Utilisation des structures de données Tables

Résumé :

Ce document décrit les routines permettant d'utiliser dans le fortran les structures de données table décrites dans le document [D4.02.05] Structure de données `sd_table` .

Table des matières

1 Généralités

1.1 Qu'est-ce qu'une table ?

Une table est une structure de données de caractère informatique permettant de stocker un ensemble de valeurs entières, réelles, complexes ou chaînes de caractères.

Une table est comparable à une base de données EXCEL (version 5), c'est-à-dire qu'on peut la voir comme une liste de colonnes (le terme de "colonne" est ici pour faire le rapprochement avec une liste EXCEL ; cela ne veut pas dire que l'impression d'une table se fasse toujours en colonne (voir le § 8 Impression d'une table)) en vis-à-vis. Chaque colonne a un nom de champ, que nous appelons paramètre, et contient des données similaires de type : I, R, C, K8, K16, K24 ou K32.

Exemple 1 : T1

NUME_ORDRE	INST	NOEUD	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.

Exemple 2 : T2

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

1.2 Quelques propriétés des tables

- Une table a un nombre limité de colonnes (ou de paramètres). Ces paramètres sont choisis par les développeurs des commandes créant des tables. Le nom d'un paramètre est une chaîne d'au plus 16 caractères.
- En revanche, le nombre de lignes d'une table est souvent « dynamique » : il dépend en général des choix de l'utilisateur : nœuds de dépouillement, instants de calculs, ...
- Les valeurs contenues dans une colonne d'une table sont toutes de même type FORTRAN : réels, complexes, entiers ou textes. On peut stocker sous forme « texte » dans une table des noms de SD Aster ou des noms d'objets JEVEUX ; par exemple des noms de fonctions.
- Une table est dite « pleine » lorsque toutes ses lignes contiennent des valeurs pour tous les paramètres de la table. La table `t1` ci-dessus est pleine. Une table qui n'est pas pleine est dite « creuse » (table `t2` ci-dessus).
- Les lignes d'une table sont naturellement ordonnées par leur ordre d'insertion dans la table (routine TBAJLI).
- Les colonnes d'une table sont naturellement ordonnées par l'ordre de déclaration de leurs paramètres (routine TBAJPA).
- Une table a au moins une ligne. Sur une ligne, il peut y avoir des cellules vides : le paramètre associé n'est pas affecté (table `t2` ci-dessus).

2 Liste des utilitaires pour les tables

Fonction	Nom
Créer une nouvelle table	TBCRSD
Déclarer les paramètres	TBAJPA
Récupérer l'existence et le type d'un paramètre	TBEXIP
Fusionner plusieurs tables en une seule	TBFUTB
Détruire une table	DETRSD
Dupliquer une table	COPISD
Construire une table en y ajoutant des lignes une à une	TBAJLI
Récupérer le numéro d'une ligne dans une table	TBNULI
Imprimer une table sur listing	TBIMPR
Filtrer les lignes d'une table en imposant des critères sur un ou plusieurs paramètres pour créer une nouvelle table de taille plus petite	TBEXTB
Trier les lignes d'une table en fonction de certains paramètres sélectionnés. Le résultat du tri est une table dont les lignes ont été réordonnées.	TBTRTB
Recueillir dans un vecteur les valeurs correspondant à un paramètre donné	TBEXVE
Créer une fonction à partir de 2 colonnes d'une table	TBEXFO
Lire la valeur associée à un paramètre donné pour une ligne d'une table	TBLIVA
Récupérer toutes les valeurs numériques d'une table dans une `SD LISTR8`	TBEXLR

3 Routines de gestion d'une table

3.1 Routine TBCRSD : Créer une nouvelle table

TBCRSD(nomtab, base)

nomtab	in	K19	Nom de la nouvelle table à créer Si elle existe déjà, on la détruit + émission <A>
base	in	K1	Base de création de la table table ('G','V',...)

