MM031 - TP6
Résolution d'équations différentielles

Maxime Chupin : chupin@ann.jussieu.fr

03 février 2016

Brève introduction à gnuplot

Wikipédia : Gnuplot est un logiciel qui sert à produire des représentations graphiques en deux ou trois dimensions de fonctions numériques ou de données. Le programme fonctionne sur de nombreux ordinateurs et systèmes d'exploitation (Linux, Windows, OS/2, VMS...) et peut envoyer les graphiques à l'écran ou dans des fichiers dans de nombreux formats.)

Une très bonne introduction à `gnuplot` se trouve à l'adresse suivante : `http://perso.ensta-paristech.fr/~kielbasi/docs/gnuplot.pdf`.

0.1 Utilisation basique

Nous allons utiliser `gnuplot` sur un script que nous allons faire écrire par notre programme C++. De plus, nous allons utiliser la fonction `system()` pour exécuter `gnuplot` sur notre script avec la commande suivante :

```
> gnuplot  monscript.gnuplot
```

Script de base Nous donnons ici un script de base permettant de représenter une fonction de $\mathbf{R}$ dans $\mathbf{R}$, plus précisément sur un segment Ω.

Il s'agit ici de lire un fichier de données (discrétisation du segment et des valeurs de la fonction). Supposons que le fichiers de données se nomme `fichier.dat` et contient deux colonnes de valeurs, la première pour la discrétisation du segment Ω et la deuxième pour les valeurs de la fonction en ces ponts.

Le tracé par `gnuplot` est alors très simple. Voici un exemple de script :

```
1 # on définit le terminal de sortie
2 set terminal x11 persist
3 # on trace
4 plot "monfichier.dat" using 1:2 with lines title "Tracé de $f$"
```

La ligne 2 `set terminal` permet d'indiquer la sortie que l'on souhaite. Ici il s'agit du serveur `x11` qui est en fait l'affichage d'une fenêtre sous Unix. On aurait pu spécifier un format d'image (`jpeg`, `png`, etc.) ou bien même du code LaTeX...Beaucoup de choses sont possibles.

Le `using 1:2` est ici facultatif, il indique que l'on trace la colonne 2 en fonction de la colonne 1. Si notre fichier de données contenait plus de colonnes, on pourrait indiquer les colonnes que l'on considère.

Le reste se comprend aisément, on peut bien entendu beaucoup plus paramétrer le tracé avec de nombreuses options que nous vous laissons rechercher (notamment la couleur de tracé, etc.).

Mise en oeuvre de la méthode des différences finies

Exercice 1

Soit $\Omega = [0, 1]$, on cherche numériquement $u \in H_0^1(\Omega)$ solution de l'équation

$$-\Delta u = f.$$

Ici $f(x)$ est un terme source et u est nulle sur le bord du domaine.

1. Définir la discrétisation de l'espace Ω en n points.
2. En utilisant la méthode des différences finies à l'ordre deux, mettre l'équation sous la forme d'un système linéaire.
3. En utilisant la classe matrice du TP précédent, résoudre le système linéaire à l'aide de la décomposition LU et de la méthode de Cholesky, pour plusieurs termes sources différents. (on pourra prendre par exemple des gaussiennes $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ pour $\sigma = 0.01$ et $0.1 < \mu < 0.9$).
4. Écrire dans des fichiers de sortie les solutions afin de les visualiser avec `gnuplot`.
5. Pour lancer un script gnuplot depuis votre programme C++ vous pouvez utiliser la fonction `system` après avoir inclus en entête `<stdlib.h>`. Ensuite, la ligne

   ```
   system ("gnuplot script.gp");
   ```

   dans votre code C++ permettra de lancer gnuplot sur le script `script.gp` s'il est placé dans le même répertoire que votre exécutable.
6. Comparer les temps de résolution de la décomposition LU et de Cholesky.

Exercice 2

On veut maintenant résoudre l'équation

$$-c\Delta u + u = f \text{ sur } \Omega = [0, 1]$$

avec $f(x) = \sin(2\pi x)$ et u nul sur le bord du domaine.

1. Calculer la solution exacte du problème.
2. Résoudre comme dans l'exercice précédent l'équation avec des différences finies, la décomposition LU et la méthode de Cholesky.
3. Écrire un fichier de sortie lisible par `gnuplot` permettant la visualisation de la solution exacte, de la solution approchée et de la différence en valeur absolue entre les deux.
4. Comparer les temps de résolution de la décomposition LU et de Cholesky.