

ICAB

MANUEL DE REFERENCE IGA

Table des Matières

1 Présentation des fichiers IGA	2
1.1 Structure générale des fichiers IGA	2
1.2 Syntaxe utilisée dans les fichiers IGA	3
1.3 Directives du préprocesseur IGA	6
2 Bibliothèque des Entités IGA	9
<i>CONSTRAINT, contraintes sur les degrés de liberté</i>	10
<i>ELEMENT, élément fini</i>	13
<i>FUNCTION, fonction $y=f(x)$</i>	23
<i>IMPORT, importer des projets ICAB</i>	25
<i>LOAD, chargement</i>	26
<i>NODE, noeud d'élément fini</i>	35
<i>NOTE, annotation</i>	36
<i>PROPERTY, propriétés et matériaux</i>	37
<i>REFERENCE, système de coordonnées ou repère</i>	49
<i>RESTRAINT, conditions limites</i>	51
<i>STEP, préparation d'un calcul</i>	53
<i>UNIT, système d'unités</i>	54
INDEX	55

1 Présentation des fichiers IGA

Les fichiers **IGA** (Initial General ASCII) contiennent les données nécessaires à la description de structures pour l'exécution de simulations numériques. Ces fichiers peuvent être créés et modifiés avec un programme d'édition de texte. La structure des fichiers est largement inspirée du format **IGES**⁽¹⁾.

Les informations du fichier IGA sont en ASCII. Le format ASCII permet la création et la modification des fichiers avec un éditeur de texte. En outre ce type de format est à l'heure actuelle le seul qui permet d'échanger des fichiers entre ordinateurs de manière fiable. Les fichiers ASCII s'opposent aux fichiers binaires qui ne sont modifiables que par des programmes et qui sont spécifiques à l'ordinateur sur lequel ils ont été créés. Les fichiers IGA garantissent ainsi une grande compatibilité entre systèmes informatiques hétérogènes.

Ce chapitre détaille la structure et la syntaxe des fichiers IGA. Le chapitre suivant explicite la bibliothèque des entités IGA, à savoir le détail des données pouvant être insérées dans un fichier IGA. Les fichiers IGA sont transformés en projets par l'application ICAB Edit.

1.1 Structure générale des fichiers IGA

Les informations contenues dans un fichier IGA sont structurées pour former une base de données comprenant une collection d'entités. Les entités, sorte de fiches élémentaires, comprennent des informations disposées selon une syntaxe imposée. Les entités sont organisées en **Données d'Entrée DE** et en **Paramètres des Données PD**. Dans la suite de ce document, les abréviations DE et PD seront utilisées.

1.1.1 Données d'Entrée DE

Les Données d'Entrées DE constituent un index qui permet de renseigner les informations contenues dans le fichier IGA. Une Donnée d'Entrée DE joue un rôle similaire à celui tenu par une *Directory Entry* de fichier IGES. Le nombre d'informations relatives aux DE sont limitées et sont semblables pour toutes les Données d'Entrées DE. Les sections DE servent d'identificateurs pour les sections Paramètres des données PD. Par exemple, une DE pour un noeud contient en particulier le numéro de ce noeud ou le nom attribué à ce noeud.

1.1.2 Paramètres des Données PD

Une section PD (*Parameter Data* pour les fichiers IGES) est obligatoirement associée à une DE et contient des informations requises pour la définition de l'entité en question. Par exemple, les paramètres PD pour un noeud contiennent les coordonnées précisant sa position.

1.1.3 Identificateur d'Entités ET

Chaque couple DE et PD constitue une **entité**. Les entités DE et PD sont regroupées en ensembles appelés ensembles de données (par exemple les noeuds, éléments, propriétés...). Avant chaque ensemble d'entités, une **Identification d'Entités ET** permet préciser des propriétés communes à tout un ensemble d'entités. La structure générale d'un fichier IGA est celle indiquée sur la figure ci-dessous.

⁽¹⁾ "Initial Graphics Exchange Specification (IGES) Version 5.1", sept. 1991, IGES/PDES Organization, National Institute of Standards and Technology, Gaithersburg, MD 20899, USA.

Ce format a été conçu pour des échanges de données entre logiciels de Conception et de Fabrication Assistées par Ordinateur (CFAO).

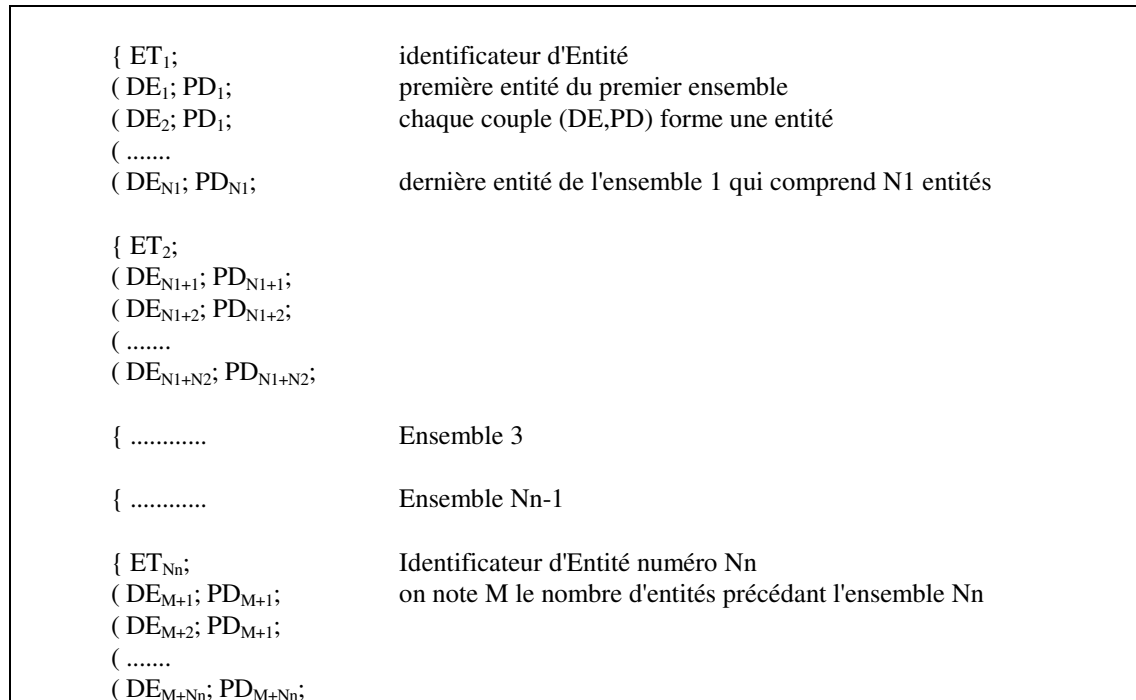


Figure 2 Structure schématique d'un fichier IGA

1.2 Syntaxe utilisée dans les fichiers IGA

Les informations contenues dans un fichier IGA apparaissent sous la forme de:

- chaînes de caractères,
- nombres entiers,
- de nombres réels,
- délimiteurs,
- de paramètres.

```

NODE();                               /* Identificateur d'Entité NOEUD */
N_A; 0.0, 0.0, 0.0;                   /* DE 1; suivie par PD 1 */
N_B; 1.0, 0.0, 0.0;
N_C; 2.0, 0.0, 0.0;

PROPERTY(TYPE=SPRING);                /* ID 2 */
spring1; K=1E5;
PROPERTY(TYPE=MASS);                   /* ID 3 */
mass1; MA=10.0;

ELEMENT(TYPE=SPRING,PROP=spring1);     /* ID 4 */
; N_A, N_B;                             /* la DE est vide et prend donc des */
; N_B, N_C;                             /* valeurs par défaut */
ELEMENT(TYPE=MASS,PROP=mass1);         /* ID 5 */
; N_A;
; N_C;

CONSTRAINT(TYPE=KINEMATICS)           /* ID 6 */
; N_A, X,Y,Z,RX,RY,RZ;
; N_B, Y,Z,RX,RY,RZ;
; N_C, Y,Z,RX,RY,RZ;

```

Figure 3 Exemple de fichier IGA

1.2.1 chaîne de caractères

Une chaîne de caractères est une suite de caractères parmi les listes ci-dessous:

caractères minuscules *a b c d e f g h i j k l m n o p q r s t u v w x y z*
 caractères majuscules *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z*
 caractères numériques *0 1 2 3 4 5 6 7 8 9*
 caractère de soulignement *_*

Une chaîne de caractères ne doit pas commencer par un caractère numérique.

exemples de chaînes de caractères: *cabine cab01 Cab_le*

1.2.2 nombre entier

Un nombre entier est une chaîne de caractères constituée de caractères numériques et précédée du signe moins *-* pour indiquer un nombre négatif, du signe plus *+* pour indiquer un nombre positif. Le signe plus est facultatif pour un nombre positif.

exemples de nombres entiers: *12 -382 0 +827*

1.2.3 nombre réel

Un nombre réel est un nombre comportant une partie décimale. La partie décimale est identifiée par le caractère point *.* et doit figurer même si la partie décimale est nulle de manière à distinguer un nombre réel d'un nombre entier. Ainsi *12* est interprété comme un nombre entier et *12.0* est comme un nombre réel. Lorsque les nombres réels sont très petits ou très grands, la puissance de 10 est précisée par la lettre majuscule **E**. La valeur un million deux cents mille (1.2×10^6) peut s'écrire *1200000.0* ou *1.2E6*. Un millionième (1.0×10^{-6}) peut s'écrire *0.000001* ou *1.0E-6*. Les nombres réels sont interprétés par ICAB Edit en simple précision, c'est-à-dire avec 7 à 8 chiffres significatifs (Mais les applications ICAB réalisent les calculs en double précision).

exemples de nombres réels: *12.0 -38.2 0.0 1.21233E4 +8.27E-4 -7.34E+6*

1.2.4 Délimiteurs

* Virgule *,*

La virgule sépare des chaînes de caractères, des nombres entiers ou des nombres réels.

exemple: *integer ,12 ,real , -12.323*

* Point-virgule *;*

Le point-virgule permet de délimiter la fin de Données d'entrées DE, de Paramètres de Données PD ou d'Identificateur d'Entités ET.

exemple: *NODE() ;
1,Node1 ; 1.0, 2.0, 3.0 ;*

* parenthèses *()*

Les parenthèses sont utilisées pour délimiter les arguments définis dans les Identificateurs d'Entités ET (cf paragraphe "syntaxe ET" ci-après).

exemple *ELEMENT (TYPE=SPRING);*

* Signe égal *=*

Le signe égal est utilisé pour l'affectation de valeurs dans un identificateur de paramètre (cf paragraphe suivant).

exemple: *TYPE=SPRING*

* dièse *#*

Le caractère dièse ne doit être employé qu'au début d'une ligne de manière à indiquer le début d'une directive (cf section sur les directives du préprocesseur ci-après).

exemple: *#define X1 1.5 /* cette ligne est une directive */
NODE()
1; X1, 0.0, 0.0;*

1.2.5 Identificateur de paramètre

Un identificateur de paramètre permet d'affecter des valeurs à des identificateurs d'entité ET, des Données d'Entrée DE ou des Paramètres de Données PD. Un paramètre, c'est-à-dire une information élémentaire (chaîne de caractères, nombres...) n'a qu'une portée limitée à l'entité dans laquelle il se trouve. Un identificateur de paramètre

comprend une chaîne de caractères suivie du signe égal =. Le nom de l'argument est une chaîne de caractères qui est interprétée en fonction du contexte. Les paramètres qui suivent le signe égal peuvent être des chaînes de caractères, des nombres entiers ou des réels.

exemple de paramètres: `ELEMENT(TYPE=BEAM_LINEAR, MAT=1)`

L'exemple comporte deux identificateurs de paramètre dont le premier a pour nom `TYPE` et pour argument une chaîne de caractères `BEAM_LINEAR`. Le deuxième identificateur de paramètre a pour nom `MAT` et son argument est un nombre entier `1`.

1.2.6 Syntaxe des Identificateurs d'Entité ET

Un identificateur d'Entité est constitué d'une chaîne de caractères et d'une suite de paramètres. Cette suite de paramètres est délimitée par des parenthèses (). Les parenthèses sont indispensables même si aucun paramètre n'est nécessaire. L'identificateur doit tenir sur une seule ligne.

Un identificateur ET a la forme générale suivante:

$$\text{identificateur } (\text{paramètre1, paramètre2, ... })$$

La chaîne de caractères placée avant la parenthèse ouvrante (est le nom de l'entité. Les paramètres indiqués entre parenthèses () permettent de préciser des propriétés qui seront communes à toutes les entités jusqu'à ce qu'un autre identificateur d'entité soit rencontré.

exemple: `NODE()` /* identificateur d'Entité sans paramètre */

`1; 1.0, 0.0, 0.0;`

`2; 2.0, 0.0, 0.0;`

`ELEMENT(TYPE=SPRING, PROP=spr1) /* avec paramètres */`

`; 1, 2;`

1.2.7 Syntaxe des Données d'Entrées DE, pointeurs

Une Donnée d'Entrée doit appartenir à un ensemble d'Entités, c'est-à-dire un ensemble dont le type d'entité est précisé par un identificateur ET. Une **DE commence en début de ligne** et se termine par un point virgule ;.

Une DE peut contenir un nombre entier et/ou une chaîne de caractère. Le nombre entier sert à donner un numéro à l'entité. La chaîne de caractère est appelée pointeur dont la longueur ne doit pas excéder 8 caractères de manière à préserver la compatibilité avec le format IGES. Le pointeur permet de nommer une entité. Pour faire référence à une entité, il est possible d'indiquer son pointeur. En l'absence de pointeur dans une DE, cette DE ne peut pas être identifiée par un pointeur. Si le numéro de DE n'est pas précisé, la valeur par défaut est zéro mais pour certaines entités une numérotation automatique est activée.

Attention: même si une DE est vide, le point virgule qui marque la fin de la DE reste nécessaire dans le fichier.

exemple: `NODE()`

`1; 1.0, 0.0, 0.0; // numéro 1`

`N2; 2.0, 0.0, 0.0; // pointeur N2`

`3,N3; 3.0, 0.0, 0.0; // numéro 3 et pointeur N3`

`ELEMENT(TYPE=SPRING, PROP=1)`

`; 1, N2; // DE vide -> numérotation automatique`

1.2.8 Syntaxe des Paramètres de Données PD

Les Paramètres de Données PD ne peuvent figurer qu'après une Donnée d'Entrée DE. Les Paramètres de Données se terminent par un point virgule ;. Ils peuvent contenir des chaînes de caractères, des nombres entiers ou réels, des paramètres. Il n'existe pas de syntaxe générale car les données dépendent du type d'entité. La description de la syntaxe des Paramètres de Données est détaillée dans le chapitre relatif à la bibliothèque des entités.

exemple: *NODE()*

```

1;      1.0, 0.0, 0.0;    // coordonnées du noeud 1
N2;     2.0, 0.0, 0.0;    // coordonnées du noeud N2
ELEMENT(TYPE=SPRING, PROP=1)
;       1, N2;           // élément contenant les noeuds 1 et N2

```

1.3 Directives du préprocesseur IGA

L'analyse d'un fichier IGA s'effectue en deux passes:

- La première passe, qui est prise en charge par le préprocesseur IGA, consiste à éliminer les commentaires et à analyser le fichier en fonction de directives. Une directive permet de paramétrer un fichier IGA, de programmer des lectures conditionnelles et de fusionner plusieurs fichiers IGA dans le même projet.
- La deuxième passe concerne l'analyse effective des entités ET, des Données d'Entrées DE et Paramètres de Données PD.

Une directive est interprétée grâce au caractère dièse # **situé au début** de la ligne contenant la directive. Les lignes de directives sont analysées lors de la première passe puis sont éliminées avant la deuxième passe. L'analyse de la première passe (préprocesseur) est séquentielle ce qui implique que seules les directives qui précèdent la ligne en cours peuvent avoir un effet sur cette ligne.

1.3.1 commentaires

Dans un fichier IGA, il est possible d'inclure des commentaires, c'est-à-dire du texte qui ne sera pas interprété mais qui permet de faire figurer des annotations utiles pour le lecteur. Les commentaires sont supprimés logiquement du fichier avant analyse.

