Examen de TP Analyse Numérique L3 Mathématiques, 5 décembre 2014
Seuls l'aide mémoire scilab distribué et l'aide en ligne de scilab sont autorisés.
A la fin de l'examen, remettre votre feuille d'examen avec les réponses aux questions et le code scilab recopié sur votre feuille.

Exercice 1 : Variante de la méthode de Newton en dimension 1

La méthode suivante permet d'approcher numériquement les solutions de problèmes du type $F(y) = 0$ où F est une fonction de $\mathbb{R}$ dans $\mathbb{R}$. Elle consiste à construire une suite $(y^{(k)})_{k\in\mathbb{N}}$ telle que

$$y^{(0)} \in \mathbb{R} \text{ donné}, \quad \text{et} \quad y^{(k+1)} = y^{(k)} - \frac{\big(F(y^{(k)})\big)^2}{F\big(y^{(k)} + F(y^{(k)})\big) - F(y^{(k)})}, \quad \forall k \geqslant 0.$$

1) Ecrire la fonction scilab qui prend y en entrée et donne $F(y)$ en sortie avec

$$F(y) = y - \cos\Big(y + \frac{3\pi}{8}\Big).$$

La constante π est obtenue dans scilab avec la commande *%pi*.

2) Ecrire le code scilab qui, étant donnés $y^{(0)} = -10$, $eps = 10^{-10}$ et $nitmax = 100$, calcule successivement les itérés $y^{(k+1)}$, $k \geqslant 0$ tant que

$$\frac{|F(y^{(k)})|}{|F(y^{(0)})|} > eps \text{ et } k < nitmax.$$

La forme programmée dans scilab ne stokera que l'itéré courant y et calculera les valeurs absolues des résidus relatifs successifs $\frac{|F(y^{(k)})|}{|F(y^{(0)})|}$ ainsi que le nombre d'itérations à convergence. La valeur absolue de $y \in \mathbb{R}$ en scilab s'obtient par la commande $abs(y)$. On utilisera une boucle *while condition then ... end*.

3) Pour la fonction F implémentée à la question 1) et pour le choix $y^{(0)} = -10$, donner l'approximation de la solution de $F(y) = 0$ obtenue avec $eps = 10^{-10}$, $nitmax = 100$. Donner les résidus relatifs $\frac{|F(y^{(k)})|}{|F(y^{(0)})|}$ successifs et le nombre d'intérations obtenus. Commentez la nature de la convergence obtenue de l'algorithme.

Exercice 2 : Méthode SOR (Successive Over Relaxation)

La méthode SOR est une méthode itérative de résolution des systèmes linéaires. Considérons le système $Ax = b$ où $A \in \mathcal{M}_n(\mathbb{R})$ et $x, b \in \mathbb{R}^n$ $(n \geqslant 1)$. La matrice A et le vecteur b sont donnés alors que $x \in \mathbb{R}^n$ est l'inconnue du problème. Le paramètre $\omega \in]0, 2]$ s'appelle le paramètre de relaxation. Il s'agit de construire une suite $(x^{(k)})_{k\in\mathbb{N}}$ de vecteurs de $\mathbb{R}^n$ définie comme suit

$$x^{(0)} \in \mathbb{R}^n \text{ donné}, \quad \text{et} \quad \big(\frac{D}{\omega} - F\big)x^{(k+1)} = \big(\frac{D}{\omega} - F\big)x^{(k)} + (b - Ax^{(k)}), \quad \forall k \geqslant 0.$$

où D, $-E$, $-F$ représentent (respectivement) les parties diagonale, inférieure stricte et supérieure stricte de $A = D - E - F$.

1) Montrer que les relations suivantes sont vérifiées à chaque itération $k \geqslant 0$: pour tous $i = n, \cdots, 1$

$$\tilde{x}_i^{(k+1)} = \frac{1}{A_{ii}}\Big(b_i - \sum_{j=i+1}^{n} A_{ij}x_j^{(k+1)} - \sum_{j=1}^{i-1} A_{ij}x_j^{(k)}\Big).$$

et

$$x_i^{(k+1)} = \omega\tilde{x}_i^{(k+1)} + (1-\omega)x_i^{(k)}.$$

C'est cette expression qui sera utilisée pour coder l'algorithme SOR en menant les calculs dans l'ordre des i décroissants à chaque itération k.

2) Nous souhaitons calculer successivement les éléments de la suite $(x^{(k)})_{k\in\mathbb{N}}$ en utilisant une seule variable x (vectorielle avec n lignes et 1 colonne) et un seul réel de stockage noté xti. Expliquer rapidement pourquoi l'expression de la question 1 est intéressante pour arriver à cet objectif. Supposons que la variable x contienne le vecteur $x^{(k)}$, écrire quelques lignes de code scilab (utilisant une boucle for) qui permettent de modifier successivement les lignes de la variable x pour que cette dernière contienne *in fine* $x^{(k+1)}$ (la matrice A, le second membre b, n et le paramètre de relaxation ω sont supposés donnés).
La boucle for pour $i = n$ à 1 par ordre décroissant s'écrit en scilab : $for\ i = n : -1 : 1$ ... end.

3) Ecrire le code scilab qui, étant donnés n, A, b, ω, $x^{(0)}$, eps et $nitmax$, calcule successivement les itérés $x^{(k+1)}$, $k \geqslant 0$ tant que $\frac{||b-Ax^{(k)}||_2}{||b-Ax^{(0)}||_2} > eps$ et $k < nitmax$. On prendra $x^{(0)} = 0$ soit donc une initialisation avec $x = zeros(n,1)$ en scilab. La forme programmée dans scilab ne stokera que l'itéré courant x et calculera les normes des résidus relatifs successifs $\frac{||b-Ax^{(k)}||_2}{||b-Ax^{(0)}||_2}$ et le nombre d'itérations nit effectuées.

4) Nous choisissons dans cette question

$$A = \begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & -1 \\ 1 & -1 & 2 \end{pmatrix}, \quad \text{et} \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Pour cette matrice A et ce second membre b, donner la solution du système $Ax = b$ calculée par l'algorithme SOR pour $eps = 10^{-10}$, $nitmax = 100$, $\omega = 1$. Indiquer le nombre d'itérations nécessaires pour arriver à convergence lorsque l'on fixe $eps = 10^{-10}$ et $\omega = 1$. Donner, par l'expérience numérique, une valeur de $\omega \in]0,2]$ qui améliore la convergence de l'algorithme (nombre d'itérations réduit) par rapport au choix $\omega = 1$.

5) Donner le coût d'une itération de l'algorithme SOR en nombre d'opérations flottantes ($+$, $-$, $\times$, $/$) en fonction de n (en supposant la matrice A pleine).