Analyse Numérique SV, Examen 2022

Durée : 3 heures

1er juillet 2022

Contents

1 Problème 1 : Équations différentielles ordinaires

On considère le problème de Cauchy suivant :

$$\begin{cases} y'(t) = 2\sqrt{y(t)}\,\dfrac{1}{1+t^2} & \forall t \in [4,\infty) \\ y(4) = 3 \end{cases}$$

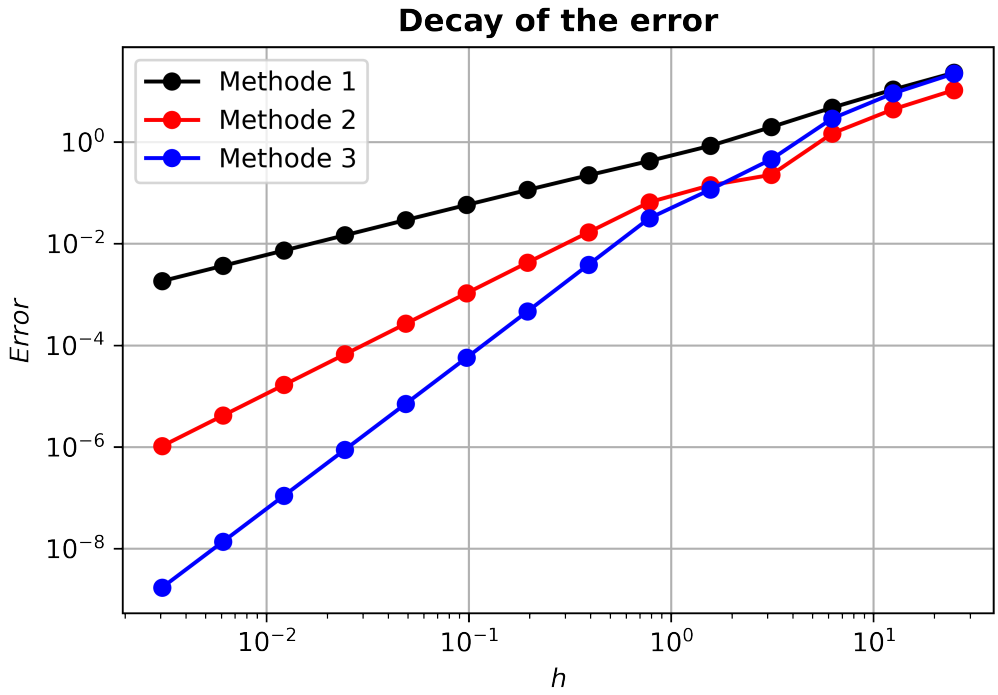
1a. Sur la feuille de réponse sont reportées quelques méthodes d'approximation d'équations différentielles ordinaires. Sous chaque colonne, écrivez respectivement si c'est implicite ou explicite, donnez l'ordre, et si c'est conditionnellement stable, inconditionnellement stable, ou instable.

1b. Écrivez le schéma de Crank-Nicholson pour l'approximation numérique de $y(t)$.

1c. Donnez une condition sur h afin que la méthode d'Euler Progressive, appliquée au problème de Cauchy mentionné, soit stable.

1d. Soient n fixé et u_n la valeur de la solution numérique au temps $t = t_n$; on applique la méthode de Crank-Nicholson et on veut trouver u_{n+1}. Vérifiez que u_{n+1} est la solution d'une équation de la forme $F(u_{n+1}) = 0$. Donnez la forme de F (en fonction de x : $F(x) = ...$) et écrivez la relation récursive qui définit les itérations $x^{(k)}$ de la méthode de Newton pour résoudre l'équation non-linéaire avec donnée initiale $x^{(0)} = u_n$. (Pas besoin de simplifier l'expression obtenue.)

