

République du Sénégal  
Un Peuple - Un But - Une Foi

# UNIVERSITE CHEIKH ANTA DIOP DE DAKAR



ECOLE SUPERIEURE POLYTECHNIQUE  
Centre de Thiès  
Département de Génie Civil

GC.0075

## PROJET DE FIN D'ETUDES

En vue de l'obtention du Diplôme d'Ingénieur de Conception

Titre : **MODELISATION PAR ELEMENTS FINIS APPLIQUEE AU  
CALCUL DES PONTS COURBES**

Auteur :

Mr Valdiodio Guirande DIOUF

Encadreur :

Dr. Moustapha NDIAYE

PFE 2002

## DEDICATIONS

A ALLAH Soubhana Wa Talla pour ses Innombrables Grâces.

Au Prophète Muhamad (P.S.L) son honorable envoyé.

A ma Très Chère Mère .

A mon Cher Père.

A mes chers Frères et Sœurs.

A tous mes Parents (sans distinction).

A mes Grands Parents décédés in Mémoriam (Que la terre leur soit légère).

A tous les Frères Musulmans et Sœurs Musulmanes.

A tous ceux qui épris de Paix et de Justice Luttent pour un Monde Meilleur.

## REMERCIEMENTS

J'adresse mes profonds remerciements :

- Au Dr, Professeur Moustapha NDIAYE pour son encadrement.
- A tous mes Educateurs m'ayant appris depuis toujours ce que je sais et ce que je saurais faire Aujourd'hui et Demain.
- Aux Contribuables Sénégalais pour m'avoir fait bénéficier de supports financiers à travers Bourses et Aides.
- A tous mes Camarades de Promotion pour la cohabitation enrichissante partagée tout au long de notre formation.

## AVANT PROPOS

Les structures de type poutre à parois minces sont d'usage très fréquent en Génie Civil car elles offrent une grande rigidité et une grande résistance pour un poids relativement faible.

Cependant, elles subissent facilement des instabilités et sont affectées d'un gauchissement non négligeable (comportement en torsion uniforme et tordue) qui modifie considérablement leur comportement.

Différentes théories ont été établies afin d'étudier le comportement particulier de ces structures, qui diffèrent selon qu'il s'agit d'une section à parois minces ouverte ou fermée.

L'objectif du travail qui suit est de présenter une formulation en éléments finis en même temps une formulation directe pour l'étude du comportement mécanique des poutres gauches chargées dont la section transversale à parois minces peut être ouverte ou fermée.

Notre domaine d'étude se limitera pour une première part à ces types de structures chargées sur deux appuis, et nous poserons avant résolution les équations fondamentales du problème différentiel rencontrées dans leur étude en prenant en compte les effets conjugués de la torsion de St Venant et de celle de Vlasov.

Notre domaine d'application reste ouvert à la modélisation des ponts courbes non nécessairement plans et qui peuvent être circulaires, hélicoïdal ou droit.

## SOMMAIRE

Dans ce projet nous avons travaillé dans le cadre de l'élasticité tridimensionnelle pour poser et résoudre le problème différentiel fondamental régissant la statique des poutres chargées.

Nous avons en première considération pris en compte des paramètres géométriques pour traiter les trois cas de poutre circulaire, hélicoïdale ou droite. Etant donné que nos hypothèses de calcul concernent en plus des poutres à parois minces à section ouverte ou fermée, où les effets dus au gauchissement sont prépondérants, nous n'avons pas manqué de prendre en compte aussi bien la torsion uniforme de Saint Venant que celle de Vlasov.

Considérant alors les trois grands groupes de relations qui régissent la statique des corps solides déformables, nous avons établi et résolu analytiquement avec des programmes de calcul sur le Logiciel Maple le système fondamental rencontré, en même temps générer d'autres paramètres tels que la matrice de rigidité ou les forces équivalentes aux nœuds.

Nous avons par ailleurs ouverts quelques brèches en ce qui concerne deux approches de résolution numérique par des méthodes variationnelles.

## LISTE DES SYMBOLES ET ABREVIATIONS

$s$	: variable d'expression de l'abscisse curviligne
$x, y, z$	: coordonnées cartésiennes locales
$X, Y, Z$	: coordonnées cartésiennes globales
$G(Y_G, z_G)$	: coordonnées locales du centre géométrique
$C(y_C, z_C)$	: coordonnées locales du centre de courbure torsion
$\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{zz}$	: déformations en coordonnées cartésiennes ( $\varepsilon_{xx} = \varepsilon_{xx}$ )
$\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$	: composantes cartésiennes des contraintes ( $\sigma_{xx} = \sigma_{xx}$ )
$\theta_x, \theta_y, \theta_z$	: rotations autour des axes locaux
$\theta_X, \theta_Y, \theta_Z$	: rotations autour des axes globaux
$Q_x, Q_y, Q_z$	: effort normal et efforts tranchants
$M_x, M_y, M_z$	: moments de torsion et de flexion
$M_w$	: Bi-moment de torsion
$I_{xx}, I_{yy}$	: torsions de St Venant et de Vlassov
$F = \{F_{bx}, F_{by}, F_{bz}\}$	: forces de chargements concentrés ou répartis
$M = \{M_{bx}, M_{by}, M_{bz}\}$	: moments de chargements concentrés ou répartis
$[B]$	: matrice reliant les déformations aux variables nodales
$[C]$	: matrice reliant les contraintes aux déformations
$[L]$	: opérateur matriciel différentiel reliant les déformations aux déplacements
$[TG]$	: matrice de transformation repère local - repère global
$[GL]$	: matrice de transformation repère global - repère local
$[R]$	: matrice de transformation géométrique totale
$N_0 = \{N_{01}, N_{02}\}$	: vecteur coefficients d'interpolation de Lagrange
$K_g$	: vecteur courbure torsion global
$K_l$	: vecteur courbure torsion local
$\psi$	: fonction de contrainte
$S_w$	: moment sectoriel de la fonction de gauchissement
$\varpi$	: pulsation ou fonction de gauchissement en torsion

## LISTE DES FIGURES

Figure 2-1	: Section, Ligne moyenne et Repère Local Principal de Fresnel	p.5
Figure 2-2	: Repères Local $\{e_1, e_2, e_3\} = \{x, y, z\}$ et Global $\{X, Y, Z\} = \{i, j, k\}$ liés à la géométrie de la poutre	p.6
Figure 2-3	: Rotations axiales pour une poutre droite spatiale	p.11
Figure 3-1	: Console soumise à un moment de Torsion concentré	p.16
Figure 3-2	: Section ouverte mince d'un élément de poutre	p.16
Figure 3-3	: Modes de résistance des sections à la torsion	p.17
Figure 3-4	: Photo d'illustration d'un pont courbe circulaire	p.19
Figure 3-5	: Élément typique poutre courbe à 2 nœuds avec degrés de libertés	p.19
Figure 3-6	: Tablier de pont caisson courbe avec Repère local et centre de courbure torsion $C (Y_c, Z_c)$	p.21
Figure 4-1	: Rotation d'une section de poutre autour de l'axe neutre X en G	p.45
Figure 4-2	: Géométrie générique d'une poutre spatiale hélicoïdale	p.47
Figure 5-1	: Poutre courbe générale approximée par trois tronçons de poutres droites	p.79

## TABLE DES MATIERES

1 INTRODUCTION .....	1
2 GEOMETRIE GENERIQUE DES POUTRES COURBES .....	4
2-1 Définition de la poutre .....	4
2-2 Grandeurs locales et globales des paramètres géométriques .....	6
2-2-1 Paramétrisation géométrique .....	6
2-2-2 Matrices de transformation de coordonnées .....	9
2-2-3 Courbures et Torsions .....	13
3 RELATIONS ET EQUATIONS DIFFERENTIELLES FONDAMENTALES .....	15
3-1 Etude de la torsion .....	15
3-1-1 La Torsion uniforme (St Venant) .....	15
3-1-2 La Torsion non uniforme (Vlassov) .....	15
3-2 Relations Cinématiques .....	20
3-2-1 Grandeurs .....	20
3-2-2 Expressions vectorielles et matricielles .....	22
3-3 Conditions d'Equilibre .....	26
3-3-1 Grandeurs .....	26
3-3-2 Expressions vectorielles et matricielles .....	26
3-4 Lois Constitutives .....	27
3-4-1 Grandeurs .....	27
3-4-2 Expressions .....	29
3-5 Equations aux dérivées partielles fondamentales du Problème .....	37
4 RESOLUTION DU PROBLEME FONDAMENTAL .....	38
4-1 Résolution Analytique .....	38
4-1-1 Programmation .....	38
4-1-2 Cas Circulaire .....	43
4-1-3 Cas Hélicoïdal .....	46
4-1-4 Cas Droit .....	49
4-2 Equations différentielles fondamentales du problème .....	53
4-2-1 Cas Circulaire .....	53
4-2-1 Cas Hélicoïdal .....	55
4-2-3 Cas Droit .....	56
4-3 Proposition des Résultats .....	57
4-3-1 Cas circulaire .....	57
4-3-2 Cas Hélicoïdal .....	64
4-3-3 Cas droit .....	65
4-4 Formulation Directe .....	71
4-5 Formulation Variationnelle pour la Résolution Numérique .....	72
4-6 Formulation mixte à deux champs .....	75
5 RECOMMANDATIONS .....	78
6 CONCLUSIONS .....	81
7 ANNEXES (Listing de Programmes et Résultats) .....	85

## 1 INTRODUCTION

Dans le domaine du Génie Civil un grand nombre de constructions (bâtiments, ponts...) sont assimilables à des assemblages de poutres. La poutre est un élément de base en Résistance des matériaux. Au sein des constructions, elle est susceptible de reprendre des efforts et des moments dans toutes les directions.

A ce titre l'étude de son comportement en tant que système mécanique s'avère d'une importance capitale. C'est ainsi que toute une théorie a été développée dans la résolution des problèmes linéaires de poutres et expose deux hypothèses fondamentales ainsi que leurs conséquences dans les limites d'un comportement élastique :

- Le principe de St Venant qui permet de définir le système des forces extérieures relatif à une section pour en déterminer les contraintes.
- Le principe de Navier Bernoulli généralisé qui permet de définir la déformation de la poutre au voisinage d'une section donnée.

Les deux problèmes majeurs qui se posent dans l'étude des poutres élastiques et homogènes ont recours à :

- la recherche du système de forces extérieures relatif à une section quelconque
- au calcul des contraintes s'exerçant sur une section connaissant le système de forces extérieures relatif à cette section.

L'ingénierie civile dans un souci de bien définir la poutre a élaboré plusieurs modèles théoriques et numériques de son comportement. Parmi ceux-ci figure la Méthode des Eléments Finis qui est de loin la plus utilisée et la meilleure en terme de précision et de convergence des algorithmes de résolution. Pour un problème donné, la première étape consiste à établir ses équations fondamentales qui se présentent sous forme d'équations différentielles ou aux dérivées partielles et qui satisfont aux conditions d'équilibre, de compatibilité géométrique et limites.

Si le modèle physique choisi pour modéliser le problème réel nous fournit les équations et les conditions aux limites, les mathématiques, par contre, sont souvent impuissantes pour en donner une solution analytique. Les mathématiques prouvent parfois l'existence et l'unicité de la solution et il est prudent de s'en assurer avant de se lancer dans la recherche d'une solution numérique. La méthode des éléments finis fournit une solution approchée de la solution exacte sous forme d'un champ défini par morceaux dans la partition du domaine global. Chercher une solution par éléments finis consiste donc à déterminer quel champ local on attribue à chaque sous domaine pour que le champ global obtenu par juxtaposition de ces champs locaux soit solution du problème.

Les hypothèses générales à prendre en compte pour l'étude des problèmes pratiques de poutres concernent les comportements en flexion avec prise en compte des effets de cisaillement transversal pour les modèles dits de Timoshenko ou de Mindlin / Reissner et qui généralisent ceux de Navier / Bernoulli. Mais aussi les hypothèses de Saint Venant sur la torsion libre, ou de Vlassov pour la torsion gênée avec prise en compte du gauchissement des sections à parois minces et ouvertes (ou fermées). Pour le cas des poutres, l'établissement des équations fondamentales se fait sur considération des relations d'équilibre, cinématiques et des lois consécutives.

Dans le domaine de la construction des ouvrages d'art, les tabliers de ponts reposent sur un système de poutres si le tablier n'est pas lui même entièrement une poutre de type caisson par exemple. Nous savons par ailleurs que ces poutres supportent et transmettent des efforts et des moments de grande envergure d'où la nécessité de bien poser et de valider les modèles qui seront établis pour son étude.

Pour une poutre droite symétrique dont les conditions d'appui et de chargement sont connues, en plus de la considération unique de la torsion uniforme de St Venant, la résolution du problème différentiel peut s'avérer moins fastidieux. Par contre, si l'on introduit les paramètres géométriques liés à la courbure et que l'on prenne en compte en plus de la torsion libre, la torsion non uniforme de Vlassov qu'on ne saurait négliger dans le comportement des poutres à parois minces (par exemple), le

problème différentiel peut s'avérer encore plus difficile dans sa résolution. C'est donc ce modèle de poutre courbe avec les cas circulaire et hélicoïdal applicables au calcul des ponts caissons courbes que nous allons développer dans le cadre de ce projet.

Dans un premier temps, nous ferons considération unique de la statique de la poutre avec des tenseurs de forces et de moments concentrés ou repartis.

Ensuite, par une paramétrisation nous introduirons les considérations géométriques dans les équations du problème en même temps nous prendrons compte des effets conjugués de la torsion libre et de la torsion gênée.

Eventuellement, nous prendrons considération des effets dynamiques d'accélération de la masse.

Enfin, après avoir établi les équations différentielles fondamentales des modèles circulaire, hélicoïdal et droit dans le cas tridimensionnel le plus général, nous passerons à leur résolution analytique et à défaut numérique avec le Logiciel de calcul formel Maple®.

## 2 GEOMETRIE GENERIQUE DES POUTRES COURBES

Nous allons lier notre poutre à deux systèmes de repères :

- un repère global  $(O, X, Y, Z)$
- un repère local  $(G, x, y, z)$

Ce choix s'explique car la forme la plus simple des équations de la théorie des poutres s'écrit dans le repère local qui sera choisi comme un repère central principal d'inertie pour des raisons simplificatrices que nous évoquerons par la suite. Toute étude sur une poutre commence par la définition de ce repère.

L'établissement de la matrice de transformation géométrique repère local -repère global nous permettra d'exprimer selon notre convenance dans l'un ou l'autre des repères les différentes équations du problème.

### 2-1 Définition de la poutre

On définit par :

- $C$  une courbe orientée = ligne moyenne de la poutre.
- $G$  son point courant = centroïde de la section
- $S$  abscisse curviligne mesurée à partir d'un point fixe  $O$  sur  $C$
- $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$  le trièdre de Fresnet orthonormé associé à  $C$  en  $G$  : c'est la base du repère local associé à la poutre. Donc  $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\} = \{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$

Le plan  $\{\mathbf{n}, \mathbf{b}\}$  est appelé *plan normal en  $G$* . Dans ce plan on définit la section droite  $S(y, z)$  de la poutre, de contour extérieur fermé  $\Omega$  et de centre de gravité sur la ligne moyenne. La section droite peut aussi avoir des contours intérieurs  $\Omega_i$ ,

On appelle poutre, le volume engendré par les sections droites lorsque  $G$  décrit la ligne moyenne. On suppose de plus que :

- Les sections droites sont constantes ou <<lentement variables>> lorsque  $G$  varie.
- Si on appelle  $d$  une dimension transversale de la section droite, on a en tout point de la poutre  $d \ll R$ .

Ces deux hypothèses sont essentielles : elles permettent d'assimiler un tronçon de poutre courbe de longueur  $dl$  à un tronçon de poutre droite. Cette approximation est à la base de tout ce qui suit.

Enfin, on se place dans le cadre de *l'élasticité linéaire isotrope en petites déformations et petits déplacements* : Le tenseur des déformations que nous utiliserons est donc  $\mathcal{E}$

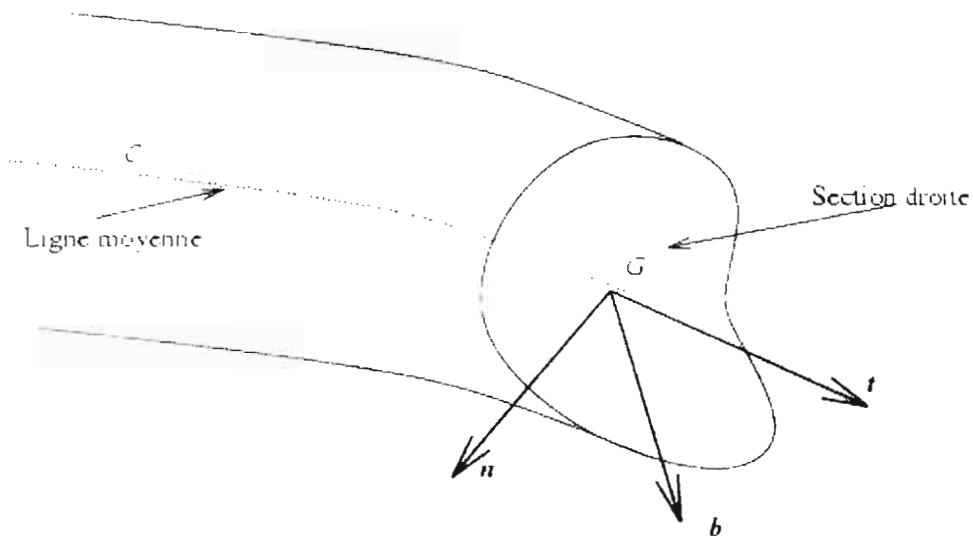


Fig 2-1: Section, Ligne moyenne et Repère Local Principal de Fresnet

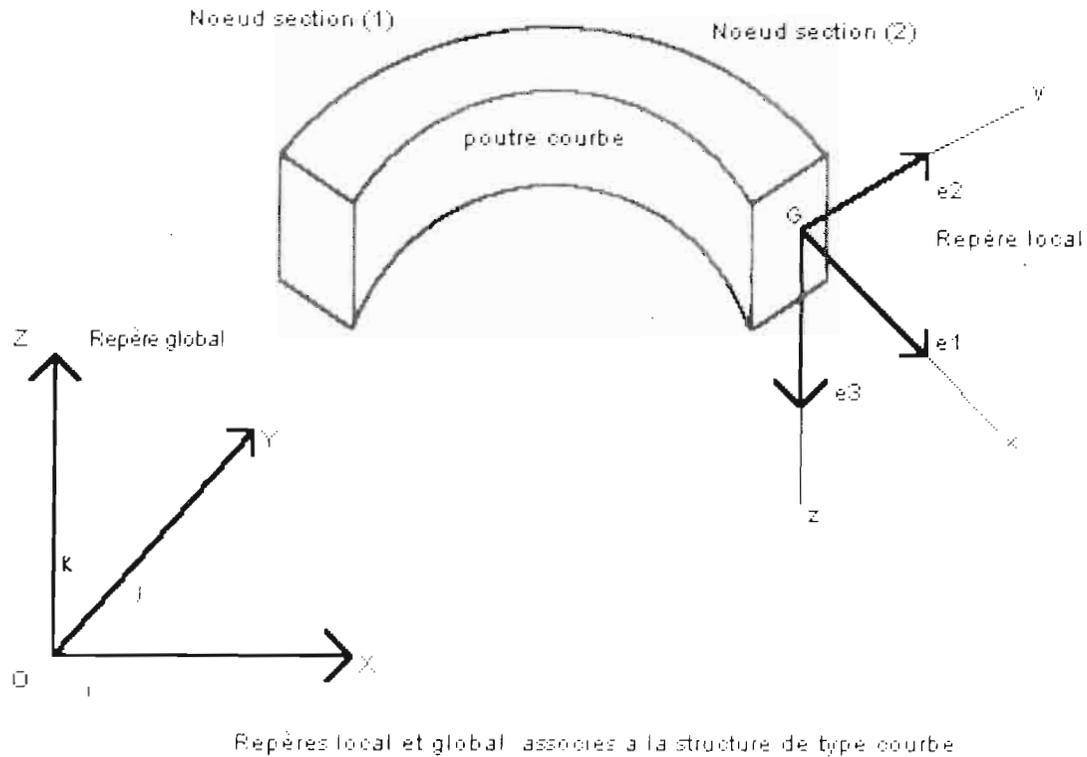


Fig 2-2 : Repères Local  $\{e_1, e_2, e_3\} = \{x, y, z\}$  et Global  $\{X, Y, Z\} = \{i, j, k\}$  liés à la géométrie de la poutre

## 2-2 Grandeurs locales et globales des paramètres géométriques

### 2-2-1 Paramétrisation géométrique

Vecteur position :

$$OM = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.1.a)$$

Paramétrisation :  $X = f(t)$        $Y = g(t)$        $Z = h(t)$

Ce sont les composantes du vecteur position dans le repère global où  $t$  est le paramètre de variation de l'abscisse curviligne  $s$

Donc le vecteur position s'exprime encore :

$$\mathbf{OM} = \begin{bmatrix} f(t) \\ g(t) \\ h(t) \end{bmatrix} \quad (2.1.b)$$

Construction de la base du repère local de Fresnet :

Première présentation :

$$\begin{aligned} \mathbf{T} &= \frac{\partial}{\partial t} \mathbf{OM}(t) \\ \mathbf{N} &= \frac{\partial}{\partial t} \mathbf{T}(t) \\ \mathbf{B} &= \mathbf{T} \wedge \mathbf{N} \end{aligned} \quad (2.2)$$

$\mathbf{T}$  : vecteur tangent non encore normé

$\mathbf{N}$  : vecteur normal non encore normé

$\mathbf{B}$  : vecteur binormal

L'opérateur ( $\wedge$ ) représente le produit vectoriel direct.

Par ailleurs nous pouvons donner les expressions des vecteurs de la base de Fresnet en fonction des paramètres  $f(t)$ ,  $g(t)$  et  $h(t)$  soit :

$$\begin{aligned} \mathbf{T} &= f'(t) \mathbf{i} + g'(t) \mathbf{j} + h'(t) \mathbf{k} \\ \mathbf{N} &= f''(t) \mathbf{i} + g''(t) \mathbf{j} + h''(t) \mathbf{k} \end{aligned} \quad (2.3)$$

Où  $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$  sont les vecteurs de la base du repère global. Nous obtenons

$$\mathbf{T} = \begin{cases} T_x = f' \\ T_y = g' \\ T_z = h' \end{cases} \quad \mathbf{N} = \begin{cases} N_x = f'' \\ N_y = g'' \\ N_z = h'' \end{cases} \quad \mathbf{B} = \begin{cases} B_x = T_y N_z - T_z N_y \\ B_y = T_z N_x - T_x N_z \\ B_z = T_x N_y - T_y N_x \end{cases} \quad (2.4)$$

On définit les normes quadratiques des vecteurs :

$$\alpha = \text{norm}(\mathbf{T}, 2) \quad \beta = \text{norm}(\mathbf{N}, 2) \quad \gamma = \text{norm}(\mathbf{B}, 2) \quad (2.5)$$

Pour construire les vecteurs de la base du Repère local :

$$\mathbf{e1} = \frac{T}{\alpha} \quad \mathbf{e2} = \frac{N}{\beta} \quad \mathbf{e3} = \frac{B}{\gamma} \quad (2.6.a)$$

$$\{\mathbf{e1}, \mathbf{e2}, \mathbf{e3}\} = \left\{ \frac{T}{\alpha}, \frac{N}{\beta}, \frac{B}{\gamma} \right\} \quad (2.6.b)$$

**Deuxième présentation :**

On définit tout d'abord le vecteur position :

$$\mathbf{r} = f(t) \mathbf{i} + g(t) \mathbf{j} + h(t) \mathbf{k} \quad (2.7)$$

Pour obtenir le vecteur tangent on différentie  $\mathbf{r}$  et on obtient

$$\frac{d\mathbf{r}}{dt} = f'(t) \mathbf{i} + g'(t) \mathbf{j} + h'(t) \mathbf{k} \quad (2.8)$$

Le vecteur unitaire tangent qui est en même temps le vecteur  $\mathbf{e1}$  de la base du repère local de Fresnet est donné par :

$$\mathbf{e1} = \frac{\frac{d\mathbf{r}}{dt}}{\left| \frac{d\mathbf{r}}{dt} \right|} \quad (2.9)$$

Cette équation nous renseigne que  $\mathbf{e1}$  fait un angle constant  $\theta$  avec l'axe Z de façon à ce que :

$$\mathbf{e1} \cdot \mathbf{k} = \cos(\theta) = \frac{f'(t)}{\left| \frac{d\mathbf{r}}{dt} \right|} \quad (2.10)$$

Nous pouvons maintenant procéder à la détermination du vecteur courbure-torsion dans le repère global soit :

$$\mathbf{K}_g = \frac{de_1}{ds} = \frac{de_1}{dt} \frac{dr}{dt} \quad (2.11)$$

Comme  $\mathbf{K}$  est parallèle au plan X-Y et a la même direction que le vecteur unitaire normal  $\mathbf{e}_2$ , ce dernier peut être exprimé par :

$$\mathbf{e}_2 = \frac{\mathbf{K}}{|\mathbf{K}|} \quad (2.12)$$

Finalement nous déterminons le vecteur unitaire de la base locale  $\mathbf{e}_3$  qui coïncide avec le vecteur unitaire binormal. A partir de  $\mathbf{e}_1$  et  $\mathbf{e}_2$  on trouve par le produit vectoriel :

$$\mathbf{e}_3 = \mathbf{e}_1 \wedge \mathbf{e}_2 \quad (2.13)$$

Nous obtenons les coefficients de la matrice de Transformation local –global en exprimant les vecteurs  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  en termes de  $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ .

### 2-2-2 Matrices de transformation de coordonnées

*Matrice de Transformation géométrique Repère local - Repère global :*

On exprime  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  en termes de  $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$  par la matrice associée

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \frac{T_1}{\alpha} & \frac{T_2}{\alpha} & \frac{T_3}{\alpha} \\ \frac{N_1}{\beta} & \frac{N_2}{\beta} & \frac{N_3}{\beta} \\ \frac{B_1}{\gamma} & \frac{B_2}{\gamma} & \frac{B_3}{\gamma} \end{bmatrix} \begin{bmatrix} i \\ j \\ k \end{bmatrix} \quad (2.14)$$

D'où la matrice :

$$\mathbf{LG} = \begin{bmatrix} \frac{T_y}{\alpha} & \frac{T_x}{\alpha} & \frac{T_z}{\alpha} \\ \frac{N_y}{\beta} & \frac{N_x}{\beta} & \frac{N_z}{\beta} \\ \frac{B_y}{\gamma} & \frac{B_x}{\gamma} & \frac{B_z}{\gamma} \end{bmatrix} \quad (2.15)$$

On définit par la transposée de cette matrice celle de Transformation Repère global – Repère local soit :

$$\mathbf{GL} = \begin{bmatrix} \frac{T_y}{\alpha} & \frac{N_y}{\beta} & \frac{B_y}{\gamma} \\ \frac{T_x}{\alpha} & \frac{N_x}{\beta} & \frac{B_x}{\gamma} \\ \frac{T_z}{\alpha} & \frac{N_z}{\beta} & \frac{B_z}{\gamma} \end{bmatrix} \quad (2.16)$$

- Pour les cas Circulaire et Hélicoïdal nous nous placerons dans l'hypothèse d'une rotation  $\alpha$  des sections autour de l'axe géométrique dans le plan  $\{y,z\}$ . On considère alors en plus une matrice de rotation :

$$\mathbf{R}_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2.17)$$

Signalons cependant pour éviter des conflits de notations que cette nouvelle valeur de l'angle alpha est différente de celle qui donne la norme de  $\mathbf{T}$ .  
Ce qui nous donne pour la nouvelle matrice de Transformation repère Local –

Repère Global :

$$T = R_{\alpha} \times LG \tag{2.18.a}$$

Soit :

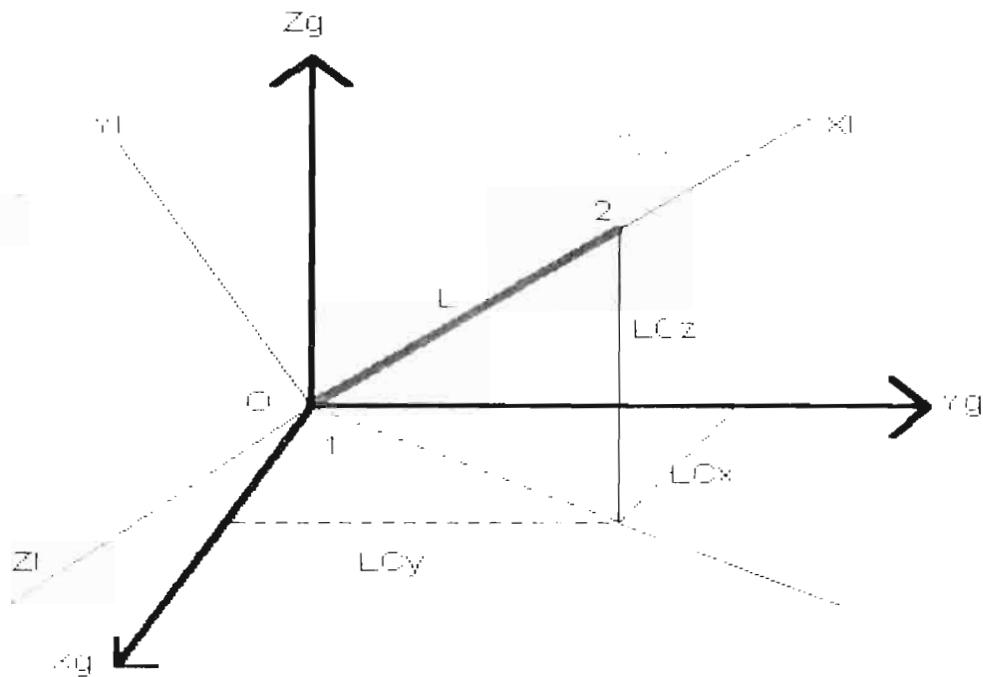
$$T = R_{\alpha} \times LG = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} \frac{T_1}{\alpha} & \frac{T_1}{\alpha} & \frac{T_2}{\alpha} \\ \frac{N_1}{\beta} & \frac{N_1}{\beta} & \frac{N_2}{\beta} \\ \frac{B_1}{\gamma} & \frac{B_1}{\gamma} & \frac{B_2}{\gamma} \end{bmatrix} \tag{2.18.b}$$

- Pour le cas droit, nous ferons une formulation de la matrice de transformation géométrique avec les cosinus directeurs des angles

Avec la paramétrisation :

$$X = at + x_0 \quad Y = bt + y_0 \quad Z = ct + z_0 \tag{2.19}$$

on pose:  $L = \sqrt{a^2 + b^2 + c^2}$  (2.20)



Rotations axiales pour une poutre droite spatiale

Fig 2-3 Rotations axiales pour une poutre droite spatiale

$$\alpha = \text{angle}(\text{OZ}_g, \text{OY}_l) \quad \beta = \text{angle}(\text{OZ}_l, \text{OX}_g) \quad \gamma = \text{angle}(1-2, \text{X}_g\text{OY}_g)$$

$$\text{Cosinus directeurs :} \quad C_x = \frac{a}{L} \quad C_y = \frac{b}{L} \quad C_z = \frac{c}{L} \quad (2.21)$$

$$\text{On définit :} \quad \cos(\beta) = \frac{C_x}{\sqrt{C_x^2 + C_z^2}} \quad \sin(\beta) = \frac{C_z}{\sqrt{C_x^2 + C_z^2}} \quad (2.22)$$

$$\text{Soit la matrice de rotation :} \quad R_\beta = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.23)$$

$$\cos(\gamma) = \sqrt{C_x^2 + C_z^2}$$

$$\sin(\gamma) = C_y \quad (2.24)$$

$$\text{Soit la matrice de rotation :} \quad R_\gamma = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

Enfin si on considère la rotation axiale d'angle  $\alpha$  avec la matrice de rotation  $R_\alpha$ , nous obtenons pour la matrice de transformation totale repère Local – repère Global :

$$\mathbf{R} = \mathbf{R}_\alpha \mathbf{R}_\gamma \mathbf{R}_\beta \quad (2.26.a)$$

Soit

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.26.b)$$

*Abscisse curviligne :*

En pratique nous allons ramener et résoudre le problème différentiel fondamental régissant le modèle de la poutre avec le paramètre de l'abscisse curviligne S.

Son expression différentielle est donnée par :

$$ds = (f'(t)^2 + g'(t)^2 + h'(t)^2)^{1/2} dt = |OM(t)| dt \tag{2.27}$$

Ce qui donne sous forme intégrale :

$$S = \int_{t_0}^t |OM(t)| dt \tag{2.28}$$

Si on prend dorénavant pour paramètre l'arc S qui définit complètement la forme de la courbe C, on peut écrire les équations de la courbure et de la torsion sous la forme de fonctions continues et dérivables en S.

### 2-2-3 Courbures et Torsions

On définit le vecteur courbure torsion K dans les deux repères local et global :

Le vecteur K exprimé dans le repère global aura pour expression :

$$K_g = \frac{\frac{\partial T(t)}{\partial t}}{\frac{ds}{dt}} \tag{2.29.a}$$

Soit

$$K_g = \frac{1}{\alpha} [f'', g'', h''] \begin{bmatrix} i \\ j \\ k \end{bmatrix} \tag{2.29.b}$$

Donc

$$\mathbf{Kg} = [K_x, K_y, K_z]^T \quad (2.29.c)$$

avec

$$K_x = \frac{f'''}{\alpha} \quad K_y = \frac{g'''}{\alpha} \quad K_z = \frac{h'''}{\alpha} \quad (2.30)$$

Son expression dans le repère local se fait par l'intermédiaire de la matrice de transformation GL soit :

$$\mathbf{K}_l = \frac{1}{\alpha} [f''', g''', h'''] [GL] \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (2.31.a)$$

$$\mathbf{K}_l = [K_x, K_y, K_z]^T \quad (2.31.b)$$

En pratique nous ferons le calcul avec le vecteur courbure torsion de la base locale sous forme matricielle ce qui donne :

$$\mathbf{K}_l = \begin{bmatrix} 0 & K_z & -K_y \\ -K_z & 0 & K_x \\ K_x & -K_x & 0 \end{bmatrix} \quad (2.31.c)$$

En faisant l'analogie entre le vecteur courbure  $\mathbf{K}$  et la vitesse angulaire de rotation suivant l'abscisse curviligne  $s$ , on peut remarquer que les axes du repère local tournent autour de l'axe local  $y$ , ce qui nous permet d'évaluer la composante :

$$K_z = - \frac{d\omega_3}{dt} \cdot e_2 \quad (2.32)$$

## 3 RELATIONS ET EQUATIONS DIFFERENTIELLES FONDAMENTALES

### 3-1 Etude de la torsion

#### Définition :

La Torsion est une sollicitation résultant de l'application à une barre (arbre de transmission, poutre prismatique, poutre coque etc.) d'un ou de plusieurs moments de torsion constituant l'action , et évidemment d'un ou de plusieurs moments de torsion de réaction maintenant l'équilibre.

Ces moments tendent à faire tourner les sections droites autour d'un axe longitudinal de l'élément poutre , dit " axe de torsion ", qui rencontre chaque plan de section en un point appelé " centre de torsion ".

Les composantes du couple équivalent au moment de torsion sont parallèles au plan des sections droites, et la torsion engendre principalement des contraintes de cisaillement (dites aussi tangentiels) dues au glissement par rotation des sections droites les unes sur les autres.

On distingue principalement deux catégories de torsion

- La Torsion gauche uniforme de St Venant
- La Torsion gauche entravée non uniforme de Vlassov

#### 3-1-1 La Torsion uniforme (St Venant)

C'est l'état de torsion pure d'une barre tel que qu'en cours de déformation les sections droites initialement planes, restent planes en tournant autour de l'axe de torsion.

#### 3-1-2 La Torsion non uniforme (Vlassov)

Les efforts correspondant à ce mode de résistance ne peuvent se développer que si la section transversale, qui aimerait gauchir, voit son gauchissement empêché par des conditions d'appui adéquates( exemple, encastrement complet ).

On dit qu'une section plane gauchit lorsqu'elle ne reste pas plane après déformation sous l'action d'un moment de torsion extérieur.

En cours de déformation, en tournant autour de l'axe de torsion les sections initialement planes subissent une distorsion encore appelée déplanation.

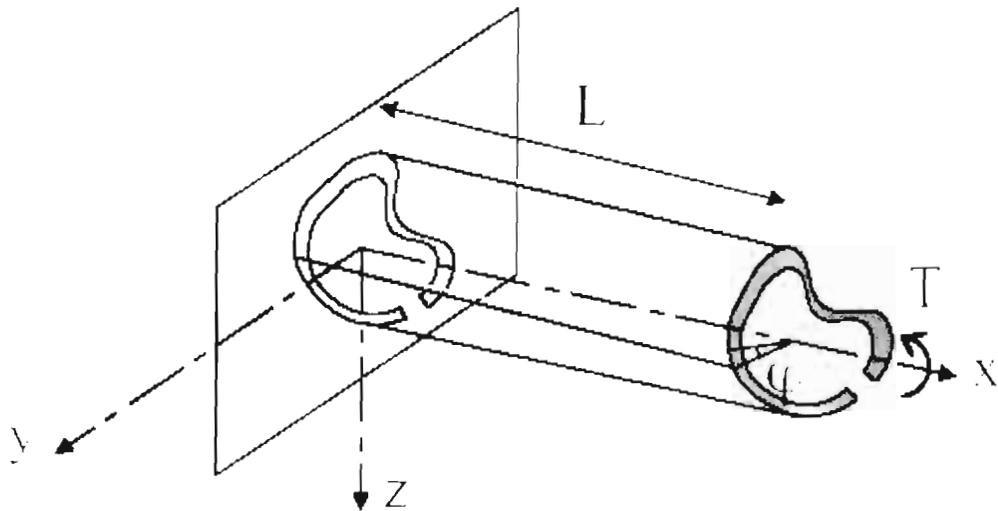
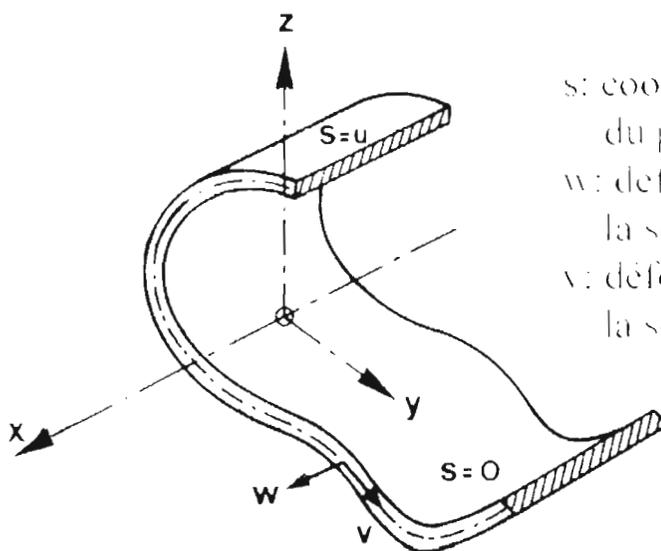


Fig 3-1 : Console soumise à un moment de Torsion concentré

### Torsion non uniforme: définitions et hypothèses



$s$ : coordonnée curviligne le long du profil

$w$ : déformation hors du plan de la section.

$v$ : déformation dans le plan de la section selon  $s$ .

Fig 3-2 : Section ouverte mince d'un élément de poutre



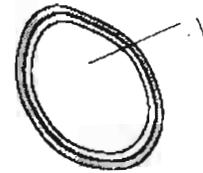
Rigidité:

sections massives  
pleines



sections fermées à parois minces

s: abscisse de la  
ligne moyenne  
t: épaisseur



formule de St-Venant:

$$J \approx \frac{A^4}{40 I_p}$$

formule de Bredt:

$$J \approx \frac{4 A^2}{\oint \frac{ds}{t(s)}}$$

Les différents domaines d'application du calcul de la torsion peuvent être résumés à :

## Torsion:

Domaines d'application:

ponts courbes et ponts biais

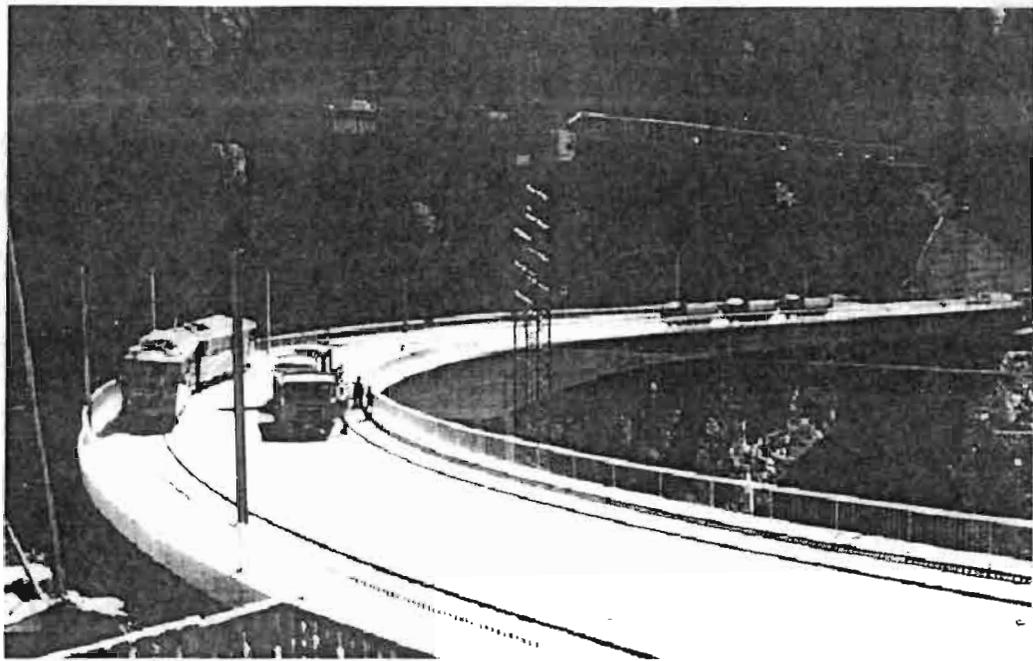
ponts avec charges excentrées

mâts

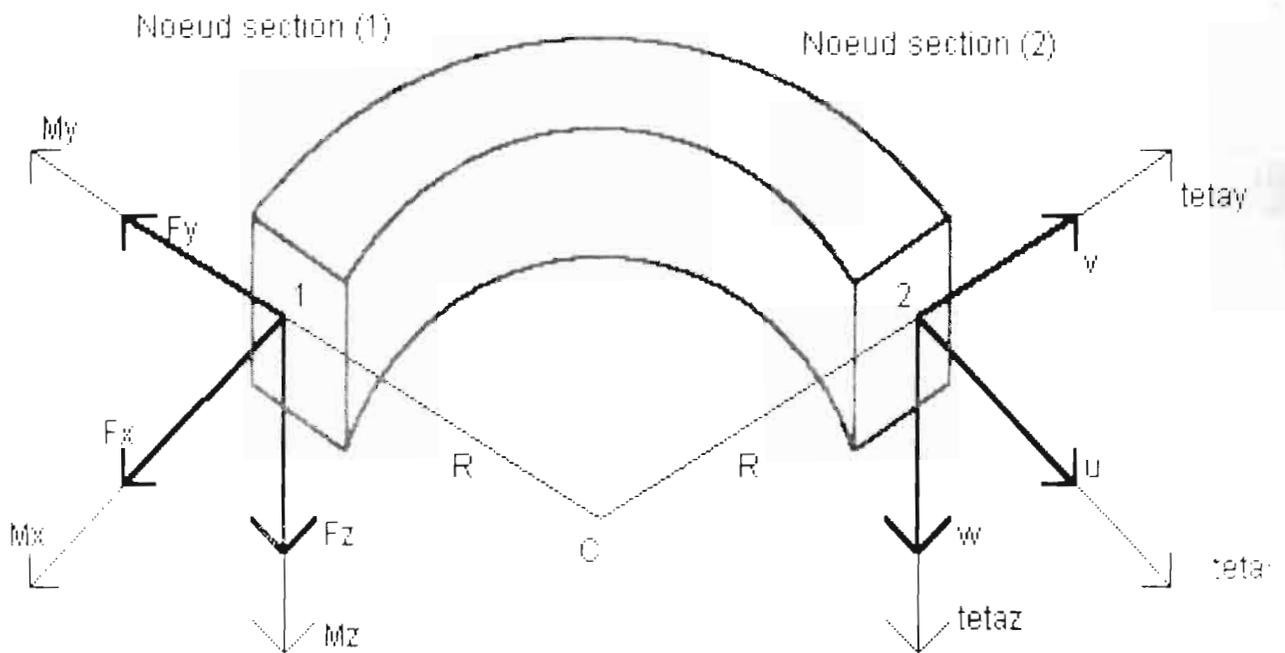
sheds, tôles pliées, constructions plissées

wagons, fuselage et ailes d'avion

Illustration: pont courbe



**Fig 3-4:** Photo d'illustration d'un pont courbe circulaire.



Element typique poutre courbe à 2 nœuds avec degrés de liberté

**Fig 3-5 :** Élément typique poutre courbe à 2 nœuds avec degrés de libertés

## 3-2 Relations Cinématiques

### 3-2-1 Grandeurs

Les relations cinématiques sont celles qui donnent les expressions des trois translations :

$$\{u_g(s), v_g(s), w_g(s)\}$$

d'un point courant de la fibre moyenne ainsi les composantes de ses trois rotations libres

$$\{\theta_x(s), \theta_y(s), \theta_z(s)\}$$

L'indice G signifiant que l'on est bien au centre géométrique de la section sur la ligne moyenne. L'indice (s) signifie que l'on recherche leurs expressions en fonction de l'abscisse curviligne.

Nous ramenons l'étude sur la ligne moyenne car en théorie des poutres, les charges sont considérées comme étant appliquées à la ligne moyenne.

On définit :

- le vecteur translations au point courant G  $\mathbf{U}_g = \{u_g(s), v_g(s), w_g(s)\}^T$
- le vecteur rotations libres  $\theta = \{\theta_x(s), \theta_y(s), \theta_z(s)\}^T$
- les vecteurs de transition  $\mathbf{r} = \{0, y, z\}^T$  et  $\mathbf{e1} = \{1, 0, 0\}^T$
- la fonction de gauchissement sur la section {y,z} soit  $\omega = \omega(y, z)$
- la rotation libre due au bi-moment (gauchissement des sections)  $\psi(s)$

En expressions sous formes matricielles nous obtenons dans le repère local :

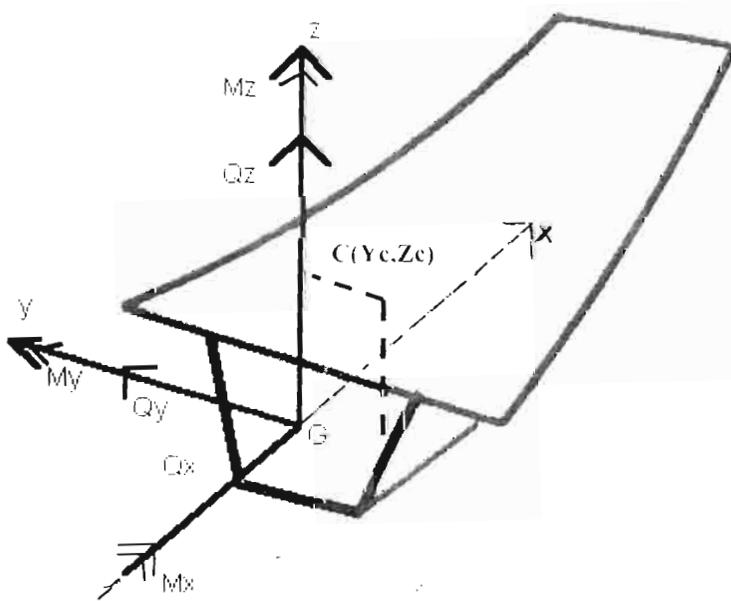
$$\theta = \begin{bmatrix} 0 & -\theta_z(s) & \theta_y(s) \\ \theta_z(s) & 0 & -\theta_x(s) \\ -\theta_y(s) & \theta_x(s) & 0 \end{bmatrix} \quad (3.1)$$

$$r := \begin{bmatrix} 0 & z & y \\ -z_b(z) & 0 & 0 \\ y_b(y) & 0 & 0 \end{bmatrix} \quad (3.2)$$

Avec  $z_b(z) = \bar{z} - \bar{z}_c$   $y_b(y) = \bar{y} - \bar{y}_c$  (3.3)

Ou,  $C (y_c, z_c)$  est le centre de Courbure Torsion qui ne coïncide pas toujours avec le centre géométrique  $G$ .

$$el := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad (3.4)$$



Poutre Caisson courbe avec Efforts et Moments de section

**Fig 3-6 :** Tablier de pont caisson courbe avec Repère local et centre de courbure torsion  $C (Y_c, Z_c)$

### 3-2-2 Expressions vectorielles et matricielles

Les relations cinématiques sont données par les expressions vectorielles du Champ  $\mathbf{U}$  { translations, rotations } ainsi que de sa dérivée  $\mathbf{U}'$  soit :

$$\mathbf{U} = \mathbf{U}_g + \mathbf{r}\theta - \varpi(y,z)\psi(s) \mathbf{e}_1 \quad (3.5.a)$$

Composantes de la dérivée de la fonction de gauchissement :

On définit sur la section  $(y,z)$  :

$$\begin{aligned} \omega_{x,x} &= \frac{\partial}{\partial x} \varpi(y,z) = 0 \\ \omega_{x,y} &= \frac{\partial}{\partial x} \varpi(y,z) + \frac{\partial}{\partial y} \omega(y,z) = \frac{\partial}{\partial y} \omega(y,z) = \omega_{,y} \\ \omega_{x,z} &= \frac{\partial}{\partial x} \varpi(y,z) + \frac{\partial}{\partial z} \omega(y,z) = \frac{\partial}{\partial z} \omega(y,z) = \omega_{,z} \end{aligned} \quad (3.6)$$

$$\mathbf{U}' = \mathbf{U}'_g + \mathbf{r}\theta' + \psi(s) \begin{bmatrix} 0 \\ \omega_{,y} \\ \omega_{,z} \end{bmatrix} - \omega \psi(s) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (3.7.a)$$

On définit le vecteur  $\mathbf{k}$  comme la dérivée du vecteur des rotations libres  $\theta$  soit :

$$\mathbf{k} = \theta' = \frac{\partial \theta}{\partial s} = \{\mathbf{k}_x, \mathbf{k}_y, \mathbf{k}_z\}^T \quad (3.8.a)$$

Sous forme matricielle nous obtenons :

Vecteur rotation  $\mathbf{k}$

$$\mathbf{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial s} \theta_y(s) \\ \frac{\partial}{\partial s} \theta_x(s) \\ \frac{\partial}{\partial s} \theta_z(s) \end{bmatrix} + \begin{bmatrix} 0 & K_z & -K_y \\ -K_z & 0 & K_x \\ K_y & -K_x & 0 \end{bmatrix} \begin{bmatrix} \theta_y(s) \\ \theta_x(s) \\ \theta_z(s) \end{bmatrix} \quad (3.8.b)$$

Champ de déplacements  $\mathbf{U}$ 

$$\mathbf{U} = \begin{bmatrix} u_z(s) \\ v_y(s) \\ w_y(s) \end{bmatrix} + \begin{bmatrix} 0 & z & -y \\ z_b(z) & 0 & 0 \\ v_b(y) & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta_1(s) \\ \theta_2(s) \end{bmatrix} - \omega(y, z) \psi(s) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (3.5.b)$$

Dérivée de  $\mathbf{U}$ 

$$\mathbf{U}' = \begin{bmatrix} \frac{\partial}{\partial s} u_z(s) \\ \frac{\partial}{\partial s} v_y(s) \\ \frac{\partial}{\partial s} w_y(s) \end{bmatrix} + \begin{bmatrix} 0 & z & -y \\ z_b(z) & 0 & 0 \\ v_b(y) & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} k_y \\ k_x \\ k_z \end{bmatrix} + \psi(s) \begin{bmatrix} 0 \\ \frac{\partial}{\partial y} \omega(y, z) \\ \frac{\partial}{\partial z} \omega(y, z) \end{bmatrix} - \omega(y, z) \psi'(s) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (3.7.b)$$

Le champ des déformations ( $\mathcal{E}$ ) est donnée par l'expression vectorielle :

$$\mathcal{E} = \frac{\partial \mathbf{U}}{\partial s} + \mathbf{e}_1 \times \theta = \mathbf{U}' + \mathbf{K} \times \mathbf{U} + \mathbf{T} \times \theta \quad (3.9.a)$$

$\mathbf{e}_1$  représente la tangente unitaire à la section dans le repère local de Fresnet

$\mathbf{K}$  le vecteur courbure torsion dans le repère local

$\theta$  le vecteur rotations libres

Le signe ( $\times$ ) est une multiplication

En expression sous forme matricielle nous obtenons :

$$\begin{aligned}
 \varepsilon &= \begin{bmatrix} \frac{\partial}{\partial s} u_g(s) \\ \frac{\partial}{\partial s} v_g(s) \\ \frac{\partial}{\partial s} w_g(s) \end{bmatrix} + \begin{bmatrix} 0 & K_z & -K_v \\ -K_z & 0 & K_v \\ K_v & -K_v & 0 \end{bmatrix} \begin{bmatrix} u_g(s) \\ v_g(s) \\ w_g(s) \end{bmatrix} + \begin{bmatrix} 0 & -v \\ -z_h(z) & 0 & 0 \\ v_h(y) & 0 & 0 \end{bmatrix} \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix} \\
 \psi(s) \begin{bmatrix} 0 \\ \frac{\partial}{\partial y} w(y,z) \\ \frac{\partial}{\partial z} w(y,z) \end{bmatrix} &= \omega(y,z) \left( \frac{\partial}{\partial s} \psi(s) \right) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \theta_y(s) \\ \theta_x(s) \\ \theta_z(s) \end{bmatrix} \quad (3.9.b)
 \end{aligned}$$

Le développement des produits matriciels donne les composantes des trois déformations principales sous la forme :

$$\begin{aligned}
 \varepsilon_s &= \left( \frac{\partial}{\partial s} u_g(s) \right) + K_z v_g(s) - K_v w_g(s) + z K_v - v K_z - \omega(y,z) \left( \frac{\partial}{\partial s} \psi(s) \right) \\
 \varepsilon_x &= \left( \frac{\partial}{\partial s} v_g(s) \right) - K_z u_g(s) + K_v w_g(s) - z_h(z) K_v - \psi(s) \left( \frac{\partial}{\partial y} w(y,z) \right) + \theta_x(s) \\
 \varepsilon_z &= \left( \frac{\partial}{\partial s} w_g(s) \right) + K_v u_g(s) - K_v v_g(s) + v_h(y) K_v - \psi(s) \left( \frac{\partial}{\partial z} w(y,z) \right) - \theta_z(s)
 \end{aligned} \quad (3.10)$$

Nous rappelons que l'indice (s) de l'abscisse curviligne porte le même sens que (x) qui est donné par la tangente unitaire du repère local de Fresnet.

