Université Paris I, Panthéon - Sorbonne

Licence M.I.A.S.H.S. deuxième année 2018 – 2019

Méthodes Numériques S4

Examen terminal, mai 2019

Examen de 2h00. Tout document ou calculatrice est interdit.

1. **(Sur 14 points)** On considère pour $n \in \mathbf{N}$, $I_n = \displaystyle\int_0^{\pi/2} \cos^{2n}(t)\, dt$.

   (a) Montrer que I_n existe pour tout $n \in \mathbf{N}$ **(0.5 pts)**. Déterminer I_0 **(0.5 pts)**.

   (b) Montrer que $(I_n)_{n\in\mathbf{N}}$ est une suite décroissante minorée **(1.5 pts)**. Que peut-on en déduire **(0.5 pts)**?

   (c) A l'aide d'intégration par parties, montrer que $I_{n+1} = \dfrac{2n+1}{2n+2}\, I_n$ pour tout $n \in \mathbf{N}$ **(2 pts)**.

   (d) En déduire que $I_n = \dfrac{(2n)!}{2^{2n}(n!)^2}\, \dfrac{\pi}{2}$ pour tout $n \in \mathbf{N}$ **(2 pts)**.

   (e) On désire calculer directement I_{10}, I_{50} et I_{100}. On écrit ainsi le programme en `R`:

   ```
   n=c(10,50,100)
   I=factorial(2*n)/((factorial(n))^2*2^(2*n))*pi/2
   ```

   Le résultat obtenu est `[1] 0.2767697 0.1250185 NaN`. Comment expliquer le dernier résultat **(1 pt)**? Comment aurait-on pu éviter cela **(0.5 pts)**? Ecrire un autre programme permettant de calculer I_{100} **(1.5 pts)**.

   (f) On va utiliser un autre programme pour approcher I_{50} et I_{100}. Soit la suite de commandes :

   ```
   m=1000
   k=c(1:m)
   I50=sum(cos((k-1)*pi/(2*m))^100+cos(k*pi/(2*m))^100)/(2*m)*pi/2
   I100=sum(cos((k-1)*pi/(2*m))^200+cos(k*pi/(2*m))^200)/(2*m)*pi/2
   I50; I100
   ```

   On a obtenu `[1] 0.1250185 [1] 0.08851198`. Expliquer ce qui a été fait avec ce programme, en écrivant notamment la formule théorique **(1 pt)**. Comment aurait-on pu présager à l'avance de la précision des calculs **(1.5 pt)**? Quelle méthode aurait-on pu utiliser pour approcher I_{100} encore plus précisément **(0.5 pts)**?

   (g) On va utiliser maintenant une autre méthode pour I_{50} et I_{100}. Soit le programme:

   ```
   m=1000
   X=runif(m,0,pi/2)
   IMC50=mean(cos(X)^100)*pi/2
   IMC100=mean(cos(X)^200)*pi/2
   IMC50; IMC100
   ```

   On obtient cette fois-ci les résultats `[1] 0.1121384 [1] 0.08072921`. Ecrire I_n comme une espérance et en déduire ce qui a été calculé en justifiant **(1.5 pts)**. Donner un contrôle de l'erreur commise **(1 pt)**. Pour approcher I_{100}, préfèrera-t-on 0.08851198 ou 0.08072921 **(0.5 pts)**?

Proof. (a) Pour tout $n \in \mathbf{N}$, la fonction $xin[0,\pi/2] \mapsto \cos^{2n}(x)$ est continue sur $[0,\pi/2]$, donc I_n existe.
On a $I_0 = \int_0^{\pi/2} dx = \frac{\pi}{2}$.

(b) On a pour tout $x \in [0,\pi/2]$, $0 \leq \cos x \leq 1$, donc $0 \leq \cos^{2n+2} x \leq \cos^{2n} x \leq 1$ et ainsi comme $f \leq g \Longrightarrow \int_a^b f(t)dt \leq \int_a^b g(t)dt$, on en déduit que $0 \leq I_{n+1} \leq I_n$ pour tout $n \in \mathbf{N}$: la suite est bien décroissante et minorée par 0.
On sait que l'on peut déduire que la suite est convergente.

