

Examen de TP – 14 Mai 2013 –14h-17h

Modalités de l'épreuve (3 heures) : *Seul le mémento Scilab est autorisé, à l'exclusion de tout autre document. Chacun rendra à la fin de l'épreuve une copie papier, faisant la liste des questions traitées et répondant aux questions demandant une réponse écrite.*

Vous ouvrirez en début de séance un fichier nommé impérativement nom-prenom.sci où nom est votre nom de famille.

Vous stockerez dans nom-prenom.sci les fonctions Scilab qui vous sont demandées.

En fin de séance, bien enregistrer la dernière version de votre fichier sur le disque D, ne pas éteindre votre poste.

Dans tous les cas, ne partez pas avant que l'enseignant ait vérifié l'enregistrement de votre fichier.

Il est bien sûr vivement recommandé d'enregistrer régulièrement vos fichiers.

Le barème [.] est indicatif. Les exercices sont indépendants.

Exercice 1. [12] Intégration numérique

Le but de l'exercice est de mettre en oeuvre la méthode de Simpson pour de l'intégration numérique.

- (1) On désigne par "intervalle de référence" de la méthode, l'intervalle $I = [0, 1]$.

Soit $g : \xi \in I \mapsto g(\xi) \in \mathbb{R}$ une fonction continue sur $I = [0, 1]$.

On définit la méthode de Simpson pour le calcul numérique de l'intégrale de g sur I par :

$$\int_0^1 g(\xi) d\xi \simeq \frac{1}{6}g(0) + \frac{4}{6}g(1/2) + \frac{1}{6}g(1).$$

- (a) [1] Montrer que la méthode est exacte si g est un polynôme de degré inférieur ou égal à 2.
- (b) [1] Ecrire une fonction Scilab `J=Simpson1(g)` qui prend en entrée une fonction `g` et rend en sortie l'approximation de $J = \int_0^1 g(\xi) d\xi$ par la méthode de Simpson précédente.
- (c) [1] Tester `Simpson1(g)` pour 3 polynômes de degré 3 et comparer avec la valeur exacte de $J = \int_0^1 g(\xi) d\xi$. (On donnera les polynômes considérés et le calcul des valeurs exactes de J). Quelle conclusion en tirez-vous ?
- (2) Soit $K = [a, b]$ un intervalle et soit $f : x \in K \mapsto f(x) \in \mathbb{R}$ une fonction continue sur K .
- (a) [1] Expliciter, en fonction de f et a, b , la fonction $g : \xi \in I = [0, 1] \mapsto g(\xi) \in \mathbb{R}$ telle que, par changement de variables affine $x = T_K(\xi)$ (T_K polynôme de degré 1 que vous donnerez),

$$\int_a^b f(x) dx = \int_0^1 g(\xi) d\xi.$$

- (b) [1] En déduire une fonction Scilab `J=Simpson2(f,a,b)` qui prend en entrée une fonction `f` et les coordonnées `a, b` de l'intervalle $K = [a, b]$. En sortie l'approximation de $J = \int_a^b f(x) dx$ via la fonction `Simpson1`.

- (c) [1] Tester `Simpson2(f,a,b)` pour vos 3 polynômes précédents, pour $a=0$ et $b=2$ et comparer avec la valeur exacte de $J = \int_0^2 f(x) dx$. (On donnera les polynômes considérés et le calcul des valeurs exactes de J). Que constatez-vous ? Expliquer pourquoi.
- (3) [2] Soit $f : x \in [c, d] \mapsto f(x) \in \mathbb{R}$ une fonction continue sur l'intervalle $[c, d]$. On se donne un maillage de l'intervalle,

$$c = x_1 < x_2 < \cdots < x_{N-1} < x_N = d.$$

On veut calculer l'intégrale $J = \int_c^d f(x) dx$ par la relation de Chasles,

$$J = \int_{x_1}^{x_2} f(x) dx + \cdots + \int_{x_{N-1}}^{x_N} f(x) dx = \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} f(x) dx,$$

et où chaque intégrale est calculée par méthode de Simpson (fonction `Simpson2`).