Ce qui signifie que :

$$\mathcal{E}_s = \mathcal{E}_{xx}, \quad \mathcal{E}_{sy} = \mathcal{E}_{xy}, \quad \mathcal{E}_{sz} = \mathcal{E}_{xz}$$

Nous donnons en outre les composantes du vecteur k :

$$\kappa_{\bar{y}} := K_{\bar{y}} \theta_{\bar{y}}(s) - K_{\bar{y}} \theta_{\bar{z}}(s) + \left( \frac{\hat{c}}{\hat{c}s} \theta_{\bar{y}}(s) \right)$$

$$\kappa_{\bar{y}} := -K_{\bar{z}} \theta_{\bar{y}}(s) + K_{\bar{y}} \theta_{\bar{z}}(s) + \left( \frac{\hat{c}}{\hat{c}s} \theta_{\bar{y}}(s) \right)$$

$$\kappa_{\bar{z}} := K_{\bar{y}} \theta_{\bar{x}}(s) - K_{\bar{x}} \theta_{\bar{y}}(s) + \left( \frac{\hat{c}}{\hat{c}s} \theta_{\bar{z}}(s) \right)$$

(3.11)

Avec de nouvelles considérations, nous pouvons écrire les expressions des déformations  $\mathcal{E}_s = \mathcal{E}_{xx}$ ,  $\mathcal{E}_{sy} = \mathcal{E}_{xy}$ ,  $\mathcal{E}_{sz} = \mathcal{E}_{xz}$  sous les formes :

$$\varepsilon_{\bar{x}} := \varepsilon_0 - y' \kappa_{\bar{z}} + z \kappa_{\bar{y}} - w(y, z) \kappa_{\bar{u}} \quad (3.12.a)$$

avec:

$$\varepsilon_0 := \left( \frac{\hat{c}}{\hat{c}s} u_g(s) \right) + K_{\bar{z}} v_g(s) - K_{\bar{y}} w_g(s) \quad (3.12.b)$$

$$\varepsilon_{\bar{xy}} := \gamma_{\bar{xy}} - \left( \left( \frac{\hat{c}}{\hat{c}y'} w(y, z) \right) + z_h(c) \right) \gamma_{\bar{xy}} + \left( \frac{\hat{c}}{\hat{c}y'} w(y, z) \right) \gamma_{\bar{u}} \quad (3.13.a)$$

avec:

$$\gamma_{\bar{xy}} = \left( \frac{\hat{c}}{\hat{c}s} v_g(s) \right) - K_{\bar{z}} u_g(s) + K_{\bar{x}} w_g(s) + \theta_{\bar{z}}(s)$$

$$\gamma_{\bar{xy}} = K_{\bar{z}} \theta_{\bar{y}}(s) - K_{\bar{y}} \theta_{\bar{z}}(s) + \left( \frac{\hat{c}}{\hat{c}s} \theta_{\bar{x}}(s) \right)$$

$$\gamma_{\bar{w}} = \gamma_{\bar{xy}} - \kappa_{\bar{w}} \quad (3.13.b)$$

$$\varepsilon_{\bar{sz}} := \gamma_{\bar{sz}} + \left( y_h(c) - \left( \frac{\partial}{\partial z} w(y, z) \right) \right) \gamma_{\bar{xy}} + \left( \frac{\partial}{\partial z} w(y, z) \right) \gamma_{\bar{w}} \quad (3.14.a)$$

où :

$$\gamma_{xz} = \left( \frac{\hat{c}}{\hat{c}y} w_g(s) \right) + K_v u_g(s) - K_v v_g(s) - O_v(s) \quad (3.14.b)$$

### 3-3 Conditions d'Equilibre

#### 3-3-1 Grandeurs

On note :

$\mathbf{Q} = \{Q_x, Q_y, Q_z\}^T$  le vecteur des efforts normal et tranchants internes respectivement dans les directions x, y et z.

$\mathbf{M} = \{M_x, M_y, M_z\}^T$  le vecteur des moments internes.

$\mathbf{F}_b = \bar{\mathbf{F}} = \{F_{bx}, F_{by}, F_{bz}\}^T$  le torseur des forces appliquées concentrées et réparties dans les trois directions

$\mathbf{M}_b = \bar{\mathbf{M}} = \{M_{bx}, M_{by}, M_{bz}\}^T$  le torseur des moments appliqués concentrés et répartis dans les trois directions.

#### 3-3-2 Expressions vectorielles et matricielles

Les expressions des relations d'équilibre se traduisent sous forme vectorielle par ces deux équations :

Equilibre des efforts normaux et tranchants internes par rapport aux forces concentrées ou réparties  $\mathbf{F}_b$  :

$$\mathbf{Q}' + \mathbf{K} \times \mathbf{Q} + \mathbf{F}_b = \mathbf{0} \quad (3.15.a)$$

Equilibre des moments internes et de chargements concentrés ou répartis :

$$\mathbf{M}' + \mathbf{K} \times \mathbf{M} + \mathbf{e}_3 \times \mathbf{Q} + \mathbf{M}_b = 0 \quad (3.16.a)$$

Nous pouvons ramener ces expressions sous formes matricielles :

$$\begin{bmatrix} \frac{\partial}{\partial s} Q_v(s) \\ \frac{\partial}{\partial s} Q_1(s) \\ \frac{\partial}{\partial s} Q_2(s) \end{bmatrix} + \begin{bmatrix} 0 & K_z & -K_y \\ -K_z & 0 & K_x \\ K_y & -K_x & 0 \end{bmatrix} \begin{bmatrix} Q_v(s) \\ Q_1(s) \\ Q_2(s) \end{bmatrix} + \begin{bmatrix} Fh_1 \\ Fh_2 \\ Fh_3 \end{bmatrix} = 0 \quad (3.15.b)$$

$$\begin{bmatrix} \frac{\partial}{\partial s} M_x(s) \\ \frac{\partial}{\partial s} M_y(s) \\ \frac{\partial}{\partial s} M_z(s) \end{bmatrix} + \begin{bmatrix} 0 & K_z & -K_y \\ -K_z & 0 & K_x \\ K_y & -K_x & 0 \end{bmatrix} \begin{bmatrix} M_x(s) \\ M_y(s) \\ M_z(s) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} Q_v(s) \\ Q_1(s) \\ Q_2(s) \end{bmatrix} + \begin{bmatrix} Mb_x \\ Mb_y \\ Mb_z \end{bmatrix} = 0 \quad (3.16.b)$$

### 3-4 Lois Constitutives

#### 3-4-1 Grandeurs

Les contraintes en toute section d'abscisse ( $x = s$ ) s'expriment par :

$$\sigma_{xx} = E \varepsilon_{xx}$$

$$\tau_{xy} = G \varepsilon_{xy}$$

$$\tau_{xz} = G \varepsilon_{xz}$$

(3-4)

Où :

$E$  : module d'élasticité du matériau de la poutre

$G$  : module de cisaillement

$\sigma_{xx}$  : contrainte normale dans la direction principale de l'abscisse curviligne  $s = x$

$\tau_{xy}$  : contrainte de cisaillement normale à  $x$  dans la direction  $y$

$\tau_{xz}$  : contrainte de cisaillement normale à  $x$  dans la direction  $z$

On définit alors le vecteur des contraintes :

$$\boldsymbol{\sigma} = \{\sigma_{xx}, \tau_{xy}, \tau_{xz}\}^T = \begin{bmatrix} E & 0 & 0 \\ 0 & G & 0 \\ 0 & 0 & G \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{xy} \\ \varepsilon_{xz} \end{bmatrix} \quad (3.5)$$

Par intégration des contraintes, on définit les différents Efforts et Moments :

$\mathbf{Q} = \{Q_x, Q_y, Q_z\}^T$  efforts normal  $Q_x$  et tranchants  $Q_y$  et  $Q_z$

$\mathbf{M} = \{M_y, M_z, M_w\}^T$  moments fléchissant  $M_y$  et  $M_z$  et Bi-moment  $M_w$

$\mathbf{T} = \{T_{sv}, T_w\}^T$  les torsions avec :

$T_{sv}$  : la Torsion uniforme de St Venant

$T_w$  : la Torsion non uniforme de Vlassov

On détermine le vecteur des efforts résultants :

$\boldsymbol{\sigma}_R = \{\mathbf{Q}, \mathbf{M}, \mathbf{T}\}^T = \{Q_x, Q_y, Q_z, M_y, M_z, M_w, T_{sv}, T_w\}^T$  avec :

L'effort normal et les efforts tranchants qui sont donnés par :

$$\mathbf{Q} = \int_A \boldsymbol{\sigma} dA \quad \text{soit} \quad \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} = \int_A \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \end{bmatrix} dA \quad (3.6.a)$$

$$Q_x = \int \sigma_{yy} dA \quad Q_y = \int \sigma_{xy} dA \quad Q_z = \int \sigma_{xz} dA \quad (3.6.b)$$

Les moments de flexion  $M_x$ ,  $M_y$  et le Bi-moment  $M_w$  qui sont donnés par :

$$\mathbf{M} = \begin{bmatrix} M_y \\ M_z \\ M_w \end{bmatrix} = \int \sigma_{xx} \begin{bmatrix} -y \\ -z \\ w(x,y,z) \end{bmatrix} dA \quad (3.7.a)$$

$$M_y = \int -z \sigma_{xx} dA \quad M_z = \int -y \sigma_{xx} dA \quad M_w = \int -w(x,y,z) \sigma_{xx} dA \quad (3.7.b)$$

Le moment de torsion total autour de x s'obtient par  $M_x = T_{sv} + T_w$  avec :

$$T_{sv} = - \int \sigma_{xy} [(z-z_c) + w_{,y}] dA + \int \sigma_{xz} [(y-y_c) - w_{,z}] dA \quad \text{Torsion de St Venant} \quad (3.8.a)$$

$$T_w = \int \int \sigma_{xy} w_{,xy} + \sigma_{xz} w_{,xz} dA \quad \text{Torsion de Vlasov} \quad (3.8.b)$$

### 3-4-2 Expressions

Etant donné :

$\sigma_R$  : vecteur conjugué des « contraintes » ( Efforts Résultants)

On définit :

$\varepsilon_R$  : vecteur conjugué des « déformations » ( Déformations Résultantes)

$$\text{avec} \quad \varepsilon_R = \{ \varepsilon_0, \gamma_{xy}, \gamma_{xz}, \kappa_y, \kappa_z, \kappa_w, \gamma_{sv}, \gamma_w \}^T \quad (3.9)$$

Les contraintes s'expriment encore par :

$$\begin{aligned}\sigma_{xx} &= E \varepsilon_{xx} = E \varepsilon_0 + z E \kappa_y - y E \kappa_z - \varpi(y,z) E \kappa_w \\ \tau_{xy} &= G \varepsilon_{xy} = G \gamma_{xy} - [\varpi_{,y} + z_b(c)] G \gamma_{sv} + \varpi_{,y} G \gamma_w \\ \tau_{xz} &= G \varepsilon_{xz} = G \gamma_{xz} - [y_b(c) - \varpi_{,z}] G \gamma_{sv} + \varpi_{,z} G \gamma_w\end{aligned}\quad (3.10)$$

avec :

$$\begin{aligned}\varepsilon_0 &= \frac{\hat{c}}{\partial s} u_g(s) + K_z v_g(s) - K_x w_g(s) \\ \gamma_{xy} &= \frac{\hat{c}}{\partial s} v_g(s) - K_z u_g(s) + K_x w_g(s) - \theta_x(s) \\ \gamma_{xz} &= \frac{\hat{c}}{\partial s} w_g(s) - K_y u_g(s) - K_x v_g(s) + \theta_y(s) \\ \kappa_x &= \frac{\hat{c}}{\partial s} \theta_x(s) - K_y \theta_z(s) + K_z \theta_y(s) \\ \kappa_y &= \frac{\hat{c}}{\partial s} \theta_y(s) - K_z \theta_x(s) + K_x \theta_z(s) \\ \kappa_z &= \frac{\hat{c}}{\partial s} \theta_z(s) - K_y \theta_x(s) - K_x \theta_y(s) \\ \kappa_x &= \frac{\hat{c}}{\partial s} \psi(s) \\ \gamma_{sv} &= \frac{\hat{c}}{\partial s} \theta_z(s) + K_z \theta_y(s) - K_y \theta_z(s) \\ \gamma_w &= \gamma_{sv} - \psi(s)\end{aligned}\quad (3.11)$$

En utilisant les équations précédentes on peut finalement écrire l'Equation constitutive caractéristique type entre les efforts et les déformations résultantes :

$$\sigma_R = \mathbf{C} \varepsilon_R \quad (3.12)$$

Où la matrice symétrique C (8x8) est le tenseur d'élasticité spatiale de la poutre.

Elle s'exprime en plus avec les inerties et les rigidités des sections

$$A = \int_A dA \quad : \text{aire de la section de la poutre avec } dA = dydz$$

$$I_y = \int_A z^2 dA \quad : \text{moment d'inertie principale par rapport à y}$$

$$I_z = \int_A y^2 dA \quad : \text{moment d'inertie principale par rapport à z}$$

$$I_{yz} = \int_A yz dA \quad : \text{produit d'inertie relativement aux axes y et z}$$

$$I_p = I_y + I_z \quad : \text{moment d'inertie polaire ou quadratique}$$

$$I_w = \int_A w^2 dA \quad : \text{moment d'inertie de gauchissement (ou sectoriel)}$$

$$S_w = \int_A \omega(y,z) dA \quad : \text{premier moment sectoriel}$$

$$I_{wy} = \int_A wy dA$$

Produits d'inertie de gauchissement

$$I_{wz} = \int_A wz dA$$

$$J = \int_A [(y_b(c) - w_z)^2 + (z_b(c) + w_y)^2] dA \quad : \text{Inertie torsionnelle}$$

Avec les hypothèses sur la section dépendant du choix du repère local central principal d'inertie:

$$\int_A z dA = \int_A y dA = 0$$

$$\int_A y_b(c) dA = -y_c$$

$$\int_A z_b(c) dA = -z_c$$

C (  $y_c$   $z_c$  ) le centre de courbure torsion

Après intégration des contraintes, on trouve pour les efforts résultants

Effort normal  $Q_x$  et efforts tranchants  $Q_y$  et  $Q_z$  :

$$Q_x = E \left[ 1 \varepsilon_0 - \int \int_A \omega(y,z) dy dz \right] \quad (3.13.a)$$

$$Q_y = G(\gamma_{xy} A + z_c \gamma_{xz}) \quad (3.13.b)$$

Les Moments:

$$M_y = -E \left[ I_x \left( \theta_x(s) + \left( \frac{\hat{c}}{\hat{\partial}s} \theta_z(s) \right) \right) - I_y \left( \frac{\hat{c}}{\hat{\partial}s} \theta_x(s) \right) + I_{xy} \left( \frac{\hat{c}}{\hat{\partial}s} \psi(s) \right) \right] \quad (3.14.a)$$

$$M_z = -E \left[ -I_x \left( \theta_x(s) + \left( \frac{\hat{c}}{\hat{\partial}s} \theta_z(s) \right) \right) + I_{yz} \left( \frac{\hat{c}}{\hat{\partial}s} \theta_x(s) \right) - \left( \frac{\hat{c}}{\hat{\partial}s} \psi(s) \right) I_{yz} \right] \quad (3.14.b)$$

$$M_w = -E \left[ \int_V \omega(y, z) dy dz - \left( \theta_x(s) + \left( \frac{\hat{c}}{\hat{\partial}s} \theta_z(s) \right) \right) I_{wy} + \left( \frac{\hat{c}}{\hat{\partial}s} \theta_x(s) \right) I_{wz} - \left( \frac{\hat{c}}{\hat{\partial}s} \psi(s) \right) I_w \right] \quad (3.14.c)$$

Les Torsions:

$$T_{xy} = G J \gamma_{xy} + G \gamma_{xy} z_i - G \gamma_{xz} y_i - G (J - I_p) \gamma_w \quad (3.15.a)$$

$$T_w = G (I_p - J) \gamma_w + G (J - I_p) \gamma_w \quad (3.15.b)$$

$$M_x = T_{xy} + T_w = G (I_x + I_z) \gamma_{xy} + G \gamma_{xy} z_i - G \gamma_{xz} y_i \quad (3.15.c)$$

Noûs poserons par la suite:

$$S_w := \int_V \omega(y, z) dy dz$$

Ce qui donne :

$$Q_x := E(A \varepsilon_0 - S_w)$$

$$M_w := -E \left( S_w - \left( \theta_x(s) + \left( \frac{\hat{c}}{\hat{\partial}s} \theta_z(s) \right) \right) I_{wy} + \left( \frac{\hat{c}}{\hat{\partial}s} \theta_x(s) \right) I_{wz} - \left( \frac{\hat{c}}{\hat{\partial}s} \psi(s) \right) I_w \right)$$

En écrivant la relation constitutive  $\sigma_R = \mathbf{C} \varepsilon_R$ , on isole la matrice C qui s'écrit dans sa forme la plus globale :

$$\mathbf{C} = \begin{bmatrix}
 E \cdot & 0 & 0 & 0 & 0 & -E S_{xy} & 0 & 0 \\
 0 & G \cdot I & 0 & 0 & 0 & 0 & G z_c \cdot A & 0 \\
 0 & 0 & G \cdot A & 0 & 0 & 0 & -G y_c \cdot A & 0 \\
 0 & 0 & 0 & E I_{yy} & -E I_{yz} & -E I_{zy} & 0 & 0 \\
 0 & 0 & 0 & -E I_{yz} & E I_{yy} & E I_{zy} & 0 & 0 \\
 E \cdot & 0 & 0 & -E I_{zy} & E I_{yy} & E I_{zz} & 0 & 0 \\
 0 & G z_c \cdot A & -G y_c \cdot A & 0 & 0 & 0 & G \cdot J & -G (J - I_p) \\
 0 & 0 & 0 & 0 & 0 & 0 & G (J - I_p) & G (J - I_p)
 \end{bmatrix} \quad (3.16.a)$$

On peut alors remarquer qu'elle est symétrique. Nous ferons d'autres considérations supplémentaires pour donner une dernière expression à cette matrice de rigidité.

- *Choix du repère local central principal d'inertie :*

L'étude de la statique des poutres est généralement faite compte tenant du choix de ce repère pour des raisons évidemment simplificatrices. Sous cette considération :

- le produit d'inertie  $I_{yz}$  est nul.
- les inerties  $I_y$  et  $I_z$  sont principales.

Pour le cas des sections constituées de parois minces de faible épaisseur  $t$ , étant donné que  $y$  et  $z$  sont des axes principaux d'inertie, nous avons pour la détermination des coordonnées du centre de torsion  $C (y_C, z_C)$  les relations suivantes :

Nous proposerons deux méthodes de détermination du centre de torsion.

Première méthode :

On détermine les coordonnées du centre de torsion par :

$$y_C = -\frac{I_{\overline{xy}}}{I_y} \quad z_C = -\frac{I_{\overline{xz}}}{I_z}$$

Soient  $y$  et  $z$  des axes principaux d'inertie pour la section de poutre.

- Si la section est symétrique par rapport à  $z$ , alors le centre de torsion est situé sur cet axe et on a  $y_C = 0$  soit  $C(0, z_C)$ .
- Si le centre géométrique  $G$  est un point de symétrie, alors  $G$  et  $C$  coïncident.

Deuxième méthode :

On définit :

$$Q_y = \int_A \alpha x y dA \quad Q_z = \int_A \alpha x z dA \quad \text{comme les efforts tranchants de section}$$

$$M_{x_C} = \int_A \alpha x y (z - z_C) dA + \int_A \alpha x z (y - y_C) dA \quad \text{le moment de torsion par rapport au point C}$$

$$M_{x_G} = \int_A \alpha x y (z) dA + \int_A \alpha x z (y) dA \quad \text{le moment de torsion par rapport à G}$$

Nous avons la relation :

$$M_{x_C} = M_{x_G} - y_C \cdot Q_z + z_C \cdot Q_y$$

Le point  $C$  est appelé centre de cisaillement / torsion si les contraintes  $\sigma_{11}$ ,  $\sigma_{22}$  dues à la flexion sont telles que:

$$M_{x_C} = M_{x_G} - y_C \cdot Q_z + z_C \cdot Q_y = 0$$

Ce qui conduit à :

$$y_C = \frac{M}{Q} \quad (\text{avec } Q_y = 0 \text{ et } Q_z \neq 0)$$

$$z_C = -\frac{M}{Q} \quad (\text{avec } Q_z = 0 \text{ et } Q_y \neq 0)$$

La démarche pour déterminer  $y_C$  ( idem pour  $z_C$  ) est la suivante pour une section donnée:

- on évalue d'abord  $\sigma_x$  et  $\sigma_y$  comme contraintes de cisaillement.
- par intégration de ces contraintes on détermine  $Q_y$  et  $Q_z$ .
- on évalue finalement  $M_{xG}$  puis enfin  $y_C$  et  $z_C$

▪ *Considérations sur la fonction de gauchissement  $\varpi(y,z)$  :*

En normalisant la cordonnée sectorielle  $\varpi(y,z)$  sur toute la section, nous avons le résultat fondamental:

$$S\varpi = \int_A \varpi(y,z)dA = 0$$

Nous allons nous placer dans l'hypothèse des sections simplement connexes (sections à parois minces ouvertes ou fermées) doublement symétriques pour lesquelles la fonction de gauchissement est nulle sur les frontières.

Nous allons appliquer le Théorème de Green au calcul des produits d'inertie de gauchissement :

$$I_{wy} = \int_A y \varpi dA \qquad I_{wz} = \int_A z \varpi dA$$

$$\text{Ainsi que des termes :} \quad \int_A \varpi_{,y} dA \qquad \int_A \varpi_{,z} dA$$

Ces considérations annulent ces quatre quantités. En conséquence, il apparaît aussi que  $y_C = 0$  et  $z_C = 0$ , d'où le centre de torsion coïncide avec le centre géométrique. Ces résultats nous donnent donc finalement pour la matrice C :

$$C = \begin{bmatrix} EA & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & GA & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & GA & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & EI_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & EI_x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & EI_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & GJ & -G(J-I_p) \\ 0 & 0 & 0 & 0 & 0 & 0 & G(I_p - J) & G(J - I_p) \end{bmatrix} \quad (3.16.b)$$

Nous pouvons aussi ramener cette matrice d'élasticité spatiale C en une matrice symétrique (7x7) en ne considérant que le moment de torsion totale :

$$M_x = T_{sv} + T_w = G(I_y + I_z)\gamma_{sv} + G\gamma_{tw} - G\gamma_{tw} \\ M_x = G I_p \gamma_{sv}$$

Le vecteur des efforts résultants se ramène donc à :

$$\sigma_R = \{ Q_x, Q_y, Q_z, M_x, M_y, M_z, M_w \}^T$$

Le vecteur des déformations résultantes se limite à :

$$\epsilon_R = \{ \epsilon_0, \gamma_{xy}, \gamma_{xz}, \gamma_{sv}, \kappa_y, \kappa_z, \kappa_w \}^T$$

Avec ces nouvelles considérations, C s'écrit :

$$C = \begin{bmatrix} EA & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & GA & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & GA & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & G I_p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & EI_y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & EI_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & EI_z \end{bmatrix} \quad (3.16.c)$$

En pratique nous allons poser et résoudre le problème différentiel de la poutre avec la matrice  $C(8 \times 8)$  établie ( 3.16.b) en tenant en compte les torsions  $T_{sv}$  et  $T_w$ .

### 3-5 Equations aux dérivées partielles fondamentales du Problème

Connaissant déjà la relation constitutive :

$$\sigma_R = \mathbf{C} \varepsilon_R \quad (3.17)$$

On construit la relation Déformations - Déplacements .

$$\varepsilon_R = \mathbf{L} \mathbf{U} \quad (3.18)$$

Avec l'opérateur matriciel rectangulaire  $L$  ( $8 \times 7$ ) qui est donné par :

$$L : \begin{bmatrix} Ds & Kz & -Ky & 0 & 0 & 0 & 0 \\ -Kz & Ds & Kx & 0 & 0 & 1 & 0 \\ Ky & -Kx & Ds & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -Kz & Ds & Kx & 0 \\ 0 & 0 & 0 & Ky & -Kx & Ds & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Ds \\ 0 & 0 & 0 & Ds & Kz & -Ky & 0 \\ 0 & 0 & 0 & Ds & Kz & -Ky & -Ds \end{bmatrix} \quad (3.19)$$

$Ds$  est un opérateur différentiel égal à :  $Ds = \frac{\partial}{\partial s}$

$\mathbf{K} = \{K_x, K_y, K_z\}^T$  est le vecteur courbure torsion local.

On écrit l'équation de Lamé :

$$\mathbf{L}^T \sigma_R + \mathbf{b} = \mathbf{0} \quad (3.20)$$

En faisant (3.17) et (3.18) dans (3.20) nous posons l'équation matricielle fondamentale du problème :

$$L^T C L U + b = 0 \quad (3.21)$$

Ses composantes lignes fournissent les 7 équations différentielles fondamentales du problème. On obtient pour les différents cas un système de 7 équations différentielles non linéaires à 7 inconnues couplées et d'ordre 2.

## 4 RESOLUTION DU PROBLEME FONDAMENTAL

### 4-1 Résolution Analytique

Nous nous proposerons dans un premier temps de faire la résolution analytique du problème différentiel posé par les trois cas de poutres circulaire, hélicoïdale et droite. En effet, le problème posé par le système des sept équations différentielles couplées et d'ordre deux est non linéaire et requière par conséquent de puissants outils mathématiques de résolution.