La syntaxe employée est similaire à celle utilisée dans le langage de programmation C++. La première possibilité consiste à faire précéder le commentaire par la paire de symboles /* et à marquer la fin par les mêmes symboles tapés dans l'ordre inverse, soit */. Ce type de commentaire peut être développé sur plusieurs lignes. Les commentaires imbriqués ne sont pas permis.

La deuxième possibilité consiste à placer deux barres obliques consécutives `//`. La fin de la ligne après la double barre `//` sera analysée comme un commentaire. Le commentaire peut démarrer à n'importe quel endroit de la ligne et non nécessairement en début de ligne.

exemples: `/* ceci est un commentaire`
 `développé sur deux lignes */`

`/* ceci est un commentaire sur une ligne */`

`// commentaire de fin de ligne après la double barre`

1.3.2 macros #define #undef

Les directives macros offrent un mécanisme de remplacement des éléments. La directive `#define` définit une macro. La syntaxe est la suivante:

`#define <identificateur> <séquence>`

Toutes les occurrences de l'identificateur de macro (chaîne de caractères) dans le fichier IGA seront remplacées par le texte défini par la séquence d'éléments (chaîne de caractères, nombre entier ou réel). Ces remplacements sont dits des développements de macro. Une séquence de macro ne doit pas elle-même contenir un identificateur de macro. Si une macro est définie deux fois, la deuxième définition est utilisée à partir de sa position dans le fichier. Il vaut mieux éviter les doubles déclarations de macros afin d'éliminer les ambiguïtés.

La définition d'une macro peut être supprimée par la directive `#undef`:

`#undef <identificateur>`

exemple: `#define X1 1.0 // chaque occurrence de X1 sera remplacée par 1.0`
 `1; X1, 0.0, 0.0; // équivalent à "1; 1.0, 0.0, 0.0;"`
 `#undef X1`
 `2; X1, 0.0, 0.0; // cette ligne est analysée sans modification`
 `// à savoir sans substitution de X1`

1.3.3 conditions #ifdef #ifndef #else #endif

Les directives conditionnelles permettent d'éliminer de l'analyse une partie du contenu du fichier. Les conditions de test portent sur la définition (`#ifdef`) ou la non-définition (`#ifndef`) de macros, c'est-à-dire si une commande `#define` a été exécutée pour cet identificateur et si celui est encore actif.

`³#ifdef <identificateur de macro>`

3... lignes analysées si la condition est vraie
 3... c'est-à-dire si la macro est définie

`³#else`

3... lignes analysées si la condition est fausse

`³#endif`

`³#ifndef <identificateur de macro>`

3... lignes analysées si la condition est vraie
 3... c'est-à-dire si la macro n'est pas définie

`³#else`

3... lignes analysées si la condition est fausse

`³#endif`

La directive `#ifdef <macro>` est vraie si la macro a été définie préalablement par la directive `#define`. La directive `#ifndef <macro>` teste la condition opposée et n'est vraie que si la macro n'a pas été définie ou bien a été détruite par la directive `#undef`. La directive `#endif` doit apparaître de manière à indiquer la fin de la portée de la directive conditionnelle. La directive `#else` permet de fournir une alternative si la condition testée est fausse.

exemple: `#define X1 1.0`

```
#ifdef X1
1; X1, 0.0, 0.0; // ligne analysée puisque X1 est défini
#else
1; 1.0, 2.0, 3.0; // ligne non interprétée
#endif

#ifdef X1 // directive sans alternative (pas de #else)
2; X1, 3.0, 0.0;
#endif
```

Attention: Dans l'application ICAB version 1.0, une macro n'est pas développée.

```
exemple: #define X1 1.0 // la macro X1 est égale à '1.0'
#define X2 X1 // la macro X2 est égale à 'X1' et non pas à '1.0'
```

1.3.4 inclusion #include

La directive `#include` permet d'insérer un fichier à l'endroit où est placée la ligne. Le nom du fichier à insérer est indiqué entre guillemets "<nom de fichier>". Le fichier à insérer doit lui-même être un fichier IGA.

exemple: `#define X1 1.0`

```
#include "test1.iga" // le fichier "test1.iga" est inclus ici */
NODE();
1; X1, 0.0, 0.0;
```


2 Bibliothèque des Entités IGA

Ce chapitre explicite la bibliothèque des entités IGA, en détaillant les données pouvant être insérées dans un fichier IGA. La lecture du chapitre précédent "Présentation des fichiers IGA" est conseillée préalablement à celui-ci.

Dans ce chapitre, chaque section est relative à une entité. Chacune des entités est présentée selon un même schéma:

* un encadré énumère d'abord les caractéristiques de l'entité en les séparant d'un point virgule, à savoir:

- le nom de l'entité suivi des paramètres de l'identificateur d'entité (ET),
- les données d'entrée (DE),
- les paramètres de données (PD).

* l'étude de l'entité est ensuite divisée en quatre paragraphes traitant respectivement des points suivants:

1. paramètres de l'Identificateur d'Entité (ET)
2. Données d'Entrée (DE)
3. Paramètres de Données (PD)
4. exemples

Par convention les inscriptions littérales sont en majuscules tandis que les données paramétrables sont en minuscules. Est par ailleurs porté en indice le type de paramètre (entier, pointeur, réel, chaîne de caractères).

Voici à titre d'exemple une partie de fichier IGA :

```
ELEMENT(TYPE=PSS_PQUAD, MAT=alu)
E2; 21, 22, 23, 24, 25, 26, 27, 28;
```

La première ligne "*ELEMENT(TYPE=PSS_PQUAD, MAT=alu)*" est explicitée dans le premier paragraphe "paramètres de l'Identificateur d'Entité ELEMENT". La partie "*E2;*" correspond aux "Données d'Entrée DE de l'entité ELEMENT". Le paragraphe "Paramètres de Données PD de l'entité ELEMENT" permettra d'analyser "*21, 22, 23, 24, 25, 26, 27, 28;*". Chaque section se termine par des exemples qui viennent illustrer les propos généraux des trois paragraphes précédents.

La liste des Entités traitées dans ce chapitre est la suivante :

CONSTRAINT	contraintes sur les degrés de liberté
ELEMENT	élément fini
FUNCTION	fonction $y=f(x)$
LOAD	chargement des noeuds et des éléments
NODE	noeud d'élément fini
PROPERTY	propriétés physiques et matériaux
REFERENCE	référentiel
RESTRAINT	déplacement imposé
STEP	préparation d'un calcul

CONSTRAINT, contraintes sur les degrés de liberté

CONSTRAINT (TYPE=contrainte,CASE=chargement)
 numéro, label; n1, X, Y, Z, RX, RY, RZ;
 numéro, label; n1, X, Y, Z, RX, RY, RZ, n2, ..., n8;
 numéro, label; n1, X=fx, Y=fy, Z=fz, RX=mx, RY=my, RZ=mz;

L'entité CONSTRAINT est utilisée pour imposer des contraintes sur des degrés de liberté de noeuds d'un maillage d'éléments finis. Il peut s'agir de l'élimination des degrés de liberté (équivalent à un blocage) ou de l'établissement de relations entre degrés de liberté. Cette entité est donc employée pour définir les conditions aux limites d'un modèle par éléments finis, conjointement avec les entités RESTRAINT et LOAD.

Paramètres de l'Identificateur d'entité CONSTRAINT

TYPE=contrainte_{chaîne}

trois types de contraintes sont possibles, à savoir :

KINEMATICS élimination et donc blocage de degrés de liberté

COUPLE fusion de degrés de liberté

MPC établissement de relations entre degrés de liberté
 (Multi Point Constraint)

CASE=chargement_{entier}

numéro d'application du chargement et des conditions aux limites (Load Case). Ce paramètre permet de traiter plusieurs cas de chargements avec le même modèle. Ce paramètre optionnel est mis à zéro par défaut et dans ce cas le chargement est appliqué en permanence.

Données d'Entrée DE pour l'entité CONSTRAINT

numéro, Label;

numéro_{entier} numéro de la contrainte.

label_{chaîne} nom de la contrainte pouvant être utilisé comme pointeur.

Remarque: ni le <numéro>, ni le <label> ne sont indispensables. Néanmoins, en l'absence de ces informations, aucune référence ne peut être faite pour la contrainte. L'indication du <label> suffit pour une référence ultérieure à la contrainte.

Paramètres de Données PD pour l'entité CONSTRAINT

Les paramètres dépendent du type de contrainte.

TYPE=KINEMATICS élimination de degré de liberté.

n1, d1, d2, d3, d4, d5, d6;

n1_{entier ou pointeur}

n1 est le numéro ou le pointeur du noeud (entité NODE) dont les degrés de liberté sont à éliminer.

REMARQUE: si <n1> est remplacé par la chaîne de caractères "ALL". Dans ce cas, l'ensemble des noeuds subira l'élimination des degrés de liberté.

d1_{X, Y, Z, RX, RY ou RZ}

Cette lettre ou mot indique le degré de liberté à supprimer. Les directions de translation X, Y ou Z sont repérées par les lettres 'X', 'Y' ou 'Z'. Les directions de rotation X, Y ou Z sont repérées par les mots 'RX', 'RY' ou 'RZ'.

Les paramètres d2, d3... d6 sont présents si plusieurs degrés de liberté doivent être éliminés sur le même noeud, soit 6 au plus.

Remarque sur le type KINEMATICS: comme cette contrainte élimine les degrés de liberté, les réactions des efforts aux noeuds ne sont pas connues. Si les réactions aux noeuds bloqués doivent être calculées, il faut employer l'entité RESTRAINT en imposant un déplacement nul.

TYPE=COUPLE fusion de degrés de liberté.

n1, d1, d2, d3, d4, d5, d6, n2, n3, ... n8;

autre syntaxe possible:

n1, n2, n3, ... n8, d1, d2, d3, d4, d5, d6;

n1_{entier ou pointeur}

n1 est le numéro ou le pointeur du noeud (entité NODE) dont les degrés de liberté sont à fusionner avec ceux d'autres noeuds.

n2_{entier ou pointeur}, ..., **n8**_{entier ou pointeur}

Les paramètres n2, n3, ..., n8 sont les numéros ou les pointeurs des noeuds dont les degrés de liberté sont fusionnés avec le noeud <n1>. Au plus 8 noeuds peuvent être liés. Si davantage de noeuds doivent être liés, il suffit rajouter une autre entité du même type en incluant à nouveau <n1>.

d1_{X, Y, Z, RX, RY ou RZ}

Cette lettre ou mot indique le degré de liberté à fusionner. Les directions de translation X, Y ou Z sont repérées par les lettres "X", "Y" ou "Z". Les directions de rotation X, Y ou Z sont repérées par les mots "RX", "RY" ou "RZ".

Les paramètres d2, d3... d6 sont présents si plusieurs degrés de liberté doivent être fusionnés sur chaque noeud, soit 6 au plus.

Nous notons <n1,d1> le déplacement du noeud <n1> dans la direction <d1>. Nous adoptons des conventions similaires pour <n2,d2>, ... <n8,d8>. L'entité CONSTRAINT(TYPE=COUPLE) permet d'établir les relations suivantes :

<n1,d1> = <n2,d1> = <n3,d1> = ... = <n8,d1>

<n1,d2> = <n2,d2> = <n3,d2> = ... = <n8,d2>

.....

<n1,d6> = <n2,d6> = <n3,d6> = ... = <n8,d6>

TYPE=MPC établissement de relations entre degrés de liberté

n1,d1,v1, n2,d2,v2, ..., n7,d7,v7;

n1_{entier ou pointeur}

numéro ou pointeur de noeud (entité NODE).

d1_{X,Y,Z,RX,RY ou RZ}

lettre ou mot indiquant une direction (un choix parmi "X", "Y", "Z", "RX", "RY" ou "RZ")

v1_{réel}

coefficient relatif au degré de liberté du noeud <n1> dans la direction <d1>.

Des conventions identiques sont reprises pour les noeuds suivants (n2, d2, v2, ...,

n7, d7, v8).

Une équation MPC établit une contrainte entre les déplacements de degrés de liberté pour des noeuds dans des directions données. Nous notons $\langle n1,d1 \rangle$ le déplacement du noeud $\langle n1 \rangle$ dans la direction $\langle d1 \rangle$. Nous avons :

$$\langle n1,d1 \rangle * v1 + \langle n2,d2 \rangle * v2 + \dots + \langle n7,d7 \rangle * v7 = 0.0$$

Jusqu'à 7 degrés de liberté peuvent être impliqués dans une équation MPC. **Le premier degré de liberté $\langle n1,d1 \rangle$ devient esclave**, c'est-à-dire qu'il ne peut appartenir ni à une autre contrainte MPC, ni à un élément du type RIGID_BAR ou RIGID_JOINT (cf entité ELEMENT ci-après). En outre le coefficient $v1$ ne doit pas être nul.

exemples pour l'entité CONSTRAINT

/ Les degrés de liberté selon les directions X, Y et RY pour le noeud 1 sont supprimés */*

CONSTRAINT(TYPE=KINEMATICS)

; 1, X,Y,RY;

/ les degrés de liberté relatifs à la translation Z et aux rotations autour des axes X et Y de tous les noeuds de la structure sont supprimés
Cette configuration est typiquement employée pour étudier un assemblage de poutres dans le plan (xy).*

**/*

CONSTRAINT(TYPE=KINEMATICS)

; ALL, Z, RX, RY;

/ Cas de fusion de degrés de liberté. La première contrainte est telle que :*

** déplacement(noeud 2, direction X) = déplacement(noeud 3, direction X)*

** = déplacement(noeud 4, direction X)*

** et*

** déplacement(noeud 2, direction Y) = déplacement(noeud 3, direction Y)*

** = déplacement(noeud 4, direction Y)*

** La deuxième contrainte assure que :*

** déplacement(noeud 2, direction X) = déplacement(noeud 1, direction X)*

**/*

CONSTRAINT(TYPE=COUPLE)

; 2, X,Y, 3, 4;

; 2, X, 1;

/ La liaison MPC permet d'établir la relation:*

** déplacement(noeud 5,direction X)*2.0 + déplacement(noeud 5, direction Y)*3.0 = 0.0*

**/*

CONSTRAINT(TYPE=MPC)

; 5, X, 2.0, 5, Y, 3.0;

ELEMENT, élément fini

ELEMENT (COLOR=couleur, MAT=matériau, PROP=propriété, TYPE=élément)
numéro, label; n1, n2,..., nn;

ELEMENT est employé pour définir un élément conformément à la méthode des éléments finis.

Paramètres de l'Identificateur d'entité ELEMENT**COLOR=couleur**_{entier}

numéro de couleur de l'élément. Ce paramètre est facultatif.

MAT=matériau_{entier ou pointeur}

matériau employé dans les éléments, référencé par un numéro entier ou un pointeur défini par une entité PROPERTY du type "caractéristiques de matériau".

PROP=propriété_{entier ou pointeur}

propriétés physiques des éléments, et référencées par un numéro entier ou un pointeur défini par une entité PROPERTY du type "propriétés physiques".

Remarque: les propriétés physiques du paramètre PROP sont différentes des caractéristiques de matériau du paramètre MAT. Ainsi, pour un élément de type poutre (BEAM_LINEAR), le paramètre PROP concerne la section, les moments d'inertie... Les paramètre MAT fait référence au module d'élasticité, au coefficient de Poisson du matériau constituant la poutre.

MESH=maillage_{entier}

Ce numéro indique que cet élément a été généré par le mailleur.

TYPE=élément_{chaîne}

chaîne de caractères indiquant le type d'élément. La liste des éléments est détaillée ci-dessous.

Liste des éléments ponctuels ou élancés (module ICAB Force).

