Diagramme d’interaction 2D/3D

Programme développé par Xavier MORIN – BE BYTP

Le programme DI permet de calculer le diagramme d’interaction de toute section composée de contours et de nuages de points associés à des lois quelconques de déformation-contraintes telles que les définissent les règlements de calculs.

Le programme est développé pour fonctionner directement sur Excel. Par l’intermédiaire de lignes de commandes, l’utilisateur doit fournir la géométrie de la section, les lois déformation-contrainte et les pivots de calcul.

La première partie de cette note présente la théorie et les principes fondamentaux retenus pour le calcul.

La seconde partie présente le fonctionnement du programme et de chacune des commandes autorisées.

La troisième partie présente l’architecture du programme et s’adresse aux développeurs désireux de comprendre le fonctionnement du programme ou souhaitant apporter des commentaires et conseils pour les développements futurs.

Pour tout bug constaté ou tout problème lié à l’utilisation du programme (si l’aide n’apporte pas la réponse à la question) : [x.morin@bouygues-construction.com](mailto:x.morin@bouygues-construction.com)

Table des matières

[Théorie et principes du programme 3](#_Toc428347586)

[Diagramme d’interaction – qu’est-ce que c’est ? 3](#_Toc428347587)

[Hypothèses 3](#_Toc428347588)

[Traitement des contours 3](#_Toc428347589)

[Fonctionnement du programme 4](#_Toc428347590)

[Repère, convention de signe et unités 4](#_Toc428347591)

[Présentation générale des lignes de commandes 5](#_Toc428347592)

[Description des fonctions 6](#_Toc428347593)

[(\*)DI 6](#_Toc428347594)

[(\*)LOI 6](#_Toc428347595)

[(+)CONTOUR 8](#_Toc428347596)

[(+)POINTS 10](#_Toc428347597)

[(\*)PIVOTS 11](#_Toc428347598)

[(-)CENTRE 12](#_Toc428347599)

[(-)NBANGLES 13](#_Toc428347600)

[(-)NBDIV 14](#_Toc428347601)

[(-)TOLC 14](#_Toc428347602)

[(\*)EOF 15](#_Toc428347603)

[Rapport d’exécution 16](#_Toc428347604)

[Exemples 16](#_Toc428347605)

[Maintenance du programme et programmation 17](#_Toc428347606)

[Structure des données 17](#_Toc428347607)

[Classe Problème 17](#_Toc428347608)

[Classe Loi 18](#_Toc428347609)

[Classe Contour 18](#_Toc428347610)

[Classe Points 18](#_Toc428347611)

[Classe Pivot 19](#_Toc428347612)

[Structure des résultats 20](#_Toc428347613)

[classe calcul\_par\_angle 20](#_Toc428347614)

[classe calcul\_par\_Pivot 20](#_Toc428347615)

[Structure du programme 21](#_Toc428347616)

[Lecture des données 21](#_Toc428347617)

[Calcul 21](#_Toc428347618)

[Sortie des résultats 21](#_Toc428347619)

[À FAIRE 21](#_Toc428347620)

# Théorie et principes du programme

## Diagramme d’interaction – qu’est-ce que c’est ?

Un diagramme d’interaction est une représentation graphique du domaine de validité des torseurs applicables à une section.

Si pour une section un torseur est défini par deux variables (N et My par exemple), alors le domaine de validité des torseurs sera analogue à une surface délimitée par un contour.

Si un torseur est défini par trois variables (N, My et Mz par exemple), alors le domaine de validité des torseurs sera analogue à un volume délimité par une surface.

Le domaine de validité des torseurs applicables à une section est calculé à partir du domaine de validité des états de déformation de la section.

Les états de déformation de la section sont limités par les critères portant sur les matériaux :

* limitation en déformation (généralement aux ELU) ;
* limitation en contrainte (généralement aux ELS).

## Hypothèses

« addition » des éléments de section. (ex : si des aciers se trouvent dans un contour, le béton au droit des armatures n’est pas supprimé).

## Traitement des contours

sens important

# Fonctionnement du programme

## Repère, convention de signe et unités

La section étudiée est décrite dans le plan (Oyz).

La convention de signe sur les déformations est la suivante :

* déformation positive = raccourcissement ;
* déformation négative = allongement.