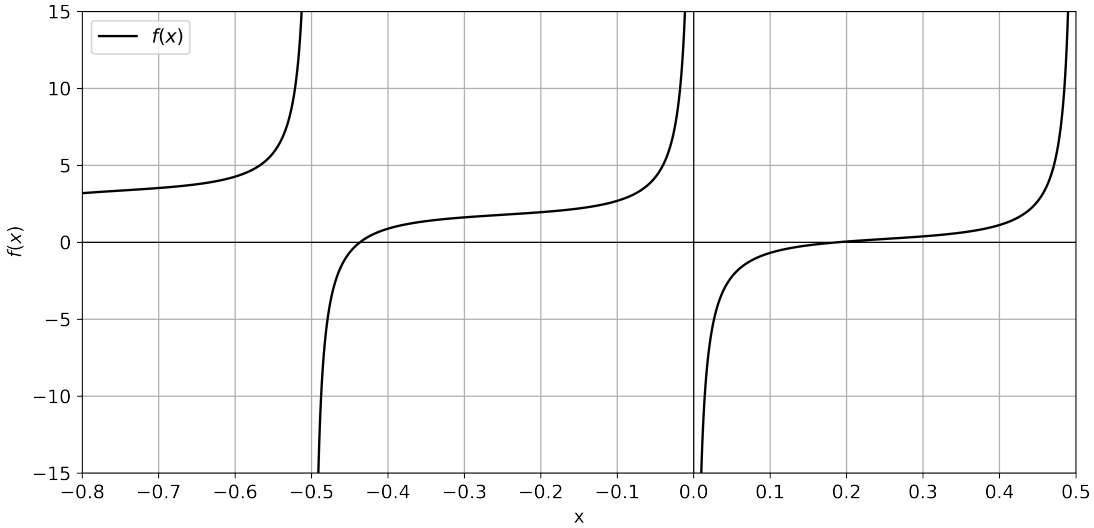
1e. Voici le graphique des erreurs pour 3 méthodes différentes. Quelle est l'ordre de convergence de chacune ? Est-ce que certaines pourraient être des méthodes que vous connaissez ? Justifiez vos réponses.



[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

2 Problème 2 : Équations non-linéaires

Soit $f : [-0.8, 0.5] \to \mathbb{R}$ la fonction représentée dans le graphique suivant. On veut utiliser la méthode de bissection pour calculer une approximation des zéros de f.



2a. (*Pour cet exercice, référez-vous au tableau sur la feuille de réponses*)

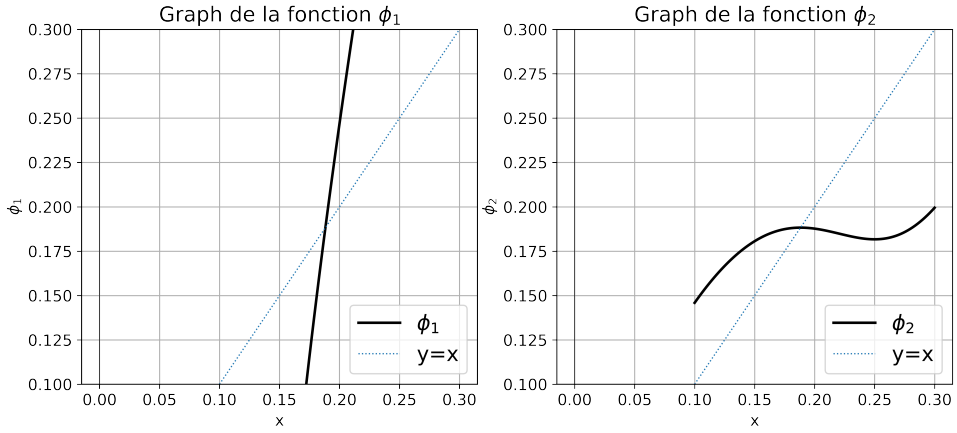
Pour chaque intervalle où il est possible d'utiliser la méthode de bissection, estimez le nombre minimal d'itérations nécessaires pour trouver un zéro avec une erreur inférieure à 10^{-6}. Marquez une croix dans la dernière colonne si la théorie ne garantit pas sa convergence.

2b. La fonction du graphique précédente possède un seul zéro α dans $[0.1, 0.3]$. On aimerait l'approcher par une méthode de point fixe à partir de l'une des fonctions ϕ_1 ou ϕ_2 dans les graphiques suivants. Pour quelle fonction la méthode de point fixe converge-t-elle localement? Justifiez votre réponse et, s'il y a convergence, donnez l'ordre de convergence.

