IUT Littoral | **2019 - 2020**
Info 2A | **Ana. Num. – C++**

TP 1 - Introduction

Pour représenter les nombres réels nous allons utiliser le type `double` du `C++`. Le type `double` étant habituellement codé sur 64 bits, nous n'aurons accès qu'à 2^{64} nombres distincts. Nous ne manipulerons donc pas directement les nombres réels mais des approximations. Le type `double` permet de représenter les nombres entre -1.7×10^{-308} et 1.7×10^{308}.

1. Fonction réelle

Une fonction mathématique $f : \mathbb{R} \to \mathbb{R}$ sera codée en `C++` par `double f(double)`.

Exercice 1. On rappel que pour tout $x \in \mathbb{R}$ la fonction exponentielle est donnée par :

$$\exp(x) = \sum_{k=0}^{+\infty} \frac{x^k}{k!}.$$

a. Ecrire une fonction `double exp_v1(double x,size_t n)` qui retourne la valeur approchée

$$\exp_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!}$$

de $\exp(x)$.

b. Tester la fonction `exp_v1` pour $x = 1$ et $n = 10, 15, 20$ et 25.

c. Ecrire une fonction `double exp_v2(double x,double p)` qui retourne une valeur approchée $\exp_n(x)$ de $\exp(x)$ vérifiant

$$|\exp_{n+1}(x) - \exp_n(x)| = \frac{|x|^{n+1}}{(n+1)!} \leqslant p$$

Exercice 2. Reprendre l'exercice 1 pour

$$\arctan(x) = \sum_{k=0}^{+\infty} (-1)^k \frac{x^{2k+1}}{2k+1}.$$

Tester les fonctions pour $x = -0.7$ (Cette méthode de calcul de $\arctan(x)$ fonctionne bien que pour des petites valeurs de x.)

Exercice 3. Soit $a > 0$ un nombre réel. On considère la suite (u_n) donnée par

$$u_0 = 1 \quad \text{