Annexe 11

Description des routines

et du programme principal

du fichier pmultianneaux.f

On se propose de décrire les procédures fortrans du programme pmultianneaux.f afin d'éclairer les lignes de programme qu'on peut y trouver. On répartit la discussion suivant des grosses rubriques. On rentre donc ici encore un peu plus dans l'explication du programme.

1 Sous-routines d'écriture dans le fichier pass.m:

La sous-routine Ouvrir permet d'ouvrir le fichier en écriture et Fermer de le fermer. Les sous-routines Passvecteur, Passmatrice et Passcalair permettent d'écrire, dans le fichier pass.m, des instructions d'affectation de variables interprétables par matlab. On leur rentre, en argument, la variable et son nom sous forme d'une chaîne de caractères.

Avec ce jeux de routines, on crée donc un fichier interprétable en matlab qui permet de faire passer les variables fortrans en variables de matlab. On peut alors utiliser la sortie graphique et les instructions graphiques de matlab.

2 Sous-routines utiles d'opérations sur des matrices et des vecteurs:

- On a une routine de recherche du maximum et du minimum de 2 nombres: max et min.

- Sur les vecteurs de $\mathbb{R}^3$:

On a une routine d'addition, de soustraction, de multiplication par un scalaire (prodscal), d'affectation, de produit scalaire, de produit vectoriel (ne pas faire: call produitvectoriel(a,b,a)) et une sous-routine qui nous donne la norme.

- Sur les matrices :

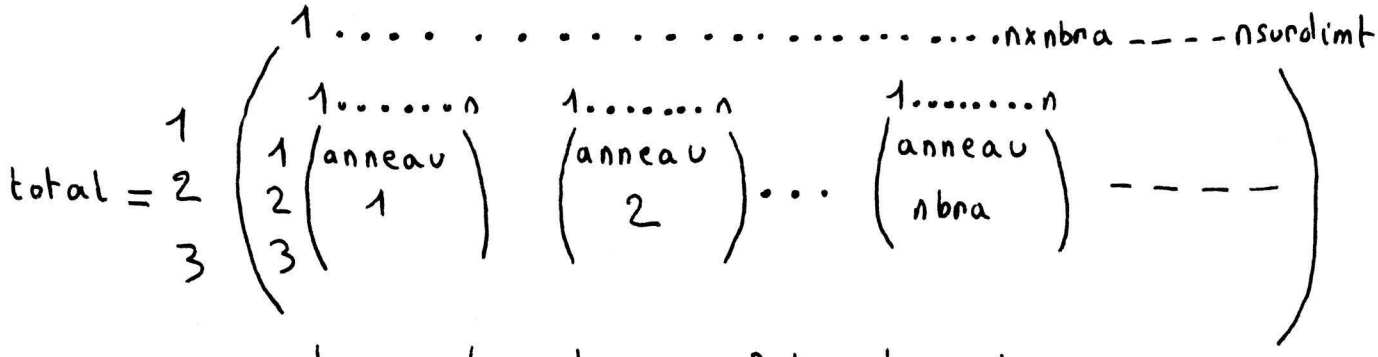
On utilise deux sous-routines d'affectation: egalmat et egalmatt.

- Sous-routines d'échange entre matrice et vecteurs:

Avec affectmatvect, on recherche le vecteur correspondant à la i-ème colonne d'une matrice et avec affectvectmat, on met un vecteur à la i-ème colonne d'une matrice.

- Sous-routines d'échange entre matrices :

Nous rappelons la structure de la variable total :



contenu = les nbra profils des nbra anneaux.

Lire(profil, total, nu) extrait le profil de l'anneau numéro nu de la matrice total et l'écrit dans la matrice profil.

ecrire(profil, total, nu) fait l'opération inverse. Elle prend le contenu de la variable profil et le range à la place des données de l'anneau numéro nu dans la matrice total.

3 Sous-routines d'échange avec le fichier profil.dat :

Il ya la routine loadprofil pour la lecture, saveprofil pour l'écriture ainsi que les procédures ouvrirf et fermerf pour ouvrir et fermer le fichier.

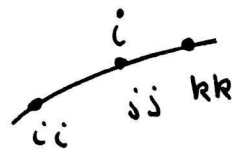
4 Sous-routines de discrétisation :

* dXs(a,c,d) : elle effectue l'opération $d = \frac{c-a}{2ds}$ utile pour l'approximation des dérivées premières.

* dXss(a,b,c,d) : elle effectue l'opération $d = \frac{a-2b+c}{(ds)^2}$ utile pour l'approximation de la dérivée seconde.

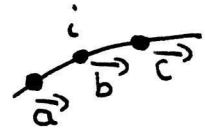
* indice(i,ii,jj,kk) :

Elle met respectivement dans ii, jj et kk, le numéro du point précédant, concomitant et succédant le point i. Elle permet donc de briser la disymétrie des équations aux noeuds 1 et n.