BEAM_LINEAR	poutre de section constante dans le sens de la longueur
BUSHING	silentbloc
GSTIFF	rigidité généralisée
MASS	masse concentrée avec possibilité d'ajouter une inertie
RIGID_BAR	barre rigide sans masse
RIGID_JOINT	barre rigide rotulée sans masse
ROD	barre ne travaillant qu'en traction ou en compression
SPRING	ressort

Liste des éléments pour problèmes avec plaque plane (Plate)

PLT_LTRIA	triangle linéaire
PLT_PTRIA	triangle parabolique
PLT_CTRIA	triangle cubique
PLT_LQUAD	quadrangle linéaire
PLT_PQUAD	quadrangle parabolique
PLT_CQUAD	quadrangle cubique

Liste des éléments pour problèmes avec membrane

MEM_LTRIA	triangle linéaire
MEM_CTRIA	triangle parabolique
MEM_CTRIA	triangle cubique
MEM_LQUAD	quadrangle linéaire
MEM_PQUAD	quadrangle parabolique
MEM_CQUAD	quadrangle cubique

Liste des éléments pour problèmes avec coque (Shell)

SHL_LTRIA	triangle linéaire
SHL_CTRIA	triangle parabolique
SHL_CTRIA	triangle cubique
SHL_LQUAD	quadrangle linéaire
SHL_PQUAD	quadrangle parabolique
SHL_CQUAD	quadrangle cubique

Liste des éléments mono-dimensionnels

L1D	élément axial linéaire
P1D	élément axial parabolique
C1D	élément axial cubique
LCYLIND	cylindre linéaire
PCYLIND	cylindre parabolique
CCYLIND	cylindre cubique
LSPHERE	sphère linéaire
PSPHERE	sphère parabolique
CSPHERE	sphère cubique

Liste des éléments pour problèmes avec contrainte plane (Plane Stress)

PSS_LTRIA	triangle linéaire
PSS_CTRIA	triangle parabolique
PSS_CTRIA	triangle cubique
PSS_LQUAD	quadrangle linéaire
PSS_PQUAD	quadrangle parabolique
PSS_CQUAD	quadrangle cubique

Liste des éléments pour problèmes avec déformation plane (Plane Strain)

PST_LTRIA	triangle linéaire
PST_CTRIA	triangle parabolique
PST_CTRIA	triangle cubique
PST_LQUAD	quadrangle linéaire
PST_PQUAD	quadrangle parabolique
PST_CQUAD	quadrangle cubique

Liste des éléments pour problèmes axisymétriques

AXI_LTRIA	triangle linéaire
AXI_CTRIA	triangle parabolique
AXI_CTRIA	triangle cubique
AXI_LQUAD	quadrangle linéaire
AXI_PQUAD	quadrangle parabolique
AXI_CQUAD	quadrangle cubique

Liste des éléments volumiques solides tri-dimensionnels

SOL_LTETRA	tétraèdre linéaire
SOL_PTETRA	tétraèdre parabolique
SOL_LWEDGE	pentaèdre linéaire

SOL_PWEDGE	pentaèdre parabolique
SOL_LBRICK	brique linéaire
SOL_PBRICK	brique parabolique

Données d'Entrée DE pour l'entité ELEMENT

numéro, label;

numéro_{entier} numéro de l'élément.

label_{chaîne} nom de l'élément pouvant être utilisé comme pointeur.

Remarque: ni le <numéro>, ni le <label> ne sont indispensables. Néanmoins, en l'absence de ces informations, aucune référence ne peut être faite pour l'élément. L'indication du <label> suffit pour une référence ultérieure à l'élément. Si le <numéro> n'est pas indiqué, il est généré par une numérotation automatique. Deux éléments ne doivent pas porter le même numéro ou le même label.

Paramètres de Données PD pour l'entité ELEMENT

n1_{entier ou pointeur}, ..., nn_{entier ou pointeur};

Les paramètres contiennent les numéros des noeuds ou les pointeurs des noeuds (entité NODE) constituant l'élément fini. Le nombre de noeuds nécessaires dépend du type d'élément. L'ordre de numérotation des noeuds dans un élément fini doit respecter certaines conventions. Les figures données ci-dessous explicitent ces conventions ainsi que les numérotations correspondant aux arêtes (EDGES) et aux faces qui sont employées pour appliquer des chargements sur les éléments (entité LOAD).

Liste des éléments ponctuels ou élanés

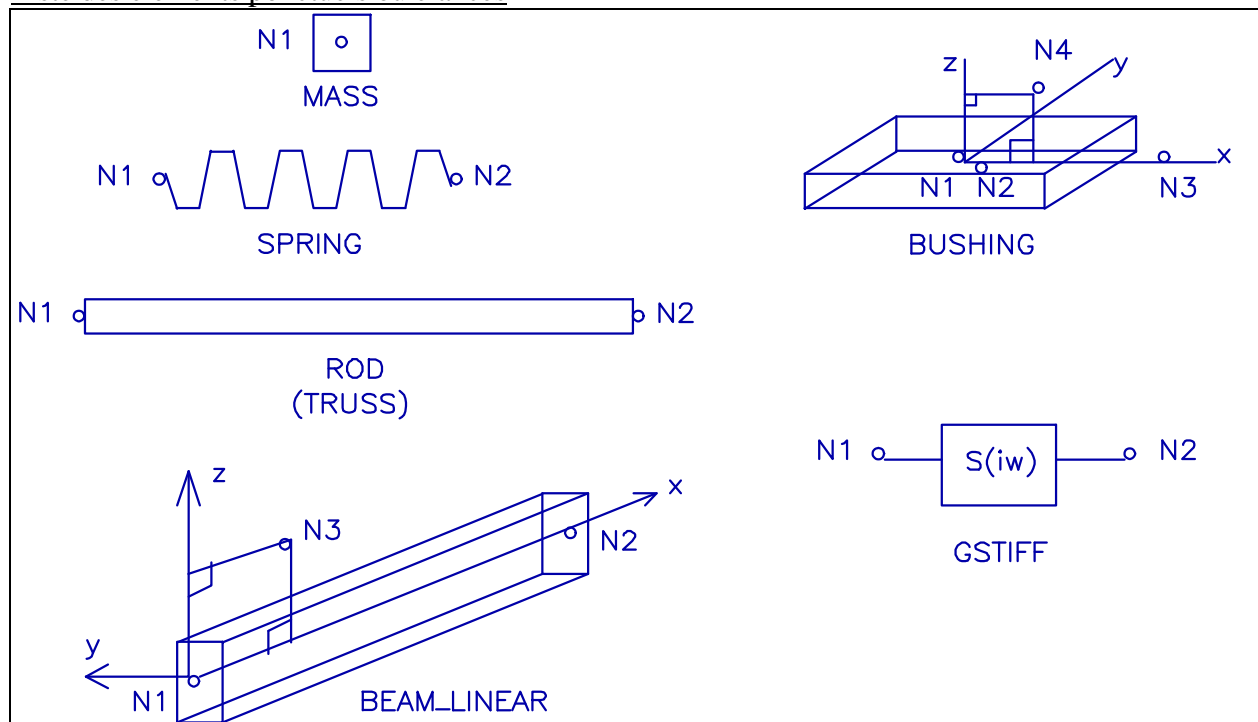


Figure 4 Eléments ponctuels ou élanés

TYPE=MASS

L'élément "MASS" définit une masse concentrée en un noeud. Cette masse peut être complétée par une matrice d'inertie. Les valeurs de la masse et de la matrice d'inertie sont indiquées dans une entité "PROPERTY(TYPE=MASS)".

TYPE=SPRING

L'élément "SPRING" définit un ressort entre deux noeuds. Ce ressort agit en traction et compression dans la direction définie par les deux noeuds formant les extrémités du ressort. Le ressort peut également avoir une masse. Ces propriétés (rigidité, masse) sont indiquées dans une entité "PROPERTY(TYPE=SPRING)".

TYPE=ROD

L'élément "ROD" définit une barre élastique entre deux noeuds ne pouvant agir qu'en traction ou compression (cet élément est souvent appelé "TRUSS" en anglais). La section de la barre est indiquée dans une entité "PROPERTY(TYPE=ROD)" et les caractéristiques du matériau dans une entité "PROPERTY(TYPE=ISO)".

TYPE=BEAM LINEAR

L'élément "BEAM_LINEAR" définit une poutre droite de section constante sur sa longueur. L'origine et l'extrémité de la poutre sont définies respectivement par le premier et le deuxième noeud (n1, n2). Le troisième noeud (n3) permet d'orienter la section de la poutre: le plan (n1, n2, n3) définit le plan (xz) du repère local associé à la poutre. Si le noeud n3 n'est pas indiqué, l'axe (z) de la poutre est parallèle à l'axe (Z) du repère global. Si la poutre est verticale (dans ce cas l'axe (x) de la poutre est parallèle à l'axe (Z) du repère global), alors l'axe (y) de la poutre est parallèle à l'axe (Y) du repère global.

Les propriétés physiques d'une poutre (section, moments d'inertie...) sont définies dans une entité "PROPERTY(TYPE=BEAM_LINEAR)" et les caractéristiques du matériau dans une entité "PROPERTY(TYPE=ISO)".

Effet du cisaillement transverse

lorsque la poutre est élancée, c'est à dire que sa longueur est grande par rapport à sa section, l'effet du cisaillement transverse peut être négligé. La formulation dite de Kirchhoff est alors employée (hypothèse des sections droites avec conservation des normales et nullité des déformations liées au cisaillement transverse).

Pour les poutres courtes, il est préférable de tenir compte du cisaillement transverse et dans ce cas la formulation de Timoshenko est appliquée. Pour que le cisaillement transverse soit pris en compte dans les calculs, il faut que les paramètres "SRY et SRZ" soient non nuls (cf entité "PROPERTY(TYPE=BEAM_LINEAR)". Le paramètre "SRY" définit le facteur de correction AR/AR_S où AR est la section de la poutre et AR_S la "section cisillée" correspondant à des efforts tranchants T_y et des moments fléchissants M_z . Ce coefficient de correction est toujours supérieur à 1 et vaut 6/5 pour des sections rectangulaires, 7/6 pour des sections circulaires, 2 pour un tube circulaire (sections homogènes).

Précision

L'élément "BEAM_LINEAR" permet de calculer exactement (conformément aux théories de Kirchhoff ou de Timoshenko) la réponse statique d'une poutre soumise à une charge uniformément répartie sur sa longueur, c'est à dire pour des variations linéaires de l'effort tranchant et des variations paraboliques du moment fléchissant.

TYPE=BEAM TAPERED

L'élément "BEAM_TAPERED" définit une poutre droite de section variable. Sa définition est similaire à celle de d'une poutre de type "BEAM_LINEAR". Toutefois, deux paramètres optionnels sont possibles:

$$\mathbf{X1} = x1_{\text{réel}}$$

$$\mathbf{X2} = x2_{\text{réel}}$$

Les paramètres <x1> et <x2> permettent d'affecter un coefficient de pondération aux valeurs numériques des propriétés des sections. Les valeurs par défaut sont nulles $x1=x2=0$.

Nous notons <S1> (respectivement <S2>) une propriété physique de la section 1 (respectivement section 2) définie dans l'entité PROPERTY (TYPE=BEAM_TAPERED) à laquelle l'élément fait référence. Les valeurs des propriétés S_a de la poutre sur la section avant (S_b sur la section arrière, $S_{1/2}$ au milieu) sont calculées comme suit:

$$S_{1/2} = S_1 \frac{1-u}{2} + S_2 \frac{1+u}{2}, \quad u = \frac{x1 - x2}{2}$$

$$S_a = S_1 \left(1 - \frac{x1}{2}\right) + S_2 \frac{x1}{2}, \quad (u = x1 - 1)$$

$$S_b = S_1 \frac{x2}{2} + S_2 \left(1 - \frac{x2}{2}\right), \quad (u = 1 - x2)$$

TYPE=BUSHING

L'élément "BUSHING" (silentbloc ou appui élastique) définit des rigidités entre deux noeuds (n1, n2) selon 6 directions (x, y, z, rx, ry, rz) dans un repère différent du repère global. Ce repère est défini par les noeuds n1, n3, n4. La droite (n1, n3) correspond à l'axe (x) et le plan (n1, n3, n4) définit le plan (xz) du repère local. Si les noeuds n3 et n4 ne sont pas indiqués, alors le repère local est identique au repère global. Le noeud n2 peut occuper la même position que le noeud n1.

Nous notons $\mathbf{F1}$ le torseur force associé au noeud n1 dans le repère local de l'élément ayant pour composantes ($F_{x1}, F_{y1}, F_{z1}, M_{x1}, M_{y1}, M_{z1}$) et $\mathbf{D1}$ le vecteur des déplacements et des rotations du noeud n1. Nous adoptons des notations analogues pour le noeud n2 ($F2, D2$). Les relations établies par l'élément "BUSHING" sont les suivantes:

$$\begin{bmatrix} \mathbf{F1} \\ \mathbf{F2} \end{bmatrix} = \begin{bmatrix} [\mathbf{K}] & -[\mathbf{K}] \\ -[\mathbf{K}] & [\mathbf{K}] \end{bmatrix} \begin{bmatrix} \mathbf{D1} \\ \mathbf{D2} \end{bmatrix}, \quad [\mathbf{K}] = \begin{bmatrix} k_x & 0 & 0 & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 & 0 & 0 \\ 0 & 0 & k_z & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{rx} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{ry} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{rz} \end{bmatrix}$$

Les rigidités d'un élément "BUSHING" sont définies dans une entité "PROPERTY(TYPE=BUSHING)"

TYPE=GSTIFF

n1, n2, d1, d2, type, k, c, m, g;

L'élément "GSTIFF" définit une rigidité généralisée entre deux degrés de liberté.

n1_{entier ou pointeur}

numéro ou pointeur d'un noeud.

d1_{X,Y,Z,RX,RY,RZ}

Cette lettre ou mot indique une direction (translation X, Y, Z ou rotation RX, RY, RZ). Le couple <n1, d1> désigne le degré de liberté du noeud <n1> dans la direction <d1>.

Des conventions analogues sont adoptées pour <n2> et <d2>.

type_{entier}

Ce numéro définit le type de couplage entre les degrés de liberté <n1,d1> et <n2,d2>. Le paramètre <type> peut prendre les valeurs 1, 2, 3 ou 4 (cf paragraphe ci-après).

k_{réel}, c_{réel}, m_{réel}, g_{réel}

Les coefficients k, c, m, g représentent respectivement la rigidité du couplage, l'amortissement visqueux du couplage, la masse du couplage et l'amortissement structural du couplage. Ces valeurs définissent le coefficient de la rigidité généralisée S_w :

$$S_w = k - m.w^2 + i(g + c.w) \quad \text{avec } (S_{(w=0)} = k)$$

où $w = 2\pi f$ est la pulsation proportionnelle à la fréquence <f> de sollicitation, <i> est le nombre complexe tel que $i^2 = -1$.

Les définitions du couplage sont données ci-dessous:

$$type = 1 \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & S_w \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$type = 2 \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & S_w \\ S_w & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$type = 3 \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} S_w & -S_w \\ 0 & S_w \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$type = 4 \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} S_w & -S_w \\ -S_w & S_w \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

f_1 est la force (ou le moment) exercée sur le degré de liberté <n1, d1>, x_1 est le déplacement (en translation ou en rotation) de ce même degré de liberté. Les notations f_2 et x_2 s'appliquent au degré de liberté <n2,d2>.

ATTENTION: Les dimensions des termes de couplage dépendent des dimensions des degrés de liberté. Les dimensions (en unités SI) sont reportées dans le tableau ci-dessous:

	d1 _{X,Y,Z} d2 _{X,Y,Z}	d1 _{X,Y,Z} d2 _{RX,RY,RZ}	d1 _{RX,RY,RZ} d2 _{X,Y,Z}	d1 _{RX,RY,RZ} d2 _{RX,RY,RZ}
k, g	N/m=kg.s ⁻²	N/rad=kg.m.s ⁻²	N.m/m=kg.m.s ⁻²	N.m/rad=kg.m ² .s ⁻²
c	N.s/m=kg.s	N.s/rad=kg.m.s ⁻¹	N.m.s/m=kg.m.s ⁻¹	N.m.s/rad=kg.m ² .s ⁻¹
m	N.s ² /m=kg	N.s ² /rad=kg.m	N.m.s ² /m=kg.m	N.m.s ² /rad=kg.m ²

Définition des barres rigides sans masse et des corps rigides

TYPE=RIGID BAR, RIGID JOINT

L'élément "RIGID_BAR" définit une barre rigide sans masse. Cet élément est utile pour modéliser des parties très raides d'une structure, parties qui peuvent être assimilées à des corps rigides, c'est à dire non susceptibles de se déformer.

