

DEVOIR À LA MAISON

Une copie, de préférence manuscrite, doit être remise par personne. Elle devra comporter les commandes Maple correspondant aux exercices (il n'est pas nécessaire d'indiquer les réponses de Maple). Dans l'exercice 3, les questions 1a et 2a sont uniquement mathématiques. Tout travail remis en retard aura une note réduite. Vous pouvez travailler à deux ou trois, mais chacun doit remettre une copie individuelle.

Les trois exercices sont indépendants. Les deux premiers peuvent être traités sans le logiciel (avec papier, crayon et documents de cours). Maple est disponible dans la salle A du Script, barre 65-66, deuxième étage. La salle n'est pas disponible tout le temps, aussi renseignez-vous sur ses horaires en libre-service !

EXERCICE 1. On considère le polynôme $P = z^{10} + z^2 + 1$ et ses racines complexes. Donner les commandes Maple qui permettent d'afficher la liste des valeurs approchées des modules de ces racines. Même question avec la liste des valeurs approchées des arguments.

EXERCICE 2. Ecrire une procédure qui prend en entrée un polynôme P de degré 2 et qui affiche en sortie ses racines réelles éventuelles.

Instructions à suivre : la procédure devra vérifier que P est de degré 2 et afficher un message d'erreur le cas échéant. Elle devra traiter tous les cas et ne devra pas utiliser les commandes de résolution d'équation comme *solve*, *fsolve*, ...

Commandes utiles : $\text{degree}(P, X)$ donne le degré en X de P
 $\text{coeff}(P, X, k)$ donne le coefficient de X^k dans P

EXERCICE 3. Approximation d'intégrale

Soit f une fonction continue positive de $[a, b]$ dans \mathbb{R} . Le but de l'exercice est d'approcher l'intégrale $J = \int_a^b f(x)dx$ par deux méthodes classiques.

1) Méthode des rectangles

On découpe $[a, b]$ en n intervalles égaux. On approche l'intégrale de f sur chacun de ces intervalles par l'aire du rectangle ayant pour base cet intervalle et pour hauteur la valeur de f en l'extrémité gauche de l'intervalle.

- (a) En vous aidant d'un dessin, exprimer l'aire totale $J_{\text{rect}}(f, a, b, n)$ calculée ainsi entre a et b , en fonction de f, a, b et n .
- (b) Dans Maple, écrire une fonction ou une procédure *rectangle*, qui prend en entrée f, a, b et n et calcule $J_{\text{rect}}(f, a, b, n)$. On utilisera la commande *sum* plutôt que *add*.

Pour les trois questions suivantes, on prendra $f(x) = x^2 \ln(x+1)$, $a = 0$, $b = 5$.

(c) Tester *rectangle* sur cet exemple en prenant différentes valeurs pour n . Comparer les valeurs approchées de vos résultats à ceux de la commande Maple qui calcule la méthode des rectangles : commande *leftsum* de la librairie *student*.

(d) Vérifier que

$$\lim_{n \rightarrow \infty} Jrect(f, a, b, n) = J.$$

(e) Expliquer pourquoi il existe un entier m tel que :

$$|J - Jrect(f, a, b, m)| \leq 0.3$$

A l'aide d'une boucle *while*, trouver le plus petit entier m vérifiant cette propriété.

2) Méthode des trapèzes

On découpe $[a, b]$ en n intervalles égaux. On approche l'intégrale de f sur chacun de ces intervalles par l'aire du trapèze ayant pour sommets les extrémités de l'intervalle et les images par f de ces extrémités.

(a) En vous aidant d'un dessin, exprimer l'aire totale $Jtrap(f, a, b, n)$ calculée ainsi entre a et b , en fonction de f, a, b et n . On rappelle que l'aire d'un trapèze de bases B_1, B_2 et de hauteur h est donnée par $(B_1 + B_2)h/2$.

(b) Dans Maple, écrire une fonction ou une procédure *trapeze*, qui prend en entrée f, a, b et n et calcule $Jtrap(f, a, b, n)$. On utilisera la commande *sum* plutôt que *add*.

Pour les trois questions suivantes, on prendra $f(x) = x^2 \ln(x+1)$, $a = 0$, $b = 5$.

(c) Tester *trapeze* sur cet exemple en prenant différentes valeurs pour n . Comparer les valeurs approchées de vos résultats à ceux de la commande Maple qui calcule la méthode des trapèzes : commande *trapezoid* de la librairie *student*.

(d) Vérifier que

$$\lim_{n \rightarrow \infty} Jtrap(f, a, b, n) = J.$$

(e) Expliquer pourquoi il existe un entier p tel que :

$$|J - Jtrap(f, a, b, p)| \leq 0.3$$

A l'aide d'une boucle *while*, trouver le plus petit entier p vérifiant cette propriété. Comparer ce résultat à celui obtenu à la question 1.e . Qu'en pensez-vous ?