

Les méthodes numériques présentées en cours seront programmées à l'aide du logiciel **FreeFem++** (<http://www.freefem.org/ff++/index.htm>). Nous utiliserons en TP la version de FreeFem++ qui propose un environnement de développement intégré; ce logiciel est disponible en téléchargement (gratuit) à l'adresse <http://www.ann.jussieu.fr/~lehyaric/ffcs/index.htm>.

À la fin de cette séance vous enverrez vos programmes à mon adresse Email, avec L3-CS-TP1 en sujet et votre nom et prénom dans le corps du message.

TP 1 : Introduction à l'utilisation du logiciel FreeFem++ Interpolation numérique 1D



Exercice 1 Utilisation du système d'exploitation Linux

- Sur un PC de la salle informatique, taper votre nom "login" et votre mot de passe dans les cases appropriées présentes sur l'écran.
- Parcourir le menu qui se trouve à gauche, en bas de l'écran. Naviguer de sous-menu en sous-menu. Relâcher le bouton de la souris sur l'option choisie qui correspond à l'exécution d'une commande, c'est à dire à l'exécution d'un logiciel.
- Explorer la barre de lancement rapide qui se trouve en bas de l'écran. Trouver comment lancer : une fenêtre de commandes `terminal`, l'éditeur `gedit`, le navigateur `firefox`, etc.
- Dans une fenêtre Terminal, essayer les commandes de base

| Le chemin vers un dossier = PATH. | |
|--|---|
| On utilise / pour séparer les noms des dossiers. | |
| Raccourcis : Répertoire courant . Répertoire parent .. Répertoire maison ~ | |
| <code>pwd</code> | affiche le PATH vers le dossier courant |
| <code>ls</code> | liste le dossier courant |
| <code>ls -la</code> | liste complète de tous les fichiers, y compris les fichiers cachés |
| <code>mkdir nomdir</code> | crée le dossier nomdir |
| <code>cd PATH</code> | pour aller dans le dossier spécifié par le chemin PATH |
| <code>cd ..</code> | pour monter d'un niveau dans la hiérarchie de dossiers |
| <code>cd ~</code> | pour aller directement dans le dossier maison |
| <code>cp fic1 fic2</code> | copie le fic1 dans fic2, le dernier - s'il existe - étant écrasé |
| <code>cp PATH1/fic1 PATH2/fic2</code> | commande plus générale |
| <code>mv fic1 fic2</code> | le fichier fic1 change de nom, devenant fic2 |
| <code>mv PATH1/fic1 PATH2/fic2</code> | le fichier fic1 change de nom, devenant fic2 qui est déplacé dans PATH2 |
| <code>rm -i fic1 fic2</code> | efface les fichiers fic1 et fic2 tout en demandant la confirmation |
| <code>rmdir nomdir</code> | efface le dossier nomdir qui doit être vide |
| <code>du -sk nomdir</code> | espace disque occupé par le dossier nomdir |
| <code>df</code> | affiche l'espace disque occupé/restant |
| <code>cat fic</code> | affiche à l'écran le contenu du fichier fic |
| <code>more fic</code> | affiche à l'écran le contenu du fichier fic |
| <code>diff PATH1/fic1 PATH2/fic2</code> | affiche les différences entre les deux fichiers |
| <code>grep -ni nomchaine nomfic</code> | cherche une chaîne de caractères dans un fichier |

À retenir : Utiliser dans le terminal :

- la touche <Tabulation> pour compléter les commandes et les noms de répertoires ou fichiers.
- les touches <Flèche vers le haut> et <Flèche vers le bas> pour naviguer et rappeler les anciennes commandes déjà utilisées.



Exercice 2 Première utilisation du logiciel *FreeFem++*

- Lancer dans un Terminal la commande `FreeFem++-cs`.
- Explorer les menus du logiciel. Charger des exemples de programmes (File/Open example) et exécuter ces programmes ; regarder les options pour la représentation graphique.