Les torseurs (*N*, *My*, *Mz*) sont calculés comme suit :

Ainsi, si l’on renseigne des lois de comportements qui traduisent une compression lorsque la déformation est positive, on obtient la convention de signe suivante sur les torseurs calculés :

* un effort normal N positif correspond à une compression ;
* un moment My positif comprime la fibre z+ ;
* un moment Mz positif comprime la fibre y-.

|  |
| --- |
| ^z  |  x |  <-----+ <-- <) (N>0; My>0 et Mz >0 si sens direct)  \  \  y |

Le programme est adimensionnel mais les unités suivantes sont recommandées :

* MPa pour les contraintes ;
* MN, MN.m pour les efforts et moments ;
* m, m2, m4 pour les dimensions/positions, surfaces et inerties.

Tout autre système d’unités est autorisé mais doit être cohérent sur l’ensemble du fichier pour que les résultats aient un sens (par exemple kN avec mm).

Par ailleurs, les déformations sont sans unité et les angles sont en radians.

## Présentation générale des lignes de commandes

Les commandes sont saisies dans les 5 premières colonnes d’une feuille de calcul Excel.

Ces commandes doivent être saisies entre une balise de début d’instructions DI et une balise de fin d’instruction EOF (End Of File). Rien de ce qui sera écrit avant DI ou après EOF ne sera interprété par le programme.

La colonne A ne peut contenir que les commandes autorisées ; toute autre saisie renvoie une erreur et force l’arrêt du programme.

Les colonnes B, C, D et E contiennent les paramètres associés à chacune des commandes.

**Bloc d’instruction**

Certaines commandes peuvent nécessiter plusieurs lignes pour leur définition (contours, nuages de points, loi déformation-contrainte, pivots) ; ces lignes sont interprétées par le programme jusqu’à l’apparition d’une ligne vide ou d’une nouvelle commande en colonne A.

On appellera par la suite « bloc d’instruction », des lignes définies par le même mot-clef en colonne A.

Légende pour la description des commandes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| COMMANDE | *Param 1* | *Param 2* |  |  |
|  | *y1* | *z1* |  |  |
|  | *y2* | *z2* |  |  |
|  | *…* | *…* |  |  |

|  |  |
| --- | --- |
| COMMANDE | Cellule contenant le nom de la commande, insensible à la casse |
| *Param1, y1,…* | Cellule contenant des paramètres relatifs à la commande appelée |
|  | Cellule interdite à l’écriture |
|  | Cellule non lue par le programme, peut contenir des commentaires |

Les lignes vides entre deux commandes sont autorisées, voire recommandées pour plus de clarté.

Les symboles suivant dans les titres des descriptions des fonctions définissent leur caractère obligatoire :

* (\*) désigne une commande obligatoire ;
* (+) signifie qu’au moins une commande de ce type est obligatoire (CONTOUR/POINTS) ;
* (-) désigne une commande facultative (paramètres de calcul pris alors par défaut).

## Description des fonctions

### (\*)DI

La commande DI précède l’ensemble des instructions que le programme interprète. Les lignes précédant la commande DI ne sont pas interprétées par le programme et sont accessibles à l’utilisateur.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| DI |  |  |  |  |

### (\*)LOI

La commande LOI permet de définir une loi déformation-contrainte.

Une loi est définie par une série de points (déformation ; contrainte) saisie dans l’ordre croissant des déformations.

La série doit contenir au moins 2 points et n’est pas limitée supérieurement.

La définition de la loi doit permettre de « passer » par l’origine du repère (0 ; 0).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | *(mot-clef)* |  |  |

#### Loi polygonale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* |  |  |  |
|  | *eps\_1* | *sig\_1* |  |  |
|  | *eps\_2* | *sig\_2* |  |  |
|  | *…* | *…* |  |  |
|  | *eps\_n* | *sig\_n* |  |  |

* *id\_loi* : identifiant de loi sous forme d’une chaine de caractères ;
* *(eps\_i ; sig\_i)* : couples de valeurs définissant la loi.

L’interprétation par le programme est la suivante :

* la présence d’une nouvelle commande en colonne A ou l’absence d’une valeur en colonne B provoque la fin de définition de la série ;
* l’absence de valeur en colonne C est interprétée comme *sig\_i*=0 ;
* interpolation linéaire entre les points ;
* renvoie *sig\_1* si la déformation calculée est inférieure à *eps\_1* ;
* renvoie *sig\_n* si la déformation calculée est supérieure à *eps\_n*.