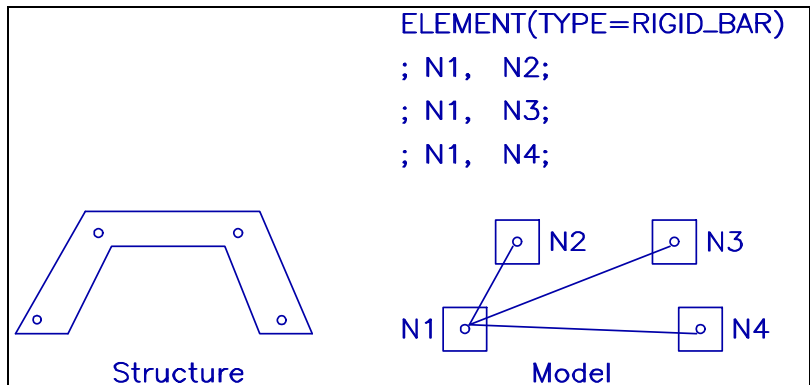


Figure 5 Définition d'un corps rigide par "RIGID_BAR"

Une barre rigide est définie par deux noeuds (n1, n2) qui jouent un rôle dissymétrique. Le deuxième noeud <n2> est le noeud esclave: ses degrés de liberté sont reportés sur le noeud maître, c'est à dire le premier noeud <n1>. Dans une liaison rotulée "RIGID_JOINT", les deux noeuds (n1, n2) ont des rotations indépendantes.

ATTENTION:

Un noeud esclave ne peut appartenir qu'à un seul corps rigide. Par ailleurs, il ne peut exister qu'un noeud maître dans un corps rigide. Ainsi toutes les barres rigides définissant un corps rigide doivent converger vers le même noeud, qui est le noeud maître.

Un déplacement imposé (entité "RESTRAINT(TYPE=DISPLACEMENT)") ne peut être appliqué sur un noeud esclave. Une contrainte cinématique "CONSTRAINT(TYPE=COUPLE, MPC)" ne peut pas être établie sur un noeud esclave. En effet, un noeud esclave a "perdu" ses degrés de liberté cédés au noeud maître.

Liste des éléments mono-dimensionnels

Les noeuds de ces éléments doivent être portés par l'axe X pour le type "1D", ou l'axe R pour les éléments "CYLIND" et "SPHERE", c'est-à-dire la première coordonnée. La coordonnée R des éléments de type "CYLIND" et "SPHERE" doit être positive.

N1	E1	N2		L1D, LCYLIND, LSPHERE	
N1		N2	N3	P1D, PCYLIND, PSPHERE	
N1		N2	N3	N4	C1D, CCYLIND, CSPHERE

Figure 6 Eléments mono-dimensionnels

Famille	1ère coordonnée	2ème coordonnée	3ème coordonnée
1D	X	0	0
CYLIND	$R^3 0$	0	0
SPHERE	$R^3 0$	0	0

Liste des éléments bi-dimensionnels

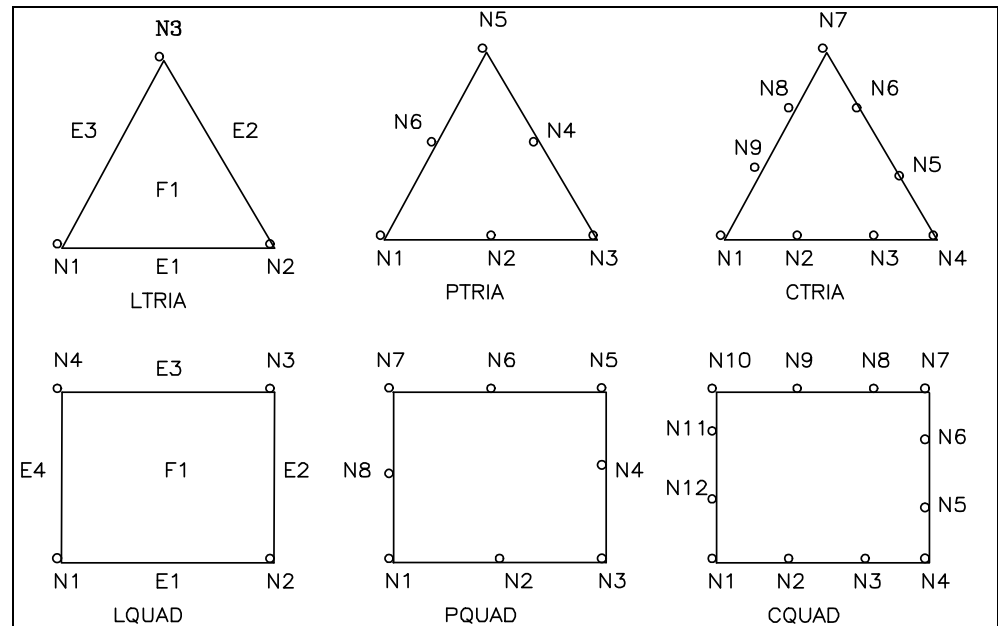


Figure 7 Eléments bi-dimensionnels, triangles et quadrangles

Les familles d'éléments bi-dimensionnels sont les éléments plans à contrainte plane "PSS", les éléments plans à déformation plane "PST", les éléments axisymétriques "AXI", les plaques planes "PLT", les membranes "MEM" et les coques "SHL".

Famille	1ère coordonnée	2ème coordonnée	3ème coordonnée
PSS, contrainte plane	X	Y	0
PST, déformation plane	X	Y	0
AXI, axisymétrique	$R^3 0$	0	Z
PLT, plaque	X	Y	0
MEM, membrane	X	Y	Z
SHL, coque	X	Y	Z

Tétraèdres pour éléments tri-dimensionnels

Ces éléments sont employés pour modéliser des structures volumiques. Les noeuds sont définis dans l'espace par leurs trois coordonnées X, Y, Z.

SOL_LTETRA : 4 noeuds

SOL_PTETRA : 10 noeuds

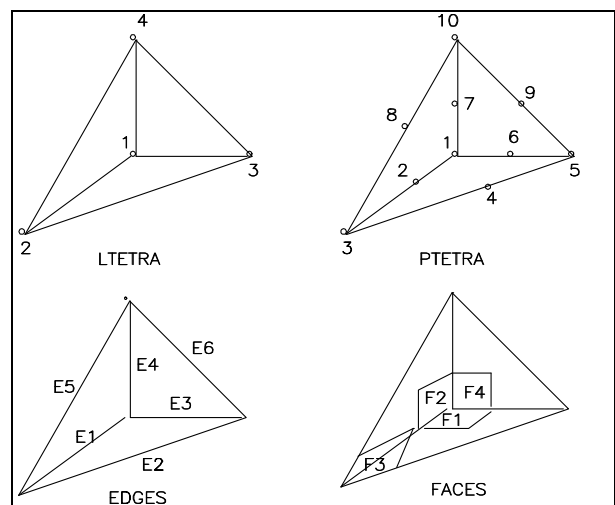


Figure 8 Eléments 3D, tétraèdres

Prismes pour éléments tri-dimensionnels

Ces éléments sont employés pour modéliser des structures volumiques. Les noeuds sont définis dans l'espace par leurs trois coordonnées X, Y, Z.

SOL_LWEDGE : 6 noeuds

SOL_PWEDGE : 15 noeuds

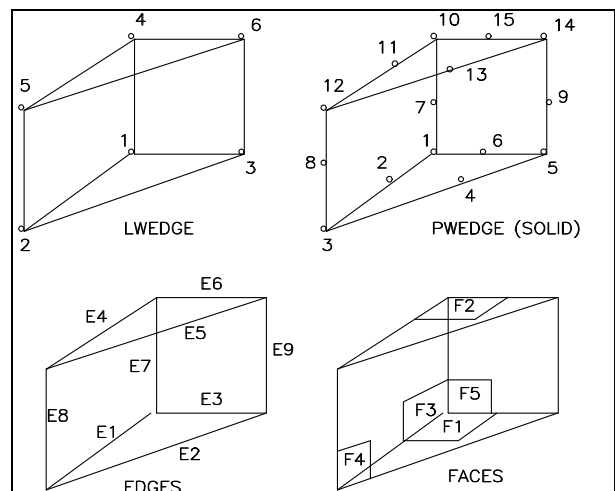


Figure 9 Eléments 3D, pentaèdres.

Briques pour éléments tri-dimensionnels

Ces éléments sont employés pour modéliser des structures volumiques. Les noeuds sont définis dans l'espace par leurs trois coordonnées X, Y, Z.

SOL_LBRICK : 8 noeuds

SOL_PBRICK : 20 noeuds

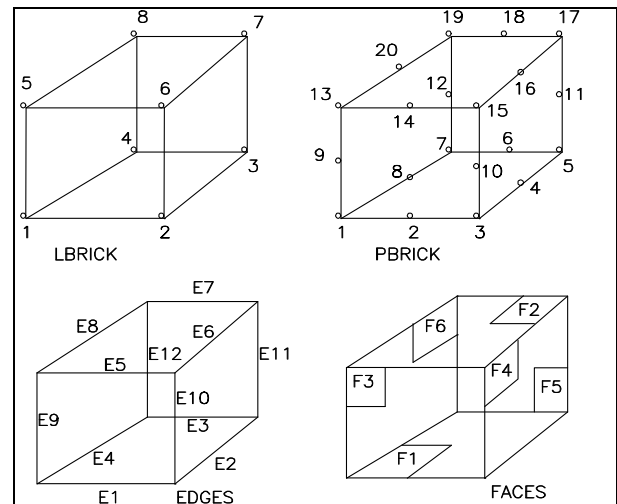


Figure 10 Eléments 3D, hexaèdres

exemples pour l'entité ELEMENT

ELEMENT(TYPE=SPRING, PROP=spring1)

S1; 1, 2;

ELEMENT(TYPE=BEAM_LINEAR, PROP=beam1, MAT=steel)

B1; 1, 2, 3;

*/**

Remarque: pour une poutre linéaire, les deux premiers noeuds définissent la poutre dans le sens de sa longueur (axe X de la poutre), le troisième noeud définit l'axe Z de la section (YZ) perpendiculaire à l'axe X.

**/*

ELEMENT(TYPE=GSTIFF)

G1; N1, N2, X, X, 4, 10.0; // équivalent à l'établissement d'un ressort entre

// N1 et N2 mais agissant dans la direction X seulement

ELEMENT(TYPE=PST_LQUAD, MAT=1)

E2; 11, 12, 13, 14;

ELEMENT(TYPE=PSS_PQUAD, MAT=alu)

E3; 21, 22, 23, 24, 25, 26, 27, 28;

FUNCTION, fonction $y=f(x)$

FUNCTION ()

numéro, label; TITLE=titre, TYPE=fonction, INCREMENT=incr, XMIN=xmin,
 XLABEL=légende1, YLABEL=légende2, Z=z, Y=y0,y1...;

L'entité FUNCTION est employée pour définir une fonction du type $y=f(x)$.

Paramètres de l'Identificateur d'entité FUNCTION

Il n'y a pas de paramètre.

Données d'Entrée DE pour l'entité FUNCTION

numéro, label;

numéro_{entier} numéro de la fonction.

label_{chaîne} nom de la fonction pouvant être utilisé comme pointeur.

Remarque: Il faut au moins indiquer le <numéro> ou le <label> de la fonction.

Paramètres de Données PD pour l'entité FUNCTION

TITLE=titre_{chaîne}

Chaîne de caractères servant de titre. Cette chaîne de caractères doit être écrite entre des guillemets si elle comporte plusieurs mots ou si sa longueur est supérieure à 8 caractères. Ce paramètre est optionnel.

TYPE=fonction_{chaîne}

Type de fonction qui peut être:

"COMPLEX", fonction $y=f(x)$ où y est complexe.

"REAL", fonction $y=f(x)$ où x et y sont des variables réelles

"SPECTRUM", fonction $Y=F(f)$, l'abscisse est la fréquence et Y est une variable complexe.

"TIME", fonction $y=f(t)$, l'abscisse est le temps, y est réel.

INCREMENT=incr_{réel}

Le paramètre INCREMENT= t est indiqué lorsque la fonction comporte des valeurs ($y[0], y[1], \dots$) équi-échantillonnées avec un incrément t . Les abscisses ($x[0]=0, x[1]=t, x[2]=2t, \dots$) ne sont pas mentionnées.

XMIN=xmin_{réel}

Pour les signaux équi-échantillonnés, c'est-à-dire lorsque le paramètre INCREMENT= t est mentionné, XMIN= X_0 précise la valeur de départ des abscisses de la fonction ($x[0]=X_0, x[1]=X_0+t, x[2]=X_0+2t, \dots$). Par défaut, X_0 est nul.

XLABEL=légende1_{chaîne}

YLABEL=légende2_{chaîne}

Chaînes de caractères utilisées pour préciser la nature de l'abscisse X et de l'ordonnée Y .

Z=z_{réel}

La valeur permet de repérer la courbe dans un espace tri-dimensionnel ($x, y=f(x), z$)

Y=y0_{réel}, , y1_{réel}, ...

Les valeurs de la fonction sont reportées à la suite du paramètre Y qui doit figurer après tous les autres paramètres. Quatre formes de codage sont employées selon que la fonction comporte des valeurs Y réelles ou complexes et que les abscisses sont équi-échantillonnées ou non (cf paramètre INCREMENT). La disposition des valeurs (x,y) est indiquée dans le tableau ci-dessous:

Fonction	échantillonnage	Y=
réelle	non	x[0], y[0], ..., x[n], y[n-1]
complexe	non	x[0], yr[0], yi[0],...,x[n-1], yr[n-1], yi[n-1]
réelle	oui	y[0], y[1], ..., y[n-1]
complexe	oui	yr[0], yi[0], ..., yr[n-1], yi[n-1]

Nous notons "yr" et "yi" les valeurs réelles et imaginaires de la fonction et "n" le nombre de points de la fonction.

exemples pour l'entité FUNCTION

FUNCTION()

*fct1; TITLE="y=f(x), fonction réelle non équi-échantillonnée",
TYPE=REAL,
Y=0.0, 0.0, // x[0],y[0]
1.0, 2.0, // x[1],y[1]
3.0, 4.0; // x[2],y[2] (3x2 valeurs)*

*fct2; TITLE="Y=f(x) fonction complexe non équi-échantillonnée",
TYPE=COMPLEX,
Y=0.0, 0.0, 1.0, // x[0], yr[0], yr[0]
1.5, 3.0, 2.0, // x[1], yr[1], yi[1]
4.0, 3.0, 4.0, // x[2], yr[2], yi[2]
5.0, 3.5, 4.5; // x[3], yr[3], yi[3] (4x3 valeurs)*

FUNCTION()

*fct3; TITLE="y=f(i) fonction réelle équi-échantillonnée",
TYPE=REAL,INCREMENT=1.0,
Y=0.5, // y[0] (x=0.0)
1.0, // y[1] (x=1.0)
2.0, // y[2] (x=2.0)
4.0; // y[3] (x=3.0)(4x1 valeurs)*

*fct4; TITLE="Y=f(i) fonction complexe équi-échantillonnée",
TYPE=COMPLEX,INCREMENT=0.5,
Y=1.0, 0.0, // Yr[0], Yi[0] (x=0.0)
0.0, 2.0, // Yr[1], Yi[1] (x=0.5)
1.0, 1.0, // Yr[2], Yi[2] (x=1.0)
2.0, 0.0; // Yr[3], Yi[3] (x=1.5) (3x2 valeurs)*

IMPORT, importer des projets ICAB

IMPORT(AGB=, PART=, LEVEL=, REF=,DMIN=,PROP)

L'entité IMPORT employée pour importer des projets ICAB. Le projet peut ainsi être constitué de plusieurs sous-projets. Pour assurer les "soudures" entre nœuds, les nœuds communs entre deux sous-modèles doivent être placés à des distances inférieures à DMIN

Paramètres de l'Identificateur d'entité IMPORT

AGB=nom du projet ICAB"chaîne"

Nom du projet ICAB. Cette chaîne de caractères doit être écrite entre des guillemets si elle comporte plusieurs mots ou si sa longueur est supérieure à 8 caractères. L'extension d'un projet ICAB est .agb

IGB=nom du projet ICAB"chaîne"

Nom du projet ICAB version 4 ou antérieure. Cette chaîne de caractères doit être écrite entre des guillemets si elle comporte plusieurs mots ou si sa longueur est supérieure à 8 caractères. Les extensions d'un projet ICAB4 sont .igb/.dgb

PART=j numéro, entier

Numéro de la partie importée. Les nœuds Ni, les éléments Ei et les propriété Pi du modèle importés ont pour label respectivement "Ni:j", "Ei:j" et "Pi:j"

LEVEL=l numéro, entier

Numéro de couche (level, layer) qui sera affectée au modèle importé ; valeur par défaut : 0

REF=Ri entier, pointeur

Référentiel utilisé pour importer le sous-modèle ; référentiel par défaut : repère global

DMIN=d réel

La distance minimale en dessous de laquelle les nœuds adjacents sont considérés comme devant être confondus. La valeur indicative par défaut correspond à 10^{-6} fois la valeur des coordonnées maximales.