A cet effet, une bibliothèque de programmes sera proposée avec le logiciel de calcul formel Maple 6.

#### 4-1-1 Programmation

Notre environnement de programmation est Maple 6 / Fortran. Avec Maple nous proposerons une bibliothèque de programmes de calcul des trois cas de poutres circulaire, hélicoïdale et droite, de façon à sortir toutes les grandeurs cinématiques algébriques, vectorielles et matricielles qui vont nous servir à effectuer des tests numériques avec des plates-formes de calcul par éléments finis en Fortran.

#### 4-1-1-1 Présentation du Logiciel de Calcul formel Maple®

Maple 6 est un système informatique compréhensible pour mathématiques avancées. Il inclue des facilités pour de l'algèbre interactive, du calcul formel, des mathématiques discrètes, des effets graphiques, de la compilation numérique ainsi que bien d'autres domaines mathématiques.

Il fournit aussi un environnement propice pour le développement de programmes mathématiques utilisant une librairie exhaustive de fonctions et d'opérateurs prédéfinis. C'est donc un puissant logiciel de calcul formel développé de chez Waterloo Maple Inc.

#### 4-1-1-2 Objectifs de la programmation

- Considérations pour la prise en compte des paramètres géométriques :
  - Vecteurs de la base locale
  - Matrices de transformation de coordonnées
  - Vecteurs courbure torsion
  
- Génération de la matrice  $\mathbf{C}$  et de l'opérateur matriciel différentiel  $\mathbf{L}$  :
  - Initialisation des moments d'inerties
  - Intégration des contraintes
  - Expressions des efforts résultants
  - Expressions des déformations résultantes
  
- Construction de l'Equation de lamé :
  - Relations cinématiques
  - Prise en compte des chargements (efforts et moments concentrés ou repartis)
  - Etablissement des équations différentielles fondamentales
  - Résolution symbolique (solutions brutes)

- Régénération de la matrice de rigidité **K**
  - Conditions limites de bord
  - Solutions homogènes du champ de déplacement
  - Détermination des constantes d'intégration
  - Raffinement des solutions
  - Forces équivalentes aux nœuds

### Génération de la Matrice de rigidité [ K ] :

*Methodologie :*

Le champ de déplacement étant trouvé, soit:

$$\{ \mathbf{U} \} = \{ u_g(s) , v_g(s) , w_g(s) , \theta_x(s) , \theta_y(s) , \theta_z(s) , \psi(s) \}^T$$

On pose:

$$\{ \alpha \} = \{ C1, C2, C3, C4, C5, C6, C7, C8, C9, C11, C12, C13, C14 \}^T$$

comme le vecteur des constantes d'intégration à déterminer.

On définit les déplacements respectifs aux nœuds :

$$\text{à } (s = 0) \quad \{ \mathbf{U}_1 \} = \{ u_1(0) , v_1(0) , w_1(0) , \theta_{x1}(0) , \theta_{y1}(0) , \theta_{z1}(0) , \psi_1(0) \}^T$$

$$\text{à } (s=L) \quad \{ \mathbf{U}_2 \} = \{ u_2(L) , v_2(L) , w_2(L) , \theta_{x2}(L) , \theta_{y2}(L) , \theta_{z2}(L) , \psi_2(L) \}^T$$

Si on définit:

$$\{ \mathbf{U} \} = \{ \{ \mathbf{U}_1 \} , \{ \mathbf{U}_2 \} \}^T$$

comme un vecteur colonne (14) de composantes respectives  $U[i]$  ( $i = 1..14$ )

On initialise les coefficients d'intégration par :

$$P(j) = \{ C_i = 0 \text{ si } i \neq j \text{ sinon } C_j = 1 \}$$

On calcule alors les coefficients de la matrice C soit  $C[i, j]$  par substitution des  $P(j)$  dans les  $U[i]$  :

$$C[i, j] = \text{subs}( P(j), U[i] ) \quad (i=1..14, j=1..14) \quad (4.7)$$

Nous obtenons par cette méthode directe une matrice C (14 x 14). C'est cette dernière méthode qui a été retenue et qui figure dans les notes de programmation.

**Détermination des forces équivalentes aux nœuds ( réactions d'appuis nodaux ) :**

Connaissant le vecteur des efforts résultants que nous posons par :

$$\{ F \} = \{ Q_x, Q_y, Q_z, M_x, M_y, M_z, M_w \}^T$$

on construit la matrice [ B ] dans la relation :

$$\{ F \} = [ B ]. \{ \alpha \} + \{ F_o \} \quad (4.8)$$

soit en faisant (4-1-1-2-5) dans cette dernière :

$$\{ F \} = [ B ]. [ C ]^{-1} \{ \Delta \} - \{ \Delta_o \} \quad (4.9)$$

soit en développant :

$$\{ F \} = [ B ]. [ C ]^{-1} \{ \Delta \} - [ B ]. [ C ]^{-1} \{ \Delta_o \} \quad (4.10)$$

posant :

$$[K] = [B] \cdot [C]^{-1} \quad \text{et} \quad \{F_o\} = [B] \cdot [C]^{-1} \cdot \{\Delta_o\} \quad (4.11)$$

où  $[K]$  et  $[B]$  sont des matrices (14x14).

$[K]$  étant la matrice de rigidité cherchée.

nous obtenons finalement :

$$\{F\} = [K] \cdot \{\Delta\} - \{F_o\} \quad (4.12)$$

Le résultat peut encore être obtenu en partitionnant au départ la matrice  $[B]$  considérant :

$$\{F_1\} = \{Q_x(s=0), Q_y(s=0), Q_z(s=0), M_x(s=0), M_y(s=0), M_z(s=0), M_w(s=0)\}^T$$

$$\{F_2\} = \{Q_x(s=L), Q_y(s=L), Q_z(s=L), M_x(s=L), M_y(s=L), M_z(s=L), M_w(s=L)\}^T$$

On partitionne aussi :  $\{\alpha\} = \{\{\alpha_1\}, \{\alpha_2\}\}^T$  avec

$$\{\alpha_1\} = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}^T$$

$$\{\alpha_2\} = \{C_8, C_9, C_{11}, C_{12}, C_{13}, C_{14}\}^T$$

$$\{F_1\} = [B_{11}(s=0), B_{12}(s=0)] \cdot \{\alpha\} + \{F_o\} \quad (4.13.a)$$

$$\text{de même:} \quad \{F_2\} = [B_{21}(s=0), B_{22}(s=0)] \cdot \{\alpha\} + \{F_o\} \quad (4.13.b)$$

ce qui donne pour

$$[\mathbf{B}] = \begin{bmatrix} B_{11}(s=0) & B_{12}(s=0) \\ B_{21}(s=L) & B_{22}(s=L) \end{bmatrix}$$

#### 4-1-2 Cas Circulaire

*Paramétrisation*       $X = R \cos(t)$        $Y = R \sin(t)$        $Z = 0$       (4.14)

$R$  désigne le rayon de courbure géométrique

$t$  est le paramètre angulaire

*Abscisse curviligne*

on choisit une origine telle que  $S(t=0) = S_0 = 0$  soit :

$$ds = R dt$$

$$S = Rt$$

D où :

$$X = R \cos\left(\frac{S}{R}\right) \quad Y = R \sin\left(\frac{S}{R}\right) \quad Z = 0 \quad (4.15)$$

Cette nouvelle paramétrisation nous permet de traduire toutes les grandeurs en fonction de l'abscisse curviligne  $S$  qui sera la variable de résolution du problème différentiel fondamental.

En particulier, la fonction de gauchissement  $\varpi(y,z)$  pourrait être exprimée avec cette variable soit  $\varpi(y,z) = \varpi(s)$

*Paramètres Géométriques :*

Base du repère local de Fresnet

Vecteur unitaire tangent       $\mathbf{e1} = \{-\sin(t), \cos(t), 0\}^T$

Vecteur unitaire normal       $\mathbf{e2} = \{-\cos(t), -\sin(t), 0\}^T$

Vecteur unitaire binormal       $\mathbf{e3} = \{0, 0, 1\}^T$       (4.16)

En ne tenant pas en compte la rotation d'angle  $\alpha$  de la section de poutre autour de l'axe neutre nous obtenons pour les matrices de transformations

Matrice de transformation Local-Global :

$$\mathbf{LG} = \begin{bmatrix} -\sin(t) & \cos(t) & 0 \\ -\cos(t) & -\sin(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.17.a)$$

Matrice de transformation Global-Local :

$$\mathbf{GL} = \begin{bmatrix} -\sin(t) & -\cos(t) & 0 \\ \cos(t) & -\sin(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.17.b)$$

Vecteur courbure-torsion global :

$$\mathbf{K}_g = \left\{ \frac{\cos(t)}{r}, -\frac{\sin(t)}{r}, 0 \right\}^T \quad (4.18)$$

Vecteur courbure- torsion local :

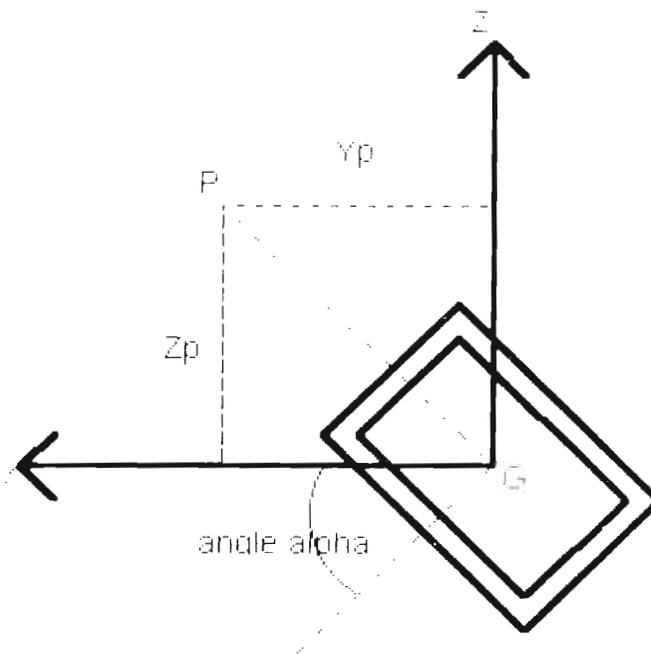
$$\mathbf{K}_l = \left\{ 0, \frac{1}{r}, 0 \right\}^T \quad (4.19.a)$$

Nous considèrerons pour alléger les calculs symboliques dans les programmes établis l'expression générale la variable  $K_y$  du vecteur courbure torsion soit :

$$\mathbf{K}_l = \{0, K_y, 0\}^T \quad (4.19.b)$$

En pratique le programme établi pour le calcul de l'élément poutre circulaire tient compte de l'effet de la matrice de rotation  $R_{\alpha}$  (justification de la torsion qui a tendance à gauchir les sections). Sous cette réserve les paramètres géométriques deviennent :

$$R_{\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (4.20)$$



Rotation d'une section de poutre autour de l'axe neutre X

**Fig 4-1 :** Rotation d'une section de poutre autour de l'axe neutre X en G

$P (y_P, z_P)$  est un point permettant de repérer globalement l'inclinaison  $\alpha$  de la section de poutre.

Matrice de transformation Local-Global .

$$LG = R_{\alpha} \times \begin{bmatrix} -\sin(t) & \cos(t) & 0 \\ -\cos(t) & -\sin(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$LG = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \times \begin{bmatrix} -\sin(t) & \cos(t) & 0 \\ -\cos(t) & -\sin(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$LG = \begin{bmatrix} -\sin(t) & \cos(\alpha)\cos(t) & -\sin(\alpha)\cos(t) \\ -\cos(t) & -\cos(\alpha)\sin(t) & \sin(\alpha)\sin(t) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (4.21.a)$$

On obtient GL par transposée :

$$GL = \begin{bmatrix} -\sin(t) & -\cos(t) & 0 \\ \cos(\alpha)\cos(t) & -\cos(\alpha)\sin(t) & \sin(\alpha) \\ -\sin(\alpha)\cos(t) & \sin(\alpha)\sin(t) & \cos(\alpha) \end{bmatrix} \quad (4.21.b)$$

Pour  $\alpha = 0$  on retrouve le cas précédent.

Vecteur courbure-torsion global :

$$K_g = \left\{ -\frac{\cos(t)}{r}, -\frac{\sin(t)}{r}, 0 \right\}^T \quad (4.22)$$

Vecteur courbure- torsion local :

$$K_l = \left\{ 0, -\frac{-\cos(t)^2 - \cos(\alpha) + \cos(\alpha)\cos(t)^2}{r}, 0 \right\}^T \quad (4.23)$$

#### 4-1-3 Cas Hélicoïdal

*Paramétrisation :*

$$X = R\cos(t) \quad Y = R\sin(t) \quad Z = bt \quad (4.24)$$

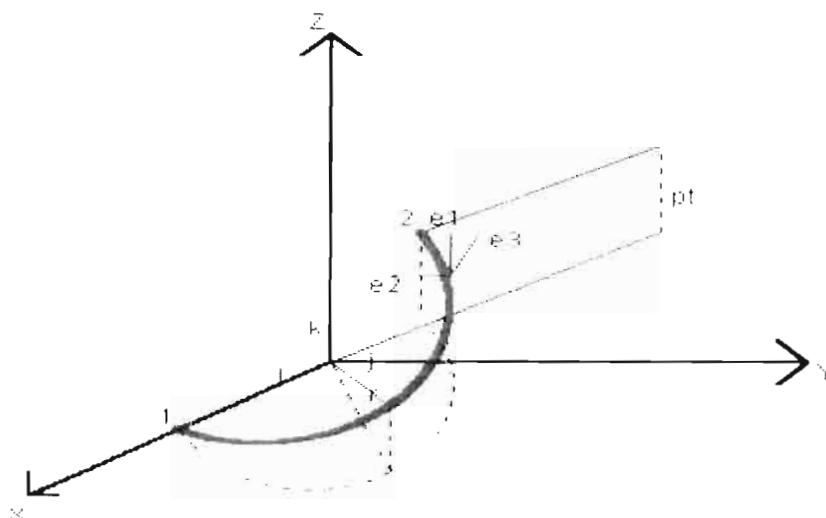
*Abscisse curviligne :*

on choisit une origine telle que  $S(t=0) = S_0 = 0$  soit :

$$ds = \sqrt{R^2 + b^2} dt$$

$$S = \sqrt{R^2 + b^2} t$$

$$X = R \cos \left( \frac{S}{\sqrt{R^2 + b^2}} \right) \quad Y = R \sin \left( \frac{S}{\sqrt{R^2 + b^2}} \right) \quad Z = bt \quad (4.25)$$



Geométrie d'une poutre hélicoïdale

Fig 4-2 : Géométrie générique d'une poutre spatiale hélicoïdale

Paramètres Géométriques :

Base du repère local de Fresnet

Vecteur unitaire tangent  $\mathbf{e1} = \left\{ -\frac{r \sin(t)}{\sqrt{r^2 + p^2}}, \frac{r \cos(t)}{\sqrt{r^2 + p^2}}, \frac{p}{\sqrt{r^2 + p^2}} \right\}^T$

Vecteur unitaire normal  $\mathbf{e2} = \{ -\cos(t), -\sin(t), 0 \}^T$

Vecteur unitaire binormal  $\mathbf{e3} = \left\{ \frac{p \sin(t)}{\sqrt{r^2 + p^2}}, -\frac{p \cos(t)}{\sqrt{r^2 + p^2}}, \frac{r}{\sqrt{r^2 + p^2}} \right\}^T$

(4.26)

En ne tenant pas en compte la rotation d'angle  $\alpha$  de la section de poutre autour de l'axe neutre nous obtenons pour les matrices de transformations :

Matrice de transformation Local-Global :

$$\mathbf{LG} = \begin{bmatrix} \frac{r \sin(t)}{\sqrt{r^2 + p^2}} & \frac{r \cos(t)}{\sqrt{r^2 + p^2}} & \frac{p}{\sqrt{r^2 + p^2}} \\ -\cos(t) & \sin(t) & 0 \\ \frac{p \sin(t)}{\sqrt{r^2 + p^2}} & -\frac{p \cos(t)}{\sqrt{r^2 + p^2}} & \frac{r}{\sqrt{r^2 + p^2}} \end{bmatrix} \quad (4.27.a)$$

Matrice de transformation Global-Local :

$$\mathbf{GL} = \begin{bmatrix} -\frac{r \sin(t)}{\sqrt{r^2 + p^2}} & -\cos(t) & \frac{p \sin(t)}{\sqrt{r^2 + p^2}} \\ \frac{r \cos(t)}{\sqrt{r^2 + p^2}} & -\sin(t) & -\frac{p \cos(t)}{\sqrt{r^2 + p^2}} \\ \frac{p}{\sqrt{r^2 + p^2}} & 0 & \frac{r}{\sqrt{r^2 + p^2}} \end{bmatrix} \quad (4.27.b)$$

Vecteur courbure-torsion global :

$$\mathbf{K}_g = \left\{ -\frac{r \cos(t)}{r^2 + p^2}, -\frac{r \sin(t)}{r^2 + p^2}, 0 \right\}^T \quad (4.28)$$

Vecteur courbure- torsion local :

$$\mathbf{K}_l = \left\{ 0, \frac{r}{r^2 + p^2}, \frac{p}{\sqrt{r^2 + p^2}} \right\}^T \quad (4.29.a)$$

Nous considérerons pour alléger les calculs symboliques dans les programmes établis l'expression générale :

$$\mathbf{K}_I = \{0, K_Y, K_Z\}^T \tag{4.29.b}$$

Considérant la rotation  $\alpha$  pour la section on obtient pour le vecteur courbure-torsion local:

$$\mathbf{K}_I = \left\{ 0, -\frac{\cos(\alpha) + \cos(\alpha) \cos(L)^2}{r}, 0 \right\}^T \tag{4.29.c}$$

#### 4-1-4 Cas Droit

*Paramétrisation :*

$$X = at + X_0 \qquad Y = bt + Y_0 \qquad Z = ct + Z_0 \tag{4.30}$$

Longueur de la poutre  $L = \sqrt{a^2 + b^2 + c^2}$

On définit les cosinus directeurs :

$$C_x = \frac{a}{L} \qquad C_y = \frac{b}{L} \qquad C_z = \frac{c}{L} \tag{4.31}$$

$$\text{Posant } C = \sqrt{C_x^2 + C_y^2}$$

Nous avons les matrices de rotations :

$$\mathbf{R}_Y = \begin{bmatrix} C & \frac{b}{L} & 0 \\ -\frac{b}{L} & C & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.32.a}$$

$$R_{\beta} = \begin{bmatrix} a & 0 & c \\ LC & 0 & LC \\ 0 & 1 & 0 \\ -c & 0 & a \\ -LC & 0 & LC \end{bmatrix} \quad (4.32.b)$$

$$R_{\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (4.32.c)$$

Soit la matrice de transformation totale :  $R = R_{\alpha} R_{\beta} R_{\gamma}$

$$R = \begin{bmatrix} a & b & c \\ L & L & L \\ -\cos(\alpha)ba - \sin(\alpha)c & \cos(\alpha)C & -\cos(\alpha)bc + \sin(\alpha)a \\ L^2C & LC & L^2C + LC \\ \sin(\alpha)ba - \cos(\alpha)c & -\sin(\alpha)C & \sin(\alpha)bc + \cos(\alpha)a \\ L^2C & LC & L^2C + LC \end{bmatrix} \quad (4.33)$$

Cette matrice de transformation est valide pour toutes les positions sauf pour le cas d'une poutre spatiale verticale où elle devient :

$$R_{\text{vert}} = \begin{bmatrix} 0 & C_y & 0 \\ -C_y & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.33)$$

On peut retrouver le résultat caractéristique pour  $\alpha = 0$  soit :

$$\mathbf{R}_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.34)$$

$$\mathbf{R} = \begin{bmatrix} \frac{a}{L} & \frac{c}{L} & \frac{c}{L} \\ -\frac{c a}{L^2 C} & C & -\frac{c^2}{L^2 C} \\ -\frac{c}{L C} & 0 & \frac{a}{L C} \end{bmatrix} \quad (4.35)$$

*Abscisse curviligne :*

on choisit une origine telle que  $S(t=0) = S_0 = 0$  soit :

$$ds = L dt$$

$$S = L t$$

$$X = a \frac{S}{L} + X_0 = C_X S + X_0$$

$$Y = b \frac{S}{L} + Y_0 = C_Y S + Y_0$$

$$Z = c \frac{S}{L} + Z_0 = C_Z S + Z_0$$

En ramenant le premier nœud (1) à l'origine du repère global on a :

$$X_0 = 0 \quad Y_0 = 0 \quad Z_0 = 0$$

*Base du repère local de Fresnet:*

$\{e_1, e_2, e_3\}$  sont respectivement les vecteurs ligne de la matrice de transformation géométrique  $\mathbf{R}$  soient :

$$\mathbf{e}_1 = \left\{ \frac{a}{L}, \frac{b}{L}, \frac{c}{L} \right\}^T$$

$$\mathbf{e2} = \left\{ -\frac{\cos(\alpha) b a}{L^2 C} - \frac{\sin(\alpha) c}{L C}, \cos(\alpha) C, -\frac{\cos(\alpha) b c}{L^2 C} + \frac{\sin(\alpha) a}{L C} \right\}^T$$

$$\mathbf{e3} = \left\{ \frac{\sin(\alpha) b a}{L^2 C} - \frac{\cos(\alpha) c}{L C}, -\sin(\alpha) C, \frac{\sin(\alpha) b c}{L^2 C} + \frac{\cos(\alpha) a}{L C} \right\}^T$$

Pour le cas  $\alpha = 0$  nous obtenons :

$$\mathbf{e1} = \left\{ \frac{a}{L}, \frac{b}{L}, \frac{c}{L} \right\}^T$$

$$\mathbf{e2} = \left\{ -\frac{c a}{L^2 C}, C, -\frac{c^2}{L^2 C} \right\}^T$$

$$\mathbf{e3} = \left\{ -\frac{c}{L C}, 0, \frac{a}{L C} \right\}^T$$

## 4-2 Equations différentielles fondamentales du problème

Nous proposerons dans ce chapitre les sept équations différentielles pour chacun des trois cas de poutre.

Nous avons une matrice C liant les contraintes aux déformations pour les trois cas identique soit :

$$C = \begin{bmatrix} EA & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & EA & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & GA & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & EI & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & EI & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & EI & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & GJ & \frac{G(I - J)}{p} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{G(I - J)}{p} & GJ \end{bmatrix}$$

### 4-2-1 Cas Circulaire

L'opérateur matriciel L est donné par :

$$L = \begin{bmatrix} Ds & 0 & -\frac{1}{r} & 0 & 0 & 0 & 0 & 0 \\ 0 & Ds & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{r} & 0 & Ds & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Ds & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{r} & 0 & Ds & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Ds \\ 0 & 0 & 0 & Ds & 0 & -\frac{1}{r} & 0 & 0 \\ 0 & 0 & 0 & Ds & 0 & -\frac{1}{r} & -Ds & 0 \end{bmatrix}$$

Les équations différentielles se posent en  $e_{qi} = 0 \quad i = 1.. 7$  avec :

eq1 :=

$$E A \left( \frac{\partial^2}{\partial s^2} u_g(s) \right) - \frac{E A \left( \frac{\partial}{\partial s} w_g(s) \right)}{r} + \frac{G A \left( \frac{\partial}{\partial s} w_g(s) \right)}{r} + \frac{G A u_g(s)}{r^2} - \frac{G A \theta_y(s)}{r} + f b x$$

$$eq2 := E A \left( \frac{\partial^2}{\partial s^2} v_g(s) \right) + E A \left( \frac{\partial}{\partial s} \theta_z(s) \right) + f b y$$

$$eq3 := - \frac{E A \left( \frac{\partial}{\partial s} u_g(s) \right)}{r} + \frac{E A w_g(s)}{r^2} + G A \left( \frac{\partial^2}{\partial s^2} w_g(s) \right) + \frac{G A \left( \frac{\partial}{\partial s} u_g(s) \right)}{r} - G A \left( \frac{\partial}{\partial s} \theta_y(s) \right) + f b z$$

$$eq4 := \frac{E I \theta_v(s)}{r^2} + \left( -2 \frac{G I}{r} + \frac{E I}{r} \right) \left( \frac{\partial}{\partial s} \theta_z(s) \right) - G I \left( \frac{\partial^2}{\partial s^2} \psi(s) \right) + 2 G I \left( \frac{\partial^2}{\partial s^2} \theta_x(s) \right) + m b x$$

$$eq5 := -G A \left( \frac{\partial}{\partial s} w_g(s) \right) - \frac{G A u_g(s)}{r} + G A \theta_v(s) + E I \left( \frac{\partial^2}{\partial s^2} \theta_v(s) \right) + m b y$$

$$eq6 := \left( 2 \frac{G I}{r^2} + E A \right) \theta_z(s) + \frac{G I \left( \frac{\partial}{\partial s} \psi(s) \right)}{r} + \left( -2 \frac{G I}{r} + \frac{E I}{r} \right) \left( \frac{\partial}{\partial s} \theta_v(s) \right) + E A \left( \frac{\partial}{\partial s} v_g(s) \right) + E I \left( \frac{\partial^2}{\partial s^2} \theta_z(s) \right) + m b z$$

$$eq7 := E I \left( \frac{\partial^2}{\partial s^2} \psi(s) \right) + \frac{G I \left( \frac{\partial}{\partial s} \theta_z(s) \right)}{r} - G I \left( \frac{\partial^2}{\partial s^2} \theta_v(s) \right) + G I \left( \frac{\partial^2}{\partial s^2} \psi(s) \right) + m b w$$

#### 4-2-1 Cas Hélicoïdal ( avec devers $K_x = 0$ )

L'opérateur matriciel différentiel L est donné par :

$$L = \begin{bmatrix} D_s & K_z & -K_y & 0 & 0 & 0 & 0 \\ -K_z & D_s & 0 & 0 & 0 & 1 & 0 \\ K_y & 0 & D_s & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -K_z & D_s & 0 & 0 \\ 0 & 0 & 0 & K_y & 0 & D_s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & D_s \\ 0 & 0 & 0 & D_s & K_z & -K_y & 0 \\ 0 & 0 & 0 & D_s & K_z & -K_y & -D_s \end{bmatrix}$$

