

Le barème proposé est sur 24 points (la note sur 24 étant considérée sur 20).

Le polycopié de cours est autorisé, à l'exception de tout autre document.

Exercice 1 (sur 8 points) Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction numérique de classe C^∞ prenant des valeurs rationnelles aux points rationnels, a et b deux nombres rationnels, et $(x_n)_{n \geq 1}$ la suite de nombres réels définie par $x_1 = a$, $x_2 = b$ et, pour tout $n > 2$,

$$x_n = \begin{cases} x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})} & \text{si } f(x_{n-1}) \neq f(x_{n-2}) \\ \frac{x_{n-1} + x_{n-2}}{2} & \text{sinon.} \end{cases}$$

1. Dites pourquoi la suite $(x_n)_{n \geq 1}$ est une suite de nombres rationnels.
2. On suppose que $f(a)f(b) < 0$. À quelle méthode algorithmique correspond la génération de cette suite $(x_n)_{n \geq 1}$? Si cette suite converge vers un nombre réel ξ tel que $f'(\xi) \neq 0$ et n'est pas stationnaire, pourquoi a-t-on $f(\xi) = 0$? Quelle est l'ordre de convergence de cette méthode algorithmique (on pourra se contenter d'en donner une minoration) ?
3. Rédiger, en complétant le code Maple suivant, une procédure récursive qui calcule x_N pour $N \geq 1$, a, b étant des rationnels fixés (la fonction f ayant été préalablement déclarée sous Maple) :

```

algorithmme := proc(f,a,b,N) option remember;
local ... ;
if N=1 then
... ;
elif N=2 then
... ;
else
... ;
... ;
if ... then
... ;
else
... ;
end if;
end if;
end proc;

```

Expliquer pourquoi les deux boucles `if ... end if` sont essentielles. Que se passerait-il en particulier si l'on omettait la première d'entre elles ? Quel sont à la fois le sens et l'intérêt de l'instruction `option remember` ?

Exercice 2 (sur 8 points)

1. Rappeler comment déclarer sous l'environnement **MATLAB** la fonction

$$f : t \in \mathbb{R} \mapsto \frac{t^2 \times (\cos(t)) \times (\ln(t^2 + 1))}{2 + t^4 - (\sin(t))^2},$$

de manière à cette fonction puisse aussi être directement évaluée aussi bien sur un vecteur ligne `t` que sur un simple scalaire `t`.

2. Rédiger un code (sous **MATLAB**) :

```
function Im = trapezes (a,b,f,m) ;
```

qui, étant donnés deux réels $a < b$ et un entier $m \in \mathbb{N}^*$, renvoie en sortie, une fois exécuté, l'intégrale approchée `Im` de `f` sur le segment $[a, b]$, calculée suivant la méthode des trapèzes composite, avec comme pas $h_m = (b - a)/M$, où $M = 2^m$. Que vaut (en vous reportant au cours) le plus grand entier $k \in \mathbb{N}^*$ tel que l'on puisse affirmer :

$$\left| \int_a^b f(t) dt - \text{Im} \right| = O\left(\left(\frac{b-a}{2^m}\right)^k\right)$$

lorsque m croît vers l'infini ?

3. On admet (suivant en cela la *formule d'Euler-MacLaurin* mentionnée en cours), que, lorsque $m \in \mathbb{N}^*$ croît vers l'infini,

$$\text{Im} = \int_a^b f(t) dt + \alpha_0[f] \left(\frac{b-a}{2^m}\right)^2 + O\left(\left(\frac{b-a}{2^m}\right)^4\right),$$

avec $\alpha_0[f] = (f'(a) - f'(b))/12$. Suivant le principe d'extrapolation de L. Richardson, modifier le code `trapezes` en un code :

```
function ImBis = trapezes2 (a,b,f,m) ;
```

de manière à ce que, cette fois, on puisse affirmer :

$$\left| \int_a^b f(t) dt - \text{ImBis} \right| = O\left(\left(\frac{b-a}{2^m}\right)^4\right)$$

lorsque m croît vers l'infini.

4. Montrer que `ImBis` correspond au calcul approché de l'intégrale de `f` sur le segment $[a, b]$, calculée suivant cette fois la méthode de Simpson composite, avec comme pas toujours $h_m = (b - a)/M$, où $M = 2^m$.

Exercice 3 (sur 8 points) On rappelle (voir le cours d'Algèbre 2) que si A désigne une matrice réelle à m lignes et n colonnes et A' sa transposée, alors, pour tout vecteur colonne $x \in \mathbb{R}^n$, on a

$$\langle A' \cdot A \cdot x, x \rangle = \|A \cdot x\|^2, \quad (*)$$

où $\| \cdot \|$ désigne la norme euclidienne sur $\mathbb{R}^n$ et $\langle \cdot, \cdot \rangle$ le produit scalaire usuel dans $\mathbb{R}^n$. Soit A une telle matrice, avec de plus $m \geq n$ et $\text{rang}(A) = n$.

1. En utilisant (*), vérifier que la matrice $M := A' \cdot A$ est une matrice symétrique réelle définie positive (c'est-à-dire dont toutes les valeurs propres sont strictement positives).
2. Soit B un vecteur colonne de $\mathbb{R}^n$. Écrire sous **MATLAB**¹ le code d'une procédure itérative (reposant sur le théorème du point fixe dans $\mathbb{R}^n$, au travers d'un algorithme vu en cours)

```
function XX = resolIterative (M,B,X,k) ;
```

qui, initiée au vecteur colonne $X \in \mathbb{R}^n$, fournit au bout de k itérations une approximation XX de la solution du système de Cramer $M \cdot XX = B$.

3. En y intégrant une boucle `while (...) && (...) ... end`, modifier le code `resolIterative` rédigé à la question 2 en un code

```
function [XX,Niter] = resolIterative2 (M,B,X,epsilon,k);
```

de manière à ce que la procédure correspondante s'arrête automatiquement lors de son exécution dès qu'au terme d'un certain nombre d'itérations $0 \leq \text{Niter} \leq k$, on trouve pour la première fois

$$\|XX(\text{Niter} + 1) - XX(\text{Niter})\| \leq \text{epsilon},$$

où `epsilon` désigne un seuil strictement positif donné. On note ici $XX(l)$, $l = 0, \dots, k$, l'état de la variable XX au terme de l itérations lors de l'exécution du code. Que signifie le fait de prendre `epsilon=eps` ?

1. Dans cette question, comme dans la suivante, on pourra, faute d'exprimer le code sous la syntaxe du logiciel **MATLAB**, se contenter de le rédiger sous forme « pseudo-algorithmique », pourvu d'en sérier clairement les instructions.