PROP=c texte= nouveau/utilise

Dans le cas « PROP=NOUVEAU », les propriétés définies précédemment ne sont pas utilisées : les propriétés et les matériaux du projet importé sont ajoutés systématiquement. Dans le cas contraire, les propriétés ayant les mêmes caractéristiques sont utilisées à nouveau.
Défaut : PROP=UTILISE

Données d'Entrée DE pour l'entité IMPORT

Il n'y a pas de données d'entrée.

Paramètres de Données PD pour l'entité IMPORT

Il n'y a pas de paramètre.

exemples pour l'entité IMPORT

UNIT() // définition des unités : N, mm

; M=0.001,KG=1000,S=1,K=1,K0=0,LENGTH="mm",MASS="T",

// importation du sous-modèle 1 défini dans la partie 1

IMPORT(AGB= "sous-modele1.agb", LEVEL=1,PART=1)

REFERENCE()2, *reference_2* ;*NO=n10 :1, NX=n11 :1 ; // origine : nœud 10 du sous-modèle 1, axe X : entre nœuds 10-11***IMPORT(AGB= "sous-modele2.agb",REF=2, LEVEL=2,PART=2)**

Le sous-modele2 est importé dans le référentiel défini à partir des nœuds du sous-modèle 1.

LOAD, chargement**LOAD** (TYPE=charge,CASE=chargement, FUNCTION=variation, DIRECT=direction)

numéro, label; G=gx,gy,gz, OMEGA=wx,wy,wz, CENTER=cx,cy,cz;

numéro, label; AM=ta, t0, t0k;

numéro, label; n1, X=fx, Y=fy, Z=fz, RX=mx, RY=my, RZ=mz;

numéro, label; n1, q1;

numéro, label; n1, t1, t2, G=gradient;

numéro, label; e11, F1=press1, F2=press2, ... , F6=press6, DIRECT=direction;

numéro, label; e11, E1=charge1a,charge1b, ..., DIRECT=direction;

L'entité LOAD est utilisée pour appliquer un chargement sur des noeuds (entité NODE) ou des éléments (entité ELEMENT). Cette entité LOAD est donc employée pour définir les conditions aux limites d'un modèle par éléments finis, conjointement avec les entités CONSTRAINT et RESTRAINT.

Paramètres de l'Identificateur d'entité LOAD**CASE=chargement**_{entier}

numéro d'application du chargement et des conditions aux limites (Load Case). Ce paramètre optionnel est nul par défaut et dans ce cas le chargement est appliqué en permanence. Ce paramètre permet de traiter plusieurs cas de chargement avec le même modèle.

FUNCTION=variation_{entier ou pointeur}

numéro ou pointeur d'une fonction (entité FUNCTION). Ce paramètre optionnel est utilisé pour les simulations de chargement variable dans le temps. L'amplitude du chargement indiquée dans la section "Paramètre de Données" est multipliée par la valeur de la fonction. Ce paramètre ne peut être utilisé que pour les calculs dynamiques seulement.

MESH=maillage_{entier}

Ce numéro indique que ce chargement a été généré par le mailleur.

TYPE=charge_{chaîne}

Type de chargement

* chargements volumiques

AM_TEMPERATURE définition de la température ambiante

ACCELERATION champ gravitationnel et forces centrifuges de rotation

VOL_HEAT génération volumique de chaleur dans les éléments

* chargements appliqués sur des noeuds

FORCE force à un noeud

NODE_HEAT source de chaleur à un noeud

TEMPERATURE	température à un noeud de coque * chargements appliqués sur des arêtes d'éléments
ED_PRESSURE	pression sur des arêtes d'éléments
ED_HEAT	flux de chaleur sur des arêtes d'éléments
ED_CONVECTION	convection de chaleur sur des arêtes d'éléments
ED_RADIATION	radiation de chaleur sur des arêtes d'éléments * chargements appliqués sur des surfaces d'éléments
PRESSURE	pression sur des surfaces d'éléments
HEAT	flux de chaleur sur des surfaces d'éléments
CONVECTION	convection de chaleur sur des surfaces d'éléments
RADIATION	radiation de chaleur sur des surfaces d'éléments

DIRECT=direction_{L,X,Y,Z}

Ce paramètre permet de définir la direction d'application des charges dans le repère global et non dans le repère local de l'élément. Cette possibilité n'est valable que pour les chargements "TYPE=ED_PRESSURE" et "TYPE=PRESSURE" (cf section suivante).

2.0.1 Données d'Entrée DE pour l'entité LOAD**numéro, label;**

numéro_{entier} numéro du chargement.

label_{chaîne} nom du chargement pouvant être utilisé comme pointeur.

Remarque: ni le <numéro>, ni le <label> ne sont indispensables. En l'absence de ces informations, aucune référence ne peut être faite au chargement. L'indication du <label> suffit pour une référence ultérieure au chargement.

Paramètres de Données PD pour l'entité LOAD

La syntaxe des Paramètres de Données dépend du type de chargement et est détaillée ci-après.

TYPE=AM TEMPERATURE définition de la température ambiante

AM=ta, t0, t0k;

AM=ta_{réel}, t0_{réel}, t0k_{réel}

Le premier paramètre <ta> correspond à la température ambiante, c'est à dire la température par défaut de la structure. Toutefois, un noeud <n1> peut être à une température différente de <ta> si des entités "LOAD (TYPE=TEMPERATURE)" ou "RESTRAINT(TYPE=TEMPERATURE)" se rapportent au noeud <n1>.

Le second paramètre <t0> indique la température de référence pour laquelle les dilatations thermiques sont nulles.

Le troisième paramètre <t0k> précise la température du zéro absolu (employée uniquement pour les calculs de radiation de chaleur).

TYPE=ACCELERATION champ gravitationnel et rotation de la structure

G=gx,gy,gz, OMEGA=wx,wy,wz, CENTER=cx,cy,cz;

G=gx_{réel}, gy_{réel}, gz_{réel}

Champ gravitationnel défini par ses trois composantes gx, gy, gz. (unité SI : m/s²)

OMEGA=wx_{réel}, wy_{réel}, wz_{réel}

Vecteur rotation défini par ses trois composantes wx, wy, wz (unité SI : rad/s)

CENTER= $\mathbf{cx}_{\text{réel}}, \mathbf{cy}_{\text{réel}}, \mathbf{cz}_{\text{réel}}$

Coordonnées cx, cy, cz du centre de rotation. Les valeurs par défaut sont (0.0, 0.0, 0.0)

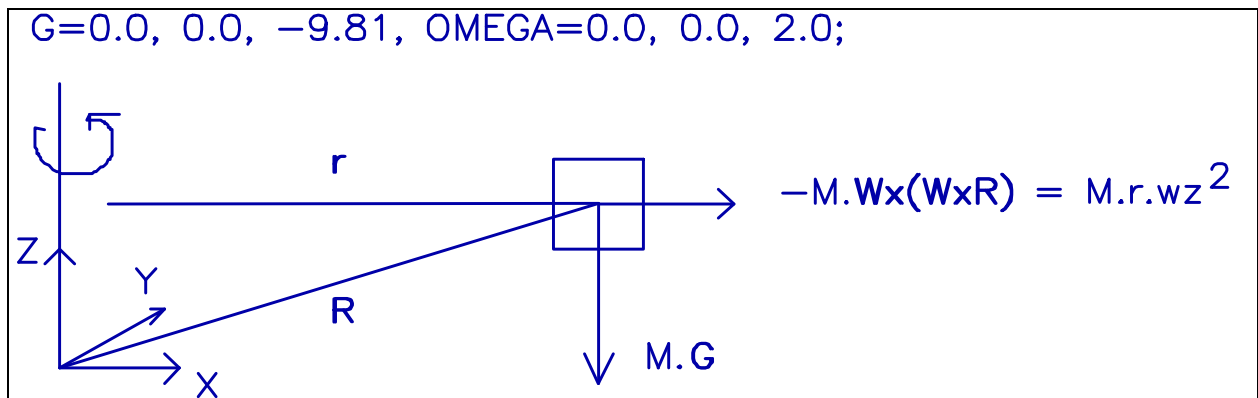


Figure 11 Définition d'un champ d'accélération

TYPE=FORCE Force ou torseur complet appliquée à un noeud.

n1, X=fx, Y=fy, Z=fz, RX=mx, RY=my, RZ=mz;

n1 entier ou pointeur

numéro ou pointeur du noeud sur lequel est appliquée la force

X=fx_{réel}

Ce paramètre permet de préciser la valeur de la force appliquée sur le noeud n1 dans la direction "X". Cette force ne peut être appliquée que si le degré de liberté du noeud <n1> dans la direction X est actif. (unité SI: N=kg.m.s⁻²)

Y=fy_{réel}

Z=fz_{réel}

Ces paramètres ont des fonctions analogues au paramètre "X=fx" dans les directions "Y" et "Z".

RX=mx_{réel}

RY=my_{réel}

RZ=mz_{réel}

Ces paramètres ont des fonctions analogues aux paramètres "X=fx, Y=fy, Z=fz" pour préciser la valeur des moments appliqués au noeud n1 autour des axes X, Y et Z. (unité SI: N.m=kg.m².s⁻²).

Remarque: seules les composantes non nulles doivent figurer.

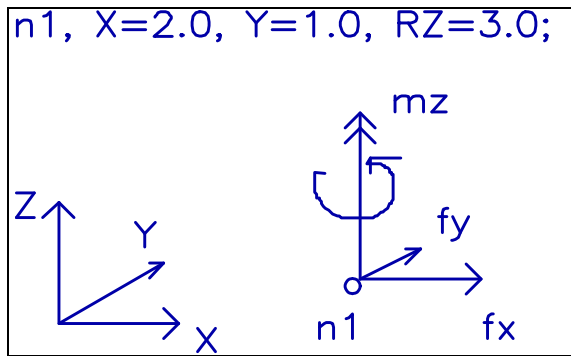


Figure 12 Charge à un noeud

TYPE=NODE HEAT source de chaleur à un noeud

n1, q1;

n1_{entier ou pointeur}

numéro ou pointeur du noeud correspondant à la source de chaleur.

q1_{réel}

Quantité de chaleur (unité SI : Watt) absorbée par le noeud, c'est-à-dire qu'une valeur positive chauffe la structure.

TYPE=TEMPERATURE températures de part et d'autre d'une coque

n1, t1, t2, G=gradient;

n1_{entier ou pointeur}

numéro ou pointeur d'un noeud sur un élément coque.

t1_{réel}, **t2**_{réel}

températures de la face supérieure (t1) et inférieure (t2) du point sur la coque.

G=gradient_{réel}

gradient thermique dans l'épaisseur de la coque (unité : SI K/m). Dans ce cas, la température <t2> est ignorée et peut ne pas figurer; la température <t1> est prise comme température moyenne au centre de la coque.

Remarque: si le noeud <n1> n'appartient pas à un élément coque, le gradient n'est pas pris en compte et la température du noeud est <t1>.

TYPE=ED PRESSURE charge répartie sur les arêtes des éléments

e1, E1=charge1a,charge1b, E2=charge2a,charge2b,...,DIRECT=direction;

e1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une arête, c'est-à-dire les éléments à contrainte ou déformation planes (PSS et PST), les éléments axisymétriques, les coques (TN), les plaques (PLT) et les poutres (BEAM).

DIRECT=direction_{L,X}

Si le paramètre "DIRECT=X" est mentionné, les charges sont prises dans le repère global. Si ce paramètre est omis, les charges sont appliquées par rapport au repère local de l'élément (par défaut DIRECT=L).

E1=charge1a_{réel}, **charge1b**_{réel}

Charge répartie par unité de longueur sur l'arête 1. Si deux valeurs <charge1a> et <charge1b> sont indiquées, la charge répartie varie de manière linéaire sur la longueur de l'arête et vaut <charge1a> à l'origine de l'arête et vaut <charge1b> à l'extrémité de l'arête.

Les paramètres E2=charge2a, charge2b, E3=charge3a,charge3b,... sont relatifs aux arêtes 2, 3 et suivantes de l'élément. Les conventions adoptées pour la signification des arêtes sont les suivantes :

Type d'élément	charge répartie par unité de longueur (en unité SI)
barreau	
ROD	1 force axiale (N/m)
poutre	
BEAM	1 force axiale (N/m) 2 charge répartie Y (N/m) 3 charge répartie Z (N/m)
Elément axisymétriques (AXI), membranes (MEM) en déformation plane (PST), en contrainte plane (PSS)	
	1 force normale sur l'arête 1 (N/m) 2 force normale sur l'arête 2 3 force normale sur l'arête 3 4 force normale sur l'arête 4 5 force de cisaillement sur l'arête 1 (N/m) 6 force de cisaillement sur l'arête 2 7 force de cisaillement sur l'arête 3 8 force de cisaillement sur l'arête 4
PLT Plaque plane	1 effort tranchant sur l'arête 1 (N/m) 2 effort tranchant sur l'arête 2 3 effort tranchant sur l'arête 3 4 effort tranchant sur l'arête 4 5 moment fléchissant sur l'arête 1 (N.m/m) 6 moment fléchissant sur l'arête 2 7 moment fléchissant sur l'arête 3 8 moment fléchissant sur l'arête 4
TN coque mince	1 force normale sur l'arête 1 (N/m) 2 force normale sur l'arête 2 3 force normale sur l'arête 3 4 force normale sur l'arête 4 5 force de cisaillement sur l'arête 1 (N/m) 6 force de cisaillement sur l'arête 2 7 force de cisaillement sur l'arête 3 8 force de cisaillement sur l'arête 4 9 effort tranchant sur l'arête 1 (N/m) 10 effort tranchant sur l'arête 2 11 effort tranchant sur l'arête 3 12 effort tranchant sur l'arête 4 13 moment fléchissant sur l'arête 1 (N.m/m) 14 moment fléchissant sur l'arête 2 15 moment fléchissant sur l'arête 3 16 moment fléchissant sur l'arête 4

Remarque 1: les éléments triangulaires ne comportent que trois arêtes. Le chargement appliqué sur l'arête N°4 est ignoré.

Remarque 2: si le paramètre "DIRECT=X" est présent, alors les arêtes 1, 2 et 3 pour les poutres correspondent aux efforts appliqués dans les directions X, Y, Z du repère global (cf figure ci-dessous). Attention, seules les contributions orthogonales à l'axe neutre sont prises en compte.

$$\vec{p} = \vec{p}_x \frac{|\vec{p}_x x \vec{e}_x|}{|\vec{p}_x|}$$

Pour l'option DIRECT=X seulement, le vecteur de la charge répartie \mathbf{p}_x est multiplié par le sinus de l'angle entre \mathbf{p}_x et la direction de l'axe neutre \mathbf{e}_x .

