

Annexe 13

Visualisation graphique sur matlab

Description des routines et du programme principal du Fichier

xpdessin.m

On se propose de décrire les procédures matlab du fichier xpdessin.m afin d'éclairer les lignes de programme qu'on peut y trouver et de présenter la façon dont on s'est pris pour représenter les anneaux.

On répartit la discussion suivant de grosses rubriques.

Il est à noter qu'on a réussi de structurer le programme en procédures en se servant de la possibilité en matlab de définir des fonctions dans des fichiers "script". Ainsi, par exemple, pour faire la procédure fonctionnelle : inverse(x) qui prend la valeur de x, l'inverse et remet le résultat dans x, on construit la fonction matlab :

function [x] = inverse(x)

Le x passe alors bien par variable et est muet.

On se sent de l'instruction : global (analogie à l'instruction fortran: common) pour ne pas surcharger les arguments des procédures.

Ce programme a été écrit en matlab et se sent de la sortie graphique de matlab, cependant il pourrait être écrit dans un autre langage et avec une autre sortie graphique. En effet, on garde toujours l'avantage des idées, des algorithmes, des procédures et de la programmation. L'utilisation d'un langage compilable permettrait un gain de vitesse.

1 Représentation d'un anneau:

11 Sous-routines d'opération sur les matrices:

* xaffete:

Elle permet d'affecter les valeurs des matrices de coordonnées, normales et binormales à d'autres matrices.

* xextract et xextractc:

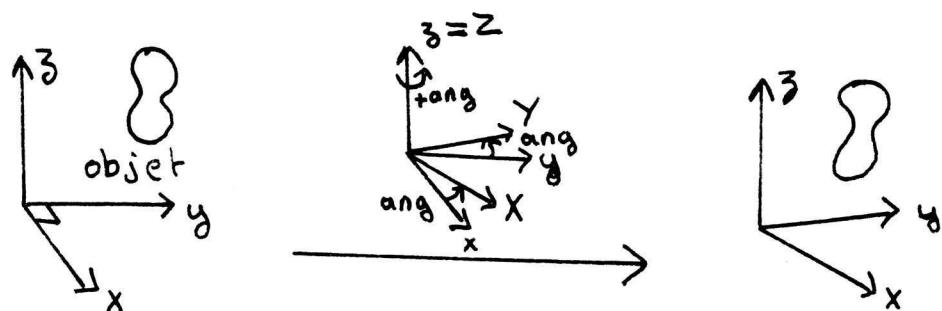
$x = \text{xextract}(xx, nu)$ extrait les informations sur l'anneau numéro nu d'une matrice de type xx et les stocke dans une matrice de type x (voir annexe 10). La sous-routine xextractc fait cette opération automatiquement pour les matrices coordonnées xx, yy, zz, les matrices normales nx, ny, nz et les matrices binormales bx, by et bz.

* xextractcv:

Elle fait la même opération que la routine xextractc mais ici entre des matrices de type xx₀ et x₀ (voir annexe 10).

12 Sous-routines de rotation:

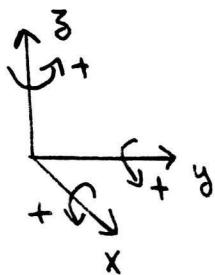
* les trois rotations élémentaires: rotax, rotay et rota_z.



`rotaz` fait tourner les axes d'un angle `tang` autour de `z`, ce qui fait tourner l'objet d'un angle `-ang` autour de `z`. Elle fait tourner un point `M`:

$$M(x, y, z) \xrightarrow{\text{rotaz}} M'(x := x, y := Y, z := Z).$$

Les routines `rotax` et `rotay` font l'opération analogue autour de `x` et de `y` sachant que les axes sont orientés comme suit (pour les angles de rotation des axes):



* Rotation par 3 rotations élémentaires:

+ xaxo et xaxon:

- xaxo fait successivement une rotation des axes d'un angle `ang1` autour de `y`, d'un angle `ang2` autour de `z` et d'un angle `ang3` autour de `y`. Elle effectue ces rotations aux trois matrices de type `x0: x0, y0, z0`.

- xaxon applique la procédure précédente successivement aux 3 matrices coordonnées `x0, y0, z0` ; aux trois matrices normales : `nx0, ny0, nz0` et aux trois matrices binormales : `bx0, by0, bz0`.

+ xaxonc: Elle fait la même opération que xaxon mais sur les matrices de type $xx: xx, \dots, nxx, \dots, bxx, \dots$

+ xaxoniV: Elle fait la même opération que xaxon mais sur les matrices de type: $xx_0, \dots, nxx_0, \dots, bxx_0, \dots$.

