Corrigé

On rappelle que l'on a : $\forall x \in \mathbb{R}, \cos(x) = \sum_{n=0}^{+\infty} (-1)^n \frac{x^{2n}}{(2n)!}$.

L'objectif principal de cet exercice est d'écrire un programme Python permettant d'obtenir une valeur approchée de $\cos(x)$ pour certaines valeurs de *x* à l'aide du développement précédent.

Pour tout *x* réel, on pose : $u_n(x) = (-1)^n \frac{x^{2n}}{(2n)!}$, $S_n(x) = \sum_{k=0}^{n} (-1)^k \frac{x^{2k}}{(2k)!}$ et $R_n(x) = \cos(x) - S_n(x)$.

1. Montrer que pour tout réel de l'intervalle $I = \left[-\sqrt{2}\,;\sqrt{2}\right]$ la série $\sum u_n(x)$ vérifie le critère spécial des séries alternées. En déduire : $\left|\cos x - S_n(x)\right| \leq \frac{x^{2n+2}}{(2n+2)!}$.

Soit *x* fixé dans $I = \left[-\sqrt{2}\,;\sqrt{2}\right]$.

On a immédiatement, l'exposant de *x* étant pair : $\forall n \in \mathbb{N}, \frac{x^{2n}}{(2n)!} \geq 0$ et donc

$$\forall n \in \mathbb{N}, u_n(x) \times u_{n+1}(x) = (-1)^n \frac{x^{2n}}{(2n)!} \times (-1)^{n+1} \frac{x^{2(n+1)}}{(2(n+1))!} = -\frac{x^{2n}}{(2n)!} \times \frac{x^{2(n+1)}}{(2(n+1))!} \leq 0.$$

La série $\sum (-1)^n \frac{x^{2n}}{(2n)!}$ est bien une série alternée.

On a immédiatement, : $\forall n \in \mathbb{N}, \left|u_n(x)\right| = \left|(-1)^n \frac{x^{2n}}{(2n)!}\right| = \frac{x^{2n}}{(2n)!}$ puis, pour *x* non nul,

$$\forall n \in \mathbb{N}, \frac{\left|u_{n+1}(x)\right|}{\left|u_n(x)\right|} = \frac{\overline{\frac{x^{2(n+1)}}{(2(n+1))!}}}{\overline{\frac{x^{2n}}{(2n)!}}} = \frac{x^{2n+2}}{(2n+2)!} \times \frac{(2n)!}{x^{2n}} = \frac{x^2}{(2n+2)(2n+1)}.$$

Or : $\forall n \in \mathbb{N}, \frac{x^2}{(2n+2)(2n+1)} \leq \frac{x^2}{(2\times 0+2)\times(2\times 0+1)} = \frac{x^2}{2} \leq \frac{2}{2} = 1$.

Ainsi, la suite $\left(\left|u_n(x)\right|\right)_{n\in\mathbb{N}}$ est décroissante.

Si $x=0$, la suite $\left(\left|u_n(0)\right|\right)_{n\in\mathbb{N}}$ est constante à partir de $n=1$ et donc décroissante…

Pour montrer enfin que l'on a $\lim\limits_{n\to+\infty}\left|u_n(x)\right|=0$ lorsque *x* est non nul (c'est immédiat lorsque *x* l'est), on peut utiliser la formule de Stirling.

On a : $(2n)!\underset{+\infty}{\sim}\sqrt{2\pi\times 2n}\times\left(\frac{2n}{e}\right)^{2n}=2\sqrt{\pi n}\times\left(\frac{2n}{e}\right)^{2n}$ et il vient alors :

$$\left|u_n(x)\right|=\frac{x^{2n}}{(2n)!}\underset{+\infty}{\sim}\frac{x^{2n}}{2\sqrt{\pi n}\times\left(\frac{2n}{e}\right)^{2n}}=\frac{1}{\sqrt{\pi n}}\times\left(\frac{ex}{2n}\right)^{2n}.$$

Mais :

$$\ln\left[\frac{1}{\sqrt{\pi n}}\times\left(\frac{ex}{2n}\right)^{2n}\right]=-\frac{1}{2}\ln\pi-\frac{1}{2}\ln n+2n\times\ln(ex)-2n\times\ln(2n)$$

$$=-2n\times\left(\frac{1}{4}\frac{\ln\pi}{n}+\frac{1}{4}\frac{\ln n}{n}-\ln(ex)+\ln(2n)\right)$$

On a : $\lim\limits_{n\to+\infty}\left(\frac{1}{4}\frac{\ln\pi}{n}+\frac{1}{4}\frac{\ln n}{n}\right)=0$ et donc $\lim\limits_{n\to+\infty}\left(\frac{1}{4}\frac{\ln\pi}{n}+\frac{1}{4}\frac{\ln n}{n}-\ln(ex)+\ln(2n)\right)=+\infty$.