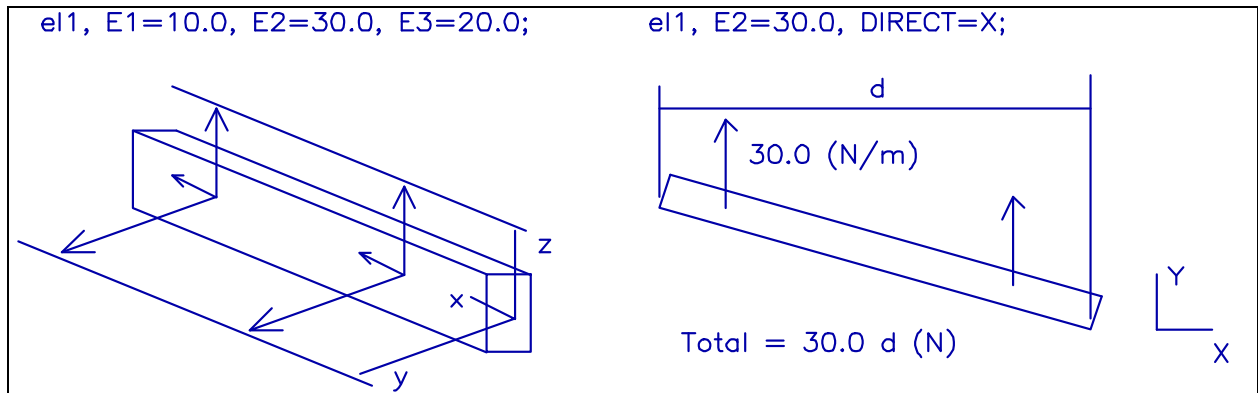


Figure 13 Définition des charges réparties sur des poutres

TYPE=ED FLUX flux de chaleur sur les arêtes des éléments

el1, E1=flux1a,flux1b, E2=flux2a,flux2b, E3=flux3a,flux3b,...;

el1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une arête, c'est-à-dire les éléments à contrainte ou déformation planes (PSS et PST), les éléments axisymétriques, les coques (TN), les plaques (PLT) et les poutres (BEAM).

E1=flux1a_{réel}, **flux1b**_{réel}

Flux de chaleur par unité de longueur sur l'arête 1. Si deux valeurs <flux1a>, <flux1b> sont indiquées, le flux varie de manière linéaire sur la longueur, vaut <flux1a> à l'origine de l'arête et <flux1b> à l'extrémité de l'arête.

Les paramètres E2=,E3=,... sont relatifs aux arêtes 2, 3 et suivantes de l'élément.

TYPE=ED CONVECTION flux de chaleur convective sur les arêtes des éléments

el1, E1=coef1, T1=temper1, E2=coef2, T2=temper2, ...;

el1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une arête, c'est-à-dire les éléments à contrainte ou déformation planes (PSS et PST), les éléments axisymétriques, les coques (TN), les plaques (PLT) et les poutres (BEAM).

E1=coef1_{réel}

Coefficient de convection par unité de longueur sur l'arête 1 (unité SI W/K/m). Les paramètres E2=,E3=,... sont relatifs aux arêtes 2, 3 et suivantes.

T1=temper1_{réel}

Température du fluide au voisinage de l'arête 1. Les paramètres T2=,T3=... sont relatifs aux arêtes 2, 3 et suivantes.

TYPE=ED RADIATION radiation de chaleur sur les arêtes des éléments

el1, E1=coef1, T1=temper1, E2=coef2, T2=temper2, ...;

e1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une arête, c'est-à-dire les éléments à contrainte ou déformation planes (PSS et PST), les éléments axisymétriques, les coques (TN), les plaques (PLT) et les poutres (BEAM).

E1=coef1_{réel}

Coefficient de radiation par unité de longueur sur l'arête 1 (unité SI:W.K⁻⁴/m). Les paramètres E2=coef2, E3=coef3,... sont relatifs aux arêtes 2, 3 et suivantes.

T1=temper1_{réel}

Température au voisinage de l'arête 1.

Les paramètres T2=temper2, T3=temper3,... sont relatifs aux arêtes 2, 3 et suivantes.

TYPE=PRESSURE pression appliquée sur les surfaces d'éléments

e1, F1=press1, F2=press2, ...,F6=press6, DIRECT=direction;

e1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une surface, c'est-à-dire une plaque (PLT), une membrane (MEM), une coque mince (TN) ou un élément solide (SOL).

F1=press1_{réel}

valeur de la pression <press1> sur la surface 1 de l'élément (unité SI : Pa=N/m²). Les paramètres F2=, F3=,..., F6= sont relatifs aux autres surfaces de l'élément, lorsque l'élément possède plus d'une surface.

DIRECT=direction_{L,X,Y ou Z}

Ce paramètre optionnel précise la direction de la pression. Si ce paramètre est omis, il prend la valeur "L". La pression correspond à l'application d'une pression hydrostatique dirigée vers la normale extérieure de l'élément lorsque la pression est positive. Cette valeur sera donc en général négative (efforts de compression). Pour un élément surfacique mince (PLT, MEM, TN), une pression positive applique des efforts dans le sens de la normale définie par la surface de l'élément.

Les valeurs "Y" ou "Z" de ce paramètre font agir la pression dans la direction "Y" ou "Z". Cette option peut être employée pour simuler l'action d'un jet d'eau ou du vent sur une surface.

TYPE=HEAT flux de chaleur appliqué sur les surfaces d'éléments

e1, F1=q1, F2=q2, ...,F6=q6;

Les conventions employées pour TYPE=HEAT sont similaires à celles de TYPE=PRESSURE. Les valeurs réelles indiquées sont en revanche relatives à des flux de chaleur (unités SI : W.m⁻²). Un flux thermique positif réchauffe la structure.

En outre, le paramètre "DIRECT=direction" n'a pas de sens pour les échanges thermiques.

TYPE=CONVECTION flux de chaleur convective appliqué sur les surfaces d'éléments
el1, F1=h1,tf1, F2=h2,tf2, ...,F6=h6,tf6;

el1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une surface, c'est-à-dire une plaque (PLT), une membrane (MEM), une coque mince (TN) ou un élément solide (SOL).

F1=h1_{réel}, **tf1**_{réel}

Valeur du coefficient de convection <h1> (unité SI : W.K.m⁻²) suivi de la valeur de la température tf1 du fluide de convection en contact avec la surface 1 de l'élément. La chaleur surfacique fournie par le fluide à la structure est donc:

$$Q1 = h1*(tf1 - Te)$$

où <Te> est la température de la structure à la surface de l'élément.

Les paramètres F2=h2,tf2, ..., F6=h6,tf6 sont relatifs aux surfaces 2,...,6 de l'élément.

TYPE=RADIATION radiation de chaleur convective sur les surfaces d'éléments
el1, F1=e1,tv1, F2=e2,tv2, ...,F6=e6,tv6;

el1_{entier ou pointeur}

numéro ou pointeur d'un élément possédant une surface, c'est-à-dire une plaque (PLT), une membrane (MEM), une coque mince (TN) ou un élément solide (SOL).

F1=e1_{réel}, **tv1**_{réel}

Valeur du coefficient de radiation <e1> (unité SI : W.K⁻⁴.m⁻²) suivie de la valeur de la température <tv1> au voisinage de la surface 1 de l'élément. La chaleur surfacique fournie par le fluide à la structure est donc :

$$Q1=e1*(tv1^4 - Te^4)$$

où <Te> est la température de la structure à la surface de l'élément. Les paramètres F2=e2,tv2,...,F6=e6,tv6 sont relatifs aux surfaces 2,...,6 de l'élément.

TYPE=VOL_HEAT génération volumique de chaleur dans les éléments
el1, q1;

el1_{entier ou pointeur}

numéro ou pointeur d'un élément.

q1_{réel}

Quantité de chaleur volumique absorbée par l'élément (unité SI : W/m³). Une valeur positive chauffe la structure.

exemples pour l'entité LOAD

LOAD(TYPE=FORCE)

;10, X=5.0, RY=1.0; //force X de 5.0 N et moment Y de 1.0 N.m au noeud 10

LOAD(TYPE=PRESSURE)

; 12, F1=10.0; //pression de 10.0 Pa sur la face 1 de l'élément 12

LOAD(TYPE=ED_PRESSURE)

// charge répartie sur la poutre 20 dans son repère local
; 20, E1=10.0, E2=30.0, E3=20.0; // charge x 10.0 N/m (axiale)
// charge y 30.0 N/m, z 20.0 N/m

LOAD(TYPE=ED_PRESSURE, DIRECT=X)// charge répartie sur la poutre 21

; 21, E2=30.0; // charge de 30.0 N/m dans la direction Y du repère global

NODE, noeud d'élément fini

.
 NODE (COLOR=couleur, COORD=ref1, DOF=ref2)
 numéro, label; x, y, z;

.
 NODE est employé pour définir un noeud devant appartenir à un élément fini.

Paramètres de l'Identificateur d'entité NODE

COLOR=couleur_{entier}
 numéro de couleur de l'élément. Cette option est facultative.

COORD=ref1_{entier ou pointeur}
 numéro ou pointeur du système de référence dans lequel sont définies les coordonnées du noeud (non disponible pour la version ICAB 1.0).

DOF=ref2_{entier ou pointeur}
 numéro ou pointeur du système de référence dans lequel sont définies les degrés de liberté. Par défaut, le système de référence utilisé est le système global.

MESH=maillage_{entier}
 Ce numéro indique que ce noeud a été généré par le mailleur.

Données d'Entrée DE pour l'entité NODE

numéro, label;
 numéro_{entier} numéro du noeud.
 label_{chaîne} nom du noeud pouvant être utilisé comme pointeur.

Remarque: Il faut au moins indiquer le <numéro> ou le <label> du noeud. Si le <numéro> n'est pas indiqué, il est généré par une numérotation automatique. Deux noeuds ne doivent pas porter le même numéro ou le même label.

Paramètres de Données PD pour l'entité NODE

X_{réel}, Y_{réel}, Z_{réel};
 Les paramètres x, y, z indiquent les coordonnées du noeud.

exemples pour l'entité NODE

```
NODE()
1;          0.0, 1.0, 2.5; // noeud portant le numéro 1
2,node2;   1.0, 1.5, 3.0; // noeud portant le numéro 2 et ayant le label node2
N3;        1.0, 1.0, 3.0; // noeud ayant le label N3
```

Dans l'exemple ci-dessus, le noeud portant le numéro 2 pourra être désigné par son label "node2", c'est à dire que "node2" définit le pointeur du noeud.

NOTE, annotation

.
 NOTE ()
 ; _____ NUMTYP=type, _____ POS=x,y,z, _____ DXY=dx,dy, _____ ELEM=nelem,ngauss, _____ NODE=n1,n2, _____
 COMMENT="texte", INVISIBLE=1;
 .

NOTE est employé pour définir un commentaire ou une cotation. _____

Paramètres de l'Identificateur d'entité NOTE

Aucun paramètre.

Données d'Entrée DE pour l'entité NODE

Aucun paramètre.

Paramètres de Données PD pour l'entité NODE

NUMTYP=type_{entier}

Les différents types de commentaires sont:

- 0 commentaire général
- 1 cotation selon l'axe X
- 2 cotation selon l'axe Y
- 3 cotation selon l'axe Z
- 4 cotation entre deux noeuds
- 5 commentaire sur un noeud
- 6 commentaire sur un élément.

POS=X_{réel}, Y_{réel}, Z_{réel}

Position du commentaire.

DXY=dx_{entier}, y_{entier}

Décalage du commentaire sur la feuille (en millimètres).

ELEM=nelem_{entier}, ngauss_{entier}

numéro d'un élément et numéro du point de Gauss de cet élément (0 pour le centre de l'élément).

NODE=n1_{entier}, n2_{entier}

numéros de noeuds.

COMMENT="texte"_{texte}

Texte du commentaire.

INVISIBLE=I_{entier 0/1}

Le commentaire n'est pas tracé si I=1.

exemples pour l'entité NOTE

NOTE()

```
; NUMTYP=0, POS=1.5, 2,0, COMMENT="commentaire à la position (1.5, 2, 0)";
; NUMTYP=1, NODE=1,2; // cotation selon X entre les noeuds 1 et 2
; NUMTYP=6, NELEM=4, DXY=10, COMMENT="commentaire sur l'élément 4";
```

PROPERTY, propriétés et matériaux

PROPERTY (TYPE=propriété, MAT=matériau)
numéro, label; P1=prop1, P2=prop2, ...;

L'entité PROPERTY est associée aux éléments (entité ELEMENT) pour définir les caractéristiques des matériaux (module d'élasticité, coefficient de Poisson pour un matériau isotrope) ou les propriétés physiques de certains éléments (moment d'inertie, section pour les poutres). Les entités ELEMENT comprennent en effet deux paramètres (MAT=, PROP=). Le paramètre MAT= indique les caractéristiques du matériau utilisé dans l'élément. Le paramètre PROP= fait référence aux propriétés physiques.

Paramètres de l'Identificateur d'entité PROPERTY

TYPE=propriété_{chaîne}

chaîne de caractères indiquant le type de propriété. La liste des type de propriété est détaillée ci-dessous.

Liste des types de matériaux.

ISO	matériau isotrope
ORTHO	matériau orthotrope
ANISO	matériau anisotrope
LAMINATE	matériau composite

Liste des types de propriétés physiques.

BEAM_LINEAR	poutre droite à section constante
BUSHING	silentbloc
MASS	masse concentrée
SPRING	ressort
SHELL	coque

MAT=matériau_{entier ou pointeur}

numéro ou pointeur du matériau qui doit être utilisé par défaut pour une propriété physique. Les caractéristiques de ce matériau seront utilisées par défaut lorsque le matériau n'est pas indiqué dans une entité ELEMENT. Ce paramètre ne peut être employé que pour les propriétés physiques et il est optionnel.

Données d'Entrée DE pour l'entité PROPERTY

numéro, label;

numéro_{entier} numéro de la propriété.

label_{chaîne} nom de la propriété pouvant être utilisé comme pointeur.

Remarque: Il faut au moins indiquer le <numéro> ou le <label> de la propriété. Deux propriétés ne doivent pas porter le même numéro ou le même label.

Paramètres de Données PD pour l'entité PROPERTY

P1=prop1, P2=prop2, ...;

Les Paramètres de Données dépendent du type de propriété. Les syntaxes possibles sont reportées ci-dessous. Sauf indication contraire, les paramètres sont des nombres réels. Toutes les propriétés indiquées ne sont pas nécessairement employées dans un calcul. Ainsi, un calcul de mécanique statique linéaire non-thermique ne prendra pas en compte les paramètres tels que l'amortissement, la dilatation thermique, le fluage...

Les paramètres opérationnels dans la version ICAB 1.0 sont indiqués en gras.

TYPE=ISO matériau isotrope (les dimensions physiques des grandeurs sont en unités SI)

E=module, **NU=poisson**, **DEN=**_, **A=a**, ...;

* Caractéristiques mécaniques

E=module_{réel}

Module d'élasticité dit module d'Young (Modulus of elasticity, Pa=N/m²)

NU=u_{réel}

coefficient de Poisson (Poisson's ratio)

G=g_{réel}

module de cisaillement (Shear Modulus, Pa), utilisé si NU n'est pas défini.

La loi de comportement d'un matériau élastique isotrope est rappelée ci-dessous:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\nu & -\nu \\ -\nu & 1 & -\nu \\ -\nu & -\nu & 1 \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{bmatrix}$$

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \end{bmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu \\ \nu & 1-\nu & \nu \\ \nu & \nu & 1-\nu \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \end{bmatrix}$$

$$\varepsilon_{ii} = \frac{\partial X_i}{\partial x_i}, \quad \gamma_{ij} = \frac{\tau_{ij}}{G}, \quad G = \frac{E}{2(1+\nu)}, \quad \gamma_{ij} = \frac{\partial X_j}{\partial x_i} + \frac{\partial X_i}{\partial x_j}$$

avec e déformation
g déformation de cisaillement
s contrainte
t contrainte de cisaillement

DEN=réel

densité massique (Mass density, kg.m⁻³)

A=a_{réel}

coefficient de dilatation thermique (Coefficient of thermal expansion).

La dilatation thermique résultant d'une différence de température ΔT entre la température de la structure et la température de référence (pour laquelle il n'y a pas de dilatation) est: $e = a \cdot \Delta T$.

