TP 2: Résolution de l'équation stationnaire de la chaleur en 1D

Le but de ce TP est de calculer la répartition de température dans une barre. On se place dans un cas où le temps est considéré comme n'intervenant plus dans le phénomène de diffusion. Le problème devient alors stationnaire.

L'équation stationnaire de la chaleur en 1D sera modélisée à l'aide d'une équation aux dérivées partielles avec les conditions aux limites suivantes:

$$\begin{cases} -\Delta u &= f & \text{dans } \Omega = [0,1] \\ u &= 0 & \text{sur les bords} \end{cases} \tag{1}$$

où f désigne une source de chaleur. Il s'agit donc ici d'une équation de Poisson dans le cas d'une source non nulle, et d'une équation de Laplace dans le cas contraire.

Question 1: Formation du système linéaire

Nous commençons par construire un maillage du domaine, c'est-à-dire un recouvrement de Ω par des segments de taille Δx. La réunion des segments est égale à Ω, l'intersection de deux segments distincts est soit vide, soit un sommet commun.

On utilise le schéma différences finies suivant:

$$\frac{-u_{i-1} + 2u_i - u_{i+1}}{\Delta x^2} = f_i \qquad \forall i / 1 \leq i \leq N \tag{2}$$

avec N le nombre de points suivant chaque direction et

$$\Delta x = \frac{1}{N+1} \tag{3}$$

La matrice A associée est tridiagonale:

$$A = (N+1)^2 \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \tag{4}$$

La matrice A est creuse, elle a, à priori, 3 éléments non nuls par ligne (sauf pour les points du bord où il y en a moins). A est symétrique définie positive (valeurs propres strictement positives).

Question 1.1 Implémentez une sous-routine qui génère la matrice A pour un N donné en argument.

Question 1.2 Vérifiez que la matrice est correcte pour $N = 2, 3, 4$.

Question 2: Résolution du système linéaire

On cherche maintenant la solution du système :

$$AX = F \tag{5}$$

où A est la matrice de la question précédente (de taille $N \times N$). X et F sont des vecteurs de taille N.

Question 2.1 A est symétrique définie positive, donc elle peut être mise sous la forme du produit d'une matrice triangulaire inférieure L et de sa transposée L^T:

$$A = LL^T \tag{6}$$

Pour ce faire, nous allons utiliser la factorisation de Cholesky. Implémentez une sous-routine pour effectuer la décomposition de Cholesky de la matrice A. Testez cette sous-routine en vérifiant que la norme de $A - LL^T$ est de l'ordre de la précision machine (en prenant $N = 2, 3, 4$).

Question 2.2 Choisissez comme second membre un vecteur F de la forme

$$F = (1, \cdots, 1, 0, \cdots, 0) \tag{7}$$

où les $N/2$ premières valeurs sont égales à 1.

Sous forme matricielle condensée, le système $Ax = F$ devient donc $LL^T = F$. Implémentez une sous-routine pour résoudre $Ax = F$ à partir des facteurs L et L^T précédemment calculés. On suppose ici que L et L^T sont donnés directement en argument de la sous-routine, et donc on doit résoudre deux systèmes triangulaires successifs puisque:

$$Ax = F \Longleftrightarrow \left\{ \begin{array}{c} Ly = F \\ L^T x = y \end{array} \right. \tag{8}$$

Tester cette sous-routine sur un second membre aléatoire et vérifier que la norme de $Ax - F$ est de l'ordre de la précision machine (en prenant $N = 2, 3, 4$).

Question 3: Visualisation des résultats

Question 3.1 Ecrivez la solution (qui est un vecteur de taille N donc) dans un fichier gnuplot.

Question 3.2 Visualisez le résultat avec gnuplot pour plusieurs valeurs de N, comme $N = 100$ et $N = 1000$.

Question 4: Optimisation de la résolution numérique

Question 4.1 La matrice L a elle aussi une structure creuse que l'on peut utiliser pour réduire le nombre d'opérations. Du fait que la factorisation de Cholesky ne nécessite pas de pivot (donc pas de permutation des lignes ou colonnes en cours de calcul), il est possible de prévoir à l'avance la structure creuse du facteur de Cholesky L. Implémentez une sous-routine pour effectuer la décomposition de Cholesky de la matrice A exploitant la structure creuse de L.

Question 4.2 Mesurez et comparez les temps CPU obtenus avec les décompositions pleine et creuse de Cholesky de la matrice A. Pour mesurer le temps CPU d'une portion de votre code, vous pouvez utiliser la fonction `CPU_TIME` de la manière suivante:

```
real::t1,t2

call CPU_TIME(t1)

. . . .

call CPU_TIME(t2)

print '("Time = ",f6.3," seconds.")',t2-t1
```

Question 5: Analyse de la qualité des résultats numériques

Question 5.1 On cherche souvent à évaluer l'erreur numérique commise par une méthode sous la forme :

$$\|u_{numerique} - u_{exact}\| = C \times N^p + O(N^{p+1}) \tag{9}$$

On appelle p l'ordre de convergence de la méthode. Notez que sous forme logarithmique, l'erreur est linéaire (en négligeant le terme d'ordre supérieur $O(N^{p+1})$):

$$\log(E) \simeq p \times \log(N) + C' \tag{10}$$

Le but de cette question est de quantifier la qualité de la solution numérique calculée à l'aide de la connaissance d'une solution analytique exacte. Considérez par exemple la solution analytique $\widehat{u}(x) = -x^4 + x^2$. On remarque que $\widehat{u}(x)$ est nulle en $x = 0$ et en $x = 1$, et que $u''(x) = 12x^2 - 2$. Le protocole de test est le suivant:

- Résolvez l'équation (1) en prenant $f(x) = 12x^2 - 2$.
- Tracez avec gnuplot la solution u obtenue et comparez-la visuellement à la solution analytique exacte $\widehat{u}(x) = x^4 - x^2$ pour différentes valeurs de N.
- Calculez la norme de la différence $u - \widehat{u}(x)$ pour les 3 valeurs de N suivantes: $N = 5$, $N = 50$ et $N = 100$. On notera $E(5)$, $E(50)$, $E(100)$ les 3 valeurs d'erreurs obtenues. Recopiez ces 3 valeurs à la main dans un fichier `convergence.dat` de la façon suivante:

  ```
  5   E(5)
  50  E(50)
  100 E(100)
  ```

  Vous pouvez visualiser la courbe de convergence via gnuplot :

  ```
  set logscale
  plot 'convergence.dat'
  ```

  Pour évaluer l'ordre de convergence, gnuplot permet de réaliser une régression linéaire avec ces trois points :

```
E(x) = p*x + C
fit E(x) 'convergence.dat' u (log($1)):(log($2)) via p, C
replot exp(C)*x**p
```

N.B. : Pour le calcul de l'erreur, on pourra utiliser la norme infinie ou/et la norme 2.

Question 5.2 Le but de cette question est de quantifier la qualité de la solution numérique calculée lorsque l'on ne dispose pas de la connaissance d'une solution analytique exacte. Le protocole de test est le suivant:

- Estimez l'erreur en calculant la différence entre deux solutions numériques obtenues respectivement avec un maillage grossier (de taille N) et un maillage deux fois plus fin.
- Comme précédemment, tracez avec gnuplot la courbe d'évolution de l'erreur en fonction du pas d'échantillonnage de Ω et calculez à nouveau l'ordre de la convergence.