Université Abou Bekr Belkaid Tlemcen **2020-2021**
Département de Mathématiques **Rattrapage**
Module: Analyse numérique 2 **Durée: 1h30**

Les calculatrices et les téléphones portables sont interdits.

Exercice 1 : Pour la résolution approchée du système

$$\begin{cases} 6x_1 & + x_2 & + x_3 & = 12, \\ x_1 & +2\ x_2 & +6x_3 & = 6, \\ 2x_1 & + 4x_2 & & = 0. \end{cases}$$

on considère la méthode itérative :

$$\begin{cases} x^{(0)} \text{ donné}, \\ x^{(k+1)} = Bx^{(k)} + c. \end{cases}$$

1) Ecrire la méthode itérative de Jacobi.
La méthode de Jacobi est-elle convergente ?
Effectuer 2 itérations de la méthode de Jacobi à partir de $x^{(0)} = (2, 2, 2)^t$.
2) Ecrire la méthode itérative de Gauss-Seidel.
La méthode de Gauss-Seidel est-elle convergente ?
Effectuer 2 itérations de la méthode de Gauss-Seidel à partir de $x^{(0)} = (2, 2, 2)^t$.

Exercice 2 : Soit le problème de Cauchy suivant :

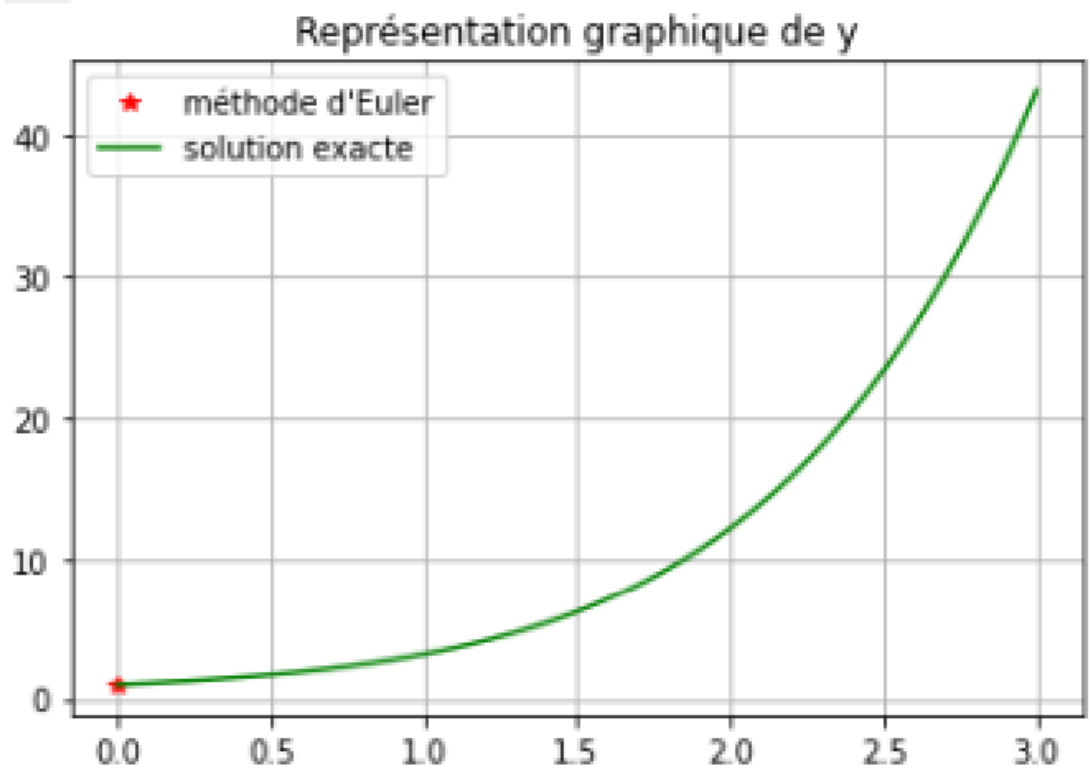
$$\begin{cases} y'(x) - y(x) = x^2 \quad x \in [0, 3] \\ y(0) = 1 \end{cases}$$