Ecrire une fonction Scilab `J=Simpson(f,COOR)` qui prend en entrée une fonction `f` et un vecteur `COOR=[x1,x2,...,xN]` donnant les coordonnées des points définissant le maillage. En sortie l'approximation de $J = \int_{x_1}^{x_N} f(x) dx$ par la méthode de Simpson.

- (4) On regarde la fonction $x \mapsto F(x) = \sqrt{1-x^2}$ sur l'intervalle $[-1, 1]$.
- (a) [1] Montrer que $\int_{-1}^1 F(x) dx = \frac{\pi}{2}$.
- (b) [1] Ecrire une fonction `Err1= err1(N)` qui prend en entrée un entier $N \geq 2$. Cette fonction calcule, pour tous les entiers $n \in [2, N]$, la valeur absolue $e_1(n) = |\text{Simpson}(F, \text{COOR}) - \pi/2|$ pour le vecteur `COOR` formant un maillage uniforme à $3n$ points de $[-1, 1]$. En sortie le vecteur `Err1` = $(e_1(2), \dots, e_1(N))$.
- (c) [1] Ecrire une fonction `Err2= err2(N)` qui prend en entrée un entier $N \geq 2$. Cette fonction calcule, pour tous les entiers $n \in [2, N]$, la valeur absolue $e_2(n) = |\text{Simpson}(F, \text{COOR}) - \pi/2|$ pour le vecteur `COOR` obtenu comme concaténation d'un maillage uniforme à n points de $[-1, -0.9]$, d'un maillage uniforme à n points de $[-0.9, 0.9]$, d'un maillage uniforme à n points de $[0.9, 1]$; En sortie le vecteur `Err2` = $(e_2(2), \dots, e_2(N))$.
- (d) [1] Ecrire une fonction `TestSimpson(N)` qui prend en entrée un entier $N \geq 2$. Cette fonction :
- trace le polygone des $N-1$ points $(n, e_1(n))$ pour $n = 2, \dots, N$,
 - trace le polygone des $N-1$ points $(n, e_2(n))$ pour $n = 2, \dots, N$,
 - les graphes des fonctions $n \mapsto (3n)^{-2}$ et $n \mapsto (3n)^{-3}$ sur l'intervalle $[2, N]$.
- On choisira convenablement le mode (normal ou logarithmique).
Quelle conclusion en tirez-vous ?

Exercice 2. [11+3] Suites dans le plan

On considère les fonctions $f : (x, y) \in \mathbb{R}^2 \mapsto x^2 - y^2 + c$ et $g : (x, y) \in \mathbb{R}^2 \mapsto 2xy + d$, où c et d sont des constantes réelles. On veut construire et représenter la suite $(u_n, v_n)_{n \geq 0}$ définie par récurrence par :

$$\begin{cases} u_{n+1} = f(u_n, v_n) \\ v_{n+1} = g(u_n, v_n) \end{cases}, \quad (u_0, v_0) \text{ fixé.}$$

- (1) On considère le cas $c = d = 0$.

- (a) [1] Pour $(u_0, v_0) = (2, 1)$, calculer (u_1, v_1) et (u_2, v_2) .
- (b) [1] Justifier que pour tout $(u_0, v_0) \in [-\frac{1}{2}, \frac{1}{2}]^2$ on a $|u_1| \leq \frac{1}{2^2}$ et $|v_1| \leq \frac{1}{2}$, puis que $(u_2, v_2) \in [-\frac{1}{2^2}, \frac{1}{2^2}]^2$.
- (c) [2] On suppose que $(u_0, v_0) \in [-\frac{1}{2}, \frac{1}{2}]^2$. Montrer par récurrence que pour tout $n \geq 2$, $(u_n, v_n) \in [-\frac{1}{2^n}, \frac{1}{2^n}]^2$ et en déduire la limite de la suite (u_n, v_n) .
- (d) [1] Ecrire une fonction Scilab `w=suite1(u0, v0, N)` avec en entrée les premiers termes `u0` et `v0`, un entier `N`. En sortie, les `N+1` valeurs de la suite (u_n, v_n) sous la forme d'une matrice : $w = \begin{pmatrix} u_0, & v_0 \\ u_1, & v_1 \\ \vdots, & \vdots \\ u_N, & v_N \end{pmatrix}$.