(c) On a $I_{n+1} = \displaystyle\int_0^{\pi/2} \cos^{2n+2}(t)\,dt = I_n - \int_0^{\pi/2} \sin^2 t \cos^{2n}(t)\,dt = I_n - \int_0^{\pi/2} \sin t\,(\sin t \cos^{2n}(t))\,dt$. Avec une intégration par parties, $\displaystyle\int_0^{\pi/2} \sin t\,(\sin t \cos^{2n}(t))\,dt = \frac{1}{2n+1}\Big(\big[-\sin t \cos^{2n+1}(t)\big]_0^{\pi/2} + \int_0^{\pi/2} \cos^{2n+2}(t)\,dt\Big) = \frac{1}{2n+1}\,I_{n+1}$. On en déduit donc que $I_{n+1} = I_n - \frac{1}{2n+1}\,I_{n+1}$, d'où $I_{n+1} = \frac{2n+1}{2n+2}\,I_n$.

(d) On montre ceci par récurrence. Vrai pour $n=0$. Et si vrai au rang n, alors $I_{n+1} = \frac{(2n+1)(2n+2)}{(2n+2)^2}\,I_n = \frac{2n+1}{2n+2}\,\frac{(2n)!}{2^{2n}(n!)^2}\,\frac{\pi}{2} = \frac{(2(n+1))!}{2^{2(n+1)}((n+1)!)^2}\,\frac{\pi}{2}$, donc vrai au rang $n+1$.

(e) On a calculé directement I_{10}, I_{50} et I_{100} avec des factorielles par la dernière formule. Or (200)! est nombre plus grand que $2^{1024} \sim 10^{307}$, plus grand nombre calculable avec R. Donc on ne peut pas calculer directement I_{100}.
On pourrait plutôt considérer la formule $I_{n+1} = \frac{2n+1}{2n+2}\,I_n$, qui donne directement I_1 à partir de $I_0 = \pi/2$ et plus généralement $I_n = \frac{\pi}{2}\prod_{k=0}^{n-1}\frac{2k+1}{2k+2}$ qui ne pose plus de problème.
On pourra ainsi utiliser le programme:

```
n=100
I=pi/2
for (k in c(0:99))
{ I=I*(2*k+1)/(2*k+2)}
I
```

(f) On utilise ici une approximation de l'intégrale I_n par la méthode des trapèzes.
On sait que si la fonction à intégrer est de classe $\mathcal{C}^2$ alors l'erreur d'approximation est $\frac{1}{12\,m^2}\,(b-a)^3 M_2$, où $M_2 = \sup_{[a,b]}|f(x)|$. Donc ici la fonction étant bien de classe $\mathcal{C}^2$, et avec $M_2 = 2n(2n-1)$, $b-a = \pi/2$, on trouve donc une majoration de l'erreur commise.
On aurait été encore plus précis avec la méthode de Simpson.

(g) On utilise ici une méthode de Monte-Carlo pour estimer I_n. On a $I_n = \frac{\pi}{2}\int_{\mathbf{R}} f(x)\,\cos^{2n} x\,dx$ avec $f(x) = \frac{1}{\pi/2}$ pour $x \in [0,\pi/2]$ et 0 sinon. Ceci est la densité de probabilité d'une loi uniforme sur $[0,\pi/2]$. Ainsi, $I_n = \frac{\pi}{2}\,\mathbb{E}\big[\cos^{2n} U\big]$, où U suit une loi uniforme sur $[0,\pi/2]$.
On approche donc cette espérance par une moyenne empirique d'un échantillon de v.a.i.i.d. $(\frac{\pi}{2}\cos^{2n} U_1, \ldots, \frac{\pi}{2}\cos^{2n} U_n)$, grâce à la loi des grands nombres.
La précision est donnée par un intervalle de confiance de niveau 95% $\big[IMCn - 1.96\times\frac{\pi}{2}\,\frac{\sigma_n}{\sqrt{m}}\,,\,IMCn + 1.96\times\frac{\pi}{2}\,\frac{\sigma_n}{\sqrt{m}}\big]$ et $\sigma_n^2 = \left(\frac{\pi}{2}\right)^2\int_0^{\pi/2}\cos^{4n}(x)\,dx - I^2(n) = \left(\frac{\pi}{2}\right) I(2n) - I^2(n)$.
La précision en $1/m^2$ de $I100$ est bien supérieure à celle en $1/\sqrt{m}$ de $IMC100$.

