esprit — Se former autrement — HONORIS UNITED UNIVERSITIES	**EXAMEN** Semestre : 1 ■ 2 ☐ Session : Principale ■ Rattrapage ☐

ETUDIANT(e) Nom et Prénom : …………………………………………………. Classe: …………………………………………………………..	Code :

Module : Calcul Scientifique (CS)
Enseignant(s) : Equipe CS de l'UP Math
Classes : 3A2→3A27 & 3B1→3B15
Documents autorisés : OUI ■ NON ☐ Nombre de pages : 7 pages
Calculatrice autorisée : OUI ■ NON ☐ Internet autorisée : OUI ☐ NON ■
Date : 06/01/2022 Heure : 11h Durée : 1h30min

*✂ ---

Code	Note /20	Nom et Signature du Surveillant	Nom et Signature du Correcteur	Observations

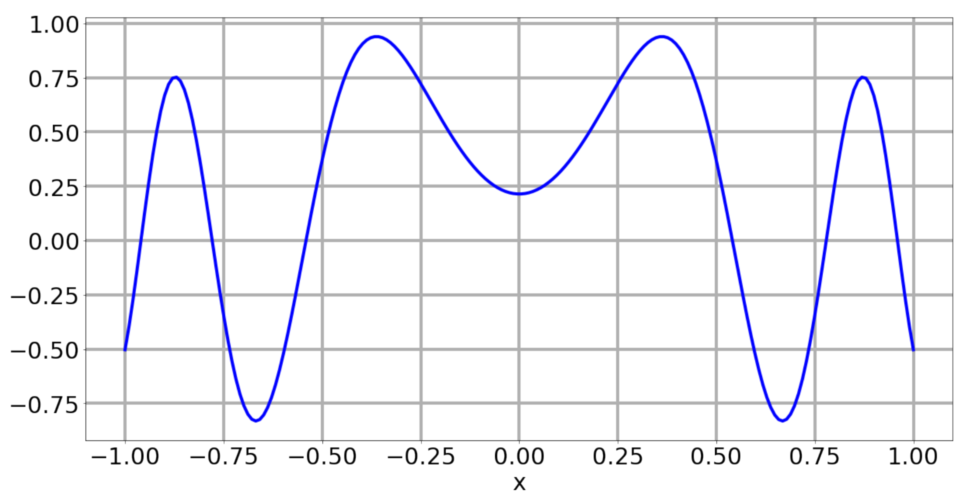
**NB : Vous êtes appelés à utiliser :

1. une flèche $\longrightarrow$ pour indiquer une indentation,
2. les importations des "`numpy`","`matplotlib.pyplot`" et "`sympy`" en utilisant les alias présentés ci-dessous:

```
[ ]: import numpy as np
     import matplotlib.pyplot as plt
     import sympy as sp
```

Exercice 1 (5 points)

Le serpent est beaucoup plus flexible que le reste des animaux vu la quantité impressionnante de vertèbres qu'il a : là où l'homme en a 32, il en a 400 ! Cela lui permet de faire onduler son corps dans de nombreuses positions. Pour bien comprendre ses déplacements, plusieurs fonctions mathématiques ont été utilisées pour les modéliser. Parmi ces dernières, on cite la fonction f introduite ci-dessous par sa courbe représentative sur l'intervalle $[-1, 1]$:



NE RIEN ECRIRE

1. En observant la représentation graphique de f, déterminer le nombre de racines de l'équation $f(x) = 0.5$. **(0.5 point)**

   ....

2. Donner le signe de f', la dérivée première f, sur l'intervalle $[-0.25, 0[$ en justifiant votre réponse. **(1.5 point)** ....

   ....

3. Donner le signe de f'', la fonction dérivée seconde de f, sur l'intervalle $[-0.25, 0[$ en justifiant votre réponse. **(1.5 point)**

   ....

   ....

4. On se propose d'utiliser la méthode de Newton pour résoudre numériquement l'équation $f(x) = 0.5$ sur l'intervalle $[-0.25, 0[$. Pour ce faire, on génère une suite récurrente de Newton notée x_n. Choisir $x_0 \in \{-0.25, 0\}$ qui assure au mieux la convergence de la suite. Justifier votre réponse. **(1.5 point)**

   ....

   ....

   ....

   ....