On testera la fonction `suite1` uniquement pour des valeurs $(u_0, v_0) \in [-\frac{1}{2}, \frac{1}{2}]^2$.

(2) On considère maintenant le cas où c et d peuvent être quelconques.

- (a) [2] On suppose que la limite $\lim_{n \rightarrow +\infty} (u_n, v_n)$ existe, est finie et vaut (u, v) . Quelles équations doivent vérifier u et v ? (Justifier).
On prend $c \geq 0$ et $d = \sqrt{c}$. Calculer les deux seules limites finies possibles. (On trouvera deux racines évidentes).

- (b) [2] Ecrire une fonction Scilab `w=suite2(u0, v0, N, c,d)` avec en entrée les coordonnées `u0` et `v0` du premier terme (u_0, v_0) , un entier `N`, et les paramètres `c` et `d`. En sortie, les `m+1` valeurs de la suite (u_n, v_n) sous la forme d'une matrice :

$$w = \begin{pmatrix} u_0, & v_0 \\ u_1, & v_1 \\ \vdots, & \vdots \\ u_m, & v_m \end{pmatrix}, \text{ où l'entier } m \text{ est le plus grand des entiers qui vérifie : } m \leq N$$

et pour tout $0 \leq n \leq m$, $\sqrt{u_n^2 + v_n^2} < 2$.

- (c) [2] Ecrire une fonction Scilab `graphe(u0, v0, N, c,d)` qui trace le polygone des points $(u_n, v_n)_{0 \leq n \leq m}$ calculé par la fonction Scilab `suite2` (autrement dit, le point de coordonnées (u_n, v_n) est relié au point suivant (u_{n+1}, v_{n+1}) par un segment de droite).

Tester avec `graphe(0.2, 0.2, 100, 0.2, 0.5)`. Que constatez-vous?

Tester avec `graphe(0.1, 0.5, 300, 0.25, 0.5)`, puis

`graphe(0, 0.5, 300, 0.25, 0.5)`. Que constatez-vous? (On fera le lien avec la question 2-a).

(3) [3] **Bonus : uniquement si vous avez du temps !**

On veut représenter l'ensemble des points (u_0, v_0) tels que la suite (u_n, v_n) reste bornée. Pour cela, pour chaque point (u_0, v_0) d'une grille de points, on garde (u_0, v_0) seulement si `suite2(u0, v0, 100, c,d)` est un vecteur à 101 lignes.

Ecrire une fonction `M=Julia(xmin, xmax, ymin, ymax, nx, ny, c,d)` avec en entrée les coordonnées $[xmin, xmax] \times [ymin, ymax]$, pour un maillage uniforme à `nx` points de $[xmin, xmax]$ et à `ny` points de $[ymin, ymax]$. (Ainsi, si $[xmin, xmax] \times [ymin, ymax] = [0, 2] \times [0, 3]$ et que `nx=3` et `ny=4`, on prend la grille des points $\{(x, y) \mid x = 0, 1, 2 \text{ et } y = 0, 1, 2, 3\}$, soit 12 points). En sortie, l'ensemble des points

(u_0, v_0) de la grille, sous la forme d'une matrice : $M = \begin{pmatrix} u_0^1 & v_0^1 \\ u_0^2 & v_0^2 \\ \vdots & \vdots \end{pmatrix}$ qui satisfont aux conditions demandées.

Ecrire une fonction `traceJulia(xmin, xmax, ymin, ymax, nx, ny, c1, c2)` qui trace en style "0" l'ensemble des points (dit de "Julia rempli") donné par la matrice M précédente.

Tester avec `traceJulia(-0.4, 0.4, -0.4, 0.4, 30, 30, 0.3, 0.5)` . Voir Fig. 1, où on a pris plus de points et du temps de calcul.

FIG. 1. Ensemble de Julia rempli associé à l'exercice 1, pour $c = 0.3$ et $d = 0.5$.