of TBEXTB:

```
npacri=6 There are 6 selection criteria because the 2nd criterion is double: (x>12) and
(x<21)
lipacr = ('K1', 'I2', 'I2', 'I3', 'I1', 'K2')
lcrpa = ('', 'WP', 'LT', 'EQ', 'NON_VIDE', 'WP')
vk (1) = ' ACTION1', VI (1) =12, VI (2) =21, VI (3) =999, vk (2) = '
III'
```

Note:

Types of criterion 'VACUUM' and 'NON\_VIDE' do not require arguments in the tables VI, vr,  
...

## 7.1.1 Typical case of the criteria of equality (or nonequality) for “floating” numbers

The equality of the floating numbers is a dangerous concept in data processing because it can depend on certain truncations: errors rounding for example. For these selection criteria, one thus uses a list of secondary criteria which specify which equality is desired. There are three possible types for the secondary criteria:

'EQUA L'	“exact” equality of the 2 floating numbers
'ABSO'	equality of the 2 floating numbers except for an epsilon (eps) in absolute comparison: truth if $ x1-x2  < eps$
'RELA'	equality of the 2 floating numbers except for an epsilon (eps) in relative comparison: truth if $ x1-x2  < eps *  x1 $

Example:

That is to say a table containing the whole parameters: I1 and I2, real parameters R1 and R2 and complex parameters C1 and C2. One wants to extract the lines from them corresponding to the following criteria:

1. I1 > 12
2. C1 = c1 to 0.01 near in “absolute”

3. R1 = r1 to 0.1 near in "relative"
4. C2/= c2 with 0.02 near in "relative" (the numbers which are not in the disc of centre c2 and of 0.002\*c2)
5. R2/= r2 to 0.2 near in "absolute" (the numbers which are not in the interval of centre r2 and of 0.2)

One will give like arguments of TBEXTB:

```
npacri = 5"  
lipacr = ('I1', 'C1', 'R1', 'C2', 'R2')"  
licrpa = ('WP', 'EQ', 'EQ', '', '')"  
VI (1) = 12, vr (1) =r1, vr (2) =r2, vc (1) =c1, vc (2) =c2"  
lcrit = ('ABSO', 'RELA', 'RELA', 'ABSO')"  
lprec = (0.01, 0.1,0.02,0.2)"
```



## 7.2 Routine TBEXTB: Filtering and extraction of a new table

**TBEXTB** (*tabin*, *basout*, *tabout*, *npacri*, *lipacr*, *lcrpa*, *VI*, *vr*, *vc*, *vk*, *lprec*, *lcrit*, *iret*)

<i>tabin</i>	in jxin	K19	Name of the table which one wants to extract from the lines
<i>basout</i>	in	K1	'G', 'V' : base creation of <i>tabout</i>
<i>tabout</i>	In jxout	K19	Name of the table which will contain the lines extracted from <i>tabin</i>
<i>npacri</i>	in	I	Many parameters implied in the criteria of extraction (dimension of <i>lipacr</i> and of <i>lcrpa</i> )
<i>lipacr</i>	in	V (K16)	List of the parameters criteria cf §7 Routines of filtering and sorting of a table
<i>lcrpa</i>	in	V (K10)	List of the selection criteria: EQ, LT, WP, IT, GE, EMPTY, NON_VIDE The significance of these criteria is given to the §7 Routines of filtering and sorting of a table.
<i>VI</i>	in	V (I)	Values of the criteria for the parameters 'I'
<i>vr</i>	in	V (R)	Values of the criteria for the parameters 'R'
<i>vc</i>	in	V (C)	Values of the criteria for the parameters 'C'
<i>vk</i>	in	V (K)	Values of the criteria for the parameters 'K'
<i>lprec</i>	in	V (R)	List of the precise details (for the criteria of equality of the floating parameters cf §7.1.1 Typical case of the criteria of equality...)
<i>lcrit</i>	in	V (K8)	List of the secondary criteria of equality for the floating parameters: 'EQUAL', 'RELA', 'ABSO' cf §7.1.1 Typical case of the criteria of equality...
<i>lst</i>	out	I	Code return: 0: OK 1: para do not exist in the table for the found line 2: pas de line found corresponding to the criteria 3: several lines found corresponding to the criteria

When one recovered in *tabin* the lines satisfying all the criteria given, one does not preserve in the table *tabout* that the columns for which there exists at least a value. The same parameter can appear several times in the list of the criteria (*lcrpa*).