La température de référence est définie dans une entité "LOAD(TYPE=AM_TEMPERATURE)"

$GE=ge_{réel}$
 amortissement structurel (Structural element damping coefficient)
 L'amortissement structurel introduit un terme complexe dans la matrice du module d'élasticité tel que le module complexe soit $[E^*] = (1 + i \cdot ge) \cdot [E]$

* Caractéristiques thermiques

$KTC=k_{réel}$
 conductivité thermique (Thermal conductivity)

$CP=cp_{réel}$
 chaleur spécifique (Specific heat)

$Q=q_{réel}$
 taux de génération de chaleur (Heat Generation rate)

$CF=cf_{réel}$
 coefficient de chaleur convective (Convective Film coefficient)

$TC=tc_{réel}$
 capacité thermique par unité de surface (Thermal Capacity for unit area)

$HF=hf_{réel}$
 taux de dégagement de chaleur (Heat Flux rate)

$SHF=s_{réel}$
 taux de dégagement de chaleur surfacique (Surface Heat Flux rate)

$AF=af_{réel}$
 facteur d'aire (Area factor)

$EM=em_{réel}$
 émissivité (Emissivity)

$AB=ab_{réel}$
 absorption (Absorbivity)

* Caractéristiques de mécanique non-linéaire et de rupture

$MU=h_{réel}$
 coefficient de friction (coefficient of friction)

$V=m_{réel}$
 viscosité (viscosity)

$YS=s_0_{réel}$
 contrainte de plastification (Yield stress), utilisée comme limite d'élasticité dans le calcul des critères de ruine pour les constructions métalliques (unité SI: Pa=N/m²)

$SC=sc_{réel}$
 coefficient de gonflement (Swelling coefficient)

$XT=s_{1+}_{réel}$
 contrainte en traction X (Allowable stress in tension in X direction)

$XC=s_{1-}_{réel}$
 contrainte en compression X (Allowable stress in compression in X direction)

$YT=s_{2+}_{réel}$
 contrainte en traction X (Allowable stress in tension in Y direction)

$YC=s_{2-}_{réel}$
 contrainte en compression X (Allowable stress in compression in Y direction)

$S=t_0_{réel}$
 contrainte en cisaillement plan (Allowable in-plane shear stress)

$F12=f12_{réel}$
 terme d'interaction pour le critère de Tsai-Wu (Interaction term for Tsai-Wu)

TYPE=ORTHO matériau orthotrope

EX=ex, EY=ey, EZ=ez, NUXY=uxy,... DEN=r, AX=ax, ...;

EX=ex_{réel}

EY=ey_{réel}

EZ=ez_{réel}

modules d'élasticité dans les directions X, Y, Z (Modulus of elasticity)

NUXY=nxy_{réel}

NUYZ=nyz_{réel}

NUXZ=nxz_{réel}

coefficients de Poisson XZ (Poisson's ratio)

GXY=gxz_{réel}

GYZ=gyz_{réel}

GXZ=gxz_{réel}

modules de cisaillement (Shear Modulus)

La loi de comportement d'un matériau élastique orthotrope est rappelée ci-dessous:

$$\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_{xx}} & -\frac{\nu_{yx}}{E_{yy}} & -\frac{\nu_{zx}}{E_{zz}} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_{xx}} & \frac{1}{E_{yy}} & -\frac{\nu_{zy}}{E_{zz}} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_{xx}} & -\frac{\nu_{yz}}{E_{yy}} & \frac{1}{E_{zz}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} \end{bmatrix} \cdot \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix}$$

$$\frac{\nu_{ij}}{E_{ii}} = \frac{\nu_{ji}}{E_{jj}}, \quad \varepsilon_{ii} = \frac{\partial X_i}{\partial x_i}, \quad \gamma_{ij} = \frac{\tau_{ij}}{G_{ij}}, \quad \gamma_{ij} = \frac{\partial X_j}{\partial x_i} + \frac{\partial X_i}{\partial x_j} \quad 2$$

avec e déformation
g déformation de cisaillement
s contrainte
t contrainte de cisaillement

$DEN = \rho_{réel}$	densité massique (Mass density)
$AX = \alpha_{x,réel}$	
$AY = \alpha_{y,réel}$	
$AZ = \alpha_{z,réel}$	coefficients de dilatation thermique en X, Y et Z (Coefficient of thermal expansion)
$GE = g_{réel}$	amortissement structurel (Structural element damping coefficient)
$KX = k_{x,réel}$	
$KY = k_{y,réel}$	
$KZ = k_{z,réel}$	conductivités thermiques en X, Y et Z (Thermal conductivity)
$Q = q_{réel}$	taux de génération de chaleur (Heat Generation rate)
$CF = c_{f,réel}$	convection thermique (convective film coefficient)
$HF = h_{f,réel}$	taux d'émission de chaleur (Heat flux rate)
$SHF = s_{réel}$	taux d'émission de chaleur surfacique (Surface Heat flux rate)
$CP = c_{p,réel}$	chaleur spécifique (Specific heat)
$V = \nu_{réel}$	viscosité (viscosity)
$MU = \mu_{réel}$	coefficient de friction (coefficient of friction)
$SCX = s_{cx,réel}$	coefficient de gonflement en X (Swelling coefficient in X)
$SCY = s_{cy,réel}$	coefficient de gonflement en Y (Swelling coefficient in Y)
$XT = s_{1+ ,réel}$	contrainte en traction X (Allowable stress in tension in X direction)
$XC = s_{1- ,réel}$	contrainte en compression X (Allowable stress in compression in X direction)
$YT = s_{2+ ,réel}$	contrainte en traction X (Allowable stress in tension in Y direction)
$YC = s_{2- ,réel}$	contrainte en compression X (Allowable stress in compression in Y direction)
$S = \tau_{réel}$	contrainte en cisaillement plan (Allowable in-plane shear stress)
$F12 = f_{12,réel}$	terme d'interaction pour le critère de Tsai-Wu (Interaction term for Tsai-Wu)

Un matériau soumis à un état de contrainte plane ne subit pas de ruine selon le critère de Tsai-Wu lorsque:

$$\frac{\sigma_{xx}^2}{X_t X_c} + \frac{\sigma_{yy}^2}{Y_t Y_c} + \frac{\sigma_{xy}^2}{S^2} - (1 + f_{12}) \frac{\sigma_{xx} \sigma_{yy}}{\sqrt{X_t X_c Y_t Y_c}} + \sigma_{xx} \left(\frac{1}{X_t} - \frac{1}{X_c} \right) + \sigma_{yy} \left(\frac{1}{Y_t} - \frac{1}{Y_c} \right) \leq 1$$

$$\text{– Von Mises, } X_t = X_c = Y_t = Y_c = \sigma_e, S = \frac{\sigma_e}{\sqrt{3}}, f_{12} = 0 \quad 3$$

Ce critère est une généralisation du critère de Von Mises applicable aux matériaux isotropes.

TYPE=ANISO matériau anisotrope

RW1=c11,c12,c13,c14,c15,c16, RW2=c22,c23,c24,c25,c26, ..., RW6=c66, ...;

RW1=c11, c12, c13, c14, c15, c16

coefficients de la première ligne de la matrice d'élasticité.

RW2=c22, c23, c24, c25, c26

coefficients de la deuxième ligne de la matrice d'élasticité.

RW3=c33, c34, c35, c36

coefficients de la troisième ligne de la matrice d'élasticité.

RW4= c44, c45, c46

coefficients de la quatrième ligne de la matrice d'élasticité.

RW5= c55, c56

coefficients de la cinquième ligne de la matrice d'élasticité.

RW6= c66

coefficient de la sixième ligne de la matrice d'élasticité.

La loi de comportement d'un matériau élastique anisotrope est rappelée ci-dessous:

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{12} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ c_{13} & c_{23} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & c_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & c_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & c_{66} \end{bmatrix} \cdot \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{bmatrix}, \quad c_{ij} = c_{ji} \quad 4$$

avec e déformation
g déformation de cisaillement
s contrainte
t contrainte de cisaillement

TEV=a1, a2, a3, a4, a5

Vecteur des coefficients de dilatation thermique (Thermal Expansion Vector)

KKM=k1, k2, k3, k4, k5, k5

Matrice de conductivité thermique (thermal conductivity Matrix)

GE=ge, DEN=_, CP=cp, XT=s1+, XC=s1-, YT=s2+, YC=s2-, S=t0, F12=f12, SCX=scx, SCY=scy

Les définitions de ces paramètres sont identiques à celles données pour l'entité:

PROPERTY(TYPE=ORTHO)

TYPE=LAMINATE matériau composite

M=m11,m12,m13,m22,m23,m33, ...;

M=m11, m12, m13, m22, m23, m33

coefficients de rigidité de membrane.

B=b11, b12, b13, b22, b23, b33

coefficients de rigidité de flexion.

C=c11, c12, c13, c22, c23, c33

coefficients de rigidité de couplage membrane-flexion.

S=s11, s12, s22

coefficients de rigidité de cisaillement transverse

LTK=h

épaisseur du composite.

La loi de comportement d'un matériau composite est rappelée ci-dessous:

$$\begin{bmatrix} N_{xx} \\ N_{yy} \\ N_{xy} \\ M_{xx} \\ M_{yy} \\ M_{xy} \\ T_{xz} \\ T_{yz} \end{bmatrix} = \begin{bmatrix} h[M] & h^2[C] & 0 \\ h^2[C]^T & \frac{h^3}{12}[B] & 0 \\ 0 & 0 & \frac{5}{6}h[S] \end{bmatrix} \cdot \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \\ -\varepsilon_{xx} \\ -\varepsilon_{yy} \\ -\varepsilon_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} \quad 5$$

$$[M] = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix}, [B] = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{12} & b_{22} & b_{23} \\ b_{13} & b_{23} & b_{33} \end{bmatrix}$$

$$[C] = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{12} & c_{22} & c_{23} \\ c_{13} & c_{23} & c_{33} \end{bmatrix}, [S] = \begin{bmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{bmatrix} \quad 6$$

- e déformation de membrane sur la surface médiane
- c courbures
- g déformation de cisaillement sur la surface médiane
- N efforts de membrane (force linéique, N/m)
- M moments fléchissants linéiques (N)
- T efforts tranchants linéiques

TYPE=BEAM LINEAR

propriétés physiques pour poutres de section constante

AR=ar, IYY=iyy, ..., SRY=sry;**AR=ar_{réel}**section de la poutre (Area, m²)**IYY=iyy_{réel}****IZZ=izz_{réel}**moments d'inertie Y et Z (Moment of inertia, m⁴). Les définitions de ces coefficients sont rappelées ci-après:

$$AR = \iint_{\partial S} ds$$

$$I_{yy} = \iint_{\partial S} z^2 ds$$

$$I_{zz} = \iint_{\partial S} y^2 ds$$

Le moment d'inertie I_{yy} est relatif à des rotations autour de l'axe (y) de la poutre (donc pour des moments fléchissants M_{yy} et des efforts tranchants T_z). Ce coefficient I_{yy} intervient pour une flexion de la poutre soumise à une charge répartie orientée dans la direction (z) du repère local de la poutre.

TC=tc_{réel}constante de torsion (Torsional constant, m⁴)**SRY=sry_{réel}****SRZ=srz_{réel}**

coefficients de correction pour la prise en compte du cisaillement transverse dans les directions du repère local de la poutre Y et Z. (Shear area Ratio Y,Z). Le coefficient <srz> est relatif aux chargements dans le plan (xz) de la poutre et va donc affecter le calcul des moments fléchissants M_{yy} .

TKY=tky_{réel}**TKZ=tkz_{réel}**

épaisseurs de la section dans les directions Y et Z du repère local de la poutre (unité SI : m).

IVY=(I/v)_y_{réel}**IVZ=(I/v)_z_{réel}**

Ces coefficients sont employés pour les calculs de contraintes normales à la section de la poutre en flexion. Si la poutre est soumise à un pur moment fléchissant M_{yy} , alors les contraintes s_{xx} dans la section de la poutre sont telles que $|s_{xx}| \leq |M_{yy}|/(I/v)_y$.

Pour un pur moment fléchissant M_{zz} , nous avons $|s_{xx}| \leq |M_{zz}|/(I/v)_z$.
(unité SI : m³)

ITC=(J/r)_{réel}

Ce coefficient est employé pour les calculs de contraintes de cisaillement associées à une torsion. Si la poutre est soumise à un pur moment de torsion M_{xx} , alors les contraintes de cisaillement t dans la section de la poutre sont telles que $|t| \leq |M_{xx}|/(J/r)$ (unité SI : m³).

ARY=ary_{réel}**ARZ=arz_{réel}**

Ces coefficients sont employés pour les calculs de contraintes de cisaillement lorsque la poutre est soumise à des efforts tranchants: $t_y = T_y/ary$, $t_z = T_z/arz$.
(unité SI : m²).

EA=ea_{réel}

Epaisseur de l'âme pour les poutres en I. Ce coefficient sert au calcul du critère de voilement (unité SI : m).

LKM=Lkm_{réel}

Si la longueur de la poutre est inférieure à Lkm, la longueur de flambement ou la portée de la poutre prise en compte est Lkm. Ce paramètre intervient dans les critères de ruine de flambement, de déversement et de flèche admissible.

LDM=Ldm_{réel}

Longueur minimale de déversement.

SP=S_{réel}

Moment statique relatif à des moments autour de l'axe (y). Le moment statique permet de tenir compte de la plastification de la poutre dans les critères de ruine de constructions métalliques avec le coefficient $y=2S/(I_{yy}/v)$. (unité SI : m³)

CFI=c0_{réel}

coût fixe de chaque poutre.

CVA=c1_{réel}

coût par unité de longueur de chaque poutre. Le coût d'utilisation d'une poutre est: $c=c_0+c_1 \times L$, où L est la longueur de la poutre.

Remarque 1: Le repère local de chaque poutre est défini dans l'entité "ELEMENT(TYPE=BEAM_LINEAR)"

NCM=Nx_c_{réel}

Effort de compression maximum admissible

NTM=Nx_t_{réel}

Effort de traction maximum admissible

NX=Nx_{réel}

Tension initiale de précontrainte

MYA=-My1_{réel}

Moment Myy initial de précontrainte à l'extrémité 1

MYB=-My2_{réel}

Moment Myy initial de précontrainte à l'extrémité 2

MZA=Mz1_{réel}

Moment Mzz initial de précontrainte à l'extrémité 1

MZB=Mz2_{réel}

Moment Mzz initial de précontrainte à l'extrémité 2

TYPE=BEAM_TAPERED propriétés physiques pour poutres de section variable

Les sections de ces poutres varient linéairement entre l'origine (notée 1) et l'extrémité (notée 2). Les propriétés sont identiques à celles des poutres de type BEAM_LINEAR mais définies 2 fois, à l'origine et à l'extrémité, pour les valeurs AR, IYY, IZZ, SRY, SRZ, ITC, SP, EA.

Les dimensions de la poutre sont définies par:

TYT: "demi"-largeur dans la direction Y+

TYB: "demi"-largeur dans la direction Y-

TZT: "demi"-hauteur dans la direction Z+

TZB: "demi"-hauteur dans la direction Z-

TYPE=BUSHING propriétés physiques pour silentblocs (appui élastique)

K=kx, ky, kz, krx, kry, krz;

K=kx_{réel}, ky_{réel}, kz_{réel}, krx_{réel}, kry_{réel}, krz_{réel}

Kij=k_{réel}, (1 ≤ i ≤ 6, i ≤ j ≤ 6)

rigidités (N/m pour kx, ky, kz), (N.m/rad pour krx, kry, krz).

Un appui élastique "BUSHING" (cf entité ELEMENT(TYPE=BUSHING)) est assimilable à une barre rigide entre les noeuds <n1,n1'> associée à une élasticité [K] localisée à l'extrémité de cette barre entre les noeuds <n1',n2>. <n1> est le premier noeud, <n2> le deuxième noeud, <n1'> est un noeud fictif localisé au même emplacement que <n2> si l'appui ne subit aucune déformation.

Nous notons F1 le torseur force associé au noeud <n1'> dans le repère local de l'élément "BUSHING" ayant pour composantes (Fx1, Fy1, Fz1, Mx1, My1, Mz1) et D1 le vecteur des déplacements et des rotations du noeud n1'. Nous adoptons des notations analogues pour le deuxième noeud <n2> (F2, D2). Les relations établies par l'élément "BUSHING" sont les suivantes:

$$\begin{bmatrix} \vec{F1} \\ \vec{F2} \end{bmatrix} = \begin{bmatrix} [K] & -[K] \\ -[K] & [K] \end{bmatrix} \begin{bmatrix} \vec{D1} \\ \vec{D2} \end{bmatrix}, [K] = \begin{bmatrix} k_x & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_y & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_z & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{rx} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{ry} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{rz} \end{bmatrix}$$

$$[K] = [K]^T$$

NB : la matrice [K] est symétrique et seuls les termes diagonaux et triangulaires supérieurs (j ≥ i) sont indiqués. Les indices utilisés sont: 1 pour la translation X (2=Y, 3=Z), 4 pour la rotation RX (5=RY, 6=RZ).

