Informatique en CPGE (2017-2018)
TD 8 : systèmes linéaires

1 Résolution d'un système tridiagonal

L'objectif est de résoudre un système du type :

$$\begin{pmatrix} b_0 & c_0 & 0 & \dots & \dots & 0 & 0 \\ a_1 & b_1 & c_1 & 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & .. & . \\ . & & & & & & . \\ . & & & & & & . \\ . & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & a_{n-2} & b_{n-2} & c_{n-2} \\ 0 & 0 & \dots & \dots & 0 & a_{n-1} & b_{n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ . \\ . \\ . \\ . \\ x_{n-2} \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ . \\ . \\ . \\ . \\ f_{n-2} \\ f_{n-1} \end{pmatrix}$$

Le système s'écrit sous la forme $MX = F$ et nous convenons que M est inversible.

M est une matrice tridiagonale définie par trois vecteurs de taille n, qui sont les diagonales a, b, c si nous posons a[0] = 0 et c[n-1] = 0, et F est le second membre, un vecteur de taille n.

Nous supposons n = 10.

1. Ecrire la définition des vecteurs a, b et c, trois listes de longueur n, en utilisant le procédé suivant :
   - donner des valeurs aléatoires réelles entre 0 et 1 aux éléments de a et c ;
   - corriger : a[0] = 0 ; c[n-1] = 0 ;
   - calculer : b[i] = a[i] + c[i] + 1, 0 <= i < n.
2. Ecrire la définition de la matrice M du système en utilisant une liste de n listes de longueur n.
3. Ecrire la définition du vecteur F de taille n, défini par : F[i] = a[i] + b[i] + c[i], 0 <= i < n.
4. Résoudre le système avec la fonction solve de scipy.linalg.
   Quels valeurs doit-on obtenir pour X[i] ? Tester le programme avec différentes valeurs de n.

2 Décomposition LU

La décomposition LU est une méthode de résolution efficace dans le cas d'un système linéaire à matrice tridiagonale. On peut la considérer comme une variante de la méthode de Gauss.

La matrice M peut se décomposer sous la forme $M = LU$ avec

$$L = \begin{pmatrix} b'_0 & 0 & 0 & \dots & \dots & 0 & 0 \\ a_1 & b'_1 & 0 & 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & .. & . \\ . & & & & & & . \\ . & & & & & & . \\ . & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & a_{n-2} & b'_{n-2} & 0 \\ 0 & 0 & \dots & \dots & 0 & a_{n-1} & b'_{n-1} \end{pmatrix} \quad \text{et} \quad U = \begin{pmatrix} 1 & c'_0 & 0 & \dots & \dots & 0 & 0 \\ 0 & 1 & c'_1 & 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & \dots & \dots & .. & . \\ . & & & & & & . \\ . & & & & & & . \\ . & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & 0 & 1 & c'_{n-2} \\ 0 & 0 & \dots & \dots & 0 & 0 & 1 \end{pmatrix}$$

Par identification nous obtenons les relations de récurrence :

$b'_0 = b_0$, $c'_0 = c_0/b_0$, et pour $k = 1, \dots, n-1$: $b'_k = b_k - a_k \times c'_{k-1}$ et $c'_k = c_k/b'_k$.

Le système $MX = F$ est alors équivalent au système $LUX = F$, soit $LY = F$, et peut se résoudre en deux étapes : $LY = F$, soit $y_0 = f_0/b'_0$, et pour $k = 1, \ldots, n-1$, $y_k = (f_k - a_k \times y_{k-1})/b'_k$;
puis $UX = Y$, soit $x_{n-1} = y_{n-1}$, et $x_k = y_k - c'_k \times x_{k+1}$, pour $k = n-2, \ldots, 1$

Les trois listes a = $[a_0, \ldots, a_{n-1}]$, b = $[b_0, \ldots, b_{n-1}]$, et c = $[c_0, \ldots, c_{n-1}]$, représentent les trois diagonales, en posant $a_0 = 0$ et $c_{n-1} = 0$.

1. Ecrire une fonction **factor_LU** qui prend en argument trois listes a, b et c comme ci-dessus et renvoie deux listes b1 et c1 construites avec les relations de récurrence données plus haut.
2. Ecrire une fonction **res_LU** qui prend en argument la liste a, les deux listes renvoyées par la fonction **factor_LU** et la liste représentant le second membre du système et renvoie la liste x solution du système.
3. Ecrire une fonction **solution** qui prend en argument quatres listes a, b, c, et F représentant le système et renvoie la liste x solution du système.
4. Tester le programme avec les listes définies dans la partie 1.
5. Comparer les temps d'exécution pour n = 1000, puis n = 5000. Déterminer à partir de quelle valeur de n la fonction **solve** n'est plus utilisable. Commentaires ?