Example:

That is to say the table: T3

With	B	C
	4	'Z'
11.	0	'A1'
24.	9	'A2'
240.	9	'A2'
12.	12.	

```
lipacr (1) = ' A'
lcrpa (1) = ' NON_VIDE'
```

```
lipacr (2) = ' B'
lcrpa (2) = ' GT'
VI (1) = 1
```

```
lipacr (3) = ' A'
lcrpa (3) = ' LE'
vr (1) = 1.d2
```

```
CAL TBEXTB (T3, 'G', T3B, 3, lipacr, lcrpa, VI, vr, vc, vk, lprec,
lcrit, iret)
```

at exit T3B contains:

Wit h	B	C
24.	9	'A2'
12.	12.	

## 7.3 Routine TBTRTB: Sorting of the table

**TBTRTB** (*tabin*, *basout*, *tabout*, *will* *npara*, *lipara*, *lcrit*, *prec*, *crit*)

<i>tabin</i>	<i>in</i> <i>jxin</i>	K19	Name of the table which one wants to classify the lines
<i>basout</i>	<i>in</i>	K1	Base creation of <i>tabout</i> : G/V/L
<i>tabout</i>	<i>in</i> <i>jxout</i>	K19	Name of the table containing all the lines of <i>tabin</i> classified according to the following criteria
<i>will</i> <i>npara</i>	<i>in</i>	I	Many parameters implied in the sort criteria (dimension of <i>will</i> <i>lipara</i> and <i>lcrit</i> )
<i>will</i> <i>lipara</i>	<i>in</i>	V (K16)	List of the parameters criteria
<i>lcrit</i>	<i>in</i>	V (K2)	List of the types of criteria: 'CR' or 'DR.' 'CR': ascending order 'DR.': decreasing order
<i>prec</i>	<i>in</i>	R	precision
<i>crit</i>	<i>in</i>	K8	criterion = 'ABSOLUTE' or 'RELATIVE'

This routine is used to create a new table (*tabout*) by permuting the order of the lines of an existing table (*tabin*) according to certain sort criteria.

### Sorting

- Values R, I : By ascending values or decreasing
- Values K8, K16, K24, K32 : Alphabetically growing or decreasing

### Example:

```
will npara = 2
will lipara = ('NODE', 'INST')
lcrit = ('CR', 'DR.')
```

The new table (tabout) will be ordered:

- firstly alphabetically crescent of the names of nodes,
- secondly by decreasing order of the moments of calculation.

Note:

The 2nd criterion does not apply that if there is equality within the meaning of the first criterion.  
The blank cells are classified in "head" (they are smallest for the relation of order).

## 8 Routine TBIMPR: Impression of a table

**TBIMPR** (table, nopase, formaz, ifr, nparim, lipaim, nbparg, lipapg, formar, formac)

IN	table	K19	name of the table which one wants to print
IN	npase	K*	Name of the possible associated significant parameter
IN	formaz	K8	format of impression of the table ('EXCEL', 'TABLE', 'MOT_CLE')
IN	ifr	K.	logical unit of impression
IN	nparim	I	Many parameters to be printed: if nparim=0, ALL the parameters are printed.
IN	lipaim	V (K16)	List of the parameters to be printed. The order of impression of the parameters is that of the list lipaim
IN	npargp	I	Many parameters of "pagination"
IN	lipapg	V (K16)	List of the parameters of pagination
IN	formar	K8	Format of writing of the actual values. If formar=' → value by default: 1PE12.5
IN	formac	K2	Convention of writing of the complex numbers: If formar = " → value by default: 'IH' If formar = 'IH' → (left real, imaginary part) If formar = 'MP' → (module, phase)

The user has the possibility of printing his results under the following formats:

### 8.1 FORMAT: 'EXCEL'

NUME_ORDRE	INST	NODE	DX	DY
1	4.	N7	3.4	3.8
1	4.	N4	2.4	2.8
1	4.	N2	1.4	1.8
4	8.	N7	3.4	3.8
4	8.	N4	2.4	2.8
4	8.	N2	1.4	1.8
7	20.	N7	3.4	3.8
7	20.	N4	2.4	2.8
7	20.	N2	1.4	1.8

### 8.2 FORMAT: 'TABLE'

DX	INST 4. 8. 20.
NODE	
N7	3.4.3.4.3.4
N4	2.4.2.4.2.4
N2	1.4.1.4.1.4

## 8.3 FORMAT: 'MOT\_CLE'

```
NUME_ORDRE: 1 INST: 4. NODE: N7 DX: 3.4 DY: 3.8  
NUME_ORDRE: 1 INST: 4. NODE: N4 DX: 2.4 DY: 2.8  
NUME_ORDRE: 1 INST: 4. NODE: N2 DX: 1.4 DY: 1.8  
NUME_ORDRE: 4 INST: 8. NODE: N7 DX: 3.4 DY: 3.8  
...
```

## 8.4 FORMAT 'EXCEL' with pagination (here definite by the parameter 'NODE')

NODE: N7

NUME_ORDRE	INST	DX	DY
1	4.	3.4	3.8
4	8.	3.4	3.8
7	20.	3.4	3.8

NODE: N4

NUME_ORDRE	INST	DX	DY
1	4.	2.4	2.8
4	8.	2.4	2.8
7	20.	2.4	2.8

NODE: N2

NUME_ORDRE	INST	DX	DY
1	4.	1.4	1.8
4	8	1.4	1.8
7	20.	1.4	1.8

By default the format of impression is format 'EXCE, i.e. presentation in columns of the various selected parameters.

## 9 Routine "TBEXLR": Transformation of a table into a "SD LISTR8"

### 9.1 Goal of the routine

To reach (the level quickly as of routines `te00ij` for example) with contained information in tables, one can transform into cubes these tables SD of the type `LISTR8`. It is the object of the routine `TBEXLR`.

Note:

The reading of this paragraph 8 can be jumped by all those which are not interested in the very particular routine which is `TBEXLR`.

## 9.2 Interface of `TBEXLR`

`TBEXLR` (`table`, `lr8`, `base`)

IN	<code>table</code>	K19	Name of the table which one wants to transform into SD LISTR8
IN JXOUT	<code>lr8</code>	K8	Name of SD LISTR8 result
IN	<code>base</code>	K1	Base creation of <code>lr8</code> : 'G ', 'V ', ...

## 9.3 Restrictions of use of `TBEXLR`

The routine `TBEXLR` cannot transform a table into `LISTR8` that if this table is "diagonal per blocks". If it is not the case, the routine will stop in fatal error.

Examples of diagonal tables per blocks

With	B	C	D	E	F
1.	2.				
2.1	3.				
			8.	9.	
			10.	11.	
					19.

With	B	C	D	E
1.				
	3.			
			10.	11.

Examples of nondiagonal tables per blocks

With	B	C	D	E	F
1.	2.				
2.1	3.		5.	7.	
			8.	9.	
			10.	11.	
					19.

Wit h	B	C	D	E	F
1.	2.				
2.1	3.				
			8.	9.	
			10.	11.	
	4.				19.

## 9.4 How a table it is transformed into LISTR8 ?

That is to say the table

Parameter	B	C	D	E	F	G	H	I	K
Values type	K1	R	I	K1	R	R	I	I	R
	W	1.	2	S	3.				
		2.	2	S	3.				
	W	3.	2	S	3.				
				S			1	2	
				S			1	3	
				S			1	4	
				S			1	5	
				S	12.				
					13.				
					14.				
				S	15.				
				S	16.				

**stage 1: one retains only the nonempty columns of type 'I' or 'R'**

Parameter	C	D	F	H	I	K
Values type	R	I	R	I	I	R
	1.	2	3.			
	2.	2	3.			
	3.	2	3.			
				1	2	
				1	3	
				1	4	
				1	5	
						12.
						13.
						14.
						15.
						16.