3.2 Routine TBAJPA : Ajouter des paramètres à la table

TBAJPA (nomtab,nbpar,nompar,typpar)

nomtab	in	K19	Nom de la table où l'on veut ajouter des paramètres
nbpar	in	I	Nombre de paramètres à ajouter
nompar	in	V(K16)	Liste des noms des paramètres à ajouter
typpar	in	V(K8)	Liste des types des paramètres : 'R','I','C','K8','K16','K24','K32'

3.3 Routine TBEXIP : Tester l'existence et le type d'un paramètre

TBEXIP (nomtab,para,exist,typpar)

nomtab	in	K19	Nom de la table à examiner
para	in	K16	paramètre à tester
exist	out	L	.TRUE. : le paramètre existe déjà dans la table nomtab
typpar	out	K8	type des paramètres s'il existe déjà dans la table : 'R','I','C','K8','K16','K24','K32'

3.4 Routine TBFUTB : Fusionner plusieurs tables en une seule table

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

tabout	in	K19	Nom de la table que l'on veut créer
basout	in	K1	Base de création de la table tabout ('G','V',...)
ntab	in	I	Nombre de tables que l'on veut fusionner
ltabin	in	K19	Noms des tables que l'on veut fusionner
para	in	K16	Nouveau paramètre (facultatif) qui permettra de distinguer l'origine de chacune des lignes de la nouvelle table si para=' ' les arguments suivants ne sont pas utilisés.
typpar	in	K8	Type du nouveau paramètre (facultatif)
vi	in	V(I)	Liste des valeurs pour le nouveau paramètre 'I' (facultatif)
vr	in	V(R)	Liste des valeurs pour le nouveau paramètre 'R' (facultatif)
vc	in	V(C)	Liste des valeurs pour le nouveau paramètre 'C' (facultatif)

vk	in	V(K*)	Liste des valeurs pour le nouveau paramètre 'K' (facultatif)

Cette routine peut être pratique pour créer une table creuse, le développeur peut réaliser chaque sous-table séparément et utiliser la routine TBFUTB pour les fusionner en une seule table.

Exemple :

On veut fusionner les 2 tables : T1 et T2

A	B	C	D
x1	x2		x3
	x4		x5
x6	x7	x8	

A	B	E
y1	y2	y3
y4		y5

Si l'on écrit :

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

On obtient la table : T3

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

Si l'on écrit :

```
ltabin(1)=T1
ltabin(2)=T2
VK(1)='ACTION1'
VK(2)='ACTION2'
CALL TBFUTB (T3,'V',2,ltabin,'N','K8',ibid,rbid,cbid,VK)
```

On obtient la table : T3

N	A	B	C	D	E
ACTION1	x1	x2		x3	
ACTION1		x4		x5	
ACTION1	x6	x7	x8		
ACTION2	y1	y2			y3
ACTION2	y4				y5

Remarques sur l'ordre des lignes et des colonnes de la nouvelle table :

Les lignes de la nouvelle table sont ordonnées en mettant bout à bout les lignes des tables que l'on fusionne.
Pour ordonner les paramètres on adopte les règles suivantes :
- le nouveau paramètre (facultatif) est numéroté en premier,
- les paramètres de la 1ère table de `ltabin` *sont ensuite ajoutés dans l'ordre qu'ils ont dans* `ltabin(1)`
- les paramètres de la 2ème table de `ltabin` *sont ensuite ajoutés (sauf ceux déjà présents dans* `ltabin(1)` *) dans l'ordre qu'ils ont dans* `ltabin(2)`
- ...

3.5 Routine DETRSD : Détruire une table

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

3.6 Routine COPISD : Dupliquer une table

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

4 Routines de préparation des valeurs à écrire dans une table

4.1 Routine TBAJVA : Ajouter une valeur associée à un paramètre

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

table	in	K19	Nom de la table où l'on veut ajouter des valeurs
nbpara	in	I	Nombre de paramètres pour la ligne
nompar	in	K16	Nom du paramètre à écrire
vi	in	I	Valeur 'I' pour le paramètre donné
livi	in	V(I)	Liste des valeurs pour les paramètres 'I'
vr	in	R	Valeur 'R' pour le paramètre donné
livr	in	V(R)	Liste des valeurs pour les paramètres 'R'
vc	in	C	Valeur 'C' pour le paramètre donné
livc	in	V(C)	Liste des valeurs pour les paramètres 'C'
vk	in	K*	Valeur 'K' pour le paramètre donné
livk	in	V(K*)	Liste des valeurs pour les paramètres 'K'

Cette routine est utile lorsqu'on ne connaît pas l'ordre des paramètres dans la table. Ainsi, on associe une valeur à un paramètre qui est écrite dans un liste (à la bonne position). Pour écrire les valeurs, dans la table, il faut donner les listes des valeurs construites à TBAJLI.