Equations différentielles:

$$\begin{aligned} eq1 : & -K_z E A \theta_v(s) + (K_y^2 G A + K_z^2 E A) u_g(s) + E A \left( \frac{\partial^2}{\partial s^2} u_g(s) \right) \\ & - E A K_y \left( \frac{\partial}{\partial s} u_g(s) \right) - K_y G A \theta_v(s) + f b x + K_y G A \left( \frac{\partial}{\partial s} w_g(s) \right) \end{aligned}$$

$$eq2 : -K_z^2 E A v_g(s) - K_z E A K_y w_g(s) + E A \left( \frac{\partial^2}{\partial s^2} v_g(s) \right) + E A \left( \frac{\partial}{\partial s} \theta_z(s) \right) + f b y$$

$$\begin{aligned} eq3 : & -K_y E A \left( \frac{\partial}{\partial s} u_g(s) \right) - K_y E A K_z v_g(s) + K_y^2 E A w_g(s) + G A \left( \frac{\partial^2}{\partial s^2} w_g(s) \right) \\ & + G A K_y \left( \frac{\partial}{\partial s} u_g(s) \right) - G A \left( \frac{\partial}{\partial s} \theta_v(s) \right) + f b z \end{aligned}$$

$$\begin{aligned} eq4 : & (K_y^2 E I_z + K_z^2 E I_v) \theta_v(s) - (2 G I_p K_z - K_z E I_v) \left( \frac{\partial}{\partial s} \theta_v(s) \right) \\ & + (K_y E I_z - 2 G I_p K_y) \left( \frac{\partial}{\partial s} \theta_z(s) \right) - G I_p \left( \frac{\partial^2}{\partial s^2} \psi(s) \right) + 2 G I_p \left( \frac{\partial^2}{\partial s^2} \theta_x(s) \right) + m b x \end{aligned}$$

$$\begin{aligned} eq5 : & -2 K_z G I_p K_y \theta_z(s) - G A K_y u_g(s) + (2 G I_p K_z^2 + G A) \theta_v(s) \\ & + (2 G I_p K_z - K_z E I_v) \left( \frac{\partial}{\partial s} \theta_v(s) \right) - G A \left( \frac{\partial}{\partial s} w_g(s) \right) + m b y - K_z G I_p \left( \frac{\partial}{\partial s} \psi(s) \right) \\ & + E I_v \left( \frac{\partial^2}{\partial s^2} \theta_y(s) \right) \end{aligned}$$

$$\begin{aligned}
 eq6 := & (E A - 2 G I_p K_y^2) \theta(s) - E A K_z u_z(s) - 2 K_y G I_p K_z \theta_1(s) \\
 & + (K_y E I_z - 2 G I_p K_y) \left( \frac{\partial}{\partial s} \theta_1(s) \right) + E A \left( \frac{\partial}{\partial s} v_g(s) \right) + m b z + E I_z \left( \frac{\partial^2}{\partial s^2} \theta_z(s) \right) \\
 & + K_y G I_p \frac{\partial}{\partial s} w(s)
 \end{aligned}$$

$$\begin{aligned}
 eq7 := & E I_p \left( \frac{\partial^2}{\partial s^2} w(s) \right) - G I_p K_z \left( \frac{\partial}{\partial s} \theta_1(s) \right) + G I_p K_y \left( \frac{\partial}{\partial s} \theta_2(s) \right) - G I_p \left( \frac{\partial^2}{\partial s^2} \theta_v(s) \right) \\
 & + G I_p \frac{\partial^2}{\partial s^2} w(s) - m b w
 \end{aligned}$$

#### 4-2-3 Cas Droit

L'opérateur L s'écrit :

$$L := \begin{bmatrix} Ds & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Ds & K_x & 0 & 0 & 1 & 0 \\ 0 & -K_x & Ds & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & Ds & K_x & 0 \\ 0 & 0 & 0 & 0 & -K_x & Ds & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Ds \\ 0 & 0 & 0 & Ds & 0 & 0 & 0 \\ 0 & 0 & 0 & Ds & 0 & 0 & 0 \end{bmatrix}$$

Les équations différentielles par :

$$eq1 := E A \left( \frac{\partial^2}{\partial s^2} u_g(s) \right) + f b x$$

$$\begin{aligned}
 eq2 := & E A \left( \frac{\partial^2}{\partial s^2} v_g(s) \right) + E A K_x \left( \frac{\partial}{\partial s} w_g(s) \right) + E A \left( \frac{\partial}{\partial s} \theta_z(s) \right) - G A K_x \left( \frac{\partial}{\partial s} w_g(s) \right) \\
 & + G A K_x^2 v_g(s) + G A K_x \theta_v(s) + f b y
 \end{aligned}$$

$$\begin{aligned}
 eq3 := & K_x E A \left( \frac{\partial}{\partial s} v_g(s) \right) + K_x^2 E A w_g(s) + K_x E A \theta_z(s) + G A \left( \frac{\partial^2}{\partial s^2} w_g(s) \right) \\
 & - G A K_x \left( \frac{\partial}{\partial s} v_g(s) \right) - G A \left( \frac{\partial}{\partial s} \theta_v(s) \right) + f b z
 \end{aligned}$$

$$eq4 := 2 G I_p \left( \frac{\partial^2}{\partial s^2} \theta_v(s) \right) - G I_p \left( \frac{\partial^2}{\partial s^2} \text{pzi}(s) \right)$$

$$eq5 := (Kx^2 E I_z + G A) \theta_1(s) + G A Kx v_y(s) + (-Kx E I_z + E I_y Kx) \left( \frac{\partial}{\partial s} \theta_z(s) \right) - G A \left( \frac{\partial}{\partial s} w_x(s) \right) + mby + E I_x \left( \frac{\partial^2}{\partial s^2} \theta_v(s) \right)$$

$$eq6 := (E A + E I_y Kx^2) \theta_2(s) + E A Kx w_x(s) + (-Kx E I_z + E I_y Kx) \left( \frac{\partial}{\partial s} \theta_v(s) \right) - E A \left( \frac{\partial}{\partial s} v_y(s) \right) + E I_z \left( \frac{\partial^2}{\partial s^2} \theta_z(s) \right) + mbz$$

$$eq7 := E I_w \left( \frac{\partial^2}{\partial s^2} \text{pzi}(s) \right) + mbw$$

### 4-3 Proposition des Résultats

#### 4-3-1 Cas circulaire

Nous résolvons le système différentiel avec l'unique composante du vecteur courbure torsion donnée par :  $K_y = K_{cy}$  et trouvons à cet effet les composantes du champ des déplacements :

Translation  $u_g(s)$  :

$$u_x(s) = -\frac{1}{2} \left( -2 C1 \cos(Kcy s) Kcy^2 G A E I_y - 2 C2 \sin(Kcy s) Kcy^2 G A E I_y + 2 \cos(Kcy s)^2 fbx E I_y - 2 \cos(Kcy s) I_y^2 C3 s Kcy^2 G E + 2 \cos(Kcy s) E^2 I_y^2 C3 s Kcy^2 - \frac{\sin(Kcy s) C3 \cos(Kcy s)^2 G A E I_y}{Kcy} + \sin(Kcy s) E^2 I_y^2 C3 \cos(Kcy s)^2 Kcy + 2 \sin(Kcy s)^2 fbx E I_y - \cos(Kcy s)^4 E^2 I_y^2 C4 Kcy + 2 \cos(Kcy s) I_y^2 C3 Kcy \left( \frac{1}{2} \sin(Kcy s) \cos(Kcy s) + \frac{1}{2} Kcy s \right) G E \right)$$









$$\begin{aligned}
 & - 4 I_p^2 G^4 J^2 - C10 \cos(Kcy \cdot s) Kcy^2 E - 4 I_p^3 G^4 J - C11 s \sin(Kcy \cdot s) Kcy^2 E \\
 & + 8 I_p^2 G^3 J - C11 s \sin(Kcy \cdot s) Kcy^2 E^2 I_p - 2 I_p^2 E^4 I_p^2 - C9 \sin(Kcy \cdot s) Kcy^2 G I_p \\
 & - 4 I_p E^3 I_p^2 - C9 \sin(Kcy \cdot s) Kcy^2 G^2 I_p J + G^3 E^2 J I_p^2 - C10 \cos(Kcy \cdot s) Kcy^2 I_p^2 \\
 & - 8 I_p^2 G^3 E^3 I_p Kcy - C11 \cos(Kcy \cdot s) + 8 I_p^2 I_p^2 G^2 E^3 I_p Kcy - C11 \cos(Kcy \cdot s) \\
 & + 4 I_p^2 I_p^2 G^2 E^3 I_p Kcy^2 - C11 s \sin(Kcy \cdot s) \\
 & + 4 I_p^2 I_p^2 G^2 E^3 I_p Kcy - C12 \cos(Kcy \cdot s) s + 8 I_p^3 I_p G^3 E^2 I_p Kcy - C12 \sin(Kcy \cdot s) \\
 & - 8 I_p^2 I_p^2 G^2 E^3 I_p Kcy - C12 \sin(Kcy \cdot s) - 4 I_p^3 G^3 I_p - C9 \sin(Kcy \cdot s) Kcy^2 E^2 I_p \\
 & + I_p^4 G^2 I_p - C10 \cos(Kcy \cdot s) Kcy^2 E - 4 I_p^3 G^3 I_p - C10 \cos(Kcy \cdot s) Kcy^2 E^2 I_p \\
 & + I_p^4 G^2 I_p - C11 s \sin(Kcy \cdot s) Kcy^2 E + I_p^4 E^3 I_p^2 - C11 s \sin(Kcy \cdot s) Kcy^2 G^2 I_p^2 \\
 & - 2 I_p^2 E^2 I_p^2 - C11 s \sin(Kcy \cdot s) Kcy^2 G I_p \\
 & - 4 I_p^2 E^2 I_p^2 - C11 s \sin(Kcy \cdot s) Kcy^2 G^2 I_p J \\
 & + I_p^2 E^2 I_p^2 - C12 \cos(Kcy \cdot s) s Kcy^2 G^2 I_p^2 - 2 I_p^2 E^4 I_p^2 - C12 \cos(Kcy \cdot s) s Kcy^2 G I_p \\
 & - 4 I_p^2 I_p^2 I_p^2 - C12 \cos(Kcy \cdot s) s Kcy^2 G^2 I_p J + 2 I_p^4 G^4 E I_p Kcy - C11 \cos(Kcy \cdot s) \\
 & - I_p^4 G^2 I_p - C9 \sin(Kcy \cdot s) Kcy^2 E + 4 I_p^2 I_p^2 G^4 J^2 - C9 \sin(Kcy \cdot s) Kcy^2 E \\
 & - 8 I_p^2 G^2 E I_p Kcy J^2 - C12 \sin(Kcy \cdot s) + 4 I_p^2 I_p^2 G^2 E^3 I_p Kcy^2 - C9 \sin(Kcy \cdot s) \\
 & + 4 I_p^2 I_p^2 G^2 E^3 I_p Kcy^2 - C10 \cos(Kcy \cdot s) - 2 G^3 E^2 J^2 I_p^2 - C10 \cos(Kcy \cdot s) Kcy^2 I_p \\
 & + G^3 E^2 J I_p^2 - C11 s \sin(Kcy \cdot s) Kcy^2 I_p^2 - 2 G^3 E^2 J^2 I_p^2 - C11 s \sin(Kcy \cdot s) Kcy^2 I_p \\
 & + G^3 E^2 J I_p^2 - C12 \cos(Kcy \cdot s) s Kcy^2 I_p^2 - 2 G^3 E^2 J^2 I_p^2 - C12 \cos(Kcy \cdot s) s Kcy^2 I_p \\
 & - G^3 E^2 J I_p^2 - C9 \sin(Kcy \cdot s) Kcy^2 I_p^2 - 2 G^3 E^2 J^2 I_p^2 - C9 \sin(Kcy \cdot s) Kcy^2 I_p \\
 & + I_p^2 E^3 I_p^2 - C10 \cos(Kcy \cdot s) Kcy^2 G^2 I_p^2 - 2 I_p^4 G^4 E I_p Kcy - C12 \sin(Kcy \cdot s) \\
 & - 2 I_p^2 E^4 I_p^2 - C10 \cos(Kcy \cdot s) Kcy^2 G I_p \Big/ (E Kcy^3 G I_p (G I_p - 2 E I_p - 2 G J) \\
 & I_p (I_p E^2 I_p - 2 I_p G I_p E + G E I_p J - 2 G^2 I_p J + G^2 I_p^2))
 \end{aligned}$$

Rotation  $\theta_y(s)$  :

$$\begin{aligned}
 \theta_y(s) = & - \frac{1}{2} (2 - C3 \sin(Kcy \cdot s) E I_x + 2 - C4 \cos(Kcy \cdot s) E I_x + s^2 Kcy^2 mby + s^2 Kcy^2 jbx \\
 & - 2 - C5 s Kcy^2 E I_x - 2 - C6 Kcy^2 E I_x) \Big/ (Kcy^2 E I_x)
 \end{aligned}$$

Rotation  $\theta_z(s)$  :

$$\begin{aligned} \theta_z(s) = & - (G^2 I_p^2 \sin(Kcy s) / C11 + \cos(Kcy s) I_p^2 G^2 / C12 + G s J fby \\ & - 2 \cos(Kcy s) I_p G^2 J / C12 + C13 Kcy^2 G^2 I_p^2 + 2 C8 Kcy^2 G E I_p I_u \\ & - 2 C8 Kcy^2 G^2 I_p J - 2 C13 Kcy^2 G E I_p I_u - 2 C13 Kcy^2 G^2 I_p J \\ & - C8 Kcy^2 G^2 I_p^2 + s I_p G mbw Kcy + s E I_w fby - 2 C12 G E I_p \cos(Kcy s) I_u \\ & - C12 G^2 I_p^2 \sin(Kcy s) Kcy s - 2 C12 G^2 I_p \sin(Kcy s) Kcy s J \\ & - 2 C10 \sin(Kcy s) Kcy G^2 I_p J - 2 C12 G E I_p \sin(Kcy s) Kcy s I_u \\ & + 2 C9 \cos(Kcy s) Kcy G E I_p I_u - 2 C11 G E I_p \sin(Kcy s) I_u \\ & - 2 C11 G E I_p Kcy s \cos(Kcy s) I_u - C11 G^2 I_p^2 Kcy s \cos(Kcy s) \\ & - 2 C9 \cos(Kcy s) Kcy G^2 I_p J - C9 \cos(Kcy s) Kcy G^2 I_p^2 \\ & - 2 C10 \sin(Kcy s) Kcy G E I_p I_u + 2 C11 G^2 I_p Kcy s \cos(Kcy s) J \\ & - C10 \sin(Kcy s) Kcy G^2 I_p^2 - 2 G^2 I_p J \sin(Kcy s) / C11) / ( \\ & Kcy^2 G I_p (G I_p - 2 E I_u - 2 G J)) \end{aligned}$$

Rotation  $\psi(s)$  :

$$\begin{aligned} \psi(s) = & \left[ C^7 Kcy G^4 I_p^4 + C^7 Kcy G^2 I_p^2 I_u E^2 I_z + C^7 Kcy G^3 I_p^2 E I_z J \right. \\ & - 4 G^3 E A J^2 / C8 s I_p + 2 Kcy^2 G^2 I_p^2 I_u I_z E^2 / C13 s + 8 Kcy^2 G^4 I_p^4 J / C8 s \\ & + 4 Kcy^2 I_w^2 I_z E^3 / C8 s G I_p - 8 G^2 mbz J E I_w s I_p + 8 Kcy^2 G^2 I_p^2 I_u^2 E^2 / C13 s \\ & + 8 Kcy^2 G^2 I_z E^2 J / C8 I_u s I_p + 2 G^2 I_z E mbz J^2 s - 8 Kcy^2 G^2 I_z E^2 J / C13 I_u s I_p \\ & - 16 Kcy^2 G^3 I_p^2 J / C8 E I_u s + 2 Kcy^2 G^4 I_p^4 / C13 s + 2 I_u^2 I_z E^3 mbz s \\ & - 8 Kcy^2 G^4 I_p^4 J / C13 s - 2 C^7 Kcy G I_p E^3 I_u^2 I_z - 2 Kcy^2 G^4 I_p^4 / C8 s \\ & - 2 C^7 Kcy G^3 I_p^2 E I_z J + 8 C^7 Kcy G^3 I_p^2 E I_u J + 4 C^7 Kcy G^2 I_p^2 E^2 I_w^2 \\ & - 4 C^7 Kcy G^2 I_p E^2 I_u I_z J + 4 C^7 Kcy G^4 I_p^2 J^2 - 4 C^7 Kcy G^3 I_p^3 I_u E \\ & - 4 C^7 Kcy G^4 I_p^3 J - 4 Kcy^2 G^3 I_z E J^2 / C13 s I_p + 2 G^2 I_z E^2 A J^2 / C8 s \\ & + 4 G^2 I_p^2 mbz E I_u s + 4 G^3 I_p^2 mbz J s + 16 Kcy^2 G^3 I_p^2 J / C13 E I_u s \\ & \left. - 8 Kcy^2 G^2 I_p^2 I_u^2 E^2 / C8 s + 4 Kcy^2 G^3 I_z E J^2 / C8 s I_p \right] \end{aligned}$$

$$\begin{aligned}
 & - 2 K_{cy}^2 G^2 I_p^2 I_w E^2 C8s + 4 G^2 I_p^2 E^2 A C8I_s - 2 K_{cy}^2 G^3 I_p^2 I_w E J C8s \\
 & - 4 I_w^2 E^2 mbz s G I_p + 4 I_w^2 E^2 mbz J I_w s G - 8 G^2 E^2 A J C8I_s I_p \\
 & + 8 K_{cy}^2 G^4 I_p^2 J^2 C13s + 2 K_{cy}^2 G^3 I_p^2 I_w E J C13s - 4 E^2 A I_w^2 C8s G I_p \\
 & - 4 G^2 mbz J^2 s I_p - 8 K_{cy}^2 G^3 I_p^3 C8E I_w s - 8 K_{cy}^2 G^4 I_p^2 J^2 C8s \\
 & + G^2 I_w E J mbw s^2 K_{cy} I_p \frac{2 G^3 I_p^3 I_w E C11 \cos(Kcy s)}{Kcy} \\
 & + \frac{4 G^2 I_p^2 I_w E J C11 \cos(Kcy s)}{Kcy} + \frac{1}{2} G^2 E I_w J fby s^2 I_p - I_w I_w E^3 A C8s G I_p \\
 & - G^3 I_p^2 E A C8s - 4 K_{cy}^2 I_w^2 I_w E^3 C13s G I_p + 4 G^2 I_p^2 E A C8I_s \\
 & + 2 I_w^2 I_w E^4 A C8s - G^2 I_w E mbz J s I_p - G^2 I_w E^2 A J C8s I_p - I_w I_w E^2 mbz s G I_p \\
 & - G^2 I_p^2 I_w E fby s^2 + G^3 I_p^3 mbw s^2 K_{cy} + \frac{4 G^2 I_p^2 I_w I_w E^2 C11 \cos(Kcy s)}{Kcy} \\
 & - \frac{4 G^2 I_p^2 I_w E J C12 \sin(Kcy s)}{Kcy} + \frac{1}{2} G^3 I_p^2 fby s^2 - 8 K_{cy}^2 G^3 I_p^3 C13 E I_w s \\
 & + 4 I_w I_w E^3 A C8I_s G + \frac{1}{2} I_w E^2 I_w fby s^2 G I_p - 2 G^3 I_p^2 J mbw s^2 K_{cy} \\
 & - G^3 I_p^2 J fby s^2 - \frac{4 G^2 I_p^2 I_w I_w E^2 C12 \sin(Kcy s)}{Kcy} + \frac{2 G^2 I_p^3 I_w E C12 \sin(Kcy s)}{Kcy} \\
 & + I_w I_w E^2 mbw s^2 K_{cy} G I_p - 2 G^2 I_p^2 I_w E mbw s^2 K_{cy} - G^3 I_p^3 mbz s \Bigg) / (Kcy G I_p \\
 & (G I_p - 2 E I_w - 2 G J)(I_w E^2 I_w - 2 I_w G I_p E + G E I_w J - 2 G^2 I_p J + G^2 I_p^2))
 \end{aligned}$$

### 4-3-2 Cas Hélicoïdal

Pour ce cas particulier, la compilation du système différentiel fournit le résultat :

« Systeme run out of memor ».

Les deux composantes du vecteur courbure torsion alourdissent significativement les calculs symboliques. En conséquence, nous ne saurions proposer de solutions analytiques au système.

### 4-3-3 Cas droit

Les solutions les plus simples ont été obtenues avec ce dernier soient :

Translation  $u_g(s)$  :

$$u_g(s) = -\frac{1}{2} \frac{fbx s^2 - 2_{C13} s EA - 2_{C14} EA}{EA}$$

Translation  $v_g(s)$  :

$$\begin{aligned} v_g(s) = & (-fb) \frac{EI}{z} Kx^2 - fb) \frac{GA}{z} + G \frac{Kx}{A} mb) + (C1 \sin(Kx s) Kx^4 \frac{GI}{z} AE \\ & + (C2 \cos(Kx s) Kx^4 \frac{GI}{z} AE + (C3 \sin(Kx s) s Kx^4 \frac{GI}{z} AE \\ & + (C4 \sin(Kx s) s^2 Kx^4 \frac{GI}{z} AE + (C5 \sin(Kx s) s^3 Kx^4 \frac{GI}{z} AE \\ & + (C6 \cos(Kx s) s Kx^4 \frac{GI}{z} AE + (C7 \cos(Kx s) s^2 Kx^4 \frac{GI}{z} AE \\ & + (C8 \cos(Kx s) s^3 Kx^4 \frac{GI}{z} AE) / (Kx^4 \frac{GI}{z} AE) \end{aligned}$$

Translation  $w_g(s)$  :

$$\begin{aligned} w_g(s) = & (3 Kx I_v^2 AE G - C5 \sin(Kx s) + Kx I_v AE mb) z G + fb) z AG I_v - fb) z AG I_v \\ & - Kx A mb) z GI_v - Kx^2 I_v^2 AE G - C4 \cos(Kx s) + 6 Kx^3 I_v^2 I_v E G - C5 \sin(Kx s) \\ & - 3 Kx^2 I_v^2 AE G - C5 \cos(Kx s) s + 3 Kx I_v^2 AE G - C8 \cos(Kx s) \\ & + 3 Kx^2 I_v^2 AE G - C8 \sin(Kx s) s + Kx^2 I_v^2 AE G - C7 \sin(Kx s) \\ & - Kx^2 I_v AE G I_v - C4 \cos(Kx s) + 3 Kx I_v AE G I_v - C5 \sin(Kx s) \\ & + 3 Kx I_v AE G I_v - C8 \cos(Kx s) + Kx^2 I_v AE G I_v - C7 \sin(Kx s) \\ & + 6 Kx^3 I_v^2 I_v E G - C8 \cos(Kx s) + 6 Kx^3 I_v^2 I_v E^2 - C8 \cos(Kx s) \\ & + 6 Kx^3 I_v^2 I_v E^2 - C5 \sin(Kx s) - 3 Kx^2 I_v AE G I_v - C5 \cos(Kx s) s \\ & + 3 Kx^2 I_v AE G I_v - C8 \sin(Kx s) s - Kx^2 fb) z GI_v^2 + Kx^2 fb) z GI_v I_v \\ & + Kx^4 I_v^2 AE G - C2 \sin(Kx s) + Kx^4 I_v^2 AE G - C6 \sin(Kx s) s \\ & + Kx^4 I_v^2 AE G - C7 \sin(Kx s) s^2 - Kx^4 E A GI_v I_v - C6 \sin(Kx s) s \\ & - Kx^4 I_v^2 AE G - C4 \cos(Kx s) s^2 - Kx^4 I_v^2 AE G - C3 \cos(Kx s) s \end{aligned}$$

$$\begin{aligned}
 & - Kx^4 \frac{E A G I}{I} \frac{C7 \sin(Kx s)}{I} s^2 - Kx^4 \frac{E A G I}{I} \frac{C8 \sin(Kx s)}{I} s^3 \\
 & + Kx^4 \frac{E A G I}{I} \frac{C1 \cos(Kx s)}{I} - Kx^4 \frac{E A G I}{I} \frac{C2 \sin(Kx s)}{I} \\
 & + Kx^4 \frac{E A G I}{I} \frac{C3 \cos(Kx s)}{I} s + Kx^4 \frac{E A G I}{I} \frac{C4 \cos(Kx s)}{I} s^2 \\
 & + Kx^4 \frac{E A G I}{I} \frac{C5 \cos(Kx s)}{I} s^3 - Kx^4 \frac{I^2}{I} \frac{A E G}{I} \frac{C5 \cos(Kx s)}{I} s^3 \\
 & + Kx^4 \frac{I^2}{I} \frac{A E G}{I} \frac{C8 \sin(Kx s)}{I} s^3 - Kx^4 \frac{I^2}{I} \frac{A E G}{I} \frac{C1 \cos(Kx s)}{I} \Big/ ( \\
 & Kx^4 \frac{A E G I}{I} (I - I))
 \end{aligned}$$

Rotation suivant x :

$$\theta_x(s) = -\frac{1}{4} \frac{mbw s^2 - 4 \frac{C9 s E I}{I} - 4 \frac{C10 E I}{I}}{E I}$$

Rotation suivant y :

$$\begin{aligned}
 \theta_y(s) = & (-fby \frac{A G I}{I} - Kx \frac{A mby G I}{I} + Kx \frac{A mby G I}{I} + I \frac{fby A G I}{I} \\
 & - 6 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{E G}{I} \frac{C8 \sin(Kx s)}{I} + Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C3 \cos(Kx s)}{I} \\
 & + 6 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{E^2}{I} \frac{C8 \sin(Kx s)}{I} - 6 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{E^2}{I} \frac{C5 \cos(Kx s)}{I} \\
 & + 3 Kx^2 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C5 \sin(Kx s)}{I} s + 3 A Kx^2 \frac{E G I I}{I} \frac{C5 \sin(Kx s)}{I} s \\
 & - A Kx^3 \frac{E G I I}{I} \frac{C3 \cos(Kx s)}{I} - 3 A Kx^3 \frac{E G I I}{I} \frac{C5 \cos(Kx s)}{I} s^2 \\
 & - 2 A Kx^3 \frac{E G I I}{I} \frac{C4 \cos(Kx s)}{I} s + A Kx^3 \frac{E G I I}{I} \frac{C6 \sin(Kx s)}{I} \\
 & + A Kx^2 \frac{E G I I}{I} \frac{C4 \sin(Kx s)}{I} + 3 A Kx^2 \frac{E G I I}{I} \frac{C8 \cos(Kx s)}{I} s \\
 & + 2 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C4 \cos(Kx s)}{I} s + Kx^2 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C4 \sin(Kx s)}{I} \\
 & + 3 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C5 \cos(Kx s)}{I} s^2 - 2 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C7 \sin(Kx s)}{I} s \\
 & + 3 Kx^2 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C8 \cos(Kx s)}{I} s + Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C7 \cos(Kx s)}{I} \\
 & - Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C6 \sin(Kx s)}{I} - 3 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{G A E}{I} \frac{C8 \sin(Kx s)}{I} s^2 \\
 & + 6 Kx^3 \frac{I}{I} \frac{I^2}{I} \frac{E G}{I} \frac{C5 \cos(Kx s)}{I} + 2 A Kx^3 \frac{E G I I}{I} \frac{C7 \sin(Kx s)}{I} s \\
 & + 3 A Kx^3 \frac{E G I I}{I} \frac{C8 \sin(Kx s)}{I} s^2 + A Kx^2 \frac{E G I I}{I} \frac{C7 \cos(Kx s)}{I} \Big/ ( \\
 & Kx^3 \frac{A E G I}{I} (I - I))
 \end{aligned}$$