#### Loi EC2-3.2 – Béton – Loi de Sargin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | **EC2-3.2** |  |  |
|  | *fck* |  |  |  |

#### Loi EC2-3.3 – Béton – Loi parabole-rectangle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | **EC2-3.3** |  |  |
|  | *fck* | *γc/αcc* |  |  |

#### Loi EC2-3.4 – Béton – Loi bilinéaire

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | **EC2-3.4** |  |  |
|  | *fck* | *γc/αcc* |  |  |

#### Loi EC2-3.8 – Aciers passifs – palier incliné

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | **EC2-3.8** |  |  |
|  | *fyk* | *γs* | Es | k |
|  | *εud* | *εuk* |  |  |

#### Loi BAEL-Béton – Loi parabole-rectangle

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | **BAEL-beton** |  |  |
|  | *fcj* | *γb\*θ/0.85* |  |  |

#### Loi BAEL-Acier – Loi bilinéaire

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| LOI | *id\_loi* | **BAEL-acier** |  |  |
|  | *fy* | *γs* | Es | εlim |

### (+)CONTOUR

La commande CONTOUR permet de définir une section polygonale.

Un polygone est défini par une série de points représentant ses sommets.

Le sens de définition du contour est important :

* défini dans le sens direct (sens trigonométrique), un contour décrira une surface positive ;
* défini dans le sens indirect (sens horaire), un contour décrira une surface négative (=évidement si ce contour est décrit à l’intérieur d’une section positive).

La série doit contenir au moins 3 points et n’est pas limitée supérieurement.

L’utilisateur doit veiller à créer des contours conformes aux prescriptions du chapitre XXX.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| CONTOUR | *id\_con* | *id\_loi* | *(mot-clef)* |  |

* *id\_con* : identifiant de contour sous forme d’une chaine de caractères ;
* *id\_loi* : identifiant de loi correspondant au contour défini (la loi doit avoir été définie avant le contour) ;
* *(mot-clef)*: commande permettant de créer un contour prédéfini (CERCLE, RECTANGLE).

L’interprétation par le programme est la suivante :

* la présence d’une nouvelle commande en colonne A ou l’absence d’une valeur en colonne B provoque la fin de définition de la série ;
* l’absence de valeur en colonne C est interprétée comme *z\_i*=0 ;
* le programme referme automatiquement le contour créé ;
* un contour décrit dans le sens direct aura des caractéristiques mécaniques positives ;
* un contour décrit dans le sens indirect aura des caractéristiques mécaniques négatives.

#### Contour polygonal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| CONTOUR | *id\_con* | *id\_loi* |  |  |
|  | *y\_1* | *z\_1* |  |  |
|  | *y\_2* | *z\_2* |  |  |
|  | *…* | *…* |  |  |
|  | *y\_n* | *z\_n* |  |  |

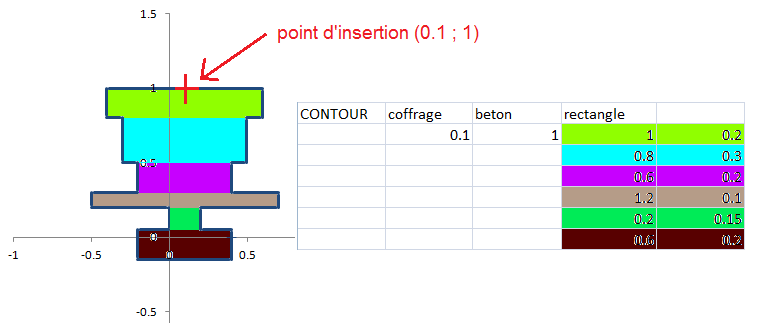
* *(y\_i ; z\_i)* : sommets du polygone.

#### Contour rectangulaire

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| CONTOUR | *id\_con* | *id\_loi* | **RECTANGLE** |  |
|  | *y\_c* | *z\_c* | *l\_y* | *l\_z* |

* *(y\_c ; z\_c)* : centre de gravité du rectangle ;
* *(l\_y*; l\_z) : respectivement longueur *y* et longueur *z* du rectangle. (si l\_y ET l\_z sont de même signe > sens trigo ; sinon > sens horaire)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| CONTOUR | *id\_con* | *id\_loi* | **RECTANGLE** |  |
|  | *y\_0* | *z\_0* | *b\_1* | *h\_1* |
|  |  |  | *b\_2* | *h\_2* |
|  |  |  | *…* | *…* |
|  |  |  | *b\_n* | *h\_n* |



Attention à ne pas laisser de contenu dans la cellule D(n+1).