CFI=c0_{réel}

coût fixe de chaque silentbloc.

TYPE=MASS propriétés physique d'une masse

MA=mass, MIM=jxx, jyy, jzz, jxy, jxz, jyz;

MA=mass_{réel}

Masse (unité SI: kg)

MIM=jxx_{réel}, jyy_{réel}, jzz_{réel}, jxy_{réel}, jxz_{réel}, jyz_{réel}

Matrice d'inertie (unité SI: kg.m²):

$$[J] = \begin{bmatrix} j_{xx} & -j_{xy} & -j_{xz} \\ -j_{xy} & j_{yy} & -j_{yz} \\ -j_{xz} & -j_{yz} & j_{zz} \end{bmatrix}$$

CFI=c0_{réel}

coût fixe de chaque masse.

TYPE=ROD propriété physique des barres élastiques

AR=ar;

AR=ar_{réel}

section de la poutre (Area, m²)

CFI=c0_{réel}

coût fixe de chaque barre.

CVA=c1_{réel}

coût par unité de longueur de chaque barre. Le coût d'utilisation d'une barre est: $c=c_0+c_1 \times L$, où L est la longueur de la barre.

Remarque: Le type de matériau doit être indiqué dans une entité PROPERTY(TYPE=ISO).

TYPE=SPRING propriétés physiques pour ressorts.

K=k, MA=mass;

K=k_{réel}

rigidité du ressort (N/m = kg.s⁻²).

MAF=mass_{réel}

masse du ressort (kg).

CFI=c0_{réel}

coût fixe de chaque ressort.

TYPE=SHELL propriétés physiques pour coques.

TK=h;

TK=h_{réel}

épaisseur de la coque (m).

Ce paramètre n'est employée que lorsque la coque est constituée d'un matériau isotrope ou orthotrope. Pour les coques composites, reportez-vous à l'entité PROPERTY(TYPE=LAMINATE).

exemples pour l'entité PROPERTY

```
/*      caractéristiques d'un matériau isotrope      */
PROPERTY(TYPE=ISO)
steel;      E=210.0E9, NU=0.3, DEN=7.8E3;
```

```
/*      propriétés physiques pour poutre      */
PROPERTY(TYPE=BEAM_LINEAR)
beam1; AR=1.0E-3, IYY=2.0E-7, IZZ=1.6E-7, TC=3.2E-7;
```

```
/*      masse ponctuelle de 15.1 avec comme matrice d'inertie
*      Jxx=1.1, Jyy=1.3, Jzz=1.1, Jxy=0.3, Jyz=0.1, Jxz=0.2
*/
PROPERTY(TYPE=MASS)
mass1; MA=15.1, MIM=1.1, 1.3, 1.1, 0.3, 0.1, 0.2;
```

```
/*      exemple de définition de propriété physique
*      avec la référence à un matériau par défaut
*      (matériau "steel", défini dans le premier exemple ci-dessus)
*/
PROPERTY(TYPE=ROD, MAT=steel)
bar1; AR=1.0E-3;
```

```
/*      ainsi, si un élément barre de type "ROD" utilise la propriété "PROP=bar1"
*      sans indiquer un matériau, alors le matériau utilisé est "steel"
*      exemple:
*/
ELEMENT(TYPE=ROD, PROP=bar1)
el1; N1, N2;
```

```
/*      élément poutre utilisant "beam1" et "steel"      */
ELEMENT(TYPE=BEAM_LINEAR, PROP=beam1, MAT=steel)
el2; N3, N4, N5;
```


REFERENCE, système de coordonnées ou repère

REFERENCE (COLOR=couleur,TYPE=ref)

numéro, label; UO= x_0, y_0, z_0 , UX= x_x, x_y, x_z , UY= y_x, y_y, y_z , UZ= z_x, z_y, z_z ,
NX= n_x , NZ= n_z , NO= n_0 ;

REFERENCE est employé pour définir un noeud devant appartenir à un élément fini.

Paramètres de l'Identificateur d'entité REFERENCE

COLOR=couleur_{entier}

numéro de couleur du référentiel. Ce paramètre est facultatif.

TYPE=ref_{entier}

numéro du type de référentiel, à savoir 1, 2 ou 3 (voir la section ci-après relative aux Paramètres de Données pour l'entité REFERENCE).

Données d'Entrée DE l'entité REFERENCE

numéro, label;

numéro_{entier} numéro du référentiel.

label_{chaîne} nom du noeud pouvant être utilisé comme pointeur.

Remarque: Il faut au moins indiquer le <numéro> ou le <label> du référentiel. Deux référentiels ne doivent pas porter le même numéro ou le même label.

Paramètres de Données PD pour l'entité REFERENCETYPE=1

Le référentiel est défini par une origine et deux vecteurs dont les paramètres sont:

UO= x_0, y_0, z_0 _{réel}
coordonnées de l'origine O.

UX= x_x, x_y, x_z _{réel}
composantes d'un vecteur définissant la direction X.

UZ= z_x, z_y, z_z _{réel}
composantes d'un vecteur définissant la direction Z de telle sorte que le vecteur UZ soit orthogonal à la direction X et Y. UZ ne doit pas être colinéaire au vecteur UX.

TYPE=2

Le référentiel est défini par une origine et trois vecteurs formant une matrice orthonormée.

UO= x_0, y_0, z_0 _{réel}
coordonnées de l'origine O.

UX= x_x, x_y, x_z _{réel}
composantes du vecteur définissant la direction X.

UY= y_x, y_y, y_z _{réel}
composantes du vecteur définissant la direction Y.

UZ= z_x, z_y, z_z _{réel}
composantes du vecteur définissant la direction Z.

Remarque: les vecteurs UX, UY et UZ doivent être normés et orthogonaux deux à deux.

TYPE=3

Le référentiel est défini par trois noeuds (entité NODE).

$NO=n0$ _{entier ou pointeur}
noeud servant d'origine au repère.

$NX=nx$ _{entier ou pointeur}
noeud définissant la direction X par la droite NO-NX.

$NZ=nz$ _{entier ou pointeur}
noeud définissant la direction Z telle que la droite NO-NZ est perpendiculaire aux directions X et Y.

exemples pour l'entité REFERENCE

```
/*_   référentiel de type 1, défini par une origine de coordonnées UO,
*     et deux directions UX et UZ
*/
```

```
REFERENCE(TYPE=1)
1,ref1; UO=1.0, 2.0, 3.0,
        UX=1.0, 0.0, 0.0,
        UZ=0.0, 1.0, 1.0;
```

```
/*_   référentiel de type 2, défini par la matrice R=[UX,UY,UZ] et
*     l'origine UO
*/
```

```
REFERENCE(TYPE=2)
2,ref;   UO=1.0, 2.0, 3.0,
        UX=1.0, 0.0, 0.0,
        UY=0.0, 0.707, -0.707,
        UZ=0.0, 0.707, 0.707;
```

```
/*_   référentiel de type 3 défini par des noeuds (entité NODE)
*     NO (origine), NX (la droite NO-NX est l'axe Ox), NZ (la droite NO-NZ est
*     perpendiculaire à l'axe Ox et Oy).
*/
```

```
REFERENCE(TYPE=3)
3,ref3; NO=10, NX=2, NZ=45;
```

RESTRAINT, conditions limites

RESTRAINT (TYPE=charge, CASE=chargement, FUNCTION=variation)
 numéro, label; n1, X=dx, Y=dy, ..., RZ=rz;
 numéro, label; n1, T=temper;

L'entité RESTRAINT est utilisée pour imposer des variations sur les déplacements ou les températures des noeuds d'un maillage d'éléments finis. Cette entité est donc employée pour définir les conditions aux limites d'un modèle par éléments finis, conjointement avec les entités CONSTRAINT et LOAD.

Paramètres de l'Identificateur d'entité RESTRAINT

CASE=chargement_{entier}

numéro d'application du chargement (Load Case). Ce paramètre optionnel est nul par défaut et dans ce cas le chargement est appliqué en permanence. Ce paramètre permet de traiter plusieurs cas de chargements avec le même modèle.

FUNCTION=variation_{entier ou pointeur}

numéro ou pointeur d'une fonction (entité FUNCTION). Ce paramètre optionnel est utilisé pour les simulations de chargement variable dans le temps. L'amplitude du chargement indiqué dans la section "Paramètre de Données" est multipliée par la valeur de la fonction.

TYPE=charge_{chaîne}

Chaîne de caractères "DISPLACEMENT" ou "TEMPERATURE" indiquant l'application d'un déplacement ou d'une température imposée.

Données d'Entrée DE pour l'entité RESTRAINT

numéro, label;

numéro_{entier} numéro de l'entité.

label_{chaîne} nom de l'entité pouvant être utilisé comme pointeur.

Remarque: ni le <numéro>, ni le <label> ne sont indispensables. En l'absence de ces informations, aucune référence ne peut être faite pour l'entité. L'indication du <label> suffit pour une référence ultérieure à cette entité de chargement.

Paramètres de Données PD pour l'entité RESTRAINT

TYPE=DISPLACEMENT déplacement imposé en translation ou en rotation

n1, X=dx, Y=dy, Z=dz, RX=rx, RY=ry, RZ=rz;

n1_{entier ou pointeur}

numéro ou pointeur d'un noeud (entité NODE) soumis à un déplacement imposé.

X=dx_{réel}

Ce paramètre permet de préciser la valeur du déplacement imposé sur le noeud <n1> dans la direction "X" (unité SI: m).

Y=dy_{réel}
Z=dz_{réel}

Les paramètres Y=dy, Z=dz correspondent aux déplacements imposés dans les directions Y et Z.

RX=rx_{réel}
RY=ry_{réel}
RZ=rz_{réel}

Les paramètres RX=rx, RY=ry et RZ=rz correspondent aux rotations imposées autour des axes X, Y et Z (unité SI: rad).

Remarque 1: Si un déplacement doit être appliqué à un noeud dans une seule direction, par exemple X, alors seul le paramètre X=dx doit figurer. Les autres degrés de liberté du noeud <n1> sont libres.

Remarque 2: Il est possible d'utiliser une entité "CONSTRAINT(TYPE=KINEMATICS)" pour supprimer des degrés de liberté, ce qui revient à les bloquer. Par contre l'entité "RESTRAINT(TYPE=DISPLACEMENT)" doit être utilisée pour permettre l'obtention des forces de réaction aux noeuds dont le déplacement est imposé (voire nul) dans une ou plusieurs directions.

TYPE=TEMPERATURE température imposée
n1, T=temper;

n1_{entier ou pointeur}

numéro ou pointeur d'un noeud (entité NODE) soumis à une température imposée.

T=temper_{réel}

valeur de la température au noeud <n1> (unité SI: K).

NB: la valeur <temper> peut figurer sans être précédée de l'identificateur "T=".

exemples pour l'entité RESTRAINT

RESTRAINT(TYPE=DISPLACEMENT)

; 1, X= 0.1, Y=0.0; // déplacement imposé de 0.1 en X et blocage en Y pour le noeud 1

RESTRAINT(TYPE=TEMPERATURE,CASE=1)

*; 2, T= 100.0; /*_ température imposée de 100.0 pour le noeud 2 */*

*; 3, 200.0; /*_ température imposée de 200.0 pour le noeud 3 */*

STEP, préparation d'un calcul

STEP ()
 numéro, label; MODEL="titre du modèle", RUN="intitulé du calcul",
 LOAD=charge1,ampli1, charge2,ampli2..., charge4, ampli4;

STEP est employé pour définir la préparation d'un calcul. Cette entité est facultative dans la mesure où la préparation d'un calcul peut être faite avec une boîte de dialogue dans l'application ICAB Force.

Paramètres de l'Identificateur d'entité STEP

Il n'y a pas de paramètre.

Données d'Entrée DE pour l'entité STEP

numéro, label;

numéro_{entier} numéro du calcul.

label_{chaîne} nom du calcul pouvant être utilisé comme pointeur.

Remarque: Il est déconseillé de faire figurer le numéro du calcul; en effet, la numérotation des calculs doit être continue et commencer par le numéro 1. Sans numéro de calcul, la numérotation est automatique.

Paramètres de Données PD pour l'entité STEP

MODEL="titre du modèle"_{chaîne de caractères}

Titre du modèle qui doit être indiqué entre des guillemets si votre texte est supérieur à 8 lettres ou si vous indiquez plusieurs mots.

RUN="intitulé du calcul"_{chaîne de caractères}

Intitulé du calcul qui doit être indiqué entre des guillemets si votre texte est supérieur à 8 lettres ou si vous indiquez plusieurs mots.

LOAD=charge1_{entier}, ampli1_{réel}, charge2_{entier}, ampli2_{réel}, ..., charge4_{entier}, ampli4_{réel}

Vous indiquez la combinaison des chargements qui sera utilisée pour le calcul. Jusqu'à 4 chargements peuvent être combinés. Les paramètres <charge1>, <charge2>, <charge3>, <charge4> sont les numéros des chargements tels qu'ils ont été indiqués dans le paramètre CASE des entités LOAD(CASE=<charge>) ainsi que pour CONSTRAINT() et RESTRAINT(). Le paramètre <ampli1> (respectivement <ampli2>, <ampli3>, <ampli4>) est le coefficient multiplicatif appliqué au chargement <charge1> (respectivement <charge2>, <charge3>, <charge4>). Si le paramètre LOAD est omis, tous les chargements sont pris en compte avec un coefficient multiplicatif de 1.

2.0.1 exemples pour l'entité STEP

STEP()

; MODEL="POUTRE EN FLEXION", RUN="effet de la neige", LOAD=1, 1.67;

Le calcul prend en compte le chargement 1 amplifié de 67%.

UNIT, système d'unités

```
.
UNIT      ()
; _____ M=coef_m, KG=coef_kg, S=coef_s, K=coef_K, _____
            LENGTH="longueur", MASS="masse", TIME="temps", FORCE="force",
            TEMPERATURE="température", TYPE="prédéfini";
```

UNIT est employé pour définir le système d'unités utilisé dans le projet.

Paramètres de l'Identificateur d'entité UNIT

Il n'y a pas de paramètre.

Données d'Entrée DE pour l'entité UNIT

Il n'y pas d'indication. Un seul système d'unités doit être défini dans un projet.

Paramètres de Données PD pour l'entité UNIT

LENGTH="longueur" chaîne de caractères
Nom de l'unité de longueur.

MASS="masse" chaîne de caractères
Nom de l'unité de masse.

TIME="temps" chaîne de caractères
Nom de l'unité de temps.

FORCE="force" chaîne de caractères
Nom de l'unité de force.

TEMPERATURE="température" chaîne de caractères
Nom de l'unité de température.

M=coef_m_{réel}
Facteur de conversion de l'unité de longueur en mètre (m).

KG=coef_kg_{réel}
Facteur de conversion de l'unité de masse en kilogramme (kg).

S=coef_s_{réel}
Facteur de conversion de l'unité de temps en seconde (s).

K="coef_K_{réel}"
Facteur de conversion de l'unité de température en Kelvin (K, 1 _K= 1 _°C).

TYPE="prédéfini" chaîne de caractères
Système d'unité prédéfini parmi : *SI* (unités par défaut du système international), *N_m* (équivalent à SI), *N_mm* (Newton, millimètre), *daN_mm* (décaNewton, millimètre), *daN_cm* (décaNewton, centimètre), *daN_m* (décaNewton, mètre), *Tf_m* (tonne-force, mètre), *kN_m* (kiloNewton, mètre), *psi_inch* (pound per squared inch, inch), *lb_foot* (pound-force, foot). Ce paramètre doit être seul ou placé en dernière position.

exemples pour l'entité UNIT

```
UNIT()
; LENGTH="mm", MASS="Mg", TIME="s", FORCE="N",
  M=0.001, KG=1000.0, S=1.0, K=1.0;
```

Choix du millimètre, du MégaGramme (Tonne), de la seconde et donc du Newton=Mg.mm.s⁻² comme unité de force, équivalent à :

```
UNIT()
; TYPE="N_mm";
```