Exercice 2 (7 points)

Soient $A \in \mathcal{M}_n(\mathbb{R})$, $b \in \mathbb{R}^n$ et $\omega \in]0, 2[$. On se propose de résoudre numériquement le système de Cramer (S) : $AX = b$ par la méthode itérative suivante :

$$
(MI)_n \begin{cases} X^{(0)} \in \mathbb{R}^n \ : \ \text{vecteur initial donné} \\ \\ (\frac{1}{\omega}D - E)X^{(k+1)} = (\frac{1-w}{w}D + F)X^{(k)} + b, \end{cases}
$$

où $A = D - E - F$, avec

$$
A = \underbrace{\begin{pmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{n,n} \end{pmatrix}}_{D} - \underbrace{\begin{pmatrix} 0 & \cdots & \cdots & 0 \\ -a_{2,1} & \ddots & & 0 \\ \vdots & \ddots & \ddots & 0 \\ -a_{n,1} & \cdots & -a_{n,n-1} & 0 \end{pmatrix}}_{E} - \underbrace{\begin{pmatrix} 0 & -a_{1,2} & \cdots & -a_{1,n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{pmatrix}}_{F}
$$

On suppose que la matrice $\frac{1}{w}D - E$ est inversible.

1. Compléter la fonction `iterative(A, b, X0, w, Tol)` prenant en entrée `A`, une matrice carrée d'ordre n, `b` le second membre, `X0` une condition initiale, `w` un paramètre et une tolérance donnée `Tol`, qui renvoie une solution approchée de $AX = b$ et le nombre d'itérations k par la méthode $(MI)_n$ en utilisant le critère d'arrêt suivant $||AX^{(k)} - b|| \leq Tol$. **(2.5 point)**

[ ]:
```
def iterative(A, b, X0,w, Tol):
    D = ...................................
    E = ...................................
    F = ...................................
    M = ...................................... #  M=(1/w)*D-E
    N =  ..................................... #  N=((1-w)/w)*D+F
    invM= ..................................... # invM c'est l'inverse de M
    B = ................................. # B=l'inverse de M multiplié par N
    C = ............................... #  C=l'inverse de M multiplié par b
    k=0
    while ............................................... :
        X0=...........................
        k+=1
    return X0,k
```

2. Soit le système de Cramer suivant (S) : $AX = b$, avec

$$
A = \begin{pmatrix} 66 & 1 & -9 & -2 \\ 1 & 35 & -5 & -9 \\ -9 & -5 & 50 & 27 \\ -2 & -9 & 27 & 85 \end{pmatrix}, \quad b = \begin{pmatrix} 97 \\ 95 \\ -196 \\ -186 \end{pmatrix}
$$

   a) Déclarer la matrice A et le vecteur b, puis donner X, la solution du système (S) en utilisant la commande `np.linalg.solve`. **(1 point)**

[ ]:

```

```

La solution du système (S) est $X = (......, ..., ..., ...)$

b) En prenant $X^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, afficher, seulement, le nombre d'itérations effectuées par la méthode $(MI)_n$ pour $\text{Tol} = 10^{-6}$ et `w=1`. **(1 point)**

[ ]:

```

```

3. On se propose d'étudier le nombre d'itérations effectuées pour atteindre la convergence par la méthode $(MI)_n$, pour la résolution du système (S), en fonction du paramètre `w`.

a) Compléter le code suivant pour tracer la courbe de nombre d'itérations pour 16 valeurs de `w` repartis uniformément sur l'intervalle $[\frac{1}{5}, \frac{9}{5}]$. **(1.5 point)**

[ ]:

```
W=..........................
N_it=[]
for w in W:
    sol=iterative(A, b, X0,w, Tol)
    N_it.append(.........)

plt.figure(figsize=(20,10))
plt.plot(....,.......,'ro--',  linewidth=3,markersize=12)
plt.yscale('log')
```

b) Identifier graphiquement la valeur de `w` assurant le nombre d'itérations minimal. On note cette valeur par `wb`. **(0.5 point)**

`wb=`...

c) Sachant que pour `w=1`, la méthode $(MI)_n$ correspond à la méthode de Gauss-Seidel, comparer les résultats pour `w=1` et `w=wb` en terme de nombre d'itérations. **(0.5 point)**

....

....

Exercice 3

On rappelle ci-dessous la forme générale d'un Problème de Cauchy (PC) défini sur $I = [t_0, t_0 + T] \subset \mathbb{R}$:

$$(PC)\begin{cases} x'(t) = f(t, x(t)), & t \in [t_0, t_0 + T] \\ x(t_0) = x_0 \in \mathbb{R} : & \text{condition initiale,} \end{cases}$$

où f est une fonction continue sur $[t_0, t_0 + T] \times \mathbb{R}$ et $T > 0$.