#### Contour circulaire

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| CONTOUR | *id\_con* | *id\_loi* | **CERCLE** |  |
|  | *y\_c* | *z\_c* | *R\_c* |  |

* *(y\_c ; z\_c)* : centre du cercle ;
* *(R\_c*) : rayon du cercle (positif pour sens trigo, négatif pour sens horaire)

### (+)POINTS

La commande POINTS permet de définir un nuage de points surfaciques.

Chaque point est défini par ses coordonnées, sa surface et sa déformation initiale.

La série doit contenir au moins 1 point et n’est pas limitée supérieurement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| POINTS | *id\_pts* | *id\_loi* | *(mot clef)* |  |
|  | *y\_1* | *z\_1* | *S\_1* | *eps0\_1* |
|  | *y\_2* | *z\_2* | *S\_2* | *eps0\_2* |
|  | *…* | *…* | *…* | *…* |
|  | *y\_n* | *z\_n* | *S\_n* | *eps0\_n* |

* *id\_pts* : identifiant de points sous forme d’une chaine de caractères ;
* *id\_loi* : identifiant de loi correspondant au nuage de points défini (la loi doit être définie avant le contour) ;
* *(y\_i ; z\_i)* : coordonnées des points ;
* *S\_i*: surface du point ;
* *eps0\_i*: déformée initiale du point.

L’interprétation par le programme est la suivante :

* la présence d’une nouvelle commande en colonne A ou l’absence d’une valeur en colonne B provoque la fin de définition de la série ;
* l’absence de valeur en colonne C est interprétée comme *z\_i*=0 ;
* l’absence de valeur en colonne D est interprétée comme *S\_i*=0 ;
* l’absence de valeur en colonne E est interprétée comme *eps0\_i*=0 ;
* le programme tient compte de la déformation initiale du point pour le calcul de la contrainte associée comme suit : sigma(point)=sigma(eps+eps0)-sigma(eps0).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| POINTS | *id\_pts* | *id\_loi* | SEGMENT |  |
|  | *y\_1* | *z\_1* | *S\_pt* | *eps0\_pt* |
|  | *y\_2* | *z\_2* | *nb\_pts* |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| POINTS | *id\_pts* | *id\_loi* | CERCLE |  |
|  | *y\_c* | *z\_c* | *S\_pt* | *eps0\_pt* |
|  | *R\_c* | *θ\_0* | *nb\_pts* |  |

### (\*)PIVOTS

La commande PIVOTS permet de définir une série de pivots de calcul de la section.

Chaque pivot est défini par un point, un type de limite et une déformation associée.

La définition d’un point se fait soit par la saisie de ses coordonnées soit par une fonction associée à une géométrie (cont

our ou points).

Le type de limite définit la contrainte associée à la déformation saisie (*eps*) :

* déformation du/des point/s supérieure à *eps* ;
* déformation du/des point/s inférieure à *eps* ;
* déformation du/des point/s égale à *eps* (tracé intermédiaire).

La série doit contenir un nombre suffisant de pivots pour former un domaine de déformation valide ; ce nombre n’est pas limité supérieurement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| PIVOTS | *y\_i* | *z\_i* | *symb\_i* | *eps\_i* |
|  | *id\_geom* | *alpha* | *symb\_j* | *eps\_j* |
|  | *id\_geom* | *id\_ptCar* | *symb\_k* | *eps\_k* |
|  | *id\_mat* | *alpha ou G* | *symb\_l* | *eps\_l* |

* *(y\_i ; z\_i)* : coordonnées du pivot ;
* *id\_geom* : identifiant de géométrie (*id\_pts* ou *id\_con*) sous forme d'une chaine de caractères ;
* *alpha*: position relative du pivot par rapport aux points extrêmes de la géométrie ;
* *id\_ptCar*: identifiant de point caractéristique associé à *id\_geom* ; sont possibles :
  + "G" : centre de gravité de *id\_geom* ;
  + "Px" : x-ième point de définition de *id\_geom*.
* *symb\_i*: type de limite du pivot ; sont possibles :
  + ">" : déformation supérieure à *eps\_i* ;
  + "<" : déformation inférieure à *eps\_i* ;
  + "=" : déformation égale à *eps\_i*.
* *eps\_i*: déformation du pivot.