L'appel à cette routine est encapsulé par les routines TBAJVC, TBAJVI, TBAJVK, TBAJVR.

4.2 Routine TBAJVC : Ajouter une valeur associée à un paramètre complexe

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

table	in	K19	Nom de la table où l'on veut ajouter des valeurs
nbpara	in	I	Nombre de paramètres pour la ligne
nompar	in	K16	Nom du paramètre à écrire
vc	in	C	Valeur 'C' pour le paramètre donné
livc	in	V(C)	Liste des valeurs pour les paramètres 'C'

4.3 Routine TBAJVI : Ajouter une valeur associée à un paramètre entier

TBAJVI (table,nbpara,nompar,vi,livi)

table	in	K19	Nom de la table où l'on veut ajouter des valeurs
nbpara	in	I	Nombre de paramètres pour la ligne
nompar	in	K16	Nom du paramètre à écrire
vi	in	I	Valeur 'I' pour le paramètre donné
livi	in	V(I)	Liste des valeurs pour les paramètres 'I'

4.4 Routine TBAJVK : Ajouter une valeur associée à un paramètre chaîne de caractères

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

table	in	K19	Nom de la table où l'on veut ajouter des valeurs
nbpara	in	I	Nombre de paramètres pour la ligne
nompar	in	K16	Nom du paramètre à écrire
vk	in	K*	Valeur 'K' pour le paramètre donné
livk	in	V(K*)	Liste des valeurs pour les paramètres 'K'

4.5 Routine TBAJVR : Ajouter une valeur associée à un paramètre réel

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

table	in	K19	Nom de la table où l'on veut ajouter des valeurs
nbpara	in	I	Nombre de paramètres pour la ligne
nompar	in	K16	Nom du paramètre à écrire
vr	in	R	Valeur 'R' pour le paramètre donné
livr	in	V(R)	Liste des valeurs pour les paramètres 'R'

4.6 Exemples d'utilisation

On souhaite ajouter dans une table les valeurs associées aux paramètres NUME_ORDRE, INST et G.

```
CALL TBAJVI(RESULT,NBPRUP,'NUME_ORDRE',IORD,LIVI)
CALL TBAJVR(RESULT,NBPRUP,'INST',TIME,LIVR)
CALL TBAJVR(RESULT,NBPRUP,'G',G,LIVR)

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

5 Routines d'écriture de valeurs dans une table

5.1 Routine TBAJLI : Ajouter une ligne à la table

TBAJLI (nomtab,nbpar,nompar,vi,vr,vc,vk,nume)

nomtab	in jxvar	K19	Nom de la table où l'on veut ajouter une ligne
nbpar	in	I	Nombre de paramètres pour la ligne
nompar	in	V(K16)	Liste des noms des paramètres de la ligne
vi	in	V(I)	Liste des valeurs pour les paramètres 'I'
vr	in	V(R)	Liste des valeurs pour les paramètres 'R'
vc	in	V(C)	Liste des valeurs pour les paramètres 'C'
vk	in	V(K*)	Liste des valeurs pour les paramètres 'K'
nume	in	I	/ 0 : on ajoute la ligne au bout de la table / i : on remplace la i^{ème} ligne de la table

Exemple :

Soit une table contenant les paramètres de type "entiers" 'I1', 'I2', 'I3', les paramètres "réels" 'R1' et 'R2' et les paramètres "chaînes de caractères" 'K1' et 'K2'

Supposons que l'on veuille y ajouter une ligne contenant : I2=i2, R1=r1, K2=k2, K1=k1.

On pourra appeler la routine TBAJLI avec les arguments :