* Obtention des trois vues:

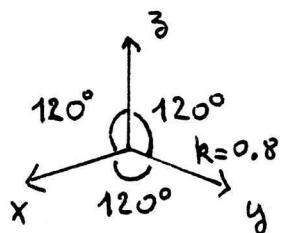
+ vue de face yz: il suffit de projeter sur le plan (y, z) sans faire de rotation $(\text{ang}_1, \text{ang}_2, \text{ang}_3) = (0, 0, 0)$

+ vue de côté xz: il suffit de projeter sur le plan (y, z) après avoir fait les rotations $(\text{ang}_1, \text{ang}_2, \text{ang}_3) = (0, \pi/2, 0)$

+ vue de haut xy: il suffit de projeter sur le plan (y, z) après avoir fait les rotations $(\text{ang}_1, \text{ang}_2, \text{ang}_3) = (-\pi/2, 0, 0)$

* Vue axonométrique:

On cherche les valeurs à donner à $(\text{ang}_1, \text{ang}_2, \text{ang}_3)$ pour avoir la représentation axonométrique suivante:



Formule de rotation autour de y_0 de θ :

$$\vec{x} \begin{vmatrix} \cos \alpha_1 \\ \sin \alpha_1 \end{vmatrix} \xrightarrow[\text{de } y_0]{\text{rotation}} \vec{x}' \begin{vmatrix} \cos(\alpha_1 - \theta) \\ \sin(\alpha_1 - \theta) \end{vmatrix}$$

d'où :

$$\vec{x}' = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \vec{x}$$

$$\begin{array}{ccc|c} x & y & z & \\ \hline 1 & 0 & 0 & \text{rotat/}z \\ 0 & 1 & 0 & \xrightarrow{\text{de } -45^\circ} \\ 0 & 0 & 1 & \end{array} \quad \begin{array}{ccc|c} x' & y' & z' & \\ \hline c & c & 0 & \text{où } c = \frac{\sqrt{2}}{2} \\ -c & c & 0 & \\ 0 & 0 & 1 & \end{array}$$

$$\begin{array}{ccc|c} x' & y' & z' & \\ \hline c & c & 0 & \text{rotat/}y \\ -c & c & 0 & \xrightarrow{\text{de } \theta} \\ 0 & 0 & 1 & \end{array} \quad \begin{array}{ccc|c} x'' & y'' & z'' & \\ \hline c & c & 0 & \\ c & -c & 0 & \\ -c \sin \theta & c \cos \theta & 1 & \end{array}$$

On a $|y''|$ projeté sur $y_3| = |z''|$ projeté sur $y_3|$

d'où $c\sqrt{1 + \sin^2 \theta} = \cos \theta$ c'est à dire

$$\cos \theta = \sqrt{\frac{2}{3}} \quad \theta = 35,2^\circ \quad k = \sin(90^\circ - 35,2^\circ) = 0,8.$$

On a donc : $(\text{ang1}, \text{ang2}, \text{ang3}) = (\underline{0}, -\pi/4, \underline{\frac{35,2}{180}\pi})$

Dans le programme on a choisi les trois différents types de points de vue suivant :

$$(\text{ang1}, \text{ang2}, \text{ang3}) = \left(-\pi/2, -\pi/4, \frac{35,2}{180}\pi\right)$$

$$\text{ou } = \left(-\pi/2, 0, \frac{35,2}{180}\pi\right)$$

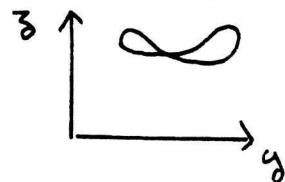
$$\text{ou } = \left(-\pi/2, -\frac{\pi}{6}, \frac{15}{180}\pi\right)$$

13 Sous-routines de dessin:

131 Représentation de la fibre sans épaisseur:

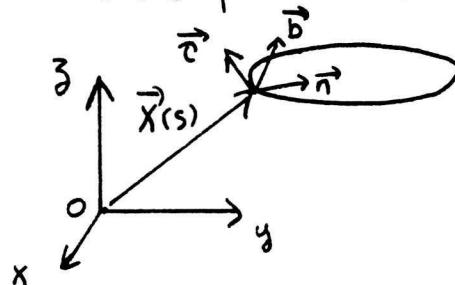
Pour représenter $\vec{X}(s)$, on se contente de projeter sur le plan (y, z) et de représenter à l'écran la courbe obtenue. C'est la procédure xgraphel qui fait cette opération.

