Université Grenoble Alpes
MAP350 – L3 MI

Examen du 11 décembre 2017

Exercice 1 : [*Neville-Aitken algorithm.*]

On considère les $n+1$ points

$$a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$$

et pour une fonction f définie sur $[a,b]$, on note $P_{i,k}(x,f)$ le polynôme d'interpolation de f aux points x_j pour $i \leq j \leq i+k$.

1. Donner le degré du polynôme $P_{i,k}(x,f)$. Justifier votre réponse.

2. Préciser les polynômes $P_{i,0}(x,f)\,, \quad i = 0,1,...,n$.

3. Montrer que pour $0 \leq k \leq n-1$, on a

$$\frac{x - x_i}{x_{i+k+1} - x_i}\, P_{i+1,k}(x,f) + \frac{x_{i+k+1} - x}{x_{i+k+1} - x_i}\, P_{i,k}(x,f) = P_{i,k+1}(x,f) \quad \text{pour } 0 \leq i \leq n-k-1\,.$$

   Indication : *on évaluera les valeurs du premier membre de cette égalité aux points x_j pour $i+1 \leq j \leq i+k$ puis aux points x_i et x_{i+k+1}.*

4. Ecrire la formule précédente pour $k = 0$ et $k = 1$, puis utiliser cette méthode pour construire le polynôme d'interpolation associé aux données

   $x_0 = -1, \quad x_1 = 0, \quad x_2 = 1,$

   pour une fonction f vérifiant

   $f(x_0) = 1, \quad f(x_1) = 0, \quad f(x_2) = 2.$

Exercice 2 :
Etant donné un intervalle non nul $[a,b]$ et un entier $n \in \mathbb{N}^*$, une suite de points $(x_{n,i})$ de cet intervalle $[a,b]$ est une suite de $n+1$ points distincts $x_{n,0}, x_{n,1}, \ldots, x_{n,n}$ de cet intervalle $[a,b]$.

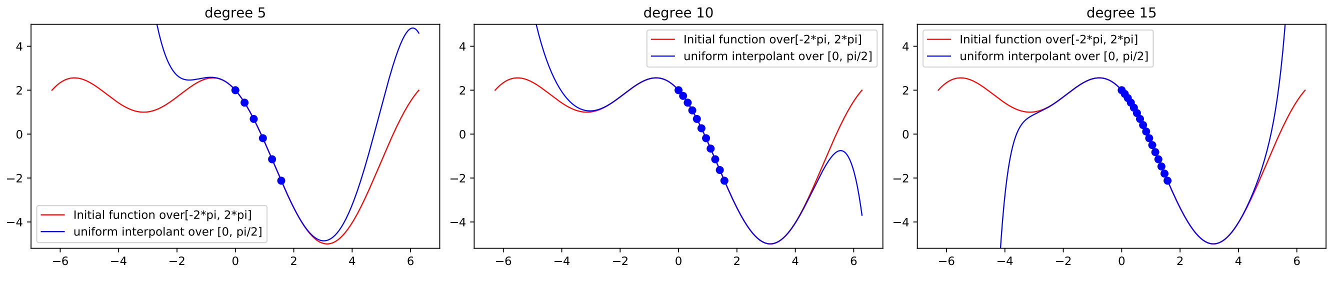
1. Donner une preuve détaillée du résultat suivant [Proposition 11, page 26, du cours sur l'interpolation].

   Soit $f \in C^\infty[a,b]$, tel que

$$\exists\, M \in \mathbb{R}^+, \quad \forall\, k \in \mathbb{N}, \quad \left|\left|f^{(k)}\right|\right| = \max_{x\in[a,b]} \left|f^{(k)}(x)\right| \;\leq\; M\,.$$

   Alors, pour toute suite de points d'interpolation $(x_{n,i})$ de $[a,b]$, la suite des polynômes d'interpolation associés $P_n(.,f)$ converge uniformément vers f sur l'intervalle $[a,b]$ quand n tend vers $+\infty$.

2. Considérons la fonction $f(x) = 2\cos(x) - 3\sin(x/2)$ définie sur $I = [-2\pi, 2\pi]$. L'interpolation uniforme de cette fonction sur l'intervalle $[0, \pi/2]$ par des polynômes de degré 5, 10 et 15 semble converger vers f sur l'ensemble de l'intervalle $[-2\pi, 2\pi]$ (voir les figures ci-dessous). Justifier cette convergence.



Exercice 3 :
Dans une expérience de biologie, on étudie la relation entre la concentration d'un substrat (noté x) et la vitesse de réaction (noté y) dans une réaction enzymatique à partir de données (x_i, y_i). Le modèle étudié est $y = \dfrac{a\,x}{b+x}$ où a et b sont les paramètres à estimer.