```
nbpar = 4
nompar = ('I2','R1','K2','K1')
vi = (i2)
vr = (r1)
vk = (k2, k1)
nume = 0
```

5.2 Routine TBNULI : Permet de récupérer un numéro de ligne dans une table

TBNULI (tabin,npacri,lipacr,vi,vr,vc,vk,lprec,lcrit,nume)

tabin	in	K19	Nom de la table dont on veut récupérer une ligne
npacri	in	I	Nombre de paramètres impliqués dans les critères de sélection de la ligne (dimension de lipacr)
lipacr	in	V(K16)	Liste des noms des paramètres critères

vi	in	V(I)	Valeurs des critères pour les paramètres 'I'
vr	in	V(R)	Valeurs des critères pour les paramètres 'R'
vc	in	V(C)	Valeurs des critères pour les paramètres 'C'
vk	in	V(K)	Valeurs des critères pour les paramètres 'K'
lprec	in	V(R)	Liste des précisions (pour les critères d'égalité des paramètres flottants [cf. §7.1.1 Cas particulier des critères d'égalité ...].)
lcrit	in	V(K8)	Liste des critères secondaires d'égalité pour les paramètres flottants : 'EGAL', 'RELA', 'ABSO' [cf. §7.1.1 Cas particulier des critères d'égalité ...]..
nume	out	I	= 0 : Il n'y a pas de ligne correspondant aux critères. = i : La ligne i est la seule qui correspond aux critères de sélection < 0 : Il y a plusieurs lignes correspondant aux critères

On recherche une ligne dans la table tabin en imposant des conditions sur ses paramètres.

Le mécanisme de sélection d'une ligne dans une table (arguments lipacr, lcrit, lprec, vi, vr, vc, vk) est expliqué au §7 Routines de filtrage et de tri d'une table.

Lorsque cette ligne est trouvée (et unique), on rend son numéro ce qui permet de modifier cette ligne à l'aide de la routine TBAJLI.

5.3 Exemples

Soit la table : T3

A	B	C
x1	x2	
x6	x7	x8

On ajoute une ligne à T3 :

```
CALL TBAJLI (T3,1,'B',ibid,z1,cbid,kbid,0)
```

On obtient alors la table :

A	B	C
x1	x2	
x6	x7	x8
	z1	

On récupère le numéro de la ligne telle que A=x1 :

```
CALL TBNULI (T3,1,'A',ibid,x1,cbid,kbid,1.d-6,'RELA',ilig)
```

On modifie la ligne ilig :

```
CALL TBAJLI (T3,1,'C',ibid,z2,cbid,kbid,ilig)
```

On obtient alors la table :

A	B	C
		z2
x6	x7	x8
	z1	

6 Routines de lecture de valeurs dans une table

6.1 Routine TBEXVE : Lecture des valeurs d'une colonne d'une table

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

nomtab	In jxin	K19	Nom de la table dans laquelle on veut extraire une colonne
para	in	K16	Paramètre désignant la colonne à extraire
nomobj	In jxout	K24	Nom de l'objet JEVEUX contenant les valeurs lues dans la table
basobj	in	K1	Base 'G','V' sur laquelle on créée le vecteur nomobj
nbval	out	I	Nombres de valeurs extraites
typval	out	K4	Type JEVEUX des valeurs extraites : I/R/C/K8,K16,...

6.2 Routine TBEXFO : Crée une fonction à partir d'une table

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

nomtab	In jxin	K19	Nom de la table dans laquelle on veut extraire une fonction.
parax	in	K16	Paramètre abscisse désignant la colonne à extraire
paray	in	K16	Paramètre ordonnée désignant la colonne à extraire
nomfo	In jxout	K24	Nom de la fonction à créer
interp	in	K1	
prolgd	in	K1	
basfon	in	K1	Base 'G', 'V' sur laquelle on crée la fonction

6.3 Routine TBLIVA : Lecture de la valeur d'une cellule

TBLIVA (nomtab,npacri,lipacr,vi,vr,vc,vk,lcrit,lprec,para,ctype,
vali,valr,valc,valk,ier)

nomtab	In jxin	K19	Nom de la table dans laquelle on veut extraire la valeur d'une cellule
npacri	in	I	Nombre de paramètres impliqués dans les critères de choix de la ligne (dimension de lipacr)
lipacr	in	V(K16)	Liste des paramètres critères
vi	in	V(I)	Valeurs des critères pour les paramètres 'I'
vr	in	V(R)	Valeurs des critères pour les paramètres 'R'
vc	in	V(C)	Valeurs des critères pour les paramètres 'C'
vk	in	V(K*)	Valeurs des critères pour les paramètres 'K'
lcrit	in	V(K8)	Liste des critères secondaires d'égalité pour les paramètres flottants : 'EGAL', 'RELA', 'ABSO' voir §7.1.1 Cas particulier des critères d'égalité
lprec	in	V(R)	Liste des précisions (pour les critères d'égalité des paramètres flottants voir au §7.1.1 Cas particulier des critères d'égalité ...)
para	in	K16	Paramètre associé à la colonne de la valeur cherchée
ctype	out	K8	Type de la valeur trouvée
vali	out	I	Valeur trouvée si paramètre 'I'
valr	out	R	Valeur trouvée si paramètre 'R'
valc	out	C	Valeur trouvée si paramètre 'C'
valk	out	K*	Valeur trouvée si paramètre 'K'
ier	out	I	Code retour : 0 : OK 1 : para n'existe pas dans la table pour la ligne trouvée 2 : pas de ligne trouvée correspondant aux critères 3 : plusieurs lignes trouvées correspondant aux critères

Cette routine permet de lire la valeur associée à un paramètre donné pour une ligne sélectionnée dans une table. On sélectionne la ligne en imposant des valeurs à certains paramètres. Le mécanisme de sélection d'une ligne dans une table (arguments lipacr, lcrit, lprec, vi, vr, vc, vk) est expliqué au §7.1.1 Cas particulier des critères d'égalité ....

Exemple :

Soit la table: T3

A	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
  CALL TBLIVA(T3,2,lipacr,vi,vr,vc,vk,lcrit,lprec,'C',ctype,vali,
&              valr,valc,valk,ier)
```