2c. La fonction f est en effet donnée par

$$f(x) = \tan\left(2\pi\left(x - \frac{1}{4}\right)\right) - \pi x + 1.$$

Donnez la méthode de Newton pour trouver une approximation du zéro de f se trouvant dans l'intervalle $[0.1, 0.3]$. Donnez explicitement un point initial pour la méthode. (Rappel : la dérivée de $\tan(x)$ est $\frac{1}{\cos^2 x}$)



[2]:
```
# importing libraries used in this book
import numpy as np
import matplotlib.pyplot as plt
```

3 Problème 3 : Approximation numérique

Métapopulations : Modèle de Levins [https://en.wikipedia.org/wiki/Metapopulation] *(Il n'est pas nécesaire de comprendre le modèle en soi, ici on reporte rapidement son explication)*

Soit N la fraction de parcelles occupées à un moment donné. Chaque parcelle occupée peut devenir inoccupée avec une probabilité d'extinction e. De plus, $1 - N$ des parcelles sont inoccupées. En supposant un taux constant c de génération à partir de chacune des N parcelles occupées, chaque parcelle inoccupée peut devenir occupée avec une probabilité de colonisation cN.

Par conséquent, le taux de changement temporel R des parcelles occupées est donné par le modèle

$$R = cN(1 - N) - eN$$

Nous aimerions estimer les constantes c et e à partir des observations suivantes :

```
# données
N = np.array([0.2, 0.08, 0.14, 0.70, 0.82, 0.26, 0.33, 0.29]);
R = np.array([ 0.020,  0.012,  0.018, -0.104, -0.17,  0.018, 0.011, 0.016])
```

3. Calculez une approximation de c et de e.

[3]:
```
# importing libraries used in this book
import numpy as np
import matplotlib.pyplot as plt
from InterpolationLib import VandermondeMatrix
```

4 Problème 4 : Systèmes linéaires

Dans cet exercice, vous pouvez utiliser Python pour justifier vos réponses. Dans ce cas, il faudra aussi transcrire les commandes utilisées dans la feuille de réponses. (3-5 commandes maximum, celles qui ne sont pas déjà dans ce notebook).

On considère le système linéaire $A\mathbf{x} = \mathbf{b}$, avec $\mathbf{b} \in \mathbf{R}^N$ et A la matrice $N \times N$

$$A = \begin{pmatrix} 5 & -\frac{4}{3} & \frac{1}{12} & 0 & 0 & \dots & 0 \\ -\frac{4}{3} & 5 & -\frac{4}{3} & \frac{1}{12} & 0 & \dots & 0 \\ \frac{1}{12} & -\frac{4}{3} & 5 & -\frac{4}{3} & \frac{1}{12} & 0 & \vdots \\ 0 & \frac{1}{12} & -\frac{4}{3} & 5 & -\frac{4}{3} & \frac{1}{12} & \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \ddots \\ \vdots & & & \ddots & \ddots & \ddots & \ddots \\ 0 & \dots & 0 & 0 & \frac{1}{12} & -\frac{4}{3} & 5 \end{pmatrix}.$$

On fixe $N = 8$. (La définition de cette matrice avec Python est donnée dans le code ci-dessous.)

4a. Donnez l'expression de la matrice d'itérations de la méthode de Jacobi appliquée à ce système, et dans un second temps calculez-la explicitement avec Python. Déterminez si la méthode de Jacobi converge. Justifiez votre réponse.

4b. Qu'en est-il de la convergence de Gauss-Seidel ? Justifiez votre réponse.

On considère la méthode de Richardson stationnaire préconditionnée $P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \alpha(\mathbf{b} - A\mathbf{x}^{(k)})$ où $\alpha \in \mathbb{R}$ et P la matrice de *Gauss-Seidel Symétrique* qui se calcule comme indiqué plus bas.

4c. Calculez la valeur optimale du paramètre α.

4d. Si on admet une erreur initiale $\|x^{(0)} - x\|_A \approx 2.5$, combien d'itérations de Richardson stationnaire préconditionnée avec paramètre α optimal faudra-t-il pour avoir une erreur $\|x^{(k)} - x\|_A < 10^{-6}$?