Le problème de cette représentation est que l'on perd complètement les informations sur x de $\vec{X}(s)$ et donc sur la vue on n'a pas la notion de la profondeur.



132 Représentation de la fibre avec épaisseur:

* L'idée de profondeur:



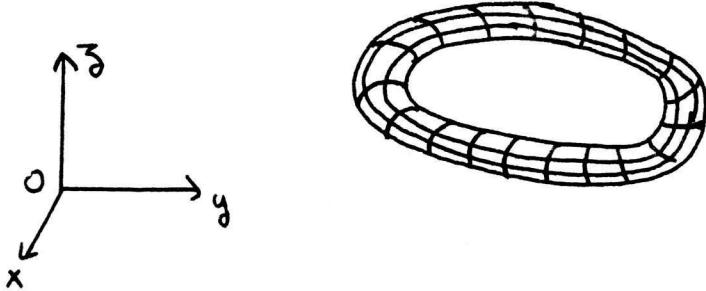
On considère la surface d'équation paramétrée:

$$\overrightarrow{OM} = \vec{X}(s) + (\cos\theta \vec{n}(s) + \sin\theta \vec{b}(s)) * \text{ep.}$$

On peut tracer deux familles de courbes sur cette surface:

- les courbes à s fixé: ce sont des cercles autour de la fibre.

- les courbes à θ fixé: ce sont des courbes "parallèles" à la fibre.



(es courbes seront tracées sous la forme de lignes polygonales à l'aide des deux pas:

$$ds = 2\pi/N \quad \text{et} \quad d\theta = 2\pi/8$$

Comme précédemment, on re présente tout en projection sur (y, z) . L'avantage de cette représentation est que l'information donnée par la coordonnée x de $X(s)$ n'est pas perdue entièrement. On a une idée de la profondeur suivant l'inclinaison et le rapprochement des cercles autour de la fibre.

Par exemple, la rotation d'un cylindre autour d'un axe dans cette représentation donne le résultat suivant:



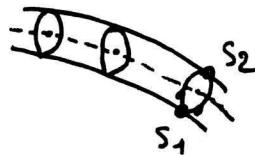
On a donc l'idée de la profondeur.

Remarquons cependant que la distance de deux points de discréétisation de la ligne centrale contigus n'est pas constante, c'est la différence d'abscisse curvilinear ds qui est constante.

* Les lignes cachées:

Afin de bien représenter l'anneau, il ne faut tracer que les portions de courbe vues.

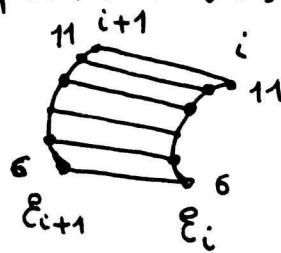
Parmis la famille des lignes "parallèles", les deux lignes parallèles extrêmes sont celles qui sont les plus éloignées l'une de l'autre. On se sent alors du fait que la projection d'un cercle est une ellipse.



Les deux points qui délimitent les lignes parallèles extrêmes et les extrémités de la portion d'ellipse à dessiner correspondent aux deux sommets de l'ellipse sur le grand axe de celle-ci. On les obtient en cherchant la distance maximale entre deux points de l'ellipse.

On détermine alors la portion d'ellipse à dessiner en comparant la position sur x des deux points contigus à S_1 et situés respectivement sur les deux portions de cercle. On représente la portion où se trouve le point contigu à S_1 qui a la plus grande valeur de coordonnée x.

Les courbes parallèles qui ne sont pas cachées sont celles qui s'appuient sur cette portion visible de l'ellipse.

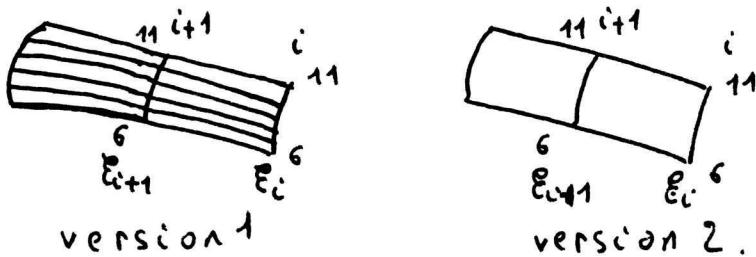


E_i : ellipse au noeud i

E_{i+1} : ellipse au noeud $i+1$.