Alors : $\lim\limits_{n\to+\infty}\left[-2n\times\left(\frac{1}{4}\frac{\ln\pi}{n}+\frac{1}{4}\frac{\ln n}{n}-\ln(ex)+\ln(2n)\right)\right]=\lim\limits_{n\to+\infty}\ln\left[\frac{1}{\sqrt{\pi n}}\times\left(\frac{ex}{2n}\right)^{2n}\right]=-\infty$ et

finalement : $\lim\limits_{n\to+\infty}\left[\frac{1}{\sqrt{\pi n}}\times\left(\frac{ex}{2n}\right)^{2n}\right]=\lim\limits_{n\to+\infty}\left|u_n(x)\right|=0$.

Pour tout réel *x* de l'intervalle $\mathrm{I}=\left[-\sqrt{2}\,;\sqrt{2}\right]$, la série $\sum u_n(x)$ vérifie le critère spécial des séries alternées.

On a alors la majoration classique : $\left|R_n(x)\right|\le\left|u_{n+1}(x)\right|$, c'est-à-dire :

$$\left|\cos(x)-S_n(x)\right|\le\frac{x^{2(n+1)}}{(2(n+1))!}=\frac{x^{2n+2}}{(2n+2)!}$$

$$\forall x\in\left[-\sqrt{2}\,;\sqrt{2}\right],\ \forall n\in\mathbb{N},\ \left|\cos(x)-S_n(x)\right|\le\frac{x^{2n+2}}{(2n+2)!}$$

2. Ecrire une fonction Python `RangMini` qui calcule le plus petit entier naturel *n* tel que $\frac{x^{2n}}{(2n)!} \leq \varepsilon$ où ε est un réel strictement positif fixé.

   ❶ La fonction `RangMini` recevra comme argument une variable `x`, correspondant au réel *x*, et une variable `epsilon` correspondant au réel ε.

   ❷ Les factorielles seront obtenues via une fonction Python `Fact` (à écrire).

On a d'abord :

```
def Fact(n):
    if n==0:
        return(1)
    else:
        return(n*Fact(n-1))
```

Pour la fonction `RangMini`, on va calculer $\frac{x^{2n}}{(2n)!}$ pour des valeurs successives de *n* (la variable correspondante sera incrémentée à chaque itération) tant que la condition $\frac{x^{2n}}{(2n)!} > 0$ est vérifée. La dernière valeur de *n* devra être retournée. On peut donc considérer le code suivant :

```
def RangMini(x,epsilon):
    n = 0
    while x**(2*n)/Fact(2*n) > epsilon:
        n += 1
    return(n)
```

Le principe du calcul de l'approximation de $\cos(x)$ pour tout réel *x* de l'intervalle I est alors :

- On fixe/demande la précision souhaitée (réel ε).
- On détermine le plus petit entier *n* tel que $\frac{x^{2n}}{(2n)!} \leq \varepsilon$.
- On calcule $S_{n-1}(x)$ qui est une valeur approchée de $\cos(x)$ à ε près.

3. Ecrire une fonction Python `EvalSommePartielle` qui recevra comme argument une variable `x`, correspondant au réel *x*, et une variable `n`, correspondant à un entier *n* et évaluera $S_n(x)$ (vous écrirez et utiliserez une fonction <u>récursive</u> `PolyHorner` mettant en œuvre la méthode de Horner pour l'évaluation des polynômes).

Pour les détails de la mise en œuvre de la méthode de Horner on se réfèrera au cours. Nous proposons le code suivant :

```
def PolyHorner(L,x):
    if len(L)==1:
        return(L[0])
    else:
        L[-2] += L[-1]*x
        del L[-1]
        return(PolyHorner(L,x))
```

L'évaluation de $S_n(x)=\sum_{k=0}^{n}(-1)^k\frac{x^{2k}}{(2k)!}$, c'est-à-dire l'évaluation du polynôme $\sum_{k=0}^{n}\frac{(-1)^k}{(2k)!}X^k$ au point x^2, consiste alors à appeler la fonction précédente avec la liste des coefficients $\frac{(-1)^k}{(2k)!}$. Dans un premier temps, on construit donc cette liste puis on appelle la fonction `PolyHorner`. D'où le code possible suivant :

```
def EvalSommePartielle(x,n):
    L = []
    for i in range(n+1):
        L.append((-1)**i/Fact(2*i))
    return(PolyHorner(L,x))
```

Remarque : la liste `L` peut également être définie comme suit :

```
L = [(-1)**i/Fact(2*i) for i in range(n+1)]
```

Ainsi, on peut aller jusqu'à écrire la fonction de façon très compacte :

```
def EvalSommePartielle(x,n):
    return(PolyHorner([(-1)**i/Fact(2*i) for i in range(n+1)],x))
```

4. Ecrire enfin une fonction Python `CosApprox` qui recevra comme argument une variable `x` correspondant au réel *x* de l'intervalle I et une variable `epsilon`, correspondant au réel ε, et qui renverra une valeur approchée de $\cos(x)$ à ε près.

L'essentiel du travail a été fait précédemment !

Le contenu de la fonction CosApprox comporte deux étapes principales :

- La détermination, grâce à la fonction `RangMini`, du plus petit indice *n* tel que $\frac{x^{2n}}{(2n)!} \leq \varepsilon$.
- Calcul de la somme partielle $S_{n-1}(x)$ (rappelons que nous avons : $S_{n-1}(x) < \frac{x^{2n}}{(2n)!} \leq \varepsilon$).