Voici quelques commandes utiles

[4]:
```
# importing libraries used in this book
import numpy as np
import matplotlib.pyplot as plt
# scipy.linalg.eig : eigenvalues of a  matric
from scipy.linalg import eig
from numpy.linalg import cond
```

[5]:
```
N = 8
A = 5*np.diag(np.ones(N)) -  4/3*np.diag(np.ones(N-1),1) - 4/3*np.diag(np.
 ↪ones(N-1),-1) \
    + 1/12*np.diag(np.ones(N-2),2)  + 1/12*np.diag(np.ones(N-2),-2)

lk, v = eig(A)
print(f'Les valeurs propres de A sont {lk}\n')
```

```
D = np.diag(np.diag(A))  # diagonal of A
L = np.tril(A,-1) # lower triangular part of A
U = np.triu(A,1) # upper triangular part of A
P = (D+L).dot(np.linalg.solve(D, D+U)) # symmetric Gauss-Seidel

lk, v = eig(P)
print(f'Les valeurs propres de P sont {lk}\n')

PA = np.linalg.solve(P,A) # P^{-1} A
lk, v = eig(PA)

print(f'Les valeurs propres de P^(-1)A sont {lk}\n')
print('+0.j signifie que la partie imaginaire est nulle')
```

```
Les valeurs propres de A sont [7.63800759+0.j 7.08748745+0.j 6.27835555+0.j
5.34267566+0.j
 4.41603618+0.j 2.62596963+0.j 3.00093401+0.j 3.61053392+0.j]

Les valeurs propres de P sont [8.02646093+0.j 7.44284578+0.j 6.59450909+0.j
5.62855871+0.j
 4.68943875+0.j 2.932826  +0.j 3.2937579 +0.j 3.88882506+0.j]

Les valeurs propres de P^(-1)A sont [1.        +0.j 0.89344166+0.j
0.90482289+0.j 0.91841341+0.j
 0.93065801+0.j 0.94006055+0.j 0.95007931+0.j 0.94644082+0.j]

+0.j signifie que la partie imaginaire est nulle
```

5 Problème 5 : Intégration numérique

On veut approximer l'intégrale

$$\int_2^5 e^{x^2} dx.$$

5a. Calculez une valeur approchée de cette intégrale en utilisant la formule simple de Simpson.

5b. Pour avoir plus de précision, on considère la formule composite de Simpson I_s^c. En fonction du nombre N de sous-intervalles, estimez l'erreur globale

$$E_t^c = \left| \int_2^5 e^{x^2} \, dx - I_s^c(e^{x^2}) \right|.$$

5c. Trouvez le nombre minimal N de sous-intervalles afin que l'erreur globale soit plus petite que 10^{-4}.

6 Problème 6 : Interpolation

6a. Exprimez l'erreur d'interpolation polynomiale, ainsi que les conditions pour sa validité.

6b. Donnez la définition d'interpolation par morceaux.

6c. Démontrez le théorème suivant, en complétant l'inégalité (*inégalité à reporter sur la feuille de réponses*).

Théorème (Interpolation linéaire par morceaux) Soit $[a, b]$ un intervalle. Il existe une constante C telle que

$$\forall f \in C^2([a,b]), \forall N \in \mathbf{N}, \text{avec } H = \frac{b-a}{N}, \text{ on a } \quad \max_{x \in [a,b]} \left| f(x) - \Pi_1^H f(x) \right| \leq$$

EDO

$$-\lambda_{max} \leq \frac{\partial f}{\partial x}(t,x) \leq -\lambda_{min}$$

$$h < \frac{2}{\max_{j=1,\dots,p}|\lambda_j|} = \frac{2}{\rho(A)}\,,$$

$$\forall n = 0,\dots,N_h \quad |u_n - y(t_n)| \leq C(h)$$

$$|y(t_n) - u_n| \leq h t_n \frac{1}{2} \max_{t\in[t_0,t_n]} |y''(t)|$$

$$u_{n+1} - u_n = \frac{h}{2}\left[f(t_n,u_n) + f(t_{n+1},u_{n+1})\right]$$

$$... = \frac{h}{2}\left[f(t_n,u_n) + f(t_{n+1},u_n + hf(t_n,u_n))\right]$$

Équations non-linéaires

$$\lim_{k\to\infty} \frac{x^{(k+1)} - \alpha}{(x^{(k)} - \alpha)^d} = \frac{1}{d!}\phi^{(d)}(\alpha)$$