□

2. **(Sur 10 points)** On sait que pour tout $x \in \mathbf{R}$, $\sin(x) = \displaystyle\sum_{k=0}^{\infty}(-1)^k\frac{x^{2k+1}}{(2k+1)!}$ et $\cos(x) = \displaystyle\sum_{k=0}^{\infty}(-1)^k\frac{x^{2k}}{(2k)!}$.
Le but de cet exercice est de donner une valeur approchée de π.

(a) Montrer que pour tout $x \in [0,2]$, $\displaystyle\Big|\sin x - \sum_{k=0}^{n}(-1)^k\frac{x^{2k+1}}{(2k+1)!}\Big| \leq \frac{2^{2n+3}}{(2n+3)!}$ **(1 pt)**.

(b) Montrer que $(2n+3)! \geq n^{n+2}$ pour tout $n \in \mathbf{N}$ **(1.5 pts)**. En déduire que pour $n=40$, $\displaystyle\Big|\sin x - \sum_{k=0}^{n}(-1)^k\frac{x^{2k+1}}{(2k+1)!}\Big| \leq \left(\frac{4}{n}\right)^{n+2} \leq 10^{-42}$ **(1.5 pt)**. De la même manière, pour $n=40$, on montre (ne pas le faire!) que $\displaystyle\Big|\cos x - \sum_{k=0}^{n}(-1)^k\frac{x^{2k}}{(2k)!}\Big| \leq \left(\frac{4}{n}\right)^{n+1} \leq 10^{-41}$.

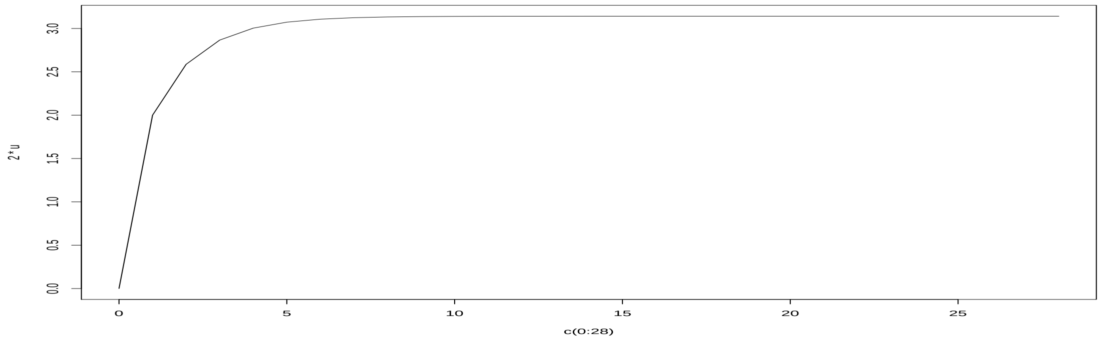
(c) Démontrer que l'équation $\sin x = 1$ admet une unique solution sur $[0,2]$ **(1 pt)**.

(d) Soit la suite de commandes en R:

```
u=0; u[2]=1; k=1; n=c(0:40);
epsilon=10^(-15)
while (abs(u[k+1]-u[k])>epsilon)
```

```
{k=k+1
si=sum((-1)^n*u[k]^(2*n+1)/factorial(2*n+1))
co=sum((-1)^n*u[k]^(2*n)/factorial(2*n))
u[k+1]=u[k]-(si-1)/co}
plot(c(0:28),2*u,"l")
```

Voici la courbe obtenue:



Expliquer ce qui a été fait par ces commandes, notamment ce qu'est l'objet `si`, la formule de calcul de `u[k]`, le rôle de `epsilon` et pourquoi la commande `plot` prend en compte `2*u` **(2 pts)**. Quel est grossièrement le coût en terme d'additions et multiplications de cette méthode d'approximation de π **(1 pt)**? Suggérer également une méthode à partir de l'approximation d'intégrales ou de séries entières, et donner le code `R` relatif **(1.5 pts)**. Est-ce plus intéressant **(0.5 pts)**?