## stage 2: recognition of the "blocks" of the "diagonal" table

Parameter	C	D	F	H	I	K
Values type	R	I	R	I	I	R
	1.	2	3.			
	2.	2	3.			
	3.	2	3.			
				1	2	
				1	3	
				1	4	
				1	5	
.						
.						
						12.
						13.
						14.
						15.
						16.

## stage 3: setting in vector of the values selected, conversion of the entireties into realities:

One puts end to end in the vector of realities the digital values found in the blocks of the table:

```
<filenb_blocs, nb_col (bloc1), nb_lig (bloc1), values (bloc1), nb_col  
(bloc2), nb_lig (bloc2), values (bloc2),...
```

The values of a block are written line by line. The table above becomes the vector of realities then below.

```
3.  
3.    3.    1.    2.    3.    2.    2.    3.    3.    2.    3.  
2.    4.    1.    2.    1.    3.    1.    4.    1.    5.  
1.    5.    12.   13.   14.   15.   16.
```

### Note:

We presented the list of realities on 4 lines to facilitate the reading of the result of the transformation, but `LISTR8` produced contains  $(3*3+2*4+1*5) + 3*2 + 1$  simply end to end put realities.



## 10 Simple example of creation and exploitation of a table

That is to say the example of the table 'T2' :

ACTION	NUME_ORDRE	INST	NODE	DX	DY	MESH	SIXX
ENTITLE 1	1	10.	N1	3.	5.		
ENTITLE 1	1	10.	N2	6.	7.		
ENTITLE 1	1	10.	N3	8.	9.		
ENTITLE 1	2	20.	N1	11.	12.		
ENTITLE 1	2	20.	N2	15.	13.		
ENTITLE 1	2	20.	N3	19.	18.		
ENTITLE 2	5	20.				MA1	-12.
ENTITLE 2	5	20.				MA2	-14.

### Notice preliminary:

The detailed interfaces of the utility routines are given to the paragraph [§2]. But the few lines of FORTRAN below can be understood without their reading.

- Declaration of the table on the basis TOTAL :

```
CAL TBCRSD ('T2', 'G')
```

- Declarations of the parameters of the table and the types of their data:

```
CAL TBAJPA ('T2', 1, 'ACTION', 'K8')  
CAL TBAJPA ('T2', 1, 'NUMÉRIQUE_ORDRE', 'I')  
CAL TBAJPA ('T2', 1, 'INST', 'R')  
CAL TBAJPA ('T2', 1, 'NODE', 'K8')  
CAL TBAJPA ('T2', 1, 'DX', 'R')  
CAL TBAJPA ('T2', 1, 'DY', 'R')  
CAL TBAJPA ('T2', 1, 'MESH', 'K8')  
CAL TBAJPA ('T2', 1, 'SIXX', 'R')
```

- Addition of the lines in the table:

```
REAL*8 VR (3)  
CHARACTER*8 VK (2)  
INTEGER VI (1)  
CHARACTER*16 LPARA 1(6), LPARA 2(5)  
DATED LPARA1/'ACTION', 'NUMÉRIQUE_ORDRE', 'INST', 'NODE', 'DX',  
'DY',  
DATED LPARA2/'ACTION', 'NUMÉRIQUE_ORDRE', 'INST', 'MESH', 'SIXX',  
VK (1) = action  
IF action = intitule1  
  C nume_ordre = 1.2  
  VI (1) = nume_ordre  
  VR (1) = T = urgent (nume_ordre)  
  C node = N1, N2, N3  
  VK (2) = node  
  VR (2) = DX (node, T)  
  VR (3) = DY (node, T)  
  CAL TBAJLI ('T2', 6, LPARA1, VI, VR, CBID, VK, 0)  
CONTINUOUS  
CONTINUOUS
```

```
ELSE IF action = intitule2
  VI (1) = 5 = nume_ordre
  VR (1) = T =instant (5)
  C nets = MA1, MA2
  VK (2) = mesh
  VR (2) = SIXX (mesh, T)
  CAL TBAJLI ('T2', 5, LPARA2, VI, VR, CBID, VK, 0)
CONTINUOUS
ENDIF
```