Partie 1 : Solution exacte

Dans ce qui suit, nous considérons le problème de Cauchy (E), un cas particulier de (PC) pour lequel $f(t, x(t)) = g(t)x(t)$:

$$(E)\begin{cases} x'(t) = g(t)x(t), & t \in [0, 2] \\ x(0) = 1 \end{cases}$$

où $g(t) = t^2e^t$ est une fonction continue sur $[0, 2]$.

La solution analytique de (E) est de la forme:

$$x(t) = e^{-2}e^{G(t)}, \quad t \in [0, 2]$$

où G désigne la fonction primitive de g, à constante nulle.

1. En utilisant le module "sympy", donner l'expression de la fonction primitive G ainsi que l'expression analytique de la solution x finale, $\forall t \in [0, 2]$.**(1,5 point)**

$$G(t) = \text{.....................................}, \quad \forall t \in [0, 2]$$

$$x(t) = \text{.....................................}, \quad \forall t \in [0, 2]$$

2. En utilisant maintenant le module "numpy", donner l'instruction nécessaire pour programmer la fonction x ainsi trouvée.**(1 point)**

[ ]:

```

```

Partie 2 : Solution approchée

On souhaite maintenant approcher numériquement la solution exacte de (E). Pour ce faire, on utilise le schéma itératif (S) suivant qui approche la solution de (PC) de façon générale :

$$(S)\begin{cases} x_{n+1} = x_n + hp_2, & n \in \{0, \cdots, N-1\} \\ p_2 = f(t_m, x_n + \frac{h}{2}p_1), & \\ p_1 = f(t_n, x_n), & \\ x_0 = t_0. & \end{cases}$$

où

- $t_n = t_0 + nh$, $n \in \{0, \cdots, N-1\}$, désigne les points de discrétisation de $I = [t_0, t_0 + T]$ avec un pas constant $h = \dfrac{T}{N}$ (N correspond au nombre de sous-intervalles de I),

- $t_m = \dfrac{t_n + t_{n+1}}{2} = t_n + \dfrac{h}{2}$, le point milieu de l'intervalle $[t_n, t_{n+1}]$,

- x_n (resp. x_m) est l'approximation de $x(t_n)$ (resp. de $x(t_m)$).

3. Programmer la fonction `Schema(f, x_0, t_0,T,N)` qui prend en paramètres la fonction f du problème (PC), la valeur initiale x_0, la valeur initiale de la subdivision t_0, T, la largeur de l'intervalle $[t_0, t_0 + T]$, et N le nombre de sous-intervalles de $[t_0, t_0 + T]$. Cette fonction doit retourner la liste des valeurs $[x_0, x_1, ..., x_N]$ calculées par le schéma numérique (S). **(2 points)**

   Ecrire les instructions nécessaires.

[ ]:
```
def Schema(f, x0, t0,T,N):
```

4. Pour $N = 20$, et sur l'intervalle $[0, 2]$, tracer sur un même graphe, les courbes de la solution exacte de (E) et la solution approchée par le schéma (S). Pour ce faire, procéder en quatre étapes:

   a) Déclarer les variables de la fonction `Schema` pour la résolution de (E). **(0.5 point)**

[ ]:
```

```

   b) Donner l'instruction qui affecte les solutions approchées $[x_0, x_1, ..., x_N]$, par (S), du problème (E), à la variable `xapp`. **(0.5 point)**

[ ]:
```
xapp=.........................................................
```

   c) Donner l'instruction qui déclare le vecteur `t`, contenant les points de discrétisation t_n, $n \in \{0, \cdots, N-1\}$ de l'intervalle $[0, 2]$. **(0.5 point)**

[ ]:
```
t=..........................................................
```

d) Compléter la cellule suivante pour tracer les deux courbes sur une échelle logarithmique. **(1 point)**

[ ]:
```
plt.figure(figsize=(20, 10))
plt.plot(....,....., 'b*',....,....., 'r*-',linewidth=2,markersize=8)
plt.legend(('Solution exacte','Solution approchée'),fontsize=30)
plt.xticks(fontsize=20)
plt.yticks(fontsize=20)
plt.yscale('.....')
plt.show()
```

5. Observez les courbes de `x` et `xapp` tracées et étudier la qualité de l'approximation sur les deux sous-intervalles : $[0, 1]$ et $[1, 2]$. Argumenter votre réponse. **(1 point)**

....

....