Rotation suivant z :

$$\begin{aligned} \theta_z(s) = & -(-Kx^3 I_v A E G I_z C3 \sin(Kx s) + 2 Kx^3 I_v^2 A E G C4 \sin(Kx s) s \\ & + 3 Kx I_v^2 A E G C5 \sin(Kx s) + Kx I_v A mbz G + fbz A G I_z - fbz A G I_v \\ & - Kx A mbz G I - 2 Kx^3 I_v A E G I_z C4 \sin(Kx s) s \\ & - 3 Kx^3 I_v A E G I_z C5 \sin(Kx s) s^2 - Kx^2 I_v^2 A E G C4 \cos(Kx s) \\ & - 6 Kx^3 I_v^2 I E G C5 \sin(Kx s) - 3 Kx^2 I_v^2 A E G C5 \cos(Kx s) s \\ & + 3 Kx^3 I_v^2 A E G C5 \sin(Kx s) s^2 + Kx^3 I_v^2 A E G C3 \sin(Kx s) \\ & + 3 Kx I_v^2 A E G C8 \cos(Kx s) + 3 Kx^2 I_v^2 A E G C8 \sin(Kx s) s \\ & + 2 Kx^3 I_v^2 A E G C7 \cos(Kx s) s + 3 Kx^3 I_v^2 A E G C8 \cos(Kx s) s^2 \\ & + Kx^3 I_v^2 A E G C6 \cos(Kx s) + Kx^2 I_v^2 A E G C7 \sin(Kx s) \\ & - Kx^2 I_v A E G I_z C4 \cos(Kx s) - Kx^3 I_v A E G I_z C6 \cos(Kx s) \\ & + 3 Kx I_v A E G I_z C5 \sin(Kx s) + 3 Kx I_v A E G I_z C8 \cos(Kx s) \\ & + Kx^2 I_v A E G I_z C7 \sin(Kx s) - 6 Kx^3 I_v^2 I E G C8 \cos(Kx s) \\ & + 6 Kx^3 I_v^2 I E^2 C8 \cos(Kx s) + 6 Kx^3 I_v^2 I E^2 C5 \sin(Kx s) \\ & - 3 Kx^2 I_v A E G I_z C5 \cos(Kx s) s + 3 Kx^2 I_v A E G I_z C8 \sin(Kx s) s \\ & - 3 Kx^3 I_v A E G I_z C8 \cos(Kx s) s^2 - 2 Kx^3 I_v A E G I_z C7 \cos(Kx s) s) / ( \\ & Kx^3 E A G I_v (I_v - I_z)) \end{aligned}$$

$$\Psi(s) = -\frac{1}{2} \frac{mbw s^2 - 2 C11 s E I_w - 2 C12 E I_w}{E I_u}$$

CM :=

**La matrice C est donnée par :**

$$\begin{aligned}
 & [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1] \\
 & [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \\
 & \left[ \begin{array}{cccccccccccc}
 0, -1, 0, -\frac{I+I}{Kx^2(I-I)}, 0, 0, 0, 3 \frac{GIA+I GA+2GII Kx^2+2III E Kx^2}{Kx^3 GA(I-I)} \\
 0, 0, 0, 0, 0, 0, 0, 0 \\
 0, 0, 0, 0, 0, 0, 0, 0 \\
 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 \\
 0, 0, -1, 0, 6 \frac{II(G-E)}{(I-I)AG}, 0, \frac{I+I}{Kx(I-I)}, 0, 0, 0, 0, 0, 0, 0, 0 \\
 0, 0, 0, \frac{I+I}{Kx(I-I)}, 0, -1, 0, 3 \frac{-GIA-I GA+2GII Kx^2-2III E Kx^2}{Kx^2(I-I)GA} \\
 0, 0, 0, 0, 0, 0, 0, 0 \\
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 \\
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1 \\
 \cos(KxL), \sin(KxL), \sin(KxL)L, \sin(KxL)L^2, \sin(KxL)L^3, L \cos(KxL), \\
 \cos(KxL)L^2, \cos(KxL)L^3, 0, 0, 0, 0, 0, 0, 0 \\
 \sin(KxL), -\cos(KxL), -L \cos(KxL), \\
 \frac{\cos(KxL)(-I Kx^2 L^2 + I Kx^2 L^2 + I + I)}{Kx^2(I-I)}, -(I GA Kx^3 \cos(KxL)L^2 \\
 - 3 GIA \sin(KxL) - 6 GII Kx^2 \sin(KxL) + 3 GIA Kx \cos(KxL)L \\
 + 3 \sin(KxL)I GA + 3 GIA Kx \cos(KxL)L - GIA Kx^3 \cos(KxL)L^2 \\
 - 6 III E Kx^2 \sin(KxL)) / (Kx^3 GA(I-I)), \sin(KxL)L, \\
 \frac{\sin(KxL)(-I Kx^2 L^2 + I Kx^2 L^2 + I + I)}{Kx^2(I-I)}, (3 GIA Kx \sin(KxL)L \\
 + 3 GIA \cos(KxL) + 3 GIA Kx \sin(KxL)L + I GA Kx^3 \sin(KxL)L^3 \\
 + 6 III E Kx^2 \cos(KxL) + 6 GII Kx^2 \cos(KxL) + 3 I GA \cos(KxL) \\
 - GIA Kx^3 \sin(KxL)L^3) / (Kx^3 GA(I-I)), 0, 0, 0, 0, 0, 0, 0
 \end{array} \right]
 \end{aligned}$$

$$[0, 0, 0, 0, 0, 0, 0, 0, L, 1, 0, 0, 0, 0]$$

$$\left[ \begin{array}{l} 0, 0, -\cos(Kx L), \\ \frac{2 I Kx \cos(Kx L) L - I \sin(Kx L) - 2 I Kx \cos(Kx L) L - I \sin(Kx L)}{Kx (I - I)} \end{array} \right] \dots -3 ($$

$$\begin{aligned} & -2 G I I Kx \cos(Kx L) - G I Kx A \cos(Kx L) L^2 + G I A Kx \cos(Kx L) L^2 \\ & + 2 I I E Kx \cos(Kx L) - G I A \sin(Kx L) L - G I A \sin(Kx L) L \Big/ ( \\ & Kx (I - I) A G), \sin(Kx L), \end{aligned}$$

$$\frac{2 I Kx \sin(Kx L) L + I \cos(Kx L) + I \cos(Kx L) - 2 I Kx \sin(Kx L) L}{Kx (I - I)} \dots 3 ($$

$$\begin{aligned} & G I A \cos(Kx L) L + G I A Kx \sin(Kx L) L^2 - 2 G I I Kx \sin(Kx L) \\ & - G I Kx A \sin(Kx L) L^2 + G I A \cos(Kx L) L + 2 I I E Kx \sin(Kx L) \Big/ ( \\ & Kx (I - I) A G), 0, 0, 0, 0, 0, 0 \end{aligned}$$

$$Kx (I - I) A G), 0, 0, 0, 0, 0, 0]$$

$$\left[ \begin{array}{l} 0, 0, -\sin(Kx L), \\ \frac{-2 I Kx \sin(Kx L) L - I \cos(Kx L) - I \cos(Kx L) + 2 I Kx \sin(Kx L) L}{Kx (I - I)} \end{array} \right] \dots -3 ($$

$$\begin{aligned} & G I A \sin(Kx L) - 2 G I I Kx^2 \sin(Kx L) - G I A Kx \cos(Kx L) L \\ & + \sin(Kx L) I G A - G I A Kx \cos(Kx L) L + 2 I I E Kx^2 \sin(Kx L) \\ & - G I A Kx^2 \sin(Kx L) L^2 + G I A Kx^2 \sin(Kx L) L^2 \Big/ (Kx^2 (I - I) G A), \\ & -\cos(Kx L), \end{aligned}$$

$$\frac{2 I Kx \cos(Kx L) L - 2 I Kx \cos(Kx L) L + I \sin(Kx L) + I \sin(Kx L)}{Kx (I - I)} \dots -3 ($$

$$\begin{aligned} & G I A Kx \sin(Kx L) L - G I A Kx^2 \cos(Kx L) L^2 + G I A Kx^2 \cos(Kx L) L^2 \\ & + 2 I I E Kx^2 \cos(Kx L) - 2 G I I Kx^2 \cos(Kx L) + G I A Kx \sin(Kx L) L \\ & + G I A \cos(Kx L) + I G A \cos(Kx L) \Big/ (Kx^2 (I - I) A G), 0, 0, 0, 0, 0, 0 \end{aligned}$$

$$[0, 0, 0, 0, 0, 0, 0, 0, 0, L, 1, 0, 0, 0]$$





#### 4-5 Formulation Variationnelle pour la Résolution Numérique

Le calcul des déplacements est fondamental en structure. En effet, dans certaines structures, la limitation des valeurs maximales des déplacements est plus importante que la limitation des valeurs maximales des contraintes.

Il existe plusieurs méthodes de calcul des déplacements notamment celle par double intégration utilisée communément en résistance des matériaux.

Nous utiliserons dans ce chapitre une méthode approximative par éléments finis plus générale basée sur les principes et théorèmes énergétiques et qui est le principe de l'énergie potentielle ou de l'énergie de déformation. (Méthode de Bubnov- Galerkin)

#### Principe de l'Energie Potentielle

Lorsqu'un système de charges agit sur un corps, le travail  $W$  effectué par ces charges est transformé en énergie cinétique  $T$  et en énergie de déformation  $U$ .

L'énergie potentielle totale  $\pi_p$  est donc donnée par la relation :

$$\pi_p = U + W + T$$

L'énergie de déformation  $U$  est donnée par :

$$U = \int_{s1}^{s2} \frac{M_x^2}{2GJ} ds + \int_{s1}^{s2} \frac{M_y^2}{2EI_y} ds + \int_{s1}^{s2} \frac{M_z^2}{2EI_z} ds + \int_{s1}^{s2} \frac{M_w^2}{2EI_w} ds +$$

$$\int_{s1}^{s2} \frac{Q_x^2}{2EA} ds + \int_{s1}^{s2} \frac{Q_y^2}{2GA_y} ds + \int_{s1}^{s2} \frac{Q_z^2}{2GA_z} ds$$

$E$  : module d'élasticité longitudinal

$G$  : module d'élasticité transversal

$I_y$ ,  $I_z$  et  $I_w$  : moments principaux centraux d'inertie

$J$  : constante de torsion

On intègre les fonctions continues des efforts résultants ( Efforts normal, tranchants et moments ) en fonction de l'abscisse curviligne  $s$ . Nous avons par ailleurs le résultat fondamental suivant :

- Parmi toutes les configurations possibles pour déplacer un système tout en satisfaisant les conditions cinématiques et les conditions frontières, les champs de déplacements qui en plus vont satisfaire les équations d'équilibre, rendront l'énergie potentielle totale stationnaire. Si cette valeur stationnaire est minimum, l'équilibre est stable.

Nous nous baserons encore sur la méthode matricielle des éléments finis pour donner les expressions des grandeurs énergétiques.

On définit tout d'abord les matrices  $[A]$  et  $[I]$  pour les relations constitutives .

$$\mathbf{Q} = \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = [A] \begin{bmatrix} \varepsilon_1 \\ \varepsilon_{21} \\ \varepsilon_3 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} = [I] \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \kappa_3 \end{bmatrix}$$

$$[A] = \begin{bmatrix} EA & 0 & 0 \\ 0 & GA & 0 \\ 0 & 0 & GA \end{bmatrix}$$

$$[I] = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

L'énergie potentielle totale de la poutre est exprimée par :

$$\pi_p = \int_{s_1}^{s_2} \frac{1}{2} \{ (\varepsilon \cdot \mathbf{A} \cdot \varepsilon) + (\mathbf{k} \cdot \mathbf{l} \cdot \mathbf{k}) \} ds - \int_{s_1}^{s_2} \{ \mathbf{u} \cdot \mathbf{F}_b + \theta \cdot \mathbf{M}_b \} ds - [\mathbf{u} \cdot \mathbf{Q} + \theta \cdot \mathbf{M}]_{s_1}^{s_2}$$

En termes de  $\mathbf{u}$ ,  $\theta$ ,  $\mathbf{K}$ ,  $\mathbf{e1}$  nous avons:

$$\pi_p = \int_{s_1}^{s_2} \frac{1}{2} \{ (\mathbf{u}' + \mathbf{K} \times \mathbf{u} + \mathbf{e1} \times \theta) \cdot \mathbf{A} \cdot (\mathbf{u}' + \mathbf{K} \times \mathbf{u} + \mathbf{e1} \times \theta) + (\theta' + \mathbf{K} \times \theta) \cdot \mathbf{l} \cdot (\theta' + \mathbf{K} \times \theta) \} ds - \int_{s_1}^{s_2} \{ \mathbf{u} \cdot \mathbf{F}_b + \theta \cdot \mathbf{M}_b \} ds - [\mathbf{u} \cdot \mathbf{Q} + \theta \cdot \mathbf{M}]_{s_1}^{s_2}$$

Le principe retenu est que l'énergie de déformation de la poutre reste constante de façon à ce que  $\delta U = 0$ .

En évaluant cette fonctionnelle et en remarquant que :

$$\begin{aligned} \mathbf{A} \cdot (\mathbf{u}' + \mathbf{K} \times \mathbf{u} + \mathbf{e1} \times \theta) &= \mathbf{Q} \\ \mathbf{l} \cdot (\theta' + \mathbf{K} \times \theta) &= \mathbf{M} \end{aligned}$$

Nous obtenons le résultat final:

$$\int_{s_1}^{s_2} \{ \delta \mathbf{u} \cdot (\mathbf{Q}' + \mathbf{K} \times \mathbf{Q} + \bar{\mathbf{F}}) + \delta \theta \cdot (\mathbf{M}' + \mathbf{K} \times \mathbf{M} + \bar{\mathbf{M}} + \mathbf{e1} \times \theta) \} ds - [\delta \mathbf{u} \cdot (\bar{\mathbf{Q}} - \mathbf{Q}) + \delta \theta \cdot (\bar{\mathbf{M}} - \mathbf{M})]_{s_1}^{s_2} = 0$$

Il apparaît dès lors que les conditions extremum de l'énergie potentielle totale sont obtenues aux nœuds d'appui de l'élément.

Cette deuxième méthode variationnelle est encore sujette à un verrouillage numérique

Ce dernier constitue une difficulté sérieuse pour toutes les formulations de structures minces. Quantité d'éléments finis de poutres sont proposés sur le marché des codes de calcul scientifique mais n'échappent pas pour la plupart à ce phénomène.

La signification communément admise semble être la dégradation de la qualité de l'approximation lorsque l'épaisseur de la structure diminue. C'est donc un phénomène relatif que l'on ne peut mettre en évidence qu'en comparant des comportements numériques pour différentes valeurs de l'épaisseur ( cas- tests ou « benchmarks »).

On effectuera les calculs sur le même cas-test pour plusieurs valeurs significativement différentes de ce paramètre.

#### 4-6 Formulation mixte à deux champs

Nous venons de voir que les deux méthodes de résolution présentées antérieurement posent quelques problèmes relatifs à la convergence des algorithmes de résolution ( limites du solveurs, verouillage numerique ).

A cet effet, on utilise dans ce chapitre un modèle de méthodes numériques « non verrouillantes » dont le principe variationnel est basé sur une fonctionnelle du type Hellinger / Reissner avec une formulation mixte à deux champs :

- un champ de déplacement approximé  $U$
- un champ de contrainte présupposé (assumed stress)  $\sigma$

On utilise donc cette méthode mixte comme intermédiaire conceptuel, pour développer une méthode numérique pour poutres courbes qui revient à utiliser des éléments finis de poutre droite de degré 1, en modifiant sélectivement certains termes de cisaillement et de membrane de la matrice de rigidité.

Si on définit :

$$\{ U \} = \{ u_g(s) , v_g(s) , w_g(s) , \theta_x(s) , \theta_y(s) , \theta_z(s) , \psi(s) \}^T$$

comme le vecteur des déplacements au point courant G centroïde de la section

$$\{d_1\} = \{u_1(s), v_1(s), w_1(s), \theta_{x1}(s), \theta_{y1}(s), \theta_{z1}(s), \psi_1(s)\}^T$$

le vecteur déplacement au premier nœud

$$\{d_2\} = \{u_2(s), v_2(s), w_2(s), \theta_{x2}(s), \theta_{y2}(s), \theta_{z2}(s), \psi_2(s)\}^T$$

le vecteur déplacement au deuxième nœud

On peut écrire la relation déformations / déplacements soit :

$$\mathcal{E} = \mathbf{L}\mathbf{U}$$

On peut interpoler linéairement U entre  $\{d_1\}$  et  $\{d_2\}$  soit :

(Interpolation linéaire de Lagrange)

$$\mathbf{U} = \mathbf{N}_u \mathbf{d}$$

avec:

$$\mathbf{N}_u = \{N_{u1}, N_{u2}\}^T$$

$$N_{u1} = \frac{1}{2}(1-s)$$

$$-1 \leq s \leq 1$$

$$N_{u2} = \frac{1}{2}(1+s)$$

$$\mathbf{d} = \{\{d_1\}, \{d_2\}\}^T$$

On pré suppose le champ de contrainte soit:

$$\sigma = \mathbf{P} \beta$$

avec  $P = I$  matrice identité par exemple

Posons 
$$\hat{\mathcal{E}} = D^{-1} \sigma$$

Nous savons déjà que 
$$\sigma = C \hat{\mathcal{E}}$$

D'où 
$$C = D^{-1}$$

L'énergie potentielle s'écrit d'ailleurs :

$$\pi_p = \frac{1}{2} \int_{\Omega} \hat{\mathcal{E}}^T C \hat{\mathcal{E}} d\Omega + \int_{\Omega} \mathbf{U}^T (L^T \sigma + \mathbf{b}) d\Omega - \int_{\Gamma_u} \mathbf{U}^T (\mathbf{t} - \bar{\mathbf{t}}) d\Gamma$$

avec une première condition frontière (Dirichlet) :

$U = \bar{U}$  sur une partition  $\Gamma_u$  du domaine.

$$H \cdot \beta + G \cdot d = f_1$$

$$G^T \cdot \beta = f_2$$

$$H = - \int_{\Omega} P^T C P d\Omega$$

$$G = \int_{\Omega} P^T V d\Omega$$

$$V = L N_u$$

On trouve alors :

$$f_1 = 0$$

$$f_2 = \int_{\Omega} N_u^T \mathbf{b} d\Omega + \int_{\Gamma_t} N_u^T \bar{\mathbf{t}} d\Omega$$

## 5 RECOMMANDATIONS

Nous avons pu établir avec des programmes de calcul les résultats concernant les solutions analytiques brutes du problème différentiel des différents cas de poutres courbes (circulaire, hélicoïdale et droite).

Nous avons pu remarquer la non simplicité des solutions pour le cas circulaire et pour le cas hélicoïdal.

Le cas droit révèle quant à lui plus de maniabilité et nous avons pu régénérer les matrices [C] et [B] de même nous avons pu établir la matrice de rigidité [K] (14 x 14). Pour ce même cas, toutes les expressions analytiques des efforts, des déformations, des contraintes peuvent être proposées sans grande ambiguïté.

En particulier nous pouvons proposer les forces équivalentes aux nœuds qui sont les réactions d'appui nodales.

Dans les différents cas nous avons remarqué que les solutions du champ de déplacement s'expriment avec quatorze constantes d'intégration.

A cet effet, on peut d'emblée proposer une solution homogène en annulant à priori toutes ces constantes.

L'abscisse curviligne a été la variable de résolution du problème différentiel et c'est avec elle que nous avons établi les expressions des résultats

Le cas de la poutre droite spatiale se révèle donc comme une satisfaction du modèle analytique. Cependant les limites de notre solveur nous font préférer des méthodes numériques variationnelles.

Ces dernières pour le cas du Principe énergétique restent encore fortement limitées par le phénomène du verrouillage numérique.

La solution immédiate devient alors l'application de la formulation variationnelle mixte comme modèle non verrouillant de base



Il nous est apparu au vu de notre démarche que la résolution analytique complète du problème différentiel posé par les poutres courbes spatiales paraît difficile pour les différents cas dans la mesure où les calculs portent sur des équations non linéaires. Le modèle de la poutre droite spatiale constitue cependant une satisfaction de la démarche analytique. Ce résultat fort intéressant nous réconforte dans une démarche de résolution des autres cas en tenant compte de ce dernier.

Dans ce contexte, les méthodes numériques qui restent cependant approximatives se dessinent comme des solutions opportunes.

En particulier, la méthode des éléments finis avec la formulation variationnelle mixte basée sur la proposition de deux champs se propose comme notre modèle de rigueur.

En pratique les différents résultats établis devraient faire l'objet de tests numériques par programmes d'éléments finis en fortran et les résultats proposés en annexes. C'est par manque de temps que nous n'avons pas pu en arriver à ce stade

## 6 CONCLUSIONS

IL nous a été imparti de poser et de résoudre le problème différentiel fondamental de la statique des poutres courbes avec prise en compte des paramètres géométriques et des effets dus au gauchissement pour les trois cas circulaire, hélicoïdal et droit .

Les fonctions solutions obtenues de types polynomiales quadratiques et sinusoïdales pour les déplacements généralisés, laissent présager des possibilités de déverouillage au moyen de la résolution directe des équations différentielles.

Cette étude porte pour l'essentiel sur des sections à paroi minces ouvertes ou fermées où les effets liés à la torsion non uniforme de Vlassov ne sauraient être négligés.

Des résultats analytiques ont été proposés pour les trois cas non sans quelques difficultés liées à la complexité des modèles rencontrés ( équations non linéaires, matrices de grand ordre...) mais aussi aux limites du solveur utilisé ( version étudiante de Maple 6 ).

Dans l'environnement informatique soft and hard actuel, seule la résolution complète du cas de la poutre droite avec torsion géométrique (  $K_x, 0, 0$  ) et le cas courbe avec (  $0, K_y, 0$  ) sont possibles. La résolution analytique du cas généralisé (  $K_x, K_y, K_z$  ) est reportée.

Il faut noter toutefois pour le cas général qu'on peut utiliser une méthode basée sur l'expression du Principe des Travaux Virtuels ( PTV, méthode de Bouvnov- Galerkin ou encore méthode  $\pi_p$  ) avec intégration sélective ou une méthode basée sur une minimisation de la fonctionnelle de Hellinger-Reissner de type  $\pi_{hr}$  .

Le cas droit s'est révélé plus simple de résolution et nous a permis d'établir quelques résultats supplémentaires concernant par exemple la matrice de rigidité ou les forces équivalentes aux nœuds pour le cas d'une poutre chargée sur deux appuis.

IL demeure par conséquent évident que la formulation analytique encore étroitement liée aux mathématiques ne sauraient donner toute la satisfaction au modèle.

C'est pour cette raison que nous avons présenté toutes les relations qui caractérisent les structures de types poutres ( paramètres géométriques, relations cinématiques, d'équilibre, conditions frontières...), en même temps nous avons proposé des méthodes éléments finis ( principes énergétiques, formulation variationnelle mixte...) permettant d'obtenir des solutions numériques précises concernant l'étude des poutres tridimensionnelles droites et courbes à parois minces ouvertes ou fermées.

Enfin, ce projet s'inscrit dans d'une optique de recherche et les résultats présentés jusqu'ici mériteraient d'être étendus. En effet, notre domaine d'application reste bien entendu le calcul d'avant projet des ponts courbes à parois minces simples ou multicellulaires qui peuvent être modéliser par des poutres du même genre. L'étude actuelle se limite à des structures sur deux appuis. La résolution complète pour une structure discrétisée en éléments droits devant se faire par la solution de l'assemblage.

Sur le plan programmation vu les limites de notre solveur ( Maple 6 ), nous recommandons à qui voudrait bien continuer le projet de se doter tout d'abord de son propre matériel informatique, de s'armer de beaucoup de volonté et de patience en même temps de s'ouvrir à la programmation orientée objet pour une résolution plus à la mode des problèmes de statique des poutres courbes ou de travailler avec d'autres solveurs du meme genre que Maple comme Matlab ou Mathematica. Enfin le volet dynamique mériterait aussi d'être prospecté pour revoir en quels termes se pose encore le problème différentiel ainsi que sa résolution.