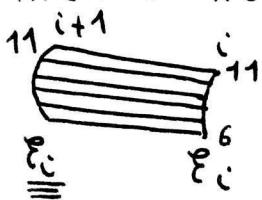
On trace alors des droites entre les points des portions d'ellipse \mathcal{E}_i et \mathcal{E}_{i+1} de même numéro.

On peut soit dessiner toutes les droites parallèles visibles : c'est la version 1, soit ne représenter que les droites parallèles extrêmes : c'est la version 2.



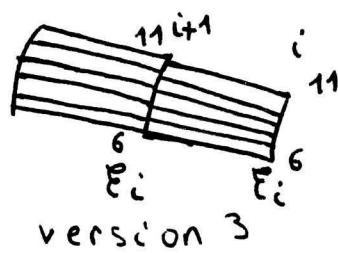
La représentation précédente se sent de la continuité des vecteurs \vec{n} et \vec{b} le long de la fibre. Cependant, le fait que dans $\vec{x}_s = \sigma \vec{c}$, σ doigt être positif entraîne une discontinuité de \vec{n} et \vec{b} (rotation de 180°). On a constaté aussi une rotation de 180° lorsque la courbure s'inverse.

Afin de palier à ce problème, on ne trace les courbes parallèles issues de \mathcal{E}_i qu'à l'aide de \mathcal{E}_i que l'on reporte au point $i+1$ afin de relier sans discontinuité les noeuds de même numéro :

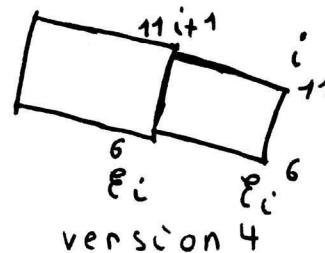


Cependant, en $i+1$ on représente bien la portion d'ellipse \mathcal{E}_{i+1} .

On a aussi deux versions suivant que l'on représente ou pas toutes les lignes parallèles :



version 3



version 4

On appelle : version 5 , la version où on ne représente que la fibre centrale en projection sur y_3 :

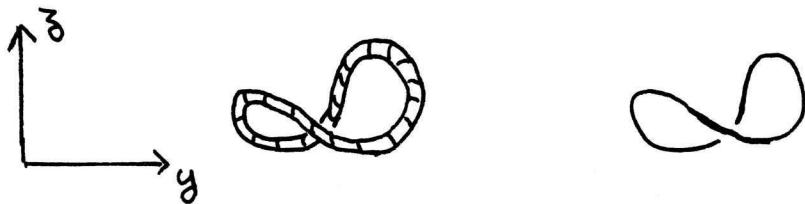


* les intersections :

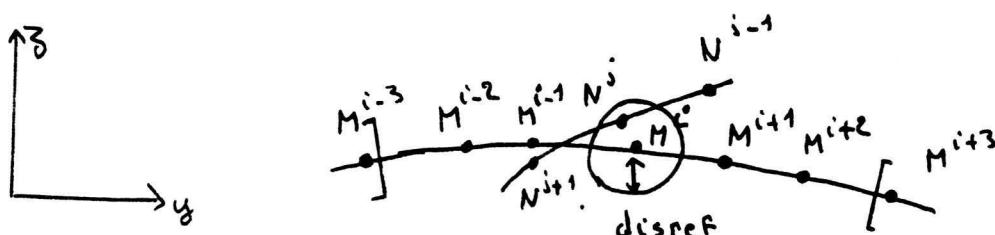
On dit qu'il ya une intersection lorsque la courbe paramétrée en projection sur (y_3) de la fibre $X(s)$ a un point double .



L'un des brins de la courbe polygonale représentant la fibre est derrière l'autre au niveau de l'intersection. On ne veut pas représenter ce brin caché lors de la représentation avec ou sans épaisseur, afin de ne pas faire apparaître des lignes cachées qui se superposeraient à des lignes vues .



Il faut donc rechercher les noeuds où il y a intersection et où le segment issu du noeud est caché. Pour cela, on itère sur les noeuds. Plaçons nous au noeud M^i lors de cette itération.



$$disref = \frac{M^i M^{i+1}}{2}$$

On itère alors en j sur les noeuds d'indice différent de $i-2, i-1, i, i+1, i+2$:

$$j \neq i-2, i-1, i, i+1, i+2.$$

On calcule: $dis = M^i N^j$. Il y a intersection entre les noeuds M^i et N^j si et seulement si $dis < disref$. C'est ce critère qu'on utilise.