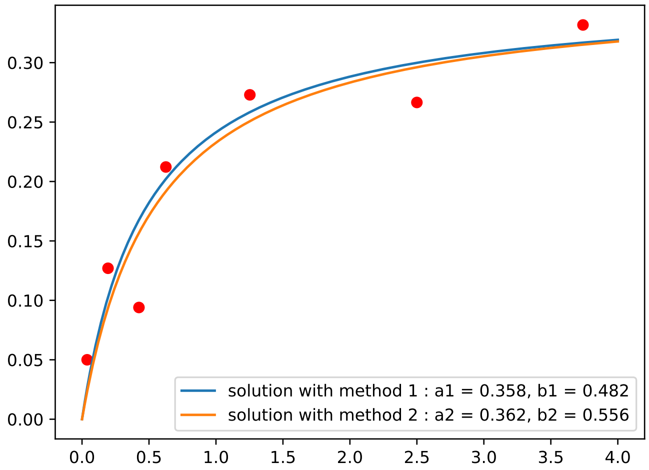
Le script Python ci-dessous fournit deux estimations de ces paramètres a et b selon deux méthodes différentes. La seconde méthode n'a pas été étudiée dans le cours de sorte que nous donnons directement le résultat.

```python
import numpy as np
import matplotlib.pyplot as plt

def f(a,b,x):
    return a * x / (b + x)

def Error1(a,b,xi,yi):
    return sum( (xi * yi - (a * xi - b * yi))**2 )

def Error2(a,b,xi,yi):
    return sum( (yi - f(a,b,xi))**2 )

# Data :
xi = np.array([0.038, 0.194, 0.425, 0.626, 1.253, 2.500, 3.740])
yi = np.array([0.050, 0.127, 0.094, 0.2122, 0.2729, 0.2665, 0.3317])
plt.plot(xi,yi,'or')

nbxi = np.size(xi)
xmin = 0 ; xmax = 4
t = np.linspace(xmin,xmax,200) # for the plotting of the solution

#    METHOD 1 :
A = np.ones((nbxi,2))
A[:,0] =  xi
A[:,1] = -yi
B = xi * yi
M = np.dot(A.T, A)
S = np.dot(A.T, B)
cf = np.linalg.solve(M, S)
a1 = cf[0]
b1 = cf[1]
y = f(a1,b1,t)
plt.plot(t,y,label='solution with method 1 : a1 = '+str(a1)+', b1 = '+str(b1))

#    METHOD 2 (Gauss_Newton) :
a2 = 0.362
b2 = 0.556
y = f(a2,b2,t)
plt.plot(t,y,label='solution with method 2 : a2 = '+str(a2)+', b2 = '+str(b2))
plt.legend(loc='best')

# residual errors :
print('Method 1 : residual error1 = ',Error1(a1,b1,xi,yi))
print('Method 1 : residual error2 = ',Error2(a1,b1,xi,yi))
print('Method 2 : residual error1 = ',Error1(a2,b2,xi,yi))
print('Method 2 : residual error2 = ',Error2(a2,b2,xi,yi))
```

Affichage console :

```
Method 1 : residual error1 =  0.0194802178024
Method 1 : residual error2 =  0.00823190626711
Method 2 : residual error1 =  0.0201322476908
Method 2 : residual error2 =  0.00784411528128
```

Sortie graphique :



Et enfin ... les questions :

1. Préciser la méthode 1 développée dans ce script Python.
2. Quelle quantité est minimisée dans cette méthode 1.
3. Idéalement, quelle quantité devrait-on minimiser pour résoudre le problème initial.
4. Comparer et commenter les 4 erreurs résiduelles obtenues.

Exercice 4 :
On considère l'espace vectoriel des splines quintiques Π_k^5 (avec $k = 2$ ou $k = 3$) associé à la famille de noeuds équidistants

$$a = x_1 < x_2 < \cdots < x_{n-1} < x_n = b,$$

avec $n \geq 2$, $x_{i+1} - x_i = h,\ i = 1, \ldots, n-1$.
On considère par ailleurs 3 suites de n réels fixés :

$$y_1, \ldots, y_n, \qquad y_1', \ldots, y_n', \qquad y_1'', \ldots, y_n''.$$

1. Justifier que le problème suivant admet une unique solution.
   `Problème 1:` *Déterminer $s \in \Pi_2^5$ tel que*

   $$s(x_i) = y_i, \quad s'(x_i) = y_i', \quad s''(x_i) = y_i'' \quad \text{pour} \quad i = 1, \ldots, n. \tag{1}$$

   On ne demande pas de construire cette solution.
   On admettra les résultats de la section 3 `[Quintic Hermite interpolation over 2 points]` *du chapitre sur l'interpolation de Hermite.*

2. On souhaite montrer que le problème suivant admet une unique solution.
   `Problème 2:` *Déterminer $s \in \Pi_3^5$ tel que*

   $$s(x_i) = y_i, \quad s'(x_i) = y_i' \quad \text{pour} \quad i = 1, \ldots, n \quad \text{et} \quad s^{(3)}(a) = s^{(3)}(b) = 0. \tag{2}$$

   2.a Donner la dimension de Π_3^5 et déterminer le nombre de contraintes d'interpolation. Que peut-on en conclure ?

   2.b Proposer une méthode pour construire une solution de ce problème. *On pourra s'inspirer de la construction des splines naturelles.* Donner la matrice du système linéaire permettant de déterminer cette solution, justifier que cette matrice est inversible, et donc en déduire l'existence et l'unicité de la solution.

Indication pour la question 2.b : Soit $p(x)$ l'unique solution du problème d'interpolation de Hermite d'ordre 2 sur un intervalle $[\alpha, \beta]$. Les dérivées d'ordre 3 aux extrémités s'expriment uniquement en fonction des données d'interpolation par

$$p^{(3)}(\alpha) = \Big(-20, -12, -3, 1, -8, 20\Big)\,\frac{3}{h^3}\begin{pmatrix} y_\alpha \\ h\, y_\alpha' \\ h^2\, y_\alpha'' \\ h^2\, y_\beta'' \\ h\, y_\beta' \\ y_\beta \end{pmatrix} \quad \text{et} \quad p^{(3)}(\beta) = \Big(-20, -8, -1, 3, -12, 20\Big)\,\frac{3}{h^3}\begin{pmatrix} y_\alpha \\ h\, y_\alpha' \\ h^2\, y_\alpha'' \\ h^2\, y_\beta'' \\ h\, y_\beta' \\ y_\beta \end{pmatrix}.$$