Proof. (a) On utilise la formule du reste d'une série à termes alternés et ainsi pour tout $x \in [0,2]$, $\left|\sin x - \sum_{k=0}^{n}(-1)^k \frac{x^{2k+1}}{(2k+1)!}\right| \leq \frac{x^{2n+3}}{(2n+3)!}$, d'où le résultat.

(b) On a $(2n+3)! = 1 \times 2 \times \cdots \times n \times (n+1) \times (n+2) \times \cdots \times (2n+3)$. De $n+2$ à $2n+3$, on a $n+2$ termes tous supérieurs à n, et ce qui précède l'est également, donc $(2n+3)! \geq n^{n+2}$.
D'après ce qui précède, $\frac{2^{2n+3}}{(2n+3)!} \leq \frac{1}{2}\,\frac{2^{2(n+2)}}{n^{n+2}} \leq \left(\frac{4}{n}\right)^{n+2}$. Pour $n \geq 40$, on a $\frac{4}{n} \leq 10^{-1}$, d'où le résultat.

(c) Soit la fonction $g(x) = \sin x - 1$ pour $x \in [0,2]$. On a $g'(x) = \cos(x)$ qui est positive jusqu'à $\pi/2$ puis négative de $\pi/2$ à 2 en s'annulant en $\pi/2$. Donc la fonction admet un unique maximum local en $\pi/2$ (qui est aussi un maximum global sur $[0,2]$) et en $\pi/2$ on a $g(\pi/2) = 0$ soit $\sin(\pi/2) = 1$. L'unique solution de $\sin x = 1$ sur $[0,2]$ est $\pi/2$.

(d) On utilise ici une méthode de Newton-Raphson pour résoudre numériquement l'équation $g(x) = 0$. On a $g'(x) = \cos x$, d'où $u_{n+1} = u_n - g(u_n)/g'(u_n) = u_n - (\sin(u_n) - 1)/\cos(u_n)$.
On utilise l'approximation précédente de $\sin(u_n)$ et $\cos(u_n)$ à partir du développement en série entière jusqu'à l'ordre 40, c'est ce que représentent `si` et `co`.
La variable `epsilon` mesure la précision de l'algorithme de Newton-Raphson. On l'arrètera donc lorsque l'écart entre `u[k+1]` et `u[k]` est suffisamment petit.
La suite des `u[k]` converge vers $\pi/2$, solution de $\sin x = 1$ sur $[0,2]$. Pour approcher π on préférera donc considérer `2 *u[k]` qui s'approche de $2 * \pi/2 = \pi$.
Pour calculer cette approximation de π, on a eu besoin de $40 * 2$ calculs de sinet cos et une trentaine d'itérations pour Newton-Raphson, soit environ 2400 opérations.
On pouvait approcher π directement avec une série entière comme celle de $4\,\mathrm{Arctan}(x)$ pris en 1. Or le DSE de $\mathrm{Arctan}(x)$ est la primitive de celui de $(1+x^2)^{-1} = \sum_{k=0}^{\infty}(-1)^k x^{2k}$ pour $|x| < 1$, et ainsi $\mathrm{Arctan}(x) = \sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{2k+1}$, DSE également valable en 1. D'où $\pi = 4\sum_{k=0}^{\infty}(-1)^k \frac{1}{2k+1}$. Avec la majoration de l'erreur obtenue pour les séries à termes alternées, l'erreur commise est en $4/(2n+3)$ si on arrête le DSE en n. Il faudrait donc calculer de l'ordre de $n = 2 \times 10^{15}$ termes pour avoir une précision en 10^{-15}: notre algorithme est bien plus rapide.
Une autre solution serait d'approcher l'intégrale $4\int_0^1 (1+x^2)^{-1}dx$ avec la méthode de Simpson et l'approximation devient ainsi bien meilleure (précision en $1/n^4$)...

□