Centrale Maths 1 PSI 2009 — Corrigé

Ce corrigé est proposé par Céline Chevalier (ENS Cachan) ; il a été relu par Benoît Landelle (Professeur en CPGE) et Gilbert Monna (Professeur en CPGE).

Ce sujet traite exclusivement de suites et de séries. Il ne demande pas beaucoup de connaissances, mais une certaine aisance dans les calculs est requise pour mener à bien la plupart des démonstrations. Trois questions algorithmiques complètent cet énoncé, séparé en deux parties indépendantes.

- La première partie a pour objectif de montrer qu'à partir d'une série semi-convergente, on peut obtenir par permutation des termes une série dont la somme est égale à n'importe quel réel x. Le problème commence par présenter une méthode algorithmique pour construire de manière effective une telle permutation des termes, et il est demandé de la programmer, au choix, en Maple ou en Mathematica. Après avoir constaté graphiquement que la méthode fonctionne, on démontre mathématiquement le résultat dans la suite de la partie.
- La deuxième partie cherche à caractériser deux types de suites (a_n). On dit qu'elles vérifient (P_1) si, pour toute suite complexe (u_n) bornée, la série de terme général $a_n u_n$ converge. Elles vérifient (P_2) si, pour toute suite réelle (u_n), la convergence de la série de terme général u_n entraîne celle de la série de terme général $a_n u_n$. On commence par caractériser les suites vérifiant (P_1). Ensuite, l'essentiel de cette partie est consacrée, pour une série $\sum a_n$ divergente, à la construction effective et algorithmique d'une suite (ε_n) de limite nulle telle que la série $\sum \varepsilon_n a_n$ reste divergente. Les deux dernières questions utilisent ce résultat pour caractériser les suites vérifiant (P_2).

Ce problème illustre donc la différence entre la semi-convergence et la convergence absolue, en montrant qu'en cas de semi-convergence on peut obtenir n'importe quelle somme en modifiant l'ordre des termes, alors que pour une série absolument convergente, toutes les séries obtenues par permutation des termes convergent vers la même somme.

Ce sujet est particulièrement déroutant et assez difficile avec plusieurs niveaux d'indices, beaucoup de notations et des références aux questions précédentes très fréquentes qui font parfois penser à un jeu de piste. En outre, il teste très peu les connaissances du programme, laissant la plus grande part à la virtuosité dans les calculs. Il peut être utilement employé pour s'entraîner à calculer sans se tromper.

Indications

Partie I

I.A.1 Effectuer une boucle `for` sur n et tester à chaque étape si $\mathrm{S}_n > x$ pour construire en fonction du résultat p_n, q_n, s_n et S_n. Stocker à chaque itération la valeur s_n dans la liste.

I.A.2 La courbe représente S_n en fonction de n. On constate qu'elle se rapproche de la droite $y = x$.

I.B Procéder par récurrence sur $n \geqslant 1$ et distinguer les cas où $\mathrm{S}_n > x$ et $\mathrm{S}_n \leqslant x$.

I.C.1 Revenir à la définition « avec des ε » d'une suite convergente.

I.C.2.a Il existe un rang n_0 à partir duquel $(p_n)_{n\in\mathbb{N}}$ est constante ; en déduire la valeur des suites $(q_n)_{n\in\mathbb{N}}$ et $(s_n)_{n\in\mathbb{N}}$. Exprimer ensuite S_n en fonction de S_{n_0}.

I.C.4 Utiliser l'égalité d'ensembles de la question I.B .

I.D.1 Distinguer les cas $\mathrm{S}_n > x$ et $\mathrm{S}_n \leqslant x$.

I.D.2 Raisonner par l'absurde et utiliser le raisonnement de la question I.C.2.a .

I.E.1 Remarquer que

$$\sum_{k=1}^{n} \frac{1}{k} - \ln n = \sum_{k=1}^{n-1} \left(\frac{1}{k} - \int_k^{k+1} \frac{1}{t}\,\mathrm{d}t \right) + \frac{1}{n}$$

I.E.3.c La limite de $(\mathrm{S}_n)_{n\in\mathbb{N}}$ est x.

I.E.3.d Effectuer un développement limité de l'expression donnée en utilisant les résultats des questions I.E.1 et I.E.2 .

Partie II

II.B.2 Développer la somme et effectuer un changement d'indice.