1) Pour $x_i = 0, 1, 2, 3$, calculer à l'aide de la méthode d'Euler y_i une approximation de $y(x_i)$.
Comparer les résultats avec la solution exacte : $y(x) = 3e^x - (x^2 + 2x + 2)$.
Indication : $e^1 = 2.718$, $e^2 = 7.389$, $e^3 = 20.086$.
2) Programme Python :
a) Écrire une fonction **euler** calculant les valeurs approchées d'une solution d'une équation différentielle par la méthode d'Euler.
b) Utiliser cette fonction euler pour retouver les valeurs calculées de l'exercice.
c) Tracer le graphe de la solution exacte $y(x)$ et indiquer les points (x_i, y_i) $(i = 0, 1, 2, 3)$.

Exercice3 : Soit A une matrice inversible.
1) Quelle relation existe-t-il en général entre $cond(A^2)$ et $(cond(A))^2$?
2) Calculer le conditionnement $cond_\infty(A)$ pour

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

```python
import matplotlib.pyplot as plt
import numpy as np
def euler(f,a,b,n,y0):
    h= (b-a)/n
    x= np.linspace(a,b,n+1)
    y=[y0]
    for i in range(n):
        y.append( _ _ _ _ _ _ _ _ _ _ _ _
    return x, y
a,b,n,y0= _ _ _ _ _ _ _ _ _ _ _ _
f= _ _ _ _ _ _ _ _ _ _ _ _ _ _
xi,yi = _ _ _ _ _ _ _ _ _ _ _ _
print(yi)
x= _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
y= _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
print(y(xi))
plt.plot(xi,yi,"r*",label="méthode d'Euler")
plt.plot _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
plt.title("Représentation graphique de y")
plt.legend()
plt.grid()
plt.show()
```



Correction

Exercice 1 : Pour la résolution approchée du système

$$\begin{cases} 6x_1 & + x_2 & + x_3 & = 12, \\ x_1 & +2\ x_2 & +6x_3 & = 6, \\ 2x_1 & + 4x_2 & & = 0. \end{cases}$$

On ne peut pas appliquer directement les méthodes de Jacobi et de Gauss-Seidel puisque l'un des coefficients diagonaux est nul. On réordonne le système de la façon suivante

$$\begin{cases} 6x_1 & + x_2 & + x_3 & = 12, \\ 2x_1 & + 4x_2 & & = 0. \\ x_1 & +2\ x_2 & +6x_3 & = 6, \end{cases}$$ 0.5

On note

$$A = D - L - U = \begin{pmatrix} 6 & 1 & 1 \\ 2 & 4 & 0 \\ 1 & 2 & 6 \end{pmatrix} \quad \text{et} \quad b = \begin{pmatrix} 12 \\ 0 \\ 6 \end{pmatrix}$$

0.5

$$D = \begin{pmatrix} 6 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 6 \end{pmatrix} \quad L = \begin{pmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ -1 & -2 & 0 \end{pmatrix} \quad \text{et} \quad U = \begin{pmatrix} 0 & -1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

1) La méthode de Jacobi s'écrit donc

$$(J) \begin{cases} x^{(0)} \text{ donné} \\ x^{(k+1)} = Bx^{(k)} + c \end{cases}$$

$$B = D^{-1}(L+U) = \begin{pmatrix} 0 & -1/6 & -1/6 \\ -1/2 & 0 & 0 \\ -1/6 & -1/3 & 0 \end{pmatrix} \quad \text{et} \quad c = D^{-1}b = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$$ 1p

La matrice A est à diagonale strictement dominante donc la méthode de Jacobi converge. 0.5
pour $x^{(0)} = (2,2,2)^t$

$$\begin{cases} x_1^{(k+1)} & = \left(\quad - 1/6 \quad x_2^{(k)} \quad - 1/6 \quad x_3^{(k)} \right) & +2, \\ x_2^{(k+1)} & = \left(- 1/2 \quad x_1^{(k)} \quad + \quad 0 \quad x_3^{(k)} \right) & +0, \\ x_3^{(k+1)} & = \left(- 1/6 \quad x_1^{(k)} \quad - 1/3 \quad x_2^{(k)} \quad \right) & +1. \end{cases}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
1	4/3	-1	0
2	13/6	$-2/3$	10/9