L’interprétation par le programme est la suivante :

* la présence d’une nouvelle commande en colonne A ou l’absence d’une valeur en colonne B provoque la fin de définition de la série ;
* l’absence de valeur en colonne C est interprétée comme *z\_i=0* ou *alpha*=0 ;
* l’absence de valeur en colonne D renvoie une erreur ;
* l’absence de valeur en colonne E est interprétée comme *eps\_i*=0 ;

saisir alpha ou 1-alpha est équivalent

Vérifier l’importance de saisir une limite systématiquement en traction

(exemple de DI où seule une limite en compression donne la même courbe que limite en compression + limite en traction très grande)

|  |  |
| --- | --- |
| rappel : ε(z)=a\*z+b  amin= « rotation » mini de l’état de def  amax= « rotation » maxi de l’état de def | 🡪 signifie que l’on peut saisir un seul pivot  et comme les matériaux sont définis avec des polygonales et, de fait, des valeurs seuil (non dépassables), le diagramme obtenu est fini (en fait, c’est comme si on regarde des matériaux qui plastifient sous grandes déformations). |

### (-)CENTRE

La commande CENTRE permet de définir le repère de calcul des torseurs du diagramme d’interaction.

Le repère est défini par un point et un angle d’inclinaison de la section.

La définition d’un point se fait soit par la saisie de ses coordonnées soit par une fonction associée à une géométrie (contour ou points).

Si la fonction CENTRE n’est pas appelée, le calcul des torseurs est effectué dans le repère général de définition de la section.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| CENTRE | *y\_c* | *z\_c* | *theta* |  |
| CENTRE | *id\_geom* | *id\_ptCar* | *theta* |  |
| CENTRE | *id\_mat* | *G par défaut* | *theta* |  |

voir pour section homogénéisée…(combinaison linéaire entre éléments/matériaux)

* *(y\_c ; z\_c)* : coordonnées du centre ;
* *id\_geom* : identifiant de géométrie (*id\_pts* ou *id\_con*) sous forme d'une chaine de caractères ;
* *id\_ptCar*: identifiant de point caractéristique associé à *id\_geom* ; sont possibles :
  + "G" : centre de gravité de *id\_geom* ;
  + "Px" : x-ième point de définition de *id\_geom*.
* *theta*: angle d’inclinaison de l’axe OY de calcul mesuré dans le sens direct.

L’interprétation par le programme est la suivante :

* la définition du centre se fait sur une seule ligne ;
* l’absence de valeur en colonne B est interprétée comme *y\_c=0* ;
* l’absence de valeur en colonne C est interprétée comme *z\_c=0* si *y\_c* est défini ou renvoie une erreur si *id\_geom* est défini ;
* l’absence de valeur en colonne D est interprétée comme *theta=0* ;
* la fonction CENTRE ne doit être appelée qu’une fois (toute nouvelle déclaration écrase la précédente).

### (-)NBANGLES

La commande NBANGLES permet de définir le nombre d’inclinaisons de section à calculer pour le diagramme d’interaction 3D.

Si la fonction NBANGLES n’est pas appelée, le programme calcule le diagramme d’interaction de la section autour de l’axe de calcul OY seulement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| NBANGLES | *nb* |  |  |  |

* *nb* : nombre d’inclinaisons de section.

L’interprétation par le programme est la suivante :

* la définition du nombre d’angles d’inclinaisons se fait sur une seule ligne ;
* l’absence de valeur en colonne B est interprétée comme *nb=1* ;
* les angles balayés par le programme sont *(i-1)\*π/nb* avec *i*∈[*1* ; *nb*] ;
* la fonction NBANGLES ne doit être appelée qu’une fois (toute nouvelle déclaration écrase la précédente).

### (-)NBDIV

La commande NBDIV permet de piloter la densité du tracé du diagramme d’interaction.

Si la fonction NBDIV n’est pas appelée, une valeur par défaut est retenue (voir ci-dessous).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| NBDIV | *valeur* |  |  |  |

* *div\_piv* : nombre minimal de divisions du plus grand tracé d’un pivot (nombre entier positif) ;
* *seuil\_con*: seuil de précision minimale pour la discrétisation des contours (nombre décimal positif : 0.001=0.1%).

L’interprétation par le programme est la suivante :

* la définition des paramètres de calcul se fait sur une seule ligne ;
* l’absence de valeur en colonne B est interprétée comme *div\_piv=10* ;
* l’absence de valeur en colonne C est interprétée comme *prec\_con=0.005* ;
* la fonction PARAM ne doit être appelée qu’une fois (toute nouvelle déclaration écrase la précédente).

### (-)TOLC

La commande TOLC permet de piloter la tolérance de discrétisation des contours.

