MT09-TP4

1. (a) Soit A une matrice carrée inversible, écrire une fonction scilab

   `[x, k] = gausseidel(A, b, tol, N, x0)`

   qui, étant donné la matrice A et le vecteur b, calcule la solution de $Ax = b$ par la méthode de Gauss-Seidel, et où

   - tol est la tolérance utilisée pour le test d'arrêt. On pourra tester par exemple
   $$\frac{\|x^{(k+1)} - x^{(k)}\|_2}{\|x^{(k+1)}\|_2} < tol,$$
   - N est un nombre maximum d'itérations,
   - x_0 est un vecteur initial,
   - k est le nombre d'itérations effectuées pour obtenir la précision voulue.

   Pour arrêter l'exécution de la fonction en cas de convergence, vous pouvez utiliser la commande `return`.

   Lorsque le nombre maximal d'itérations est atteint avant que la précision voulue ne soit obtenue, afficher un message d'erreur signalant que la méthode de Gauss-Seidel n'a pas convergé.

   (b) Tester GS avec $A = \begin{pmatrix} 1 & -2 & 2 \\ -1 & 1 & -1 \\ -2 & -2 & 1 \end{pmatrix}$ $b = \begin{pmatrix} 7 \\ -4 \\ 2 \end{pmatrix}$, puis $A = \begin{pmatrix} 1 & -1 & -2 \\ -2 & 1 & 3 \\ 0 & 2 & 1 \end{pmatrix}$ $b = \begin{pmatrix} -2 \\ 3 \\ 0 \end{pmatrix}$.

   Que se passe-t-il ?

   (c) Application : on définit la matrice $A \in \mathcal{M}_{nn}$ et le vecteur b tels que
   $$A = \begin{pmatrix} 3 & 1 & 0 & \dots & \dots & 0 \\ 1 & 3 & 1 & 0 & \dots & 0 \\ 0 & 1 & 3 & 1 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 3 & 1 \\ 0 & \dots & \dots & 0 & 1 & 3 \end{pmatrix} \text{ et } b = \begin{pmatrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{pmatrix}.$$

   i. Montrer que la méthode de Gauss-Seidel pour résoudre $Ax = b$ va converger.

   ii. Écrire une fonction `[x, k] = tridiag(n)` dépendant de la taille de la matrice n, qui construit la matrice A (on pourra utiliser la commande `diag`) et le vecteur b, puis qui détermine la solution de $Ax = b$ par la méthode de Gauss-Seidel.
   Tester $n = 10, 100, 1000$.

2. On veut résoudre $f(x) = 0$ par la méthode de Newton, x étant le vecteur inconnu de $\mathbb{R}^n$ et f est fonction connue de $\mathbb{R}^n$ dans $\mathbb{R}^n$.

   Écrire une fonction scilab :

   `[x,k]=newton (foncjac,tol,N,x0)`

   qui calcule une solution de $f(x) = 0$ par la méthode de Newton, et où

   - tol est la tolérance utilisée pour le test d'arrêt. On pourra tester par exemple
   $$\frac{\|x^{(k+1)} - x^{(k)}\|_2}{\|x^{(k+1)}\|_2} < tol,$$
   - N est un nombre maximum d'itérations,
   - x_0 est un vecteur initial,
   - `foncjac` est une fonction scilab

     `[F,J]=foncjac(x)`

     qui, étant donné le vecteur x, calcule le vecteur F défini par $F = f(x)$, et la matrice jacobienne J définie par $J = Df(x)$ (on écrira cette fonction par la suite).
   - k est le nombre d'itérations effectuées pour obtenir la précision voulue.

Pour arrêter l'exécution de la fonction en cas de convergence, vous pouvez utiliser la commande `return`.

Lorsque le nombre maximal d'itérations est atteint avant que la précision voulue ne soit obtenue, afficher un message d'erreur signalant que la méthode de Newton n'a pas convergé.

3. (a) On définit $f(x_1, x_2) = \begin{pmatrix} x_1^2 + x_2^2 - 4 \\ x_2 - e^{x_1} \end{pmatrix}$. Lorsque l'on résout $f(x) = 0$, quelle intersection de courbes étudie-t-on ? Écrire une fonction scilab `courbe(N)` qui permet de représenter graphiquement ces 2 courbes (N est le nombre de points utilisés pour le tracé).

   (b) Écrire la fonction `foncjac` dans ce cas.

   (c) Utiliser la fonction `newton` pour résoudre $f(x) = 0$ en choisissant plusieurs vecteurs initiaux.

4. On veut résoudre le problème

$$\text{Trouver la fonction } u \text{ vérifiant}$$
$$\begin{cases} -u''(x) + g(u(x)) = b(x), \text{ pour } x \in]0,1[\\ u(0) = \alpha \\ u(1) = \beta \end{cases}$$

où α et β sont des réels connus et g et b sont des fonctions réelles données.

Pour cela on discrétise, n étant donné : on pose $h = \frac{1}{n},\ x_i = ih$ et v_i l'approximation de $u(x_i)$.

On est conduit à la résolution de $f(v_1, v_2, \ldots, v_{n-1}) = 0$ avec

$$f(v_1, v_2, \ldots, v_{n-1}) = \begin{pmatrix} -\alpha + 2v_1 + h^2 g(v_1) - v_2 - h^2 b(x_1) \\ \ldots \\ -v_{i-1} + 2v_i + h^2 g(v_i) - v_{i+1} - h^2 b(x_i) \\ \ldots \\ -v_{n-2} + 2v_{n-1} + h^2 g(v_{n-1}) - \beta - h^2 b(x_{n-1}) \end{pmatrix}$$

   (a) Ecrire la fonction `foncjac` dans ce cas.
   Remarque : les fonctions qui à un vecteur $(u_1\ u_2\ \ldots\ u_{n-1})^T$ associent respectivement les vecteurs
$$\begin{pmatrix} b(u_1) \\ b(u_2) \\ \vdots \\ b(u_{n-1}) \end{pmatrix}, \begin{pmatrix} g(u_1) \\ g(u_2) \\ \vdots \\ g(u_{n-1}) \end{pmatrix}, \begin{pmatrix} g'(u_1) \\ g'(u_2) \\ \vdots \\ g'(u_{n-1}) \end{pmatrix}$$
   peuvent être définies à l'aide de la commande `deff` (voir le polycopié page 27). Revoir également les opérations élément par élément définies page 15.
   On choisira $\alpha = 5$, $\beta = 5$, $b(x) = -x(x-1)$ et $g(x) = \dfrac{10x}{1+x}$.

   (b) Résoudre le problème par la méthode de Newton : on choisira $n = 10$ puis $n = 20$.

   (c) Ecrire une fonction scilab permettant de tracer sur une même figure
   - d'une part les points de coordonnées $(x_0, \alpha), (x_1, v_1), (x_2, v_2)..., (x_9, v_9), (x_{10}, \beta)$ obtenus pour $n = 10$,
   - d'autre part les points de coordonnées $(x_0, \alpha), (x_1, v_1), (x_2, v_2)..., (x_{19}, v_{19}), (x_{20}, \beta)$ obtenus pour $n = 20$,

   Comparer.