2p

2) La méthode de Gauss-Seidel s'écrit donc

$$(G)\begin{cases} x^{(0)} \text{ donné} \\ x^{(k+1)} = Bx^{(k)} + c \end{cases}$$

$$B = (D-L)^{-1}U = \begin{pmatrix} 0 & -1/6 & -1/6 \\ 0 & 1/12 & 1/12 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{et} \quad c = (D-L)^{-1}b = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$$ 1p

A est à diagonale strictement dominante donc la méthode de Gauss-Seidel converge. 0.5
pour $x^{(0)} = (2,2,2)^t$

$$\begin{cases} x_1^{(k+1)} &= \left(\quad -1/6 \quad x_2^{(k)} \quad -1/6 \quad x_3^{(k)} \right) \quad +2, \\ x_2^{(k+1)} &= \left(-1/2 \quad x_1^{(k+1)} \quad + \quad 0 \quad x_3^{(k)} \right) \quad +0, \\ x_3^{(k+1)} &= \left(-1/6 \quad x_1^{(k+1)} \quad -1/3 \quad x_2^{(k+1)} \right) \quad +1. \end{cases}$$

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
1	4/3	$-2/3$	1
1	35/18	35/36	1

2p

Exercice3 : 1) la matrice A est inversible et donc A^2 l'est également, et par définition,

$$Cond(A^2) = \left\|A^2\right\| \ \left\|(A^{-1})^2\right\| \leq \|A\| \left\|A^{-1}\right\| \|A\| \ \left\|A^{-1}\right\| = (cond(A))^2$$ 2p

2) Pour

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad \text{et} \quad A^{-1} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ 0 & -2 & 2 \end{pmatrix}$$ 0.5

0.5

le conditionnement $cond_\infty(A) = \|A\|_\infty \ \|A^{-1}\|_\infty = 3 \times 2 = 6$ avec $\|A\|_\infty = \max\limits_{1\leq i\leq n}\Big(\sum\limits_{j=1}^{n} |a_{ij}|\Big)$ 0.5

0.5

Exercice 2 : Soit le problème de Cauchy suivant :

$$\begin{cases} y'(x) - y(x) = x^2 \quad x \in [0,3] \\ y(0) = 1 \end{cases}$$

1) La méthode d'Euler avec un pas $h = 1$.

$$\begin{cases} x_0 = 0, y_0 = 1 \\ x_{i+1} = x_i + h \\ y_{i+1} = y_i + h(y_i + x_i^2) \end{cases}$$ 1

La solution exacte : $y(x) = 3e^x - (x^2 + 2x + 2)$

x_i	y_i	$y(x_i)$	$abs(y_i - y(x_i))$
0	1	1	1
1	2	$3e^1 - 5 \ = 3.154$	1.154
2	5	$3e^2 - 10 = 12.167$	7
3	14	$3e^3 - 17 = 43.258$	29

1

1

1

2) Programme Python :

```python
import matplotlib.pyplot as plt
import numpy as np
def euler(f,a,b,n,y0):
    h= (b-a)/n
    x= np.linspace(a,b,n+1)
    y=[y0]
    for i in range(n):
        y.append(y[i]+h*f(x[i],y[i]))   0.5
    return x, y   0.5
a,b,n,y0=0,3,3,1   0.5
f=lambda x,y:x**2+y   0.5
xi,yi =euler(f,a,b,n,y0)   0.5
print(yi)
x= np.linspace(a,b,50)   0.5
y= lambda x: 3*np.exp(x)-(x**2+2*x+2)   0.5
print(y(xi))
plt.plot(xi,yi,"r*",label="méthode d'Euler")
plt.plot(x,y(x),"g-",label="solution exacte")   0.5
plt.title("Représentation graphique de y")
plt.legend()
plt.grid()
plt.show()
```