Systèmes linéaires

$$B = P^{-1}(P - A) = I - P^{-1}A \quad , \quad \mathbf{g} = P^{-1}\mathbf{b}.$$

$$P(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) = \alpha_k \mathbf{r}^{(k)}$$

$$\alpha_k = \alpha_{opt} = \frac{2}{\lambda_{min}(P^{-1}A) + \lambda_{max}(P^{-1}A)}$$

$$\begin{aligned}
&P\mathbf{z}^{(k)} = \mathbf{r}^{(k)}\\
&\alpha_k = \frac{(\mathbf{z}^{(k)})^T\mathbf{r}^{(k)}}{(\mathbf{z}^{(k)})^T A\mathbf{z}^{(k)}}\\
&\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{z}^{(k)}\\
&\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{z}^{(k)}
\end{aligned}$$

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_A \leq \left(\frac{\mathrm{Cond}(P^{-1}A) - 1}{\mathrm{Cond}(P^{-1}A) + 1}\right)^k \|\mathbf{x}^{(0)} - \mathbf{x}\|_A$$

$$\begin{aligned}
&\alpha_k = \frac{\mathbf{p}^{(k)^T}\mathbf{r}^{(k)}}{\mathbf{p}^{(k)^T}A\mathbf{p}^{(k)}}\\
&\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k\mathbf{p}^{(k)}\\
&\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{p}^{(k)}\\
&P\mathbf{z}^{(k+1)} = \mathbf{r}^{(k+1)}\\
&\beta_k = \frac{(A\mathbf{p}^{(k)})^T\mathbf{z}^{(k+1)}}{(A\mathbf{p}^{(k)})^T\mathbf{p}^{(k)}}\\
&\mathbf{p}^{(k+1)} = \mathbf{z}^{(k+1)} - \beta_k\mathbf{p}^{(k)}
\end{aligned}$$

$$\|\mathbf{x}^{(k)} - \mathbf{x}\|_A \leq \frac{2c^k}{1 + c^{2k}}\|\mathbf{x}^{(0)} - \mathbf{x}\|_A\,, \quad c = \frac{\sqrt{K_2(P^{-1}A)} - 1}{\sqrt{K_2(P^{-1}A)} + 1}$$

$$\frac{\|\mathbf{x}^{(k)} - \mathbf{x}\|}{\|\mathbf{x}\|} \leq \mathrm{Cond}\,(P^{-1}A)\frac{\|P^{-1}\mathbf{r}^{(k)}\|}{\|P^{-1}\mathbf{b}\|}$$

$$\sqrt{\frac{\lambda_{\max}(C^TC)}{\lambda_{\min}(C^TC)}} \quad \text{ou} \quad \frac{\lambda_{\max}(C)}{\lambda_{\min}(C)}$$

Interpolation

$$\varphi_k(x) = \prod_{j=0, j\neq k}^{n} \frac{(x - x_j)}{(x_k - x_j)}.$$

$$\max_{x\in I}|E_n f(x)| \leq \frac{1}{4(n+1)}(h)^{n+1}\max_{x\in I}|f^{(n+1)}(x)|,$$

$$\max_{x\in I} |\ E_1^H f(x)\ | \leq \frac{H^2}{8}\max_{x\in I}|f''(x)|.$$

$$\max_{x\in I} |\ E_n^H f(x)\ | \leq \frac{H^{n+1}}{4(n+1)}\max_{x\in I}|f^{(n+1)}(x)|\,.$$

Intégration numérique

$$J^{xx}(f) = (b-a)f\left(\frac{a+b}{2}\right); \quad H\sum_{k=0}^{N-1} f(\overline{x}_k)$$

$$J^{xx}(f) = (b-a)\frac{f(a)+f(b)}{2}; \quad \frac{H}{2}\sum_{k=0}^{N-1}\left[f(x_k) + f(x_{k+1})\right]$$

$$J^{xx}(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right];$$

$$\frac{H}{6}\sum_{k=0}^{N-1}\left[f(x_k) + 4f(\overline{x}_k) + f(x_{k+1})\right]$$

$$|I(f) - I_{xx}^c(f)| \leq \frac{b-a}{24}H^2\max_{x\in[a,b]}|f''(x)|$$

$$|I(f) - I_{xx}^c(f)| \leq \frac{b-a}{12}H^2\max_{x\in[a,b]}|f''(x)|$$

$$|I(f) - I_{xx}^c(f)| \leq \frac{b-a}{180\cdot 16}H^4\max_{x\in[a,b]}|f''''(x)|$$