II.D.1 Effectuer une boucle `for` sur i. À chaque étape, tester la valeur de A_i en fonction de p_i et en déduire p_{i+1}, ε_{i+1} et A_{i+1}.

II.D.2.a Raisonner par l'absurde et montrer l'existence d'un entier N tel que l'on ait l'égalité $\mathrm{A}_n = \mathrm{A}_\mathrm{N} + \varepsilon_\mathrm{N}(a_{\mathrm{N}+1} + \cdots + a_n)$.

II.D.2.b Raisonner par récurrence.

II.D.3.a La procédure est presque la même que la fonction `exemple`. On maintient également un compteur et, quand on se trouve dans le premier cas, on stocke ce compteur ainsi que la valeur de l'itérateur de la boucle.

II.D.3.b Partir de l'égalité $\mathrm{A}_{n_k-1} = \mathrm{A}_{n_k-2} + a_{n_k-1}\varepsilon_{n_k-1}$. Calculer ensuite A_{n_k} et A_{n_k+1} en traduisant la condition $n_{k+1} - 2 > n_k$ avec $p_{n_k+2} = p_{n_k+1}$ puis l'inégalité $\mathrm{A}_{n_k+1} < p_{n_k+1}$.

II.D.3.c Utiliser la définition donnée dans la question II.D, puis une comparaison série-intégrale.

II.D.3.d On peut appliquer l'inégalité de la question II.D.3.b au rang suivant.

II.D.3.e Étudier la convergence de la série de terme général $\ln n_{k+1} - 2^{k+1} - \ln n_k + 2^k$ et encadrer n entre n_k et $n_k + 1$.

II.E.b Raisonner par l'absurde et appliquer la question II.D .

II.F.1 Revenir à la définition « avec des ε » : la limite de $(x_n)_{n\in\mathbb{N}}$ et de $(a_n x_n)_{n\in\mathbb{N}}$ est 0.

II.F.2 Séparer la somme et effectuer un changement d'indice ; appliquer ensuite la question II.E .

I. Réorganisation des termes d'une série semi-convergente

I.A.1 Pour écrire la fonction `suite`, on commence par initialiser les différentes variables. La liste est construite à l'aide d'une boucle `for` dans laquelle, à chaque étape, on applique la définition de la construction des suites (p_n), (q_n), (s_n) et (S_n).

```
suite := proc(x,n)
    local p,q, s, i, L, S;
    p[0] :=0; q[0] := 0; L := []; S[0] := 0;
    for i from 0 to n-1 do
        if (S[i] > x)
        then q[i+1] := 1 + q[i]; p[i+1] := p[i];
             s[i+1] := 2 * q[i+1] - 1;
        else q[i+1] := q[i]; p[i+1] := 1 + p[i];
             s[i+1] := 2 * p[i+1];
        fi;
        L := [op(L),s[i+1]];
        S[i+1] := S[i] + u(s[i+1]);
    od;
    L;
end;

u := proc(n) (-1)^n/n; end;
```

> Le programme consommerait moins de mémoire en utilisant toujours la même variable (par exemple seulement p pour à la fois $p[0]$, $p[i]$ et $p[i+1]$). On a toutefois fait ici le choix d'utiliser des tableaux par souci de lisibilité. En outre, ces derniers sont utiles pour représenter graphiquement les valeurs, comme cela est fait dans la question suivante de l'énoncé.

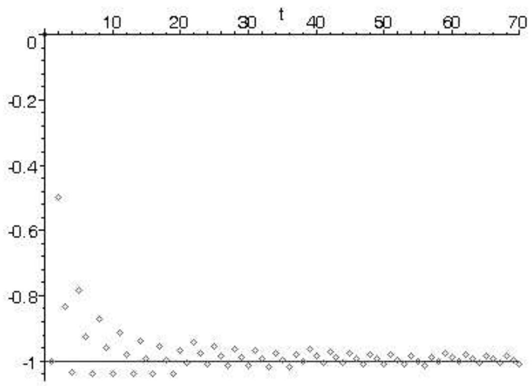
I.A.2 On constate sur le dessin que les points de la suite se rapprochent de plus en plus de la valeur limite x :