D'où le code possible :

```
def CosApprox(x,epsilon):
    return(EvalSommePartielle(x**2, RangMini(x,epsilon)-1))
```

En définitive, on aura le code complet possible suivant (avec une version « standard » de la fonction `EvalSommePartielle`) :

```
def Fact(n):
    if n==0:
        return(1)
    else:
        return(n*Fact(n-1))

def PolyHorner(L,x):
    if len(L)==1:
        return(L[0])
    else:
        L[-2] += L[-1]*x
        del L[-1]
        return(PolyHorner(L,x))

def RangMini(x,epsilon):
    n = 0
    while x**(2*n)/Fact(2*n) > epsilon:
        n += 1
    return(n)

def EvalSommePartielle(x,n):
    L = []
    for i in range(n+1):
        L.append((-1)**i/Fact(2*i))
    return(PolyHorner(L,x))

def CosApprox(x,epsilon):
    return(EvalSommePartielle(x**2,RangMini(x,epsilon)-1))
```

5. Dans cette question on étend l'intervalle de validité de la méthode ci-dessus à $J=[-\pi\,;\pi]$.

   a. Montrer que l'on a : $\forall n\in\mathbb{N}^*, \forall x\in\mathbb{R}^*, \left|\dfrac{u_{n+1}(x)}{u_n(x)}\right|\leq\dfrac{x^2}{12}$.

   b. En déduire que pour tout *x* de J, la majoration $\left|\cos x - S_n(x)\right|\leq\dfrac{x^{2n+2}}{(2n+2)!}$ est valable pour tout entier naturel *n* non nul.

Retour sur la première question.

On a travaillé sur l'intervalle I car il garantissait la décroissance de la suite $\left(\left|u_n(x)\right|\right)_{n\in\mathbb{N}}$ et nous permettait donc de majorer simplement la différence $\left|\cos x - S_n(x)\right|$.

Mais idéalement, on souhaiterait pouvoir disposer d'une méthode numérique valable sur un intervalle de longueur 2π. C'est l'objet de cette dernière question.

a. A la première question, on a vu que l'on avait, pour tout *x* réel non nul :

$$\forall n\in\mathbb{N}, \frac{\left|u_{n+1}(x)\right|}{\left|u_n(x)\right|}\leq\frac{x^2}{(2n+2)(2n+1)}$$

On en déduit : $\forall n\in\mathbb{N}^*, \forall x\in\mathbb{R}^*, \dfrac{\left|u_{n+1}(x)\right|}{\left|u_n(x)\right|}\leq\dfrac{x^2}{(2\times1+2)(2\times1+1)}=\dfrac{x^2}{12}$.

$$\forall n\in\mathbb{N}^*, \forall x\in\mathbb{R}^*, \frac{\left|u_{n+1}(x)\right|}{\left|u_n(x)\right|}\leq\frac{x^2}{12}$$

b. Pour tout réel *x* non nul de J, on a : $\dfrac{x^2}{12}\leq\dfrac{\pi^2}{12}<1$ et la suite $\left(\left|u_n(x)\right|\right)_{n\in\mathbb{N}^*}$ (notez bien l'astérisque) est strictement décroissante. Pour $x=0$ elle est constante (et donc décroissante) à partir de $n=1$.

Posons alors : $\cos(x)=1+f(x)$ (on a donc : $\forall x\in\mathbb{R}, f(x)=\sum_{n=1}^{+\infty}(-1)^n\dfrac{x^{2n}}{(2n)!}$) et, pour tout entier naturel *n* non nul : $S_n(x)=1+\sum_{k=1}^{n}(-1)^k\dfrac{x^{2k}}{(2k)!}=1+T_n(x)$.

Pour tout réel *x* de J, on peut appliquer le critère spécial des séries alternées à la série $\sum_{n\in\mathbb{N}^*}u_n(x)$. On a alors, pour tout entier naturel *n* non nul, la majoration :

$\left|f(x)-T_n(x)\right|\leq\dfrac{x^{2n+2}}{(2n+2)!}$.

En tenant compte de $f(x)-T_n(x)=\left(\cos(x)-1\right)-\left(S_n(x)-1\right)=\cos(x)-S_n(x)$, il vient finalement : $\forall n\in\mathbb{N}^*, \forall x\in \mathrm{J}, \left|\cos(x)-S_n(x)\right|\leq\frac{x^{2n+2}}{(2n+2)!}$.

$$\forall n\in\mathbb{N}^*, \forall x\in \mathrm{J}, \left|\cos(x)-S_n(x)\right|\leq\frac{x^{2n+2}}{(2n+2)!}$$

Notre algorithme reste donc valable pour tout réel *x* de J dès lors que l'on ne souhaite pas retenir $S_0(x)=1$ comme valeur approchée du cosinus recherché…

Ceci dit, on peut très facilement montrer que la majoration précédente est encore valable pour $n=0$ et pour tout réel *x* de J ! ☺ MAIS en toute rigueur, ce n'est pas le critère spécial des séries alternées qui nous la fournit…