Série 9

Cette serie est la même chose que le Notebook 4.2 et est à compléter le mardi 29 avril. Le mardi 15 avril en classe il y aura une introdution aux m´thodes itératives.

Exercice 1

(Notebook 4.2, Exemple 1)
On considère la matrice A et le terme de droite $\mathbf{b}$ suivants

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \end{pmatrix}.$$

1. A l'aide de Python, calculez les premières itération de la méthode de Jacobi. Est-ce que les itérations convergent ?
2. Calculez la matrice d'iteration associée et son rayon spectral, ensuite expliquez le résultat du premier point
3. Répetez l'exercice avec la méthode de Gauss-Seidel

Reprenez ces deux méthodes avec les matrices

$$\begin{pmatrix} 3 & 2 & 1 & 0 \\ 3 & 4 & 3 & 2 \\ 3 & 2 & 5 & 2 \\ 1 & 2 & 3 & 4 \end{pmatrix} \text{ et } \begin{pmatrix} 3 & 2 & 1 \\ 1 & 4 & 1 \\ 2 & 4 & 8 \end{pmatrix}, \text{ ici avec } \mathbf{b} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}.$$

(Notebook 4.2, Exercice et Exemple 2)

Exercice 2

(Notebook 4.2, Exemple 3, Jacobi et Gauss-Seidel avec relaxation)
Nous avons vu que la méthode de Jacobi appliquée à la matrice

$$A = \begin{pmatrix} 3 & 2 & 1 & 0 \\ 3 & 4 & 3 & 2 \\ 3 & 2 & 5 & 2 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

ne converge pas, alors que celle de Gauss-Seidel converge.

Nous allons utiliser la méthode de Richardson stationnaire avec un paramètre de relaxation α constant pour voir si on peut améliorer la convergence.

1. Calculez la matrice d'itérations associé à la méthode de Richardson avec relaxation α constant
2. A l'aide de Python, déssinez, en fonction de α, le rayon spectrale associé à cette matrice dans le cas de $P = D$ la diagonale de A et P la partie triangulaire inférieure de A.

Exercice 3

(Notebook 4.2, Partie 2 – Implémentation de la Méthode de Richardson)

Définissez une fonction Python qui implémente la méthode de Richardson stationnaire avec la structure suivante :

```
def Richardson(A, b, x0, P, alpha, maxIterations, tolerance) :
    # Stationary Richardson method to approximate the solution of Ax=b
    # INPUT         :
    # x0            : initial guess
    # P             : preconditioner
    # alpha         : constant relaxation parameter
    # maxIterations : maximum number of iterations
    # tolerance     : tolerance for relative residual
    # OUTPUT        :
    # xk            : approximate solution to the linear system
    # rk            : vector of relative norm of the residuals
```

La méthode de Richardson, comme toute méthode iterative a besoin d'un critère d'arrêt. Dans ce cas, le plus simple est de poser les critères suivants :

1. On fixe le nombre maximal d'itérations
2. Le résidu relatif est plus petit qu'une tolérance ε :

$$\frac{\|\mathbf{b} - A\mathbf{x}^{(k)}\|}{\|\mathbf{b}\|} < \varepsilon$$

On s'arrête dès que l'un des ces critières est rempli.

(Notebook 4.2, Partie 3 – Utilisation de la Méthode de Richardson)

Utilisez la fonction `Richardson` pour approcher la solution de $A\mathbf{x} = \mathbf{b}$ pour $\mathbf{b} = (0, 1, -1, 1)^T$ avec une tolérance sur le résidu relatif de 10^{-6} et le vecteur nul comme point initial et un nombre maximum d'itérations de 200

1. En utilisant $P = D$, avec : $\alpha = 0.7, 0.75, 0.79, 0.81, 0.9$
2. En utilisant P égal au triangle inférieur de A, avec : $\alpha = 1, 1.5, 1.95, 2.05$

Comment interprétez-vous ces résultats ?

Exercice 4

(Notebook 4.2, Parties 4 et 5 - Implémentation de la Méthode du Gradient Préconditionné et du Gradient Conjugué)

Définissez une fonction Python qui implémente la méthode de Gradient Préconditionné et du Gadient Conjugué Préconditionné qui ont les structures suivantes

```python
def PrecGradient(A, b, x0, P, maxIterations, tolerance) :
    # Preconditioned Gradient method to approximate the solution of Ax=b
    # INPUT         :
    # x0            : initial guess
    # P             : preconditioner
    # maxIterations : maximum number of iterations
    # tolerance     : tolerance for relative residual
    # OUTPUTS       :
    # xk            : approximate solution to the linear system
    # rk            : vector of relative norm of the residuals
```

```python
def PrecConjugateGradient(A, b, x0, P, maxIterations, tolerance) :
    # Preconditionate Conjugate Gradient method to approximate the solution
       of Ax=b
    # INPUT         :
    # x0            : initial guess
    # P             : preconditioner
    # maxIterations : maximum number of iterations
    # tolerance     : tolerance for relative residual
    # OUTPUTS       :
    # xk            : approximate solution to the linear system
    # rk            : vector of relative norm of the residuals
```

(Notebook 4.2, Partie 6 - Utilisation de la Méthode du Gradient Preconditioné et du Gradient Conjugué Préconditionné

Utilisez ces fonctions pour approcher la solution de $A\mathbf{x} = \mathbf{b}$ pour $\mathbf{b} = (0, 1, -1, 1)^T$ avec une tolérance sur le résidu relatif de 10^{-6} et le vecteur nul comme point initial, et un nombre maximum d'itérations de 200

1. En utilisant $P = D$
2. En utilisant P égal au triangle inférieur de A

Comment interprétez-vous ces résultats ?

Peut-on utiliser une méthode de Richardson pour résoudre le problème du mauvais conditionnement de la matrice de Hilbert ?

```python
n = 10
A = hilbert(n)
xexact = np.ones(n)
b = A.dot(xexact)

# Define the initial guess
x0 = b

tolerance = 1e-6
maxIter = 10
```