$$\boxed{\text{La suite } (\mathrm{S}_n) \text{ tend vers } x = -1.}$$

L'algorithme construit la suite (S_n) en utilisant trois suites auxiliaires pour les indices k des u_k. On réordonne les termes en sommant plutôt les u_{s_k} que les u_k. Ainsi, les indices sont les s_k. Plus précisément, la suite (q_n) est un compteur pour les indices impairs $2k+1$ et la suite (p_n) pour les indices pairs $2k$. Suivant la valeur de S_n, s_n est pair (et vaut $2p_n$) ou impair (et vaut $2q_n - 1$). Lorsque $\mathrm{S}_n \leqslant x$, l'algorithme ajoute un terme pair (qui est positif) et lorsque $\mathrm{S}_n > x$, il ajoute un terme impair (qui est négatif). Dans les deux cas, on fait en sorte que S_n ne s'éloigne pas de x (en changeant le signe des termes à additionner dès que S_n dépasse x, dans un sens ou dans l'autre), et on montre dans la suite de la partie qu'elle s'en rapproche même.

> À titre indicatif, on pouvait écrire le code de l'affichage des points par exemple comme dans la fonction ci-après, en modifiant la liste `L` pour qu'elle contienne les couples (i, S_i), et en faisant afficher les points de la liste ainsi obtenue ainsi que la droite horizontale d'ordonnée `x`.

```
suite := proc(x,n)
    local p,q, s, i, L, S;
    p[0] :=0; q[0] := 0; L := [[0,0]]; S[0] := 0;
    for i from 0 to n-1 do
        if (S[i] > x)
        then q[i+1] := 1 + q[i]; p[i+1] := p[i];
             s[i+1] := 2 * q[i+1] - 1;
        else q[i+1] := q[i]; p[i+1] := 1 + p[i];
             s[i+1] := 2 * p[i+1];
        fi;
        S[i+1] := S[i] + u(s[i+1]);
        L := [op(L),[i+1,S[i+1]]];
    od;
    L;
    plot([x,L],t=0..n,style=[line,point]);
end;
```



I.B On procède par récurrence sur $n \geqslant 1$ pour montrer que la propriété

$$\mathscr{P}(n) : \quad \begin{cases} \{s(1),\ldots,s(n)\} = \{2,4,\ldots,2p_n\} \cup \{1,3,\ldots,2q_n-1\} \\ p_n + q_n = n \\ \mathrm{S}_n = u_{s(1)} + \cdots + u_{s(n)} \end{cases}$$

est vraie pour tout $n \geqslant 1$.

- $\underline{\mathscr{P}(1)}$: si $x \geqslant \mathrm{S}_0 = 0$, on a $q_1 = q_0 = 0$, $p_1 = 1 + p_0 = 1$ d'où $p_1 + q_1 = 1$. Comme $s(1) = s_1 = 2p_1 = 2$, $\{s(1)\} = \{2, 2p_1\}$ car, comme $2q_1 - 1 = -1$, le deuxième ensemble proposé est vide. En outre, $\mathrm{S}_1 = \mathrm{S}_0 + u_{s_1} = u_{s(1)}$.
  Si $x < \mathrm{S}_0 = 0$, on a $q_1 = 1 + q_0 = 1$, $p_1 = p_0 = 0$ d'où $p_1 + q_1 = 1$. Comme $s(1) = s_1 = 2q_1 - 1 = 1$, on a $\{s(1)\} = \{1, 2q_1 - 1\}$ car, comme $2p_1 = 0$, le premier ensemble proposé est vide. En outre, $\mathrm{S}_1 = \mathrm{S}_0 + u_{s_1} = u_{s(1)}$.
  Dans les deux cas, $\mathscr{P}(1)$ est vraie.
- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$: si $x \geqslant \mathrm{S}_n$, on a $q_{n+1} = q_n$, $p_{n+1} = 1 + p_n$ d'où l'égalité $p_{n+1} + q_{n+1} = 1 + p_n + q_n = 1 + n$ d'après l'hypothèse de récurrence. Comme $s(n+1) = s_{n+1} = 2p_{n+1}$,
  $$\{s(1),\ldots,s(n+1)\} = \{2, 2p_1, \ldots, 2p_{n+1}\} \cup \{1, \ldots, 2q_n - 1, \overbrace{2q_{n+1} - 1}^{=2q_n-1}\}$$
  puisque les deux derniers éléments du second ensemble sont égaux. En outre, $\mathrm{S}_{n+1} = \mathrm{S}_n + u_{s_{n+1}} = u_{s(1)} + \cdots + u_{s(n)} + u_{s(n+1)}$ par hypothèse de récurrence.