en sortie on a :

```
  valk='Z'
  ier=0
```

```
ctype='K8'
```

7 Routines de filtrage et de tri d'une table

7.1 Mécanisme de filtrage de lignes dans une table

Dans les routines TBLIVA, TBNULI et TBEXTB, il est nécessaire de “filtrer” une table existante pour n’en retenir qu’une (ou plusieurs) lignes. C’est ce mécanisme que nous expliquons ici.

Pour filtrer une table, l’utilisateur impose des critères à certains paramètres. Il dira par exemple de ne retenir que les lignes de la table pour lesquelles NUME_ORDRE=1. Il peut utiliser plusieurs critères de sélection des lignes et un même paramètre peut apparaître plusieurs fois dans la liste des critères.

Le “types” de critère de choix possibles sont :

EQ	“égalité” pour les entiers, les textes, les réels ou les complexes. Pour les nombres flottants (réels ou complexes), ce critère est complété par un critère “secondaire” expliqué ci-dessous Cf. §7.1.1 Cas particulier des critères d'égalité ...).
NE	“non-égalité” (Cf. EQ)
LT	“plus petit que” Relations d’ordre : - naturelle pour les entiers et les réels - alphabétique pour les textes - invalide pour les complexes
GT	“plus grand que” (Cf. LT)
LE	“plus petit ou égal à” (Cf. LT)
GE	“plus grand ou égal à” (Cf. LT)
VIDE	cellule vide
NON_VIDE	Cellule non vide

Remarque :

Dans les routines TBLIVA et TBNULI , le type de critère de sélection est (pour l’instant) toujours l’égalité ('EQ'). La liste des critères donnée dans ces 2 routines (lcrit) est donc la liste des critères “secondaires” (Cf. § 7.1.1 Cas particulier des critères d'égalité ...).

Les arguments : vi, vr, vc, vk de ces routines contiennent les valeurs associées à ces critères (selon le type associé à chacun des paramètres sur lesquels portent les critères).

Nous allons expliquer l’utilisation de tous ces arguments sur un exemple (sans nombres flottants) :

Soit une table contenant des paramètres I1, I2,I3 de type “entier” et K1,K2 de type “caractère”.

