Séance 4 : Programmation modulaire - Problèmes non linéaires

Pour faire ce TD, il faut prendre à l'adresse

http://www.inln.cnrs.fr/~laure/EF/TD2/Seance4/index.html

les fichiers suivant :

- `P1.sci` et `P2.sci`, des fonctions qui donnent les fonctions de Lagrange.
- `gauss.sci`, une fonction qui donne les points de Gauss pour l'intégration numérique.
- `maillage_1D.sci`, fonction qui calcule le maillage en 1D.
- `trace_u_1D.sci`, pour tracer la solution u .

Ex 1 *Ecoulement de Poiseuille*
On veut résoudre $-\nabla.(\mu\nabla u) = \nabla P$ sur un domaine $[0,1]$ avec une condition aux limites de Diriclet ($u(0) = u(1) = 0$).
Le terme ∇P (la perte de charge) est constant (pour simplifier, on peut prendre $\nabla P = 1$) et μ est fonction du taux de cisaillement.

1) Ecrire la formulation faible du problème.

2) Afin d'avoir un programme plus simple et plus modulaire, on a définit les tableaux `PG, XG, COOR, CONNEC, NUMER` et `ADRESS` comme des variables globales. De la même façon, on a écrit des petites fonctions qui font l'assemblage (`assemblage.sci`), le calcul du système élementaire (`cal_Melm_?`), le maillage (`maillage_1D.sci`) et le calcul de la solution dans le cas linéaire où μ est constant (`cal_sol_lin.sci`). Prendre ces programmes à la même adresse et lire le programme `exemple_1D` qui utilisent ces fonctions. Ce dernier programme calculent la solution du problème de Poiseuille dans le cas 1D pour $\mu = 1$.

3) On suppose que μ suit la loi de Carreau-Yasuda

$$\mu(\dot{\bar{\epsilon}}) = \mu_o \ (1 + (\lambda\dot{\bar{\epsilon}})^a)^{\frac{m-1}{a}} \qquad \text{avec} \qquad \dot{\bar{\epsilon}} = \left|\frac{\partial u}{\partial x}\right|$$

On peut prendre $\mu_0 = 1.$, $\lambda = 2.$, $m = .2$ et $a = 2$. Puis les valeurs pour le polyéthylène : $\mu_0 = 1.$, $\lambda = 13.47$, $m = 0.294$ et $a = .381$.

3.1) On va utiliser une méthode de point fixe. Modifier la fonction `cal_Melm_1D.sci` afin de calculer le système élémentaire.

3.2) Ecrire une fonction que l'on peut appeller `cal_sol_ptfixe.sci` qui appelle la fonction `cal_sol_lin.sci` et la fonction écrite en **3.1)** (par exemple `cal_Melm_1D_a.sci`).

3.3) Ecrire la formulation faible à résoudre si on veut utiliser une méthode itérative de Newton.
Ecrire une nouvelle fonction (par exemple `cal_Melm_1D_b`) qui calcule le système élementaire.
Ecrire la fonction qui calcule la solution par une méthode de Newton (`cal_sol_newton.sci`). Cette fonction appellera la fonction `cal_sol_lin.sci` et la fonction `cal_Melm_1D_b.sci`.

3.4) Comparer les deux méthodes quand on augmente λ.

4) On peut regarder le cas 2D avec maintenant

$$\dot{\bar{\epsilon}} = \sqrt{\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}}$$

Les programmes analogues à ceux du cas unidimensionnel se trouve à l'adresse