On cherche alors celui des deux brins qui est derrière l'autre en déterminant celui des noeuds M^i et N^j non projetés qui a la plus petite coordonnée x .

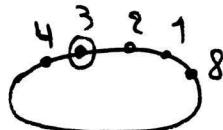
On enregistre son numéro I . Il ne faudra pas représenter ni l'arc d'ellipse en ce point ni les courbes parallèles issues de ce point.

* les procédures:

- xpermut:

C'est l'équivalent de la procédure permut du programme Fortran.

Si $i=3$:



elle donne le vecteur $v = [1, 2, 3, \dots, 8]$
 $v = [3, 4, 5, \dots, 8, 1, 2]$

Elle est très utile car elle ramène le noeud i comme si son numéro était 1.

- xinter:

Elle effectue les deux itérations présentées dans le paragraphe intersection afin de trouver les points d'intersection cachés.

Son résultat est le vecteur st des numéros de noeud d'intersection caché: $st = [4, 12]$ par exemple, ainsi que le nombre lg de ceux-ci. Si il n'y a pas de point d'intersection $lg = 0$.

Elle utilise xpermut.

- xappartient:

$sol = xappartient(i, st, lg)$.
sol a la valeur 0 si $lg = 0$ ou si i est et est égal à 1 sinon.

Elle permet donc, après avoir effectué xinter, de déterminer si le noeud i est un noeud d'intersection caché.

- x cercle:

Elle permet de calculer les coordonnées d'un point d'angle: la du cercle d'équation:

$$\overrightarrow{OM} = X(s_k) + ep(\cos \alpha \vec{a}(s_{kk}) + \sin \alpha \vec{b}(s_{kk}))$$

où s_k et s_{kk} sont donnés.

s_{kk} sera soit égale à s_k : graphe des versions 1 et 2

soit égale à s_k ou s_{k-1} : graphe des versions 3 et 4.

- x graph:

Elle effectue le graphisme épaisse de l'anneau comme il a été présenté dans les paragraphes: "idées de profondeur" et : "lignes cachées", tout en ne représentant pas les brins cachés de l'anneau.

Elle utilise toutes les procédures précédentes et permet d'obtenir les 4 versions du graphisme: 1, 2, 3 et 4 suivant la valeur correspondante de la variable: version.

2 Représentation de l'évolution:

Lors de l'évolution dans le temps, on a fait nbrconf clichés qu'il faut représenter. Chaque un des clichés est représenté à l'aide d'une des versions de dessin d'anneau précédemment présentée.

On a construit deux types de représentation des résultats :

- une représentation dite par "multivues"

- une représentation dite par "vues"

Dans les deux cas, quelque soit le nombre d'anneaux, le nombre des configurations et la taille des anneaux, le programme détermine automatiquement les coordonnées de la fenêtre de visualisation à l'écran et l'agencement des différents clichés selon le premier type de représentation ou le second et ceci de telle sorte que tout soit visible à l'écran.

2) Représentation par "multivues":

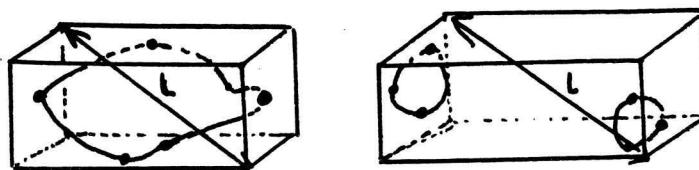
* Détermination de la longueur caractéristique:

On le fait sur la configuration initiale : $x_{x_0}, y_{y_0}, z_{z_0}$ des anneaux. On cherche la plus petite valeur x_{\min} de x_{x_0} ainsi que la plus grande x_{\max} et on fait de même pour z_{z_0} et y_{y_0} .

Ces valeurs définissent le plus petit parallélépipède parallèle aux axes de coordonnées qui contienne la configuration initiale.

La longueur caractéristique est alors définie comme la longueur de la diagonale du parallélépipède .

Exemples:



C'est la procédure `xlong1` qui effectue cette opération.

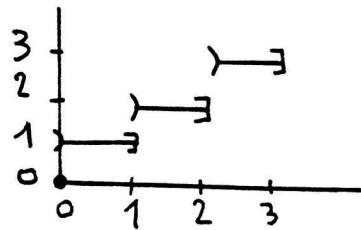
* Partition et dimensions de la fenêtre de visualisation :

On définit une fenêtre de visualisation qui a les dimensions que l'on a représentées sur la figure suivante et qui est découpée en carrés de coté l_c
 $l_c = 2L$ où L est la longueur caractéristique.