Si la fonction TOLC n’est pas appelée, une valeur par défaut est retenue (voir ci-dessous).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| TOLC | *valeur* |  |  |  |

* *valeur* : seuil de précision minimale pour la discrétisation des contours (nombre décimal positif : 0.001=0.1%).

L’interprétation par le programme est la suivante :

* l’absence de valeur en colonne B est interprétée comme *prec\_con=0.005* ;
* la fonction PARAM ne doit être appelée qu’une fois (toute nouvelle déclaration écrase la précédente).

Les paramètres de calcul permettent de contrôler le nombre de points du diagramme d’interaction et le raffinement de la discrétisation des contours.

### (\*)EOF

La commande EOF clôt l’ensemble des instructions que le programme interprète. Les lignes suivant la commande EOF ne sont pas interprétées par le programme et sont accessibles à l’utilisateur.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** |
| EOF |  |  |  |  |

## Rapport d’exécution

## Exemples

# Maintenance du programme et programmation

## Structure des données

### Classe Problème

Contient :

* table de lois
* table de contours
* table de points
* table de pivots
* paramètres de calcul :
  + centre
  + nbre d’angles
  + nbre de div
  + tolérance Contours
* table de résultats de calculs

+ fonctions associées :

* ajout de loi, contour, point
* ajout pivot (type 1=(y,z) ; 2=(IDgeom) ; 3=(IDgeom+IDpt)
* ajout centre (charge = (y,z) ; calc=(IDgeom+IDpt)
* vérif IDgeom/IDloi
* trouveNumCont/NumPoints/YZ/numLoi
* recentrage (translation/rotation)

+ fonctions de calcul :

* préparation du calcul
* calcul (calcul torseur, insertion torseur

+fonction de sorties :

* impression résultats

### Classe Loi

Contient :

* ID loi
* liste eps
* liste sig

+ fonctions associées :

* chargeLoi
* ok (fonction test)
* sigma(epsilon) retourne par interpolation

### Classe Contour

Contient :

* ID geom
* ID loi
* liste y
* liste z

+ fonctions associées :

* chargeContour
* lecture Y/Z
* ok (fonction test)
* calcul de carac
* calcul de b(z)
* translation/rotation
* fonctions de copie

### Classe Points

Contient :

* ID geom
* ID loi
* liste y
* liste z
* liste S
* liste eps0

+ fonctions associées :

* chargePoints
* lecture Y/Z/S/eps0
* ok (fonction test)
* calcul de carac
* translation/rotation
* fonctions de copie

### Classe Pivot

contient :

* type (-1 ; 0 ; 1) pour eps(> ;= ;<)valeur [signe de (valeur-eps)]
* alpha (pivot de type C)
* eps
* liste y
* liste z

+ fonctions associées :

* charge pivot
* translation/rotation
* fonctions de copie

## Structure des résultats

Inclus dans la classe\_problème : table de résultats triés par angle de calcul

### classe calcul\_par\_angle

contient :

* informations sur le calcul (angle, facteurs d’échelle pour le calcul de distance entre torseurs)
* liste résultats de calculs par pivot

### classe calcul\_par\_Pivot

contient :

* informations sur le pivot (type, z, …)
* liste aP, Nx, My, Mz

## Structure du programme

### Lecture des données

Mots clefs reconnus :

commandes obligatoires

* DI
* EOF
* LOI
* CONTOUR
* POINTS (nuage de points)
* PIVOTS

paramètres optionnels

* CENTRE (centre de calcul), saisie brute ou via ID geom/pt caractéristique
* NBANGLES (nbre d’angles de calcul pour le DI 3D), saisie brute
* NBDIV (nbre de points mini par pivot), saisie brute
* TOLC (tolérance sur la discrétisation des contours), saisie brute

## Calcul

recentrage du problème sur le centre de calcul (translation/rotation)

Boucle sur les angles

Rotations des contours/points/pivots

Discrétisation des contours

préparation des pivots

boucle sur les pivots

calcul (Boucle sur les nuages de points)

## Sortie des résultats

# Développements futurs

|  |  |
| --- | --- |
| **Titre** | **Fait/à faire** |
| Section sous forme de série de rectangles empilés | Fait |
| Identifiant « géométriques » basés sur les groupes matériaux | à faire |
| Mise en forme des graphes (paramétrage, commande MEFPIVOTS, MEFSECTION, MEFLOI etc…) | à faire |
|  |  |
|  |  |