## REFERENCES BIBLIOGRAPHIQUES

- [ 1 ] : J.M. Gere, W.Jr. Weaver, Analysis of Framer Structures, D.Van Nostrand, N.Y , (1965)
- [ 2 ] : J.L. Batoz, DAHTT, Gouri, Modélisation des structures par elements finis, Vol.2, edition Hermer, (1990)
- [ 3 ] : B. Tabarrok, M. Farshad, Hi. Yi, Computer Methods in Applied Mechanics and Engineering, 70 (11980) 275-299, North Holland, (1980)
- [ 4 ] : J. Jirousek and M.Ndiaye, Solution of orthotropic plates based on p-extension of the hybrid-Trefftz finite element model. Computers and structures, Vol.34, N°1, (1990)
- [ 5 ] : J.Jirousek, International Journal for Numerical Methods in Engineering, Vol.17, 171-186,(1981)
- [ 6 ] : J.T. Oden, Mechanics of elastic structures, Mc GrawHill, (1967)
- [ 7 ] : R.D. Cook, Concepts and application of finite element analysis, second édition, USA, (1976)
- [ 8 ] : O.C. Zienkiewicz, The finite element Method, 3<sup>rd</sup> edition. London, Mc Graw-Hill, (1977)
- [ 9 ] : V. de Ville de Goyet, Etude du comportement à la ruine de structures spatiales composées de poutres à section non symetrique, Construction Metallique, n°4, (1987)

## **7 ANNEXES ( Listing de Programmes et Résultats )**

**7-1 Programme génération C et L**

**7-2 Programme calcul poutre circulaire**

**7-3 Programme calcul poutre hélicoïdale**

**7-4 Programme calcul poutre droite**

## 7-1 Programme génération C et L

Programme calcul de la matrice C (contraintes/deformations) et L (deformations/deplacements) :

```
# Chargement des fonctions de Maple :
with(student) :
with(linalg) :

# On definit les deplacements au centre de gravité comme :

Ug:=<u[g](s),v[g](s),w[g](s)>:

# et les rotations comme :
rots:=<theta[x](s), theta[y](s), theta[z](s)>:

# Les matrices de transitions r et e1 comme :

r:=matrix(3,3,0):
r[2,1]:=- (z[b](z)):
r[3,1]:= (y[b](y)):
r[1,2]:= z:
r[1,3]:= -y:
r:=evalm(r):

e1:=matrix(3,3,0):
e1[2,3]:= 1:
e1[3,2]:=-1:
e1:=evalm(e1):

# Le champ des deplacements U :

U := evalm(Ug + r &* rots - w(y,z)*psi(s)*<1,0,0>):

# La matrice de courbure-torsion :

K = matrix(3,3,0):
K[1,2] := Kctz:
K[1,3] := -Kcty:
K[2,1] := -Kctz:
K[2,3] := Kctx:
K[3,1] := Kcty:
K[3,2] := -Kctx:
K:=evalm(K):

rotsprime:=<diff(rots[1],s),diff(rots[2],s),diff(rots[3],s)>:
```

```
Ugprime:=<diff(Ug[1],s),diff(Ug[2],s),diff(Ug[3],s)>:
wyz := <0,diff(w(y,z),y),diff(w(y,z),z)>:
```

```
k := evalm(rotsprime + evalm(K &* rots)):
kappa[x] := k[1]:
kappa[y] := k[2]:
kappa[z] := k[3]:
kappa[w] := diff(psi(s),s):
```

### # Relations Cinematiques

```
epsilon := evalm(evalm(Ugprime)+evalm((K) &* Ug)+evalm((r) &* k)-
evalm( psi(s) *(wyz)) - evalm((w(y,z)) *diff( (psi(s)),s)* <1,0,0> +
evalm((e1) &* rots)):
```

```
eps[s] := epsilon[1]:
eps[sy] := epsilon[2]:
eps[sz] := epsilon[3]:
```

```
eps[s]:=f(y,z)*(eps0-y*kappaz+z*kappay-w(y,z)*kappaw):
eps[sy]:=f(y,z)*(gamasy-
(diff(w(y,z),y)+z[b](z))*gamasv+diff(w(y,z),y)*gamaw):
eps[sz]:=f(y,z)*(gamasz+(-
diff(w(y,z),z)+y[b](y))*gamasv+diff(w(y,z),z)*gamaw):
```

```
eps[0]:= diff(u[g](s),s) + Kct[z]*v[g](s) - Kct[y]*w[g](s) :
gama[sy]:= diff(v[g](s),s) - Kct[z]*u[g](s) + Kct[x]*w[g](s) +
theta[z](s) :
gama[sz]:= diff(w[g](s),s) + Kct[y]*u[g](s) - Kct[x]*v[g](s) -
theta[y](s) :
gama[sv]:= Kct[z]*theta[y](s) - Kct[y]*theta[z](s) +
diff(theta[x](s),s) :
gama[w]:=gama[sv]-kappa[w] :
```

### # Expressions des Contraintes :

```
sigma(s) :=E*eps[s]:
tau(sy) :=G*eps[sy]:
tau(sz) :=G*eps[sz]:
```

### # Initialisation des Moments d'inerties :

```
init:={expand(Doubleint(f(y,z),y,z,A))=A,expand(Doubleint(y*f(y,z),y,z,
A))=0,expand(Doubleint(f(y,z)*z,y,z,A))=0,expand(Doubleint(y[b](y)*f(y,
z),y,z,A))=-y[c]*A,expand(Doubleint(z[b](z)*f(y,z),y,z,A))=-
z[c]*A,expand(Doubleint(y*f(y,z)*z[b](z),y,z,A))=I[yz],expand(Doubleint(
y*f(y,z)*z,y,z,A))=I[yz],expand(Doubleint(y[b](y)*f(y,z)*z,y,z,A))=I[y
z],expand(Doubleint(f(y,z)*z[b](z)*z,y,z,A))=I[y],expand(Doubleint(f(y,
z)*z[b](z)^2,y,z,A))=I[y],expand(Doubleint(f(y,z)*z^2,y,z,A))=I[y],expa
```

```

nd(Doubleint(f(y,z)*y^2,y,z,A))=I[z],expand(Doubleint(y[b](y)^2*f(y,z),
y,z,A))=I[z],expand(Doubleint(y[b](y)*y*f(y,z),y,z,A))=I[z],expand(Doub
leint(z[b](z)*w(y,z)*f(y,z),y,z,A))=I[wz],expand(Doubleint(y[b](y)*w(y,
z)*f(y,z),y,z,A))=I[wy],expand(Doubleint(z*w(y,z)*f(y,z),y,z,A))=I[wz],
expand(Doubleint(y*w(y,z)*f(y,z),y,z,A))=I[wy],expand(Doubleint(f(y,z)*
((y[b](y)-
diff(w(y,z),z))^2+(z[b](z)+diff(w(y,z),y))^2),y,z,A))=J,(expand(Doublei
nt(f(y,z)*diff(w(y,z),z)^2,y,z,A))+expand(Doubleint(f(y,z)*diff(w(y,z),
y)^2,y,z,A)))=J-
I[p],expand(Doubleint(w(y,z)^2*f(y,z),y,z,A))=I[w],(I[y]+I[z])=I[p],exp
and(Doubleint(f(y,z)*((y[b](y)-
diff(w(y,z),z))^2+(z[b](z)+diff(w(y,z),y))^2),y,z,A))=J,expand(Doublèin
t(diff(w(y,z),y)*f(y,z),y,z,A))=0,expand(Doubleint(w(y,z)*f(y,z),y,z,A)
)=S[omega],expand(Doubleint(diff(w(y,z),z)*f(y,z),y,z,A))=0,expand(Doub
leint(y[b](y)*diff(w(y,z),z)*f(y,z),y,z,A))=0,expand(Doubleint(z[b](z)*
diff(w(y,z),y)*f(y,z),y,z,A))=0,(-
expand(Doubleint(f(y,z)*diff(w(y,z),z)^2,y,z,A))-
expand(Doubleint(f(y,z)*diff(w(y,z),y)^2,y,z,A))-I[y]-I[z])=-J,(-
expand(Doubleint(f(y,z)*diff(w(y,z),z)^2,y,z,A))-
expand(Doubleint(f(y,z)*diff(w(y,z),y)^2,y,z,A)))=I[p]-J}:

```

EPR:={epso,gamasy,gamasz,gamasv,gamaw}:

# Integration des contraintes et Substitution de l'initialisation dans les expressions des Efforts :

# Effort Normal et Tranchants :

```

qx:=factor(simplify(expand(Doubleint(sigma(s),y,z,A))))
Qx:=subs(init,qx):
qy:=factor(simplify(expand(Doubleint(tau(sy),y,z,A)))):
Qy:=subs(init,qy):
qz:=factor(simplify(expand(Doubleint(tau(sz),y,z,A)))):
Qz:=subs(init,qz):

```

# Moments de Flexion :

```

my:=factor(simplify(expand(Doubleint(sigma(s)*z,y,z,A)))):
My:=subs(init,my):
mz:=factor(simplify(expand(Doubleint(sigma(s)*(-y),y,z,A)))):
Mz:=subs(init,mz):

```

# Bi-moment de Torsion :

```

mw:=factor(simplify(expand(Doubleint(sigma(s)*(-w(y,z)),y,z,A)))):
Mw:=subs(init,mw):

```

# Torsion de St Venant :

```
tsv1:=collect(factor(expand(Doubleint(-tau(sy)*z[b](z),y,z,A))),EPR):
      Tsv1:=subs(init,tsv1):
tsv2:=collect(factor(expand(Doubleint(-tau(sy)*wyz[2],y,z,A))),EPR):
      Tsv2:=subs(init,tsv2):
tsv3:=collect(factor(expand(Doubleint(tau(sz)*y[b](y),y,z,A))),EPR):
      Tsv3:=subs(init,tsv3):
tsv4:=collect(factor(expand(Doubleint(-tau(sz)*wyz[3],y,z,A))),EPR):
      Tsv4:=subs(init,tsv4):

      tsv:=collect(factor(expand(Doubleint(-
tau(sy)*(z[b](z)+wyz[2])+tau(sz)*(y[b](y)-wyz[3]),y,z,A))),EPR):
      Tsv:=subs(init,subs(init,tsv)):
```

# Torsion de Vlassov :

```
tw1:=collect(factor(expand(Doubleint(tau(sy)*wyz[2],y,z,A))),EPR):
tw2:=collect(factor(expand(Doubleint(tau(sz)*wyz[3],y,z,A))),EPR):
      tw:=collect(factor(expand(tw1+tw2)),EPR):
      Tw:=subs(init,subs(init,tw)):
```

# Moment de Torsion total suivant x :

```
mx:=collect(factor(expand(tsv1+tsv3)),EPR):
      Mx:=subs(init,mx):
```

# Construction de la matrice C dans la relation constitutive :

```
eff1:=<Qx,Qy,Qz,My,Mz,Mw,Tsv,Tw>:
def1:=<epso,gamasy,gamasz,kappay,kappaz,kappaw,gamasv,gamaw>:
```

```
C := matrix(8,8,0):
      for i to 8 do
      for j to 8 do
C[i,j]:=coeff(eff1[i],def1[j])

      end do
      end do:
```

# Test symetrie de C :

```
testsym(C):=simplify(expand(evalm(C-transpose(C)))):
```

# Efforts et Déformations résultantes :

```
EFR := vector(8) :
```

```
EPS := vector(8) .
```

```
EPS[1] := eps[0] :
```

```
EPS[2] := gama[sy] :
```

```
EPS[3] := gama[sz] :
```

```
EPS[4] := kappa[y] :
```

```
EPS[5] := kappa[z] :
```

```
EPS[6] := kappa[w] :
```

```
EPS[7] := gama[sv] :
```

```
EPS[8] := gama[w] :
```

# Construction de l'équation de Lamé et détermination de l'opérateur L :

```
DEPR := [u[g](s), v[g](s), w[g](s), theta[x](s), theta[y](s), theta[z](s), psi(s)] :
```

```
EPR := vector(8) .
```

```
EPR[1] := Ds*u[g](s) + Kz*v[g](s) - Ky*w[g](s) :
```

```
EPR[2] := Ds*v[g](s) - Kz*u[g](s) + Kx*w[g](s) + theta[z](s) :
```

```
EPR[3] := Ds*w[g](s) + Ky*u[g](s) - Kx*v[g](s) - theta[y](s) :
```

```
EPR[4] := -Kz*theta[x](s) + Kx*theta[z](s) + Ds*theta[y](s) :
```

```
EPR[5] := Ky*theta[x](s) - Kx*theta[y](s) + Ds*theta[z](s) :
```

```
EPR[6] := Ds*psi(s) :
```

```
EPR[7] := Kz*theta[y](s) - Ky*theta[z](s) + Ds*theta[x](s) :
```

```
EPR[8] := Kz*theta[y](s) - Ky*theta[z](s) + Ds*theta[x](s) - Ds*psi(s) :
```

```
L := matrix(8, 7, 0) :
```

```
for i from 1 to 8 do
```

```
for j from 1 to 7 do
```

```
L[i, j] := coeff(EPR[i], DEPR[j]) :
```

```
end do :
```

```
end do ;
```

```
C := evalm(C) ;
```

```
testsym(C) := simplify(expand(evalm(C - transpose(C)))) .
```

```
L := evalm(L) .
```

$$C := \begin{bmatrix} EA & 0 & 0 & 0 & 0 & -ES_{\omega} & 0 & 0 \\ 0 & GA & 0 & 0 & 0 & 0 & Gz_c A & 0 \\ 0 & 0 & GA & 0 & 0 & 0 & -Gy_c A & 0 \\ 0 & 0 & 0 & EI_y & -EI_{yz} & -EI_{wz} & 0 & 0 \\ 0 & 0 & 0 & -EI_{yz} & EI_z & EI_{wy} & 0 & 0 \\ -ES_{\omega} & 0 & 0 & -EI_{wz} & EI_{wy} & EI_w & 0 & 0 \\ 0 & Gz_c A & -Gy_c A & 0 & 0 & 0 & GJ & -G(J-I)_p \\ 0 & 0 & 0 & 0 & 0 & 0 & G(I-J)_p & G(J-I)_p \end{bmatrix}$$

$$L := \begin{bmatrix} Ds & Kz & -Ky & 0 & 0 & 0 & 0 \\ -Kz & Ds & Kx & 0 & 0 & 1 & 0 \\ Ky & -Kx & Ds & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -Kz & Ds & Kx & 0 \\ 0 & 0 & 0 & Ky & -Kx & Ds & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Ds \\ 0 & 0 & 0 & Ds & Kz & -Ky & 0 \\ 0 & 0 & 0 & Ds & Kz & -Ky & -Ds \end{bmatrix}$$

## 7-2 Programme calcul poutre circulaire

```

> with(student):
> with(linalg):
>
# Paramétrisation Géométrique :
X:=vector(3):
> X[1]:=r*cos(t):
X[2]:=r*sin(t):
X[3]:=0:

# Repère Local de Frenet :
T:=vector(3):N:=vector(3):B:=vector(3):

for i from 1 by 1 to 3 do
> T[i]:=diff(X[i],t):
A[i]:=diff(X[i],t,t):
end do:
L:=simplify(expand(norm(T,2),trig),assume=positive):

Kg:=vector(3):
for i from 1 to 3 do
Kg[i]:=A[i]/L^2:
end do:

alfa:=simplify(expand(norm(T,2),trig),assume=positive):
beta:=simplify(expand(norm(Kg,2),trig),assume=positive):

for i from 1 to 3 do
N[i]:=Kg[i]/beta:
end do:

B:=simplify(expand(crossprod(T,N),trig),assume=positive):
gama:=simplify(expand(norm(B,2),assume=real),assume=positive):

e1:=vector(3): e2:=vector(3) :e3:=vector(3):

for i from 1 by 1 to 3 do
e1[i]:=T[i]/alfa:
end do:

e2:=N:

for i from 1 to 3 do
e3[i]:=B[i]/gama:
end do:

```

```

R[alpha]:=matrix(3,3,0):
R[alpha][1,1]:=1:
R[alpha][2,2]:=cos(alpha):
R[alpha][2,3]:=sin(alpha):
R[alpha][3,2]:=-R[alpha][2,3]:
R[alpha][3,3]:=R[alpha][2,2]:
R[alpha]:=evalm(R[alpha]):

> M:=concat(e1,e2,e3):
GL:=evalm(R[alpha]&*M):
LG:=transpose(GL):
>
K1:=vector(3):
K1:=simplify(evalm(Kg&*GL)):
C:=vector(3):
for i from 1 to 3 do
C[i]:=-diff(e3[i],t):
end do:

Kre:=vector(3):
Kre[1]:=0:
Kre[2]:=K1[2]:
Kre[3]:=simplify(expand(dotprod(C, e2), trig), assume=positive):

Kr:=vector(3):
Kr[1]:=0:
Kr[2]:=Kcy:
Kr[3]:=0:

```

## Relations Cinematiques

# Considerant les déplacements au centre de gravite des sections comme :

```

>
> Ug:=<u[g](s), v[g](s), w[g](s)>:

```

et les rotations sont définies comme

```

rots:=<theta[x](s), theta[y](s), theta[z](s)>:

```

```

> r:=matrix(3,3,0):
> r[2,1]:=- (z[b](z)):
r[3,1]:= (y[b](y)):
r[1,2]:= z:
r[1,3]:= -y:
> r:=evalm(r):

```

```

> e1:=matrix(3,3,0):
    e1[2,3]:= 1:
    e1[3,2]:=-1:
    e1:=evalm(e1):
> U := evalm(Ug + r &* rots - w(y,z)*psi(s)*<1,0,0>):
> K := matrix(3,3,0):
    K[1,2] := Kr[3]:
    K[1,3] := -Kr[2]:
    K[2,1] := -Kr[3]:
    K[2,3] := Kr[1]:
    K[3,1] := Kr[2]:
    K[3,2] := -Kr[1]:
    K:=evalm(K):

> rotsprime:=<diff(rots[1],s),diff(rots[2],s),diff(rots[3],s)>:
    Ugprime:=<diff(Ug[1],s),diff(Ug[2],s),diff(Ug[3],s)>:
    wyz := <0,diff(w(y,z),y),diff(w(y,z),z)>:
    > k := evalm(rotsprime + evalm(K &* rots)):
        kappa[x] := k[1]:
        kappa[y] := k[2]:
        kappa[z] := k[3]:
    kappa[w] := diff(psi(s),s):

> epsilon := evalm(evalm(Ugprime)+evalm((K) &* Ug)+evalm((r) &*
k) - evalm( psi(s) *(wyz)) - evalm((w(y,z)) *diff( psi(s)),s)*
    <1,0,0>) + evalm((e1) &* rots)):
    >
    eps[s] :=(epso-y*kappaz+z*kappay-w(y,z)*kappaw):
eps[sy]:= (gamasy - (diff(w(y,z),y)+z[b](z))*gamasv+diff(w(y,z),y)*
    gamaw):
eps[sz]:= (gamasz+(-diff(w(y,z),z)+y[b](y))*gamasv+diff(w(y,z),z)
    *gamaw):
    >
eps[0] := diff(u[g](s),s) + Kr[3]*v[g](s) - Kr[2]*w[g](s) :
gam[sy] :=diff(v[g](s),s) - Kr[3]*u[g](s) + Kr[1]*w[g](s) +
    theta[z](s) :
gam[sz] :=diff(w[g](s),s) + Kr[2]*u[g](s) - Kr[1]*v[g](s) -
    theta[y](s) :
gam[sv] :=Kr[3]*theta[y](s) - Kr[2]*theta[z](s) +
    diff(theta[x](s),s):
gam[w] :=gam[sv]-kappa[w] :

# Matrice de Rigidité Normalisée dans le repere central principal d'inertie :
> C:=matrix(8,8,0):
    C[1,1]:=E*A:

```

```

C[2,2]:=E*A:
C[3,3]:=G*A:
C[4,4]:=E*I[y]:
C[5,5]:=E*I[z]:
C[6,6]:=E*I[w]:
C[7,7]:=G*J:
C[7,8]:=G*(I[p]-J):
C[8,7]:=G*(I[p]-J):
C[8,8]:=G*J:
C:=evalm(C):

```

```

> EFR := vector(8):
> EPS := vector(8):
EPS[1]:=eps[0]:
EPS[2]:=gam[sy]:
EPS[3]:=gam[sz]:
EPS[4]:= kappa[y]:
EPS[5]:= kappa[z]:
EPS[6]:= kappa[w]:
EPS[7]:=gam[sv]:
EPS[8]:=gam[w]:

```

# Calcul des Efforts Resultants :

```

> EFR := evalm(C&* EPS):
dep:={u[g](s),v[g](s),w[g](s),theta[x](s),theta[y](s),theta[z](s),
diff(theta[x](s),s),diff(theta[y](s),s),diff(theta[z](s),s),di
ff(psi(s),s)}:

```

```

> qx:=collect(factor(expand(EFR[1])),dep):
> qy:=collect(factor(expand(EFR[2])),dep):
> qz:=collect(factor(expand(EFR[3])),dep):
> my:=collect(factor(expand(EFR[4])),dep):
> mz:=collect(factor(expand(EFR[5])),dep):
> mw:=collect(factor(expand(EFR[6])),dep):
> tsv:=collect(factor(expand(EFR[7])),dep):
> tw:=collect(factor(expand(EFR[8])),dep):

```

## # Construction de L'Equation de Lamé

```
> DEPR:=[u[g](s),v[g](s),w[g](s),theta[x](s),theta[y](s),theta[z](
      s),psi(s)]:
      EPR:=vector(8):
EPR[1] := Ds*u[g](s) + Kr[3]*v[g](s) - Kr[2]*w[g](s):
EPR[2] := Ds*v[g](s) - Kr[3]*u[g](s) + Kr[1]*w[g](s) +
      theta[z](s):
EPR[3] := Ds*w[g](s) + Kr[2]*u[g](s) - Kr[1]*v[g](s) -
      theta[y](s):
EPR[4] := -Kr[3]*theta[x](s) + Kr[1]*theta[z](s) +
      Ds*theta[y](s):
EPR[5] := Kr[2]*theta[x](s) - Kr[1]*theta[y](s) +
      Ds*theta[z](s):
      EPR[6] := Ds*psi(s):
EPR[7] := Kr[3]*theta[y](s) - Kr[2]*theta[z](s) +
      Ds*theta[x](s):
EPR[8] := Kr[3]*theta[y](s) - Kr[2]*theta[z](s) + Ds*theta[x](s)
      - Ds*psi(s):

      > L := matrix(8,7,0):
      for i from 1 to 8 do
      for j from 1 to 7 do
L[i,j]:=coeff(EPR[i],DEPR[j]):

      end do:
      end do:
      L:=evalm(L):

> Bbar:=<fbx, fby, fbz, mbx, mby, mbz, mbw>:

> EQU:=evalm(transpose(L) &*EFR+Bbar):
      > for i from 1 to 7 do
      equ(i):=EQU[i]:
      end do:
      >

      # Substitution de Ds par l'operateur differentiel :
> equal:=E*A*diff((diff(u[g](s),s)-Kcy*w[g](s)),s)+Kcy*G*A*(diff(w
      [g](s),s)+Kcy*u[g](s)-theta[y](s))+fbx:
      eq1:=collect(expand(equal),dep):

> equa2:=E*A*diff((diff(v[g](s),s)+theta[z](s)),s)+fby.
      eq2:=collect(expand(equa2),dep):

> equa3:=-Kcy*E*A*(diff(u[g](s),s)-Kcy*w[g](s))+G*A*diff((diff(w[g
```

```

] (s), s) + Kcy*u[g] (s) - theta[y] (s) , s) + fbz :
eq3 := collect (expand (equa3) , dep) :

```

```

> equa4 := Kcy*E*I[z] * (Kcy*theta[x] (s) + diff (theta[z] (s) , s)) + diff ((G*
J* (-Kcy*theta[z] (s) + diff (theta[x] (s) , s)) + G*(I[p] - J) * (-Kcy*theta[
z] (s) + diff (theta[x] (s) , s) - diff (psi (s) , s))) , s) + diff ((G*(I[p] - J) * (
-Kcy*theta[z] (s) + diff (theta[x] (s) , s)) + G*J* (-Kcy*theta[z] (s) + diff
(theta[x] (s) , s) - diff (psi (s) , s))) , s) + mbx :
eq4 := collect (expand (equa4) , dep) :

```

```

> equa5 := -G*A*(diff (w[g] (s) , s) + Kcy*u[g] (s) - theta[y] (s) ) + E*I[y] * dif
f (diff (theta[y] (s) , s) , s) + mby :
eq5 := collect (expand (equa5) , dep) :

```

```

> equa6 := E*A*(diff (v[g] (s) , s) + theta[z] (s) ) + E*I[z] * diff ((Kcy*theta[
x] (s) + diff (theta[z] (s) , s)) , s) - Kcy*(G*J* (-Kcy*theta[z] (s) + diff (th
eta[x] (s) , s)) + G*(I[p] - J) * (-Kcy*theta[z] (s) + diff (theta[x] (s) , s) - d
iff (psi (s) , s))) - Kcy*(G*(I[p] - J) * (-Kcy*theta[z] (s) + diff (theta[x] (
s) , s)) + G*J* (-Kcy*theta[z] (s) + diff (theta[x] (s) , s) - diff (psi (s) , s)
)) + mbz :
eq6 := collect (expand (equa6) , dep) :

```

```

> equa7 := E*I[w] * diff (diff (psi (s) , s) , s) - diff ((G*(I[p] - J) * (-Kcy*thet
a[z] (s) + diff (theta[x] (s) , s)) + G*J* (-Kcy*theta[z] (s) + diff (theta[x]
(s) , s) - diff (psi (s) , s))) , s) + mbw :
eq7 := collect (expand (equa7) , dep) :

```

```

> syst := {eq1=0, eq2=0, eq3=0, eq4=0, eq5=0, eq6=0, eq7=0} :
inc := {u[g] (s) , v[g] (s) , w[g] (s) , theta[x] (s) , theta[y] (s) , theta[z] (s
) , psi (s)} :

```

```

> sol := dsolve (syst, inc) :

```

```

>
for i from 1 to 7 do
print (simplify (expand (sol [i]))) ;
end do ;

```

```

>

```

### **7-3 Programme calcul poutre hélicoïdale**

**Le meme programme s'applique au cas circulaire avec prise en compte des deux courbures torsion**

**Et meme pour ce cas nous n'arrivons qu'à poser les equations differentielles**

### 7-4 Programme calcul poutre spatiale droite

Programme calcul poutre droite :

```
with(student):
with(linalg):
Warning, the protected names norm and trace have been redefined and
unprotected

assume(r>0):

# Interpolation de Lagrange:

N1:=(L-t)/L: N2:=t/L:
alpha:=N1*alpha[1] + N2*alpha[2]:

# Paramétrisation Géométrique :

X[1]:=a*t+x1 + 1/2*alpha*t:
X[2]:=b*t+x2:
X[3]:=c*t+x3:
#L:=simplify(sqrt(a^2+b^2+c^2), assume=positive)#

Cx:=a/L :Cy:=b/L :Cz:=c/L:
#Co:=simplify(sqrt(Cx^2+Cz^2), assume=positive)#:
cos(beta).:=Cx/Co :sin(beta):=Cz/Co :cos(gamma):=Co :sin(gamma):=Cy:

R[beta]:=matrix(3,3,0):
R[beta][1,1]:=cos(beta):
R[beta][1,3]:=sin(beta):
R[beta][2,2]:=1:
R[beta][3,1]:=-R[beta][1,3]:
R[beta][3,3]:=R[beta][1,1]:
R[beta]:=evalm(R[beta]):

R[gamma]:=matrix(3,3,0):
R[gamma][1,1]:=cos(gamma):
R[gamma][1,2]:=sin(gamma):
R[gamma][2,1]:=-R[gamma][1,2]:
R[gamma][2,2]:=R[gamma][1,1]:
R[gamma][3,3]:=1:

R[gamma]:=evalm(R[gamma]):

R[alpha]:=matrix(3,3,0):
R[alpha][1,1]:=1:
R[alpha][2,2]:=cos(alpha):
```

```

R[alpha][2,3]:=sin(alpha):
R[alpha][3,2]:=-R[alpha][2,3]:
R[alpha][3,3]:=R[alpha][2,2]:
R[alpha]:=evalm(R[alpha]):

# Matrice transfo Local-Global:

LG:=evalm(R[alpha]&*evalm(R[gamma]&*R[beta])) :

GL:=evalm(transpose(LG)):
e1:=vector(3): e2:=vector(3): e3:=vector(3):
for i from 1 to 3 do
  e1[i]:=LG[1,i]:
  e2[i]:=LG[2,i]:
  e3[i]:=LG[3,i]:
end do:

H:=vector(3):N:=vector(3):B:=vector(3):P:=vector(3):

for i from 1 by 1 to 3 do
  P[i]:=diff(X[i],t,t):
end do:
Kg:=vector(3):
for i from 1 to 3 do
  Kg[i]:=P[i]:
end do:

Kl:=vector(3):
Kl[1]:=Kx:
Kl[2]:=0:
Kl[3]:=0:

print(vecteur, unitaire, tangent_e1,e1) :
print(vecteur, unitaire ,normale_e2,e2) :
print(vecteur, unitaire, binormal_e3,e3) :
print(matrice ,transformation_LG, lg,LG) :
print(matrice, transformation_GL, gl,GL) :

print(vecteur, courbure, torsion,glob_Kg,evalm(LG&*Kl)) :
print(vecteur, courbure, torsion,loc_Kl, loc,Kl) :

# Relations Cinematiques :

# Considerant les déplacements au centre de gravite des sections comme :

Ug:=<u[g](s),v[g](s),w[g](s)>:

```

et les rotations sont definies comme

```
rots:=<theta[x](s), theta[y](s), theta[z](s)>:
```

```
  r:=matrix(3,3,0):
  r[2,1]:=- (z[b](z)):
  r[3,1]:= (y[b](y)):
  r[1,2]:= z:
  r[1,3]:= -y:
  r:=evalm(r):
```

```
  e1:=matrix(3,3,0):
  e1[2,3]:= 1:
  e1[3,2]:=-1:
  e1:=evalm(e1):
```

```
U := evalm(Ug + r &* rots - w(y,z)*pzi(s)*<1,0,0>):
```

```
  K := matrix(3,3,0):
  K[1,2] := K1[3]:
  K[1,3] := -K1[2]:
  K[2,1] := -K1[3]:
  K[2,3] := K1[1]:
  K[3,1] := K1[2]:
  K[3,2] := -K1[1]:
  K:=evalm(K):
```

```
rotsprime:=<diff(rots[1],s),diff(rots[2],s),diff(rots[3],s)>:
```

```
Ugprime:=<diff(Ug[1],s),diff(Ug[2],s),diff(Ug[3],s)>:
```

```
wyz := <0,diff(w(y,z),y),diff(w(y,z),z)>:
```

```
k := evalm(rotsprime + evalm(K &* rots)):
```

```
kappa[x] := k[1]:
```

```
kappa[y] := k[2]:
```

```
kappa[z] := k[3]:
```

```
kappa[w] := diff(pzi(s),s):
```

```
epsilon := evalm(evalm(Ugprime)+evalm((K) &* Ug)+evalm((r) &* k)-
evalm( pzi(s) *(wyz)) - evalm((w(y,z)) *diff( (pzi(s)),s)* <1,0,0>) +
evalm((e1) &* rots)):
```

```
eps[s] :=(epso-y*kappaz+z*kappay-w(y,z)*kappaw):
```

```
eps[sy]:=(gamasy-(diff(w(y,z),y)+z[b](z))*gamasv+diff(w(y,z),y)*gamaw):
```

```
eps[sz]:=(gamasz+(-
```

```
diff(w(y,z),z)+y[b](y))*gamasv+diff(w(y,z),z)*gamaw):
```

```
EFR := vector(8):
```

```

    eps[0]:=diff(u[g](s),s) + Kl[3]*v[g](s) - Kl[2]*w[g](s) :
gam[sy]:=diff(v[g](s),s) - Kl[3]*u[g](s) + Kl[1]*w[g](s) + theta[z](s) :
gam[sz]:=diff(w[g](s),s) + Kl[2]*u[g](s) - Kl[1]*v[g](s) - theta[y](s) :
gam[sv]:=Kl[3]*theta[y](s) - Kl[2]*theta[z](s) + diff(theta[x](s),s) :
gam[w]:=gam[sv]-kappa[w] :

```

# Matrice contraintes / deformations normalisée dans le repere central principal d'inertie :

```

CC:=matrix(8,8,0) :
    CC[1,1]:=E*A :
    CC[2,2]:=E*A :
    CC[3,3]:=G*A :
    CC[4,4]:=E*I[y] :
    CC[5,5]:=E*I[z] :
    CC[6,6]:=E*I[w] :
    CC[7,7]:=G*J :
    CC[7,8]:=G*(I[p]-J) :
    CC[8,7]:=G*(I[p]-J) :
    CC[8,8]:=G*J :
CC:=evalm(CC) :

```

```

EFR := vector(8) :
EPS := vector(8) :
    EPS[1]:=eps[0] :
    EPS[2]:=gam[sy] :
    EPS[3]:=gam[sz] :
    EPS[4]:=kappa[y] :
    EPS[5]:=kappa[z] :
    EPS[6]:=kappa[w] :
    EPS[7]:=gam[sv] :
    EPS[8]:=gam[w] :

```

# Calcul des efforts resultants :

```

EFR := evalm(CC&*EPS) :
dep:={u[g](s),v[g](s),w[g](s),theta[x](s),theta[y](s),theta[z](s),diff(theta[x](s),s),diff(theta[y](s),s),diff(theta[z](s),s),diff(pzi(s),s)} :

```

```

qx:=collect(factor(expand(EFR[1])),dep) :

```

```

qy:=collect(factor(expand(EFR[2])),dep) :

```

```

qz:=collect(factor(expand(EFR[3])),dep):
my:=collect(factor(expand(EFR[4])),dep):
mz:=collect(factor(expand(EFR[5])),dep):
mw:=collect(factor(expand(EFR[6])),dep):
tsv:=collect(factor(expand(EFR[7])),dep):
tw:=collect(factor(expand(EFR[8])),dep):

```

### # Construction de L'Equation de Lamé

```

DEPR:=[u[g](s),v[g](s),w[g](s),theta[x](s),theta[y](s),theta[z](s),psi(
s)]:
EPR:=vector(8):
EPR[1]:=Ds*u[g](s)+Kl[3]*v[g](s)-Kl[2]*w[g](s):
EPR[2]:=Ds*v[g](s)-Kl[3]*u[g](s)+Kl[1]*w[g](s)+theta[z](s):
EPR[3]:=Ds*w[g](s)+Kl[2]*u[g](s)-Kl[1]*v[g](s)-theta[y](s):
EPR[4]:=-Kl[3]*theta[x](s)+Kl[1]*theta[z](s)+Ds*theta[y](s):
EPR[5]:=Kl[2]*theta[x](s)-Kl[1]*theta[y](s)+Ds*theta[z](s):
EPR[6]:=Ds*psi(s):
EPR[7]:=Kl[3]*theta[y](s)-Kl[2]*theta[z](s)+Ds*theta[x](s):
EPR[8]:=Kl[3]*theta[y](s)-Kl[2]*theta[z](s)+Ds*theta[x](s)-
Ds*psi(s):

LL:=matrix(8,7,0):
for i from 1 to 8 do
for j from 1 to 7 do
LL[i,j]:=coeff(EPR[i],DEPR[j]):
end do:
end do:
LL:=evalm(LL):

Bbar=<fbx,fbz,mbx,mbz,mbw>:
EQU:=evalm(transpose(LL)*EPR+Bbar):

for i from 1 to 7 do
equ(i):=EQU[i]:
end do:

```

# Substitution de Ds par l'operateur differentiel dans les equations:

```

equal:=E*A*diff(diff(u[g](s),s),s)+fbx:
eq1:=collect(expand(equal),dep):
equa2:=E*A*diff((diff(v[g](s),s)+Kx*w[g](s)+theta[z](s)),s)-
Kx*G*A*(diff(w[g](s),s)-Kx*v[g](s)-theta[y](s))+fby:
equa3:=Kx*E*A*(diff(v[g](s),s)+Kx*w[g](s)+theta[z](s))+G*A*diff((diff(w
[g](s),s)-Kx*v[g](s)-theta[y](s)),s)+fbz:
equa4:=diff((G*J*diff(theta[x](s),s)+G*(I[p]-J)*(diff(theta[x](s),s)-
diff(pzi(s),s))),s)+diff((G*(I[p]-
J)*diff(theta[x](s),s)+G*J*(diff(theta[x](s),s)-diff(pzi(s),s))),s):
eq4:=collect(expand(equa4),dep):

equa5:=-G*A*(diff(w[g](s),s)-Kx*v[g](s)-
theta[y](s))+E*I[y]*diff((Kx*theta[z](s)+diff(theta[y](s),s)),s)-
Kx*E*I[z]*(-Kx*theta[y](s)+diff(theta[z](s),s))+mby:
eq5:=collect(expand(equa5),dep):

equa6:=E*A*(diff(v[g](s),s)+Kx*w[g](s)+theta[z](s))+Kx*E*I[y]*(Kx*theta
[z](s)+diff(theta[y](s),s))+E*I[z]*diff((-
Kx*theta[y](s)+diff(theta[z](s),s)),s)+mbz:

equa7:=E*I[w]*diff(diff(pzi(s),s),s)+mbw:

eq1:=collect(expand(equal),dep);
eq2:=collect(expand(equa2),dep);
eq3:=collect(expand(equa3),dep);
eq4:=collect(expand(equa4),dep);
eq5:=collect(expand(equa5),dep);
eq6:=collect(expand(equa6),dep);
eq7:=collect(expand(equa7),dep);

syst:={eq1=0,eq2=0,eq3=0,eq4=0,eq5=0,eq6=0,eq7=0}:
inc:={u[g](s),v[g](s),w[g](s),theta[x](s),theta[y](s),theta[z](s),pzi(s)
}):

sol:=dsolve(syst,inc):
term:=[s,s^2,s^3,s^4]:
for i from 1 to 7 do
print(simplify(collect(factor(sol[i]),term)));
end do:

tetaz:=s->(2*G*I[y]*I[z]*E*A*Kx^3*_C7*cos(Kx*s)*s+G*I[z]*A*mbz*Kx-
G*I[z]*fbz*A-G*I[y]*A*mbz*Kx+G*I[y]*fbz*A-
G*I[y]*I[z]*E*A*Kx^2*_C7*sin(Kx*s)-3*G*I[y]*I[z]*E*A*Kx*_C5*sin(Kx*s)-
3*G*I[y]*I[z]*E*A*Kx*_C8*cos(Kx*s)-

```

$$\begin{aligned}
& 3 * G * I [y]^2 * E * A * Kx^3 * \_C8 * \cos (Kx * s) * s^2 - \\
& 3 * G * I [y]^2 * E * A * Kx^2 * \_C8 * \sin (Kx * s) * s - \\
& 6 * I [y]^2 * I [z] * E^2 * Kx^3 * \_C5 * \sin (Kx * s) - \\
& 2 * G * I [y]^2 * E * A * Kx^3 * \_C4 * \sin (Kx * s) * s - \\
& 3 * G * I [y]^2 * E * A * Kx^3 * \_C5 * \sin (Kx * s) * s^2 - G * I [y]^2 * E * A * Kx^2 * \_C7 * \sin (Kx * s) - \\
& G * I [y]^2 * E * A * Kx^3 * \_C6 * \cos (Kx * s) + 3 * G * I [y]^2 * E * A * Kx^2 * \_C5 * \cos (Kx * s) * s - \\
& 2 * G * I [y]^2 * E * A * Kx^3 * \_C7 * \cos (Kx * s) * s - 3 * G * I [y]^2 * E * A * Kx * \_C5 * \sin (Kx * s) - \\
& 3 * G * I [y]^2 * E * A * Kx * \_C8 * \cos (Kx * s) + G * I [y] * I [z] * E * A * Kx^3 * \_C3 * \sin (Kx * s) + G * I [y] \\
& * I [z] * E * A * Kx^2 * \_C4 * \cos (Kx * s) + G * I [y] * I [z] * E * A * Kx^3 * \_C6 * \cos (Kx * s) + 3 * G * I \\
& [y] * I [z] * E * A * Kx^3 * \_C8 * \cos (Kx * s) * s^2 - \\
& 3 * G * I [y] * I [z] * E * A * Kx^2 * \_C8 * \sin (Kx * s) * s + 6 * G * I [y]^2 * I [z] * E * Kx^3 * \_C5 * \sin (K \\
& x * s) + 6 * G * I [y]^2 * I [z] * E * Kx^3 * \_C8 * \cos (Kx * s) - \\
& G * I [y]^2 * E * A * Kx^3 * \_C3 * \sin (Kx * s) + G * I [y]^2 * E * A * Kx^2 * \_C4 * \cos (Kx * s) - \\
& 6 * I [y]^2 * I [z] * E^2 * Kx^3 * \_C8 * \cos (Kx * s) + 2 * G * I [y] * I [z] * E * A * Kx^3 * \_C4 * \sin (Kx * \\
& s) * s + 3 * G * I [y] * I [z] * E * A * Kx^3 * \_C5 * \sin (Kx * s) * s^2 + 3 * G * I [y] * I [z] * E * A * Kx^2 * \_C \\
& 5 * \cos (Kx * s) * s) / (Kx^3 * G * I [y] * E * A * (I [y] - I [z])) :
\end{aligned}$$

$$\begin{aligned}
& \text{vg} := s \rightarrow (-\text{fby} * A * G - \\
& \text{fby} * E * Kx^2 * I [z] + G * A * Kx * \text{mby} + \_C1 * \cos (Kx * s) * Kx^4 * G * E * A * I [z] + \_C2 * \sin (Kx * s) * \\
& Kx^4 * G * E * A * I [z] + \_C3 * \sin (Kx * s) * s * Kx^4 * G * E * A * I [z] + \_C4 * \sin (Kx * s) * s^2 * Kx^4 * \\
& G * E * A * I [z] + \_C5 * \sin (Kx * s) * s^3 * Kx^4 * G * E * A * I [z] + \_C6 * \cos (Kx * s) * s * Kx^4 * G * E * A \\
& * I [z] + \_C7 * \cos (Kx * s) * s^2 * Kx^4 * G * E * A * I [z] + \_C8 * \cos (Kx * s) * s^3 * Kx^4 * G * E * A * I [ \\
& z]) / (Kx^4 * G * E * A * I [z]) :
\end{aligned}$$

$$\text{ug} := s \rightarrow -1/2 * (\text{fbx} * s^2 - 2 * \_C13 * s * E * A - 2 * \_C14 * E * A) / (E * A) :$$

$$\begin{aligned}
& \text{tetay} := s \rightarrow (-G * I [z] * A * \text{fby} + G * I [y] * A * \text{fby} + G * I [z] * A * \text{mby} * Kx - G * I [y] * A * \text{mby} * Kx - \\
& 2 * G * I [z]^2 * Kx^3 * A * E * \_C7 * \sin (Kx * s) * s + G * I [z]^2 * Kx^2 * A * E * \_C7 * \cos (Kx * s) + G * I \\
& [y] * I [z] * A * E * Kx^2 * \_C7 * \cos (Kx * s) + 2 * G * I [z]^2 * Kx^3 * A * E * \_C4 * \cos (Kx * s) * s + G * I \\
& [z]^2 * Kx^2 * A * E * \_C4 * \sin (Kx * s) + G * I [y] * I [z] * A * E * Kx^3 * \_C6 * \sin (Kx * s) - \\
& G * I [y] * I [z] * A * E * Kx^3 * \_C3 * \cos (Kx * s) + 3 * G * I [y] * I [z] * A * E * Kx^3 * \_C8 * \sin (Kx * s) \\
& * s^2 + G * I [y] * I [z] * A * E * Kx^2 * \_C4 * \sin (Kx * s) + 3 * G * I [z]^2 * Kx^2 * A * E * \_C5 * \sin (Kx * \\
& s) * s + 6 * G * I [y] * I [z]^2 * E * Kx^3 * \_C5 * \cos (Kx * s) - \\
& G * I [z]^2 * Kx^3 * A * E * \_C6 * \sin (Kx * s) + 3 * G * I [y] * I [z] * A * E * Kx^2 * \_C5 * \sin (Kx * s) * s + \\
& 2 * G * I [y] * I [z] * A * E * Kx^3 * \_C7 * \sin (Kx * s) * s - \\
& 3 * G * I [y] * I [z] * A * E * Kx^3 * \_C5 * \cos (Kx * s) * s^2 + 6 * I [y] * I [z]^2 * E^2 * Kx^3 * \_C8 * \sin \\
& (Kx * s) - \\
& 2 * G * I [y] * I [z] * A * E * Kx^3 * \_C4 * \cos (Kx * s) * s + G * I [z]^2 * Kx^3 * A * E * \_C3 * \cos (Kx * s) - \\
& 3 * G * I [z]^2 * Kx^3 * A * E * \_C8 * \sin (Kx * s) * s^2 + 3 * G * I [z]^2 * Kx^2 * A * E * \_C8 * \cos (Kx * s) \\
& * s - \\
& 6 * G * I [y] * I [z]^2 * E * Kx^3 * \_C8 * \sin (Kx * s) + 3 * G * I [z]^2 * Kx^3 * A * E * \_C5 * \cos (Kx * s) * \\
& s^2 + 3 * G * I [y] * I [z] * A * E * Kx^2 * \_C8 * \cos (Kx * s) * s - \\
& 6 * I [y] * I [z]^2 * E^2 * Kx^3 * \_C5 * \cos (Kx * s)) / (G * I [z] * E * A * Kx^3 * (I [y] - I [z])) :
\end{aligned}$$

$$\begin{aligned}
& \text{wg} := s \rightarrow (G * I [y] * I [z] * \text{fbz} * Kx^2 - G * I [y]^2 * \text{fbz} * Kx^2 - \\
& G * I [y] * I [z] * E * A * Kx^4 * \_C7 * \sin (Kx * s) * s^2 + I [y]^2 * G * E * A * Kx^4 * \_C8 * \sin (Kx * s) * \\
& s^3 + G * I [y] * I [z] * E * A * Kx^4 * \_C5 * \cos (Kx * s) * s^3 - \\
& G * I [y] * I [z] * E * A * Kx^4 * \_C1 * \sin (Kx * s) + G * I [y] * I [z] * E * A * Kx^4 * \_C3 * \cos (Kx * s) * s \\
& + G * I [y] * I [z] * E * A * Kx^4 * \_C2 * \cos (Kx * s) - \\
& G * I [y] * I [z] * E * A * Kx^4 * \_C8 * \sin (Kx * s) * s^3 - \\
& G * I [y] * I [z] * E * A * Kx^4 * \_C6 * \sin (Kx * s) * s + I [y]^2 * G * E * A * Kx^4 * \_C7 * \sin (Kx * s) * s^
\end{aligned}$$

```

2+G*I[y]*I[z]*E*A*Kx^4*_C4*cos(Kx*s)*s^2+I[y]^2*G*E*A*Kx^4*_C1*sin(Kx*s)
)-I[y]^2*G*E*A*Kx^4*_C2*cos(Kx*s)-I[y]^2*G*E*A*Kx^4*_C3*cos(Kx*s)*s-
I[y]^2*G*E*A*Kx^4*_C4*cos(Kx*s)*s^2-
I[y]^2*G*E*A*Kx^4*_C5*cos(Kx*s)*s^3+I[y]^2*G*E*A*Kx^4*_C6*sin(Kx*s)*s-
G*I[z]*A*mbz*Kx+G*I[z]*fbz*A+G*I[y]*A*mbz*Kx-
G*I[y]*fbz*A+G*I[y]*I[z]*E*A*Kx^2*_C7*sin(Kx*s)+3*G*I[y]*I[z]*E*A*Kx*_C
5*sin(Kx*s)+3*G*I[y]*I[z]*E*A*Kx*_C8*cos(Kx*s)+3*G*I[y]^2*E*A*Kx^2*_C8*
sin(Kx*s)*s+6*I[y]^2*I[z]*E^2*Kx^3*_C5*sin(Kx*s)+G*I[y]^2*E*A*Kx^2*_C7*
sin(Kx*s)-
3*G*I[y]^2*E*A*Kx^2*_C5*cos(Kx*s)*s+3*G*I[y]^2*E*A*Kx*_C5*sin(Kx*s)+3*G
*I[y]^2*E*A*Kx*_C8*cos(Kx*s)-
G*I[y]*I[z]*E*A*Kx^2*_C4*cos(Kx*s)+3*G*I[y]*I[z]*E*A*Kx^2*_C8*sin(Kx*s)
*s+6*G*I[y]^2*I[z]*E*Kx^3*_C5*sin(Kx*s)+6*G*I[y]^2*I[z]*E*Kx^3*_C8*cos(
Kx*s)-
G*I[y]^2*E*A*Kx^2*_C4*cos(Kx*s)+6*I[y]^2*I[z]*E^2*Kx^3*_C8*cos(Kx*s)-
3*G*I[y]*I[z]*E*A*Kx^2*_C5*cos(Kx*s)*s)/(Kx^4*G*I[y]*E*A*(I[y]-I[z])):

```

```

tetax:=s->1/4*(-s^2*mbw+4*_C9*s*E*I[w]+4*_C10*E*I[w])/(E*I[w]):

```

```

ppi:=s->1/2*(-s^2*mbw+2*_C11*s*E*I[w]+2*_C12*E*I[w])/(E*I[w]):

```

```

u1:=simplify(expand(ug(0)));
v1:=simplify(expand(vg(0)));
w1:=simplify(expand(wg(0)));
theta[x1]:=simplify(expand(tetax(0)));
theta[y1]:=simplify(expand(tetay(0)));
theta[z1]:=simplify(expand(tetaz(0)));
psi[1]:=simplify(expand(ppi(0)));
u2:=simplify(expand(ug(L)));
v2:=simplify(expand(vg(L)));
w2:=simplify(expand(wg(L)));
theta[x2]:=simplify(expand(tetax(L)));
theta[y2]:=simplify(expand(tetay(L)));
theta[z2]:=simplify(expand(tetaz(L)));
psi[2]:=simplify(expand(ppi(L)));

```

```

# Homogeneisation solutions déplacements nodaux :

```

```

homo:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,_C1
1=0,_C12=0,_C13=0,_C14=0}:
u01:=subs(homo,u1):
v01:=subs(homo,v1):
w01:=subs(homo,w1):
teta0x1:=subs(homo,theta[x1]):
teta0y1:=subs(homo,theta[y1]):
teta0z1:=subs(homo,theta[z1]):
psi01:=subs(homo,psi[1]):
u02:=subs(homo,u2):
v02:=subs(homo,v2):
w02:=subs(homo,w2):
teta0x2:=subs(homo,theta[x2]):
teta0y2:=subs(homo,theta[y2]):

```

```
teta0z2:=subs(homo,theta[z2]):
psi02:=subs(homo,psi[2]):
```

```
hom[1]:={_C1=1,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[2]:={_C1=0,_C2=1,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[3]:={_C1=0,_C2=0,_C3=1,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[4]:={_C1=0,_C2=0,_C3=0,_C4=1,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[5]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=1,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[6]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=1,_C7=0,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[7]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=1,_C8=0,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[8]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=1,_C9=0,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[9]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=1,_C10=0,
          C11=0,_C12=0,_C13=0,_C14=0}:
hom[10]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=1,
           C11=0,_C12=0,_C13=0,_C14=0}:
hom[11]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
           C11=1,_C12=0,_C13=0,_C14=0}:
hom[12]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
           C11=0,_C12=1,_C13=0,_C14=0}:
hom[13]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
           C11=0,_C12=0,_C13=1,_C14=0}:
hom[14]:={_C1=0,_C2=0,_C3=0,_C4=0,_C5=0,_C6=0,_C7=0,_C8=0,_C9=0,_C10=0,
           C11=0,_C12=0,_C13=0,_C14=1}:
CM:=matrix(14,14):
for i from 1 to 14
do
    CM[1,i]:=factor(subs(hom[i],u1-u01)):
    CM[2,i]:=factor(subs(hom[i],v1-v01)):
    CM[3,i]:=factor(subs(hom[i],w1-w01)):
    CM[4,i]:=factor(subs(hom[i],theta[x1]-teta0x1)):
    CM[5,i]:=factor(subs(hom[i],theta[y1]-teta0y1)):
    CM[6,i]:=factor(subs(hom[i],theta[z1]-teta0z1)):
    CM[7,i]:=factor(subs(hom[i],psi[1]-psi01)):

    CM[8,i]:=factor(subs(hom[i],u2-u02)):
    CM[9,i]:=factor(subs(hom[i],v2-v02)):
    CM[10,i]:=factor(subs(hom[i],w2-w02)):
    CM[11,i]:=factor(subs(hom[i],theta[x2]-teta0x2)):
    CM[12,i]:=factor(subs(hom[i],theta[y2]-teta0y2)):
    CM[13,i]:=factor(subs(hom[i],theta[z2]-teta0z2)):
    CM[14,i]:=factor(subs(hom[i],psi[2]-psi02)):
```

```

end do:

CM:=evalm(CM);

# Inversion de MC :

MC:=evalm(CM^(-1));

# Construction de la matrice [B] :

> Qxx:=s->E*A*diff(ug(s),s);
Qyy:=s->E*A*diff(vg(s),s)+E*A*Kx*wg(s)+E*A*tetaz(s);
Qzz:=s->G*A*diff(vg(s),s)-G*A*Kx*vg(s)-G*A*tetay(s);
Myy:=s->E*I[y]*Kx*tetaz(s)+E*I[y]*diff(tetay(s),s);
Mzz:=s->-E*I[z]*Kx*tetay(s)+E*I[z]*diff(tetaz(s),s);
Mww:=s->E*I[w]*diff(ppi(s),s);
Mxx:=s->2*G*I[p]*diff(tetax(s),s)-G*I[p]*diff(ppi(s),s);

```

#### Efforts nodaux :

```

Qx1:=simplify(expand(Qxx(0)));
Qy1:=simplify(expand(Qyy(0)));
Qz1:=simplify(expand(Qzz(0)));
Mx1:=simplify(expand(Mxx(0)));
My1:=simplify(expand(Myy(0)));
Mz1:=simplify(expand(Mzz(0)));
Mw1:=simplify(expand(Mww(0)));
Qx2:=simplify(expand(Qxx(L)));
Qy2:=simplify(expand(Qyy(L)));
Qz2:=simplify(expand(Qzz(L)));
Mx2:=simplify(expand(Mxx(L)));
My2:=simplify(expand(Myy(L)));
Mz2:=simplify(expand(Mzz(L)));
Mw2:=simplify(expand(Mww(L)));

```

#### Efforts nodaux homogènes :

```

Qx01:=subs(homo,Qx1);
Qy01:=subs(homo,Qy1);
Qz01:=subs(homo,Qz1);
Mx01:=subs(homo,Mx1);
My01:=subs(homo,My1);
Mz01:=subs(homo,Mz1);
Mw01:=subs(homo,Mw1);
Qx02:=subs(homo,Qx2);

```

```

Qy02:=subs(homo,Qy2):
Qz02:=subs(homo,Qz2):
Mx02:=subs(homo,Mx2):
My02:=subs(homo,My2):
Mz02:=subs(homo,Mz2):
Mw02:=subs(homo,Mw2):

```

# Construction matrice B :

```

BM:=matrix(14,14):
for i from 1 to 14
do
BM[1,i]:=factor(subs(hom[i],Qx1-Qx01)):
BM[2,i]:=factor(subs(hom[i],Qy1-Qy01)):
BM[3,i]:=factor(subs(hom[i],Qz1-Qz01)):
BM[4,i]:=factor(subs(hom[i],Mx1-Mx01)):
BM[5,i]:=factor(subs(hom[i],My1-My01)):
BM[6,i]:=factor(subs(hom[i],Mz1-Mz01)):
BM[7,i]:=factor(subs(hom[i],Mw1-Mw01)):

BM[8,i]:=factor(subs(hom[i],Qx2-Qx02)):
BM[9,i]:=factor(subs(hom[i],Qy2-Qy02)):
BM[10,i]:=factor(subs(hom[i],Qz2-Qz02)):
BM[11,i]:=factor(subs(hom[i],Mx2-Mx02)):
BM[12,i]:=factor(subs(hom[i],My2-My02)):
BM[13,i]:=factor(subs(hom[i],Mz2-Mz02)):
BM[14,i]:=factor(subs(hom[i],Mw2-Mw02)):
end do:

```

# Inversion de BM :

```

MB:=evalm(BM):

```

# Matrice de Rigidité [ K ] :

```

> KK:=evalm(BM*MC)

```