On veut en extraire les lignes qui satisfassent aux critères suivants :

- la valeur de K1 est différente de ‘ACTION1’
- la valeur de I2 est comprise entre 12 et 21
- la valeur de I3 vaut 999
- la valeur de I1 est “non vide”
- la valeur de K2 est “supérieure” (au sens de l’ordre alphabétique) à ‘III’

On donnera comme arguments de TBEXTB :

```
npacri=6 Il y a 6 critères de choix car le 2ème critère est double : (x>12) et (x<21)
lipacr = ('K1', 'I2', 'I2', 'I3', 'I1', 'K2')
lcrpa = ('NE', 'GT', 'LT', 'EQ', 'NON_VIDE', 'GT')
vk(1)='ACTION1', vi(1)=12, vi(2)=21, vi(3)=999, vk(2)='III'
```

Remarque :

Les types de critère 'VIDE' et 'NON_VIDE' ne nécessitent pas d'arguments dans les tableaux vi, vr, ...

7.1.1 Cas particulier des critères d'égalité (ou de non égalité) pour des nombres "flottants"

L'égalité des nombres flottants est une notion dangereuse en informatique car elle peut dépendre de certaines troncatures : erreurs d'arrondi par exemple. Pour ces critères de sélection, on utilise donc une liste de critères secondaires qui précisent quelle égalité est voulue. Il y a trois types possibles pour les critères secondaires :

'EGAL'	égalité "exacte" des 2 nombres flottants
'ABSO'	égalité des 2 nombres flottants à un epsilon près (eps) en comparaison absolue : vrai si `|x1-x2| < eps`
'RELA'	égalité des 2 nombres flottants à un epsilon près (eps) en comparaison relative : vrai si `|x1-x2| < eps * |x1|`

Exemple :

Soit une table contenant les paramètres entiers : `I1` et `I2`, les paramètres réels `R1` et `R2` et les paramètres complexes `C1` et `C2`. On veut en extraire les lignes correspondant aux critères suivants :

1. `I1 > 12`
2. `C1 = c1` à 0.01 près en "absolu"
3. `R1 = r1` à 0.1 près en "relatif"
4. `C2 /= c2` à 0.02 près en "relatif" (les nombres qui ne sont pas dans le disque de centre c2 et de rayon 0.002*c2)
5. `R2 /= r2` à 0.2 près en "absolu" (les nombres qui ne sont pas dans l'intervalle de centre r2 et de rayon 0.2)

On donnera comme arguments de TBEXTB :

```
npacri  =  5''
lipacr  =  ('I1', 'C1', 'R1', 'C2', 'R2')''
licrpa  =  ('GT', 'EQ', 'EQ', 'NE', 'NE')''
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 : Filtrage et extraction d'une nouvelle table

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

tabin	in jxin	K19	Nom de la table dont on veut extraire des lignes
basout	in	K1	'G','V' : base de création de tabout
tabout	In jxout	K19	Nom de la table qui contiendra les lignes extraites de tabin
npacri	in	I	Nombre de paramètres impliqués dans les critères d'extraction (dimension de lipacr et de lcrpa)
lipacr	in	V(K16)	Liste des paramètres critères cf. §7 Routines de filtrage et de tri d'une table
lcrpa	in	V(K10)	Liste des critères de sélection : EQ , LT, GT , NE, LE, GE, VIDE, NON_VIDE La signification de ces critères est donnée au §7 Routines de filtrage et de tri d'une table.
vi	in	V(I)	Valeurs des critères pour les paramètres 'I'
vr	in	V(R)	Valeurs des critères pour les paramètres 'R'
vc	in	V(C)	Valeurs des critères pour les paramètres 'C'
vk	in	V(K)	Valeurs des critères pour les paramètres 'K'
lprec	in	V(R)	Liste des précisions (pour les critères d'égalité des paramètres flottants cf. §7.1.1 Cas particulier des critères d'égalité ...)
lcrit	in	V(K8)	Liste des critères secondaires d'égalité pour les paramètres flottants : 'EGAL', 'RELA', 'ABSO' cf. §7.1.1 Cas particulier des critères d'égalité ...
ier	out	I	Code retour : 0 : OK 1 : para n'existe pas dans la table pour la ligne trouvée 2 : pas de ligne trouvée correspondant aux critères 3 : plusieurs lignes trouvées correspondant aux critères

Lorsque l'on a récupéré dans tabin les lignes satisfaisant à tous les critères donnés, on ne conserve dans la table tabout que les colonnes pour lesquelles il existe au moins une valeur. Un même paramètre peut apparaître plusieurs fois dans la liste des critères (lcrpa).

Exemple :

Soit la table: T3

A	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

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

en sortie T3B contient :

A	B	C
24.	9	'A2'
12.	12.	

7.3 Routine TBTRTB : Tri de la table

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

tabin	in jxin	K19	Nom de la table dont on veut classer les lignes
basout	in	K1	Base de création de tabout : G/V/L
tabout	in jxout	K19	Nom de la table contenant toutes les lignes de tabin classées selon les critères suivants
npara	in	I	Nombre de paramètres impliqués dans les critères de tri (dimension de lipara et de lcrit)
lipara	in	V(K16)	Liste des paramètres critères
lcrit	in	V(K2)	Liste des types de critères :'CR' ou 'DR' 'CR' : ordre croissant 'DR' : ordre décroissant
prec	in	R	précision
crit	in	K8	critère = 'ABSOLU' ou 'RELATIF'

Cette routine sert à créer une nouvelle table (tabout) en permutant l'ordre des lignes d'une table existante (tabin) selon certains critères de tri.

Tri

- Valeurs R,I : Par valeurs croissantes ou décroissantes
- Valeurs K8,K16,K24,K32 : Par ordre alphabétique croissant ou décroissant

Exemple :