* discret(i, profil, a, b, c) :

Elle met respectivement dans a, b et c les coordonnées du point précédant, concom itant et succédant le point i. Elle se sert de la sous-routine indice et cherche les coordonnées dans la matrice Profil. Ces coordonnées sont utiles au calcul de la dérivée première et seconde à l'aide de dX_s et dX_{ss}.



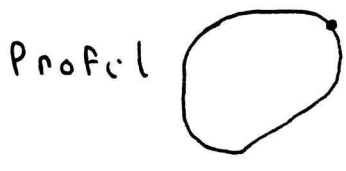
5 Fonctions de calcul sur le profil :

* sigma(a, c, l) : Elle permet de calculer $\left|\frac{c-a}{2\,ds}\right|$ en se servant de la routine dX_s.

Précédée de l'instruction : call discret(i, profil, a, b, c), elle permet donc de calculer $\sigma = |X_s|$ au point i.

* longueur(Profil, S) :

Elle calcule $S = \int_{-\pi}^{\pi} \sigma\, ds$ qui est la longueur de la ligne centrale de Profil.
Elle se sert des routines : discret et sigma.



* caractéristique (Profil, ...) :

En se servant des sous-routines des paragraphes 2 et 4, elle calcule :

X_s, X_{ss}, $G = |X_s|$, $\vec{\tau} = X_s/G$, $K\vec{b} = X_s \wedge X_{ss}/G^3$,
$K = |K\vec{b}|$, $\vec{n} = \vec{b} \wedge \vec{\tau}$, $T = -\frac{1}{G}\frac{\partial \vec{b}}{\partial s}.\vec{n}$ et $\frac{\partial K}{\partial s} = \frac{K^+ - K^-}{2\,ds}$

en tous les points de la fibre :

Profil  $G, \vec{\tau}, \ldots$

6 Fonctions de calcul liées à l'équation intégrodifférentielle :

61 Le terme $\ln(1/\varepsilon) + C_v(t) + C_w(t)$:

C'est la sous-routine : coefficient (Profil, t_1, coeff) qui le calcule.

C_w est déterminé par la formule :

$$C_w = -2\,\frac{m_0^2}{\Gamma^2}\,\frac{\varepsilon^2}{\delta^2}\left(\frac{S_0}{S}\right)^4$$

et C_v par : $C_v = \ln(\varepsilon/\delta) + 0.5(1 + \text{euler} - \ln 2)$

S_0 est calculé dans le programme principal à l'aide de la procédure longueur et est transmis à la procédure coefficient. De la même façon, S est calculé à partir de la matrice Profil à l'aide de la routine longueur.
On utilise également $\delta = (4\nu\tau_1^{(n+1)}/S)^{1/2}$ et $\nu = \varepsilon^2\Gamma$
Pour déterminer $\tau_1(n+1) = \int_0^{t=(n+1)dt} S(t')\,dt' + \tau_{10}$,
on se sert de $t_1 = \int_0^{t=ndt} S(t')\,dt' + \tau_{10}$ qui est passé en argument de la procédure et

on fait : $\tau_1(n+1) = t_1 + dt \times S$

La valeur de t_1 est initialisée dans le programme principal en faisant $t_1 = t_{10}$ après avoir calculé :

$$t_{10} = S_0 \quad (\text{car } \tau_{20} = 1)$$

La méthode numérique d'intégration utilisée est la méthode des trapèzes.

Il ne reste alors plus qu'à faire :

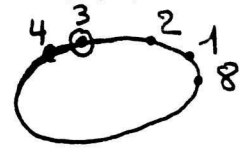
$$\text{coeff} = \tau_v + \tau_w + \ln(1/\varepsilon).$$

6.2 Le calcul de $|\lambda(t,s',s)| = \left| \int_s^{s'} G(t,s^*)\,ds^* \right|$:

* <u>permut(i,v,w)</u> :

Cette procédure permet de refaire une numérotation pratique des noeuds à partir du noeud i qui aura pour valeur 1.

Suivant que l'on numérote les noeuds dans le sens du paramétrage ou dans le sens contraire, on obtient le vecteur v ou le vecteur w de changement de numéros tels que l'on voit sur l'exemple suivant :



$$v = [\underset{1}{3}, \underset{2}{4}, \underset{3}{5}, \ldots\ldots, \underset{8}{1}, \underset{\ }{2}] \quad \begin{matrix} 1\ 2\ 3\ \ldots\ldots\ 8 \end{matrix}$$

: numérotation locale dite "positive"

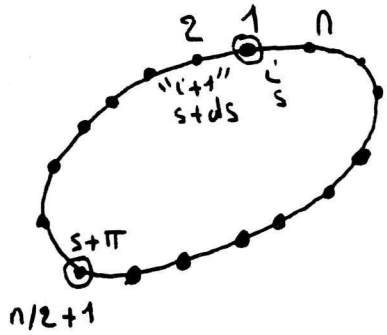
$$w = [3, 2, 1, 8, \ldots\ldots, 4]$$

: numérotation locale dite "négative".

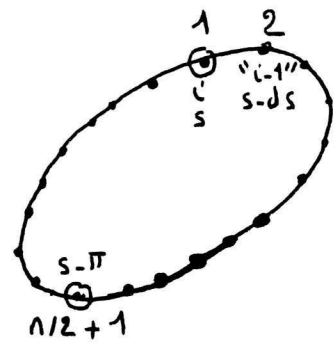
en positif

en négatif.

C'est une procédure très utile, car quelque soit le noeud i auquel on s'interesse, on le ramène comme si il était le noeud numéro 1 et il suffira donc d'écrire des procédures pour un noeud de numéro 1.



en positif



en négatif.

On choisit de prendre <u>n pair</u> afin que s+π et s-π correspondent à un point de la discrétisation de la fibre.

* <u>lambda(Profil, Fsigma, a, lamb) :</u>

Cette procédure sert à calculer la fonction :

$$|\lambda| = \left| \int_{s}^{s'} \sigma(t, s^*)\, ds^* \right|$$

en tous les points de la fibre pour des s' de l'intervalle $[-\pi, \pi]$

On fait : call permut(i,v,w) puis :

- Si $s' \in [0,\pi]$:

On utilise la numérotation positive : $a := v$ ($:=$ signe d'affectation) et on calcule λ par une méthode de trapèzes successivement appliquée du noeud 1 à tous les noeuds jusqu'au noeud $n/2+1$ et on range les différents résultats dans le vecteur lamb.

On obtient donc $|\lambda| = \text{lamb}(i+1) = \int_s^{s+ids} \sigma(t,s^*)\, ds^*$

- Si $s' \in [-\pi,0]$:

On utilise la numérotation négative : $a := w$ et on applique la même Formule des trapèzes que précédement.

On obtient donc : $|\lambda| = \text{lamb}(i+1) = -\int_s^{s-ids} \sigma(t,s^*)\, ds^*$

63 Le calcul de $\int_{-\pi}^{\pi} G(t,s+\bar{s},s)\, d\bar{s}$:

D'après la définition de G :

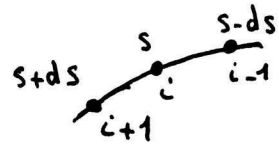
$$G(t,s',s) = F(t,s',s) - \sigma(s')\frac{K(s)}{2}\vec{b}(s)/|\lambda(s',s)| \qquad s' \neq s$$

$$= \frac{\vec{T}(s) \wedge \vec{B}(s)}{3}\,\sigma(s')\,\text{sgn}(s'-s) \qquad s' = s \pm 0$$

On doit donc choisir un voisinage V de s.

On choisit le voisinage suivant :

$$V = [s-ds, s+ds]$$



* La sous-routine Fonctg calcule alors :

$$F(t, s', s) = \frac{\sigma(s')\, K \vec{b}}{2} / |\lambda|$$

- sur $[0, \pi]$:

en se servant de la numérotation positive et de la fonction lamb calculée sur celle-ci.

- sur $[-\pi, 0]$:

en se servant de la numérotation négative et de la fonction lamb calculée sur celle-ci.

On détermine alors :

$$\int_{[-\pi,\pi]\setminus V} F - \frac{K\vec{b}\sigma}{2} / |\lambda| \, d\bar{s} = \int_{s+ds}^{s+\pi} F - \frac{K\vec{b}\sigma}{2} / |\lambda| \, d\bar{s} + \int_{s-\pi}^{s-ds} F - \frac{K\vec{b}\sigma}{2} / |\lambda| \, d\bar{s}$$

en calculant chacun des deux termes de droite par une méthode de trapèzes se servant de la fonction Fonctg. C'est la procédure : intégrale qui fait cette opération.

* La sous-routine Fonctglocal calcule $\frac{\sigma(s')\, \vec{T}(s) \wedge \vec{B}(s)}{3}$.

A l'aide de la procédure integrallocale, on calcule $\int_V \frac{\sigma(s')\, \vec{T}(s) \wedge \vec{B}(s)}{3}$ de la même façon que l'on a fait pour la sous-routine : intégrale.

64 Calcul du terme Q_2 :

$$Q_2(X) = \frac{1}{4\pi} \sum_{\substack{j=1 \\ j \neq i}}^{nbra} \Gamma_j \int_{C_j} \frac{X_j - X}{|X_j - X|^3} \wedge X_{j,s}\, ds$$

La sous-routine $fQ_2(j, profil, b, Q_2)$ calcule :

$\frac{X_j - b}{|X_j - b|^3} \wedge X_{j,s}$ au point b et range le résultat dans Q_2.

La sous-routine $termeQ_2$ calcule alors l'intégrale curviligne sur C_j au point b et met le résultat dans Q_2.

La sous-routine $sterme Q_2$ fait la sommation $\frac{1}{4\pi} \sum_{\substack{j=1 \\ j \neq i}}^{nbra}$ à l'aide de la routine $termeQ_2$ et range le résultat dans SQ_2.

65 Le calcul complet :

Le reste des calculs est fait dans la procédure schéma à l'aide des résultats de la procédure caractéristiques et des sous-routines d'intégration précédentes

La procédure recurrence effectue schéma successivement en chaque point de la discrétisation.

7 Calcul des coefficients de validité de la méthode asymptotique :

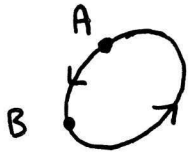
Il faut calculer $para1 = R_{min}/\delta$ et $para2 = d_{min}/2/\delta$ (δ est l'épaisseur de la fibre au temps t).

Le rayon de courbure minimum R_{min} est calculé par la procédure : rayon.

dmin est la distance minimum entre 2 points "distincts". C'est la procédure : distanceminimum qui fait le calcul du minimum de distance en parcourant tous les couples de points.

Deux points sont distincts si leur distance curviligne est supérieure à $4\pi R_{min}$. Pour voir si 2 points sont distincts, on se sert du calcul de distance curviligne de la procédure lambda etendu qui est la même procédure que lambda si ce n'est qu'elle calcule de 0 à 2π et de -2π à 0 au lieu de 0 à π et de $-\pi$ à 0.

On calcule alors l'intégrale curviligne de A à B : $\int_A^B \sigma ds$ puis l'intégrale curviligne de B à A : $\int_B^A \sigma ds$ et on prend le minimum des deux résultats que l'on compare ensuite à $4\pi R_{min}$



8 Programme principal

Il est décrit dans l'algorithme de l'annexe 9. Il se sert essentiellement de la procédure recurrence qui elle même se sert de la procédure schéma.

En plus de sortir le profil de l'anneau à différents temps, il donne la valeur des coefficients de validité para 1 et para 2

(on doit avoir para1>1 et para2>1) ainsi que le champ de vitesse de la fibre à différents temps.

2 Remarques de language:

On a programmé en fortran et on a jugé utile de faire les remarques suivantes:

- Il ne faut pas affecter une fraction à une variable, à une constante ou dans les arguments d'une procédure. Il faudrait écrire par exemple 1./3 et non la fraction 1/3.

- L'instruction parameter(aa=2D00) définit une constante aa qui est un réel non étendu (real*4 et non real*8!).

- Lors du passage des paramètres dans une sous-routine, les paramètres doivent avoir le même type que celui qui est défini dans la sous-routine.
  Ainsi, si on travaille en real*8 dans la sous-routine prodscal(k,a,b), il faut faire l'instruction: call prodscal(0.5D00, a, b) et non call prodscal(0.5, a, b)!
  Par contre, si l est un real*8, l=0.5 est bien compris.

- De la même façon, on doit toujours avoir les mêmes dimensions pour les matrices passées en paramètre d'une procédure et la déclaration de type matrice dans la procédure.
  On a choisit de surdimensionner nos matrices lors de leur déclaration par rapport à la dimension effective utilisée.

On définit donc, dans le programme principal, une constante de surdimension puis on affecte sa valeur à une variable que l'on a choisit de passer à l'intérieur des procédures à l'aide d'une instruction : common (pour ne pas surchargier la liste des arguments d'appel des procédures), afin de pouvoir surdi-mensionner les matrices des procédures avec la même valeur que dans le programme principal. De plus, il faut obligatoirement faire passer la matrice dans les arguments de la procédure : il n'y a pas possibilité de définir des matrices locales avec des surdimensions passées en paramètre.

On définit alors une variable n dans le programme principal qui correspond à la dimension effective des matrices (carrées) et on la fait passer aux sous-routines à l'aide de l'instruction common.

De cette manière, pour changer la surdimen-sion et la dimension des matrices dans tout le programme principal et dans toutes les sous-routines, il suffit de changer la constante de surdimension dans le programme principal ainsi que la variable de dimension du programme principal, en la modifiant dans le programme ou à l'aide d'un appel clavier : read (*, *) 'n=', n.

— Pour exécuter un programme à une centaine heure (de la nuit par exemple) en étant "délogué" d'une machine, il faut utiliser l'instruction unix : nohup.