		CONF 13			
		CONF 9	CONF 10	CONF 11	CONF 12
		CONF 5	CONF 6	CONF 7	CONF 8
configuration numéro		CONF 1	CONF 2	CONF 3	CONF 4
$\frac{l_c}{2}$	F_0				
$-\frac{l_c}{2}$	colonne 1	colonne 2	colonne 3	colonne 4	$\frac{l_c}{2} + 4 \times l_c$

$$A = \text{ceil}(\text{nbrconf}/4) * l_c - (c/2)$$

où ceil est la fonction suivante :



On numérote les carrés comme sur le dessin.

* La procédure xmultipuest :

Pour représenter une configuration :

- on cherche son barycentre G : c'est le but de la procédure xcentre. Elle se sert de la routine xmoy qui calcule $(\sum x_i / n)$
- on déplace la fibre de telle sorte que G soit à l'origine $o(0,0)$ du repère : c'est la procédure xcentrage.
- on dessine alors (par l'une des 5 versions de dessin d'anneau) tous les anneaux de la configuration en projection sur(yz) en les décalant suivant $y \rightarrow z$ de telle sorte que le barycentre soit au centre du carré de numéro correspondant à la configuration.

c'est la procédure xmultivuest qui fait ces différentes opérations pour toutes les configurations, après avoir:

- calculé la longueur caractéristique
- défini la fenêtre de visualisation
- choisi $\text{ep} = 0,1 \cdot l$
- fait tourner toutes les configurations selon les angles: $\text{ang}_1, \text{ang}_2, \text{ang}_3$.

La configuration initiale est comme toujours traitée à part. On obtient un résultat qui a la forme suivante:

	00		
	00	00	0
	0	0	0
	0	0	0
	0	0	0

* Les procédures xmultivuesto, xmultivuestoe et xmultivuestom:

Ce sont les mêmes procédures que xmultivuest, si ce n'est que, dans ces trois procédures, les 3 angles ($\text{ang}_1, \text{ang}_2, \text{ang}_3$) sont automatiquement définis par des valeurs données (voir fin du paragraphe 12 de cet annexe).

22 Représentation par "vues"

* longueur caractéristique:

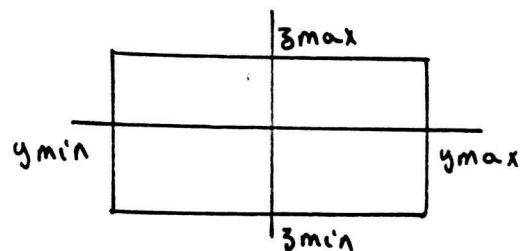
Elle est déterminée comme précédemment sur la configuration initiale.

On définit alors : $\text{ep} = 0,1 * \frac{L}{4}$.

* Dimensions de la fenêtre de visualisation:

On cherche la plus petite valeur y_{min} de y_0 et y_3 ainsi que la plus grande y_{max} et on fait de même pour z_0 et z_3 .

Ces 4 valeurs : y_{min} , y_{max} , z_{min} et z_{max} définissent les dimensions de la fenêtre de visualisation.

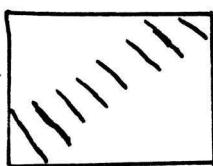


* La procédure xvuest :

Cette procédure effectue successivement les opérations suivantes :

- rotation de toutes les configurations selon les 3 angles : $ang1$, $ang2$, $ang3$
- détermination de la fenêtre de visualisation
- calcul de la longueur caractéristique l
- calcul de $ep = 0.1 * \frac{l}{4}$
- elle dessine alors (par une des 5 versions de dessin d'anneau) en projection sur (y_3) tous les anneaux des configurations de numéro contenu dans le vecteur num.

On obtient un résultat qui a la forme suivante :



* les procédures xvuesto, xvuestoe et xvuestom:

Ce sont les mêmes procédures que xvuest si ce n'est que dans ces 3 procédures les 3 angles (ang1, ang2, ang3) sont automatiquement définis par des valeurs données (voir fin du paragraphe 12 de cet annexe).

3 Le Programme principal:

Il effectue différentes "multivues" en utilisant les procédures : xmultivuest, xmultivuesto, xmultivuestoe et xmultivuestom.

Il effectue également différentes "vues" en utilisant les procédures : xuest, xuesto, xuestoe et xuestom.

Son algorithme est décrit dans le paragraphe 4 de l'annexe 3.