```
          npara = 2
lipara = ('NOEUD','INST')
lcrit = ('CR','DR')
```

La nouvelle table (tabout) sera ordonnée :

- premièrement par ordre alphabétique croissant des noms de nœuds,

- deuxièmement par ordre décroissant des instants de calcul.

Remarques :

Le 2ème critère ne s'applique que s'il y a égalité au sens du premier critère.
Les cellules vides sont classées en "tête" (ce sont les plus petites pour la relation d'ordre).

8 Routine TBIMPR : Impression d'une table

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

IN	table	K19	nom de la table que l'on veut imprimer
IN	npase	K*	Nom de l'éventuel paramètre sensible associé
IN	formaz	K8	format d'impression de la table ('EXCEL','TABLEAU','MOT_CLE')
IN	ifr	K..	unité logique d'impression
IN	nparim	I	Nombre de paramètres à imprimer: si nparim=0, on imprime TOUS les paramètres.
IN	lipaim	V(K16)	Liste des paramètres à imprimer. L'ordre d'impression des paramètres est celui de la liste lipaim
IN	nparpg	I	Nombre de paramètres de "pagination"
IN	lipapg	V(K16)	Liste des paramètres de pagination
IN	formar	K8	Format d'écriture des valeurs réelles. Si formar=' ' → valeur par défaut : 1PE12.5
IN	formac	K2	Convention d'écriture des nombres complexes : Si formar = ' ' → valeur par défaut : 'RI' Si formar = 'RI' → (partie réelle, partie imaginaire) Si formar = 'MP' → (module, phase)

L'utilisateur a la possibilité d'imprimer ses résultats sous les formats suivants :

8.1 FORMAT : 'EXCEL'

```
NUME_ORDRE    INST    NOEUD    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 : 'TABLEAU'

DX	INST 4. 8. 20.
NOEUD 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. NOEUD: N7 DX: 3.4 DY: 3.8
 NUME_ORDRE: 1 INST: 4. NOEUD: N4 DX: 2.4 DY: 2.8
 NUME_ORDRE: 1 INST: 4. NOEUD: N2 DX: 1.4 DY: 1.8
 NUME_ORDRE: 4 INST: 8. NOEUD: N7 DX: 3.4 DY: 3.8
 ...
```

8.4 FORMAT 'EXCEL' avec pagination (définie ici par le paramètre 'NOEUD')

```
NOEUD: N7

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

NOEUD: N4

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

NOEUD: N2

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

Par défaut le format d'impression est le format 'EXCEL', c'est à dire présentation en colonnes des différents paramètres sélectionnés.

9 Routine "TBEXLR" : Transformation d'une table en une "SD LISTR8"

9.1 But de la routine

Pour accéder rapidement (au niveau des routines te00ij par exemple) à des informations contenues dans des tables, on peut transformer ces tables en des SD de type LISTR8 .C'est l'objet de la routine TBEXLR.

Remarque :

> *La lecture de ce paragraphe 8 peut être sautée par tous ceux qui ne s'intéressent pas à la routine très particulière qu'est TBEXLR .*

9.2 Interface de TBEXLR

TBEXLR (table,lr8,base)

IN	table	K19	Nom de la table que l'on veut transformer en SD LISTR8
IN JXOUT	lr8	K8	Nom de la SD LISTR8 résultat
IN	base	K1	Base de création de lr8 : 'G', 'V', ...

9.3 Restrictions d'utilisation de TBEXLR

La routine TBEXLR ne peut transformer une table en LISTR8 que si cette table est « diagonale par blocs ». Si ce n'est pas le cas, la routine s'arrêtera en erreur fatale.

Exemples de tables diagonales par blocs

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

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

Exemples de tables non diagonales par blocs

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

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

9.4 Comment une table est-elle transformée en LISTR8 ?

Soit la table

Paramètre	B	C	D	E	F	G	H	I	K
Type valeurs	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.				

étape 1 : on ne retient que les colonnes non vides de type 'I' ou 'R'

Paramètre	C	D	F	H	I	K
Type valeurs	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.

étape 2 : reconnaissance des « blocs » de la table « diagonale »

Paramètre	C	D	F	H	I	K
Type valeurs	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.

étape 3 : mise en vecteur des valeurs retenues, conversion des entiers en réels :

On met bout à bout dans le vecteur de réels les valeurs numériques trouvées dans les blocs de la table :

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

Les valeurs d'un bloc sont écrites ligne par ligne. La table ci-dessus devient alors le vecteur de réels ci-dessous.

```
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.
```

Remarque :

Nous avons présenté la liste de réels sur 4 lignes pour faciliter la lecture du résultat de la transformation, mais la `LISTR8` *produite contient (3*3+2*4+1*5) + 3*2 + 1 réels simplement mis bout à bout.*

10 Exemple simple de création et d'exploitation d'une table

Soit l'exemple de la table `'T2'` :

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

Remarque préliminaire :

Les interfaces détaillées des routines utilitaires sont données au paragraphe [§2]. Mais les quelques lignes de fortran ci-dessous peuvent se comprendre sans leur lecture.

- Déclaration de la table sur la base `GLOBALE` :

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

- Déclarations des paramètres de la table et des types de leurs données :

```
CALL TBAJPA('T2',1,'ACTION','K8')
CALL TBAJPA('T2',1,'NUME_ORDRE','I')
CALL TBAJPA('T2',1,'INST','R')
CALL TBAJPA('T2',1,'NOEUD','K8')
CALL TBAJPA('T2',1,'DX','R')
CALL TBAJPA('T2',1,'DY','R')
CALL TBAJPA('T2',1,'MAILLE','K8')
CALL TBAJPA('T2',1,'SIXX','R')
```

- Ajout des lignes dans la table :

```
                    REAL*8 VR(3)
CHARACTER*8 VK(2)
INTEGER VI(1)
CHARACTER*16 LPARA1(6), LPARA2(5)
DATA LPARA1 /'ACTION','NUME_ORDRE','INST','NOEUD','DX','DY'/
DATA LPARA2 /'ACTION','NUME_ORDRE','INST','MAILLE','SIXX'/
VK(1) = action
IF action = intitule1
   DO nume_ordre = 1,2
      VI(1) = nume_ordre
      VR(1) = t = instant(nume_ordre)
      DO noeud = N1, N2, N3
         VK(2) = noeud
         VR(2) = DX(noeud,t)
         VR(3) = DY(noeud,t)
         CALL TBAJLI('T2',6,LPARA1,VI,VR,CBID,VK,0)
      CONTINUE
   CONTINUE
ELSE IF action = intitule2
```

```
      VI(1)= 5 = nume_ordre
      VR(1)= t =instant(5)
      DO maille = MA1, MA2
         VK(2) = maille
         VR(2) = SIXX(maille,t)
         CALL TBAJLI('T2',5,LPARA2,VI,VR,CBID,VK,0)
      CONTINUE
   ENDIF
```