Q1 Ecrire dans la partie "éditeur" du logiciel le programme suivant

```
real L1=2, L2=4;
int nbseg= 20;
real aspratio = L2/L1;

border Gamma1(t=0,L1) {label=1;x=t ;y=0;};
border Gamma2(t=0,L2) {label=2;x=L1;y=t;};
border Gamma3(t=L1,0) {label=3;x=t ;y=L2;};
border Gamma4(t=L2,0) {label=4;x=0 ;y=t;};

//=====
// Maillage
//=====

mesh Th=buildmesh(Gamma1(nbseg)+Gamma2(nbseg*aspratio)
                  +Gamma3(nbseg)+Gamma4(nbseg*aspratio));

plot(Th, wait=1);

//=====
// Une fonction elements finis
//=====
espace Vh(Th,P1); // espace P1
Vh f = x*x+y*y;

plot(f,fill=1,wait=1,cmm="Graph de f");

//=====
// Valeur au point P
//=====

real xp=L1/2, yp=L2/2;
cout << "valeur au point P(" << xp << "," << yp << ")=" << f(xp,yp) <<endl;
```

- Créer le dossier TP1 (commande `mkdir` dans un Terminal). Sauver le programme comme `TP1_ex2_q1.edp`, dans le dossier TP1.

Q2 Rajouter un trou circulaire au domaine. Sauver le nouveau programme comme TP1_ex2_q2.edp.

Q3 En utilisant la représentation paramétrique des courbes frontières, construire le maillage d'un *smiley*. Sauver le nouveau programme comme TP1_ex2_q3.edp.



Exercice 3 Interpolation linéaire par morceaux (1D)

Considérons un échantillonnage de la fonction $f(x) : [a, b] \rightarrow \mathbb{R}$, représenté par n valeurs connues $y_i, i = 0, \dots, n-1$, correspondant aux abscisses $a = x_0 < x_1 < \dots < x_{n-1} = b$.

La fonction $f(x)$ sera approchée par la fonction linéaire par morceaux définie sur l'intervalle $[a, b]$ par

$$\tilde{f}(x) = y_i + \frac{x - x_i}{x_{i+1} - x_i} (y_{i+1} - y_i), \quad \forall x \in [x_i, x_{i+1}] \quad \forall i = 0, \dots, n-2 \quad (1)$$

- Commençons par définir un tableau `xx` contenant les abscisses x_i équidistantes, suivant la syntaxe

```
// abscisses
real a=0, b=1, pas=0.2;
real [int] xx(a:pas:b);
cout <<"abscisses " << xx << endl;
cout <<"dimension vecteur n=" << xx.n << endl;
```

- Définissons maintenant le tableau `yy` contenant les valeurs de la fonction

```
// valeurs
int n=xx.n ; // dimension du vecteur
real [int] yy(n);
for(int i=0; i < n; i++)
    yy(i) =sin(pi*xx(i))+cos(pi*xx(i));

cout <<"valeurs yy " << yy << endl;
```

- L'échantillonnage sera sauvé dans le fichier `data_ex3.dat`

```
// ecriture dans un fichier
{
ofstream fout("data_ex3.dat");
for(int i=0; i < n ; i++)
    fout << xx(i) << " " << yy(i) << endl ;
}
```

- Pour la représentation graphique, nous allons utiliser un autre logiciel, Gnuplot, car FreeFem++ n'est pas adapté aux tracés 1D.

Dans un Terminal, commencez par vous placer dans le dossier contenant les programmes écrits (cd TP1).

Lancez la commande `gnuplot`; vous êtes maintenant dans l'environnement Gnuplot, avec des commandes spécifiques. Pour visualiser la courbe, testez les commandes suivantes :

```
plot "data_ex3.dat"
plot "data_ex3.dat" w l
plot "data_ex3.dat" w lp
plot "data_ex3.dat" w lp pt 7
```

Commentez les commandes suivantes :

```
set xrange [0:1]
plot "data_ex3.dat" w lp pt 7, sin(pi*x)+cos(pi*x)
```

Changez le pas de discrétisation de 0.2 à 0.02 et refaire tous les calculs. Commenter.

Q1 Compléter le programme afin de pouvoir réaliser l'interpolation linéaire :

- définir une variable xP pour l'abscisse du point d'interpolation, $a < xP < b$;
- écrire un algorithme qui va trouver le sous-intervalle tel que $xP \in [xx_i, xx_{i+1}]$ (utiliser le Help pour trouver la syntaxe des instructions `if` et `break`);
- appliquer la formule (1) pour calculer la valeur interpolée fxP ;
- comparer avec la valeur exacte;
- vérifier graphiquement que la valeur trouvée est bien l'interpolé linéaire.

Sauver le nouveau programme comme `TP1_ex3_q1.edp`.

Programmes à envoyer par Email :

TP1_ex2_q2.edp

TP1_ex2_q3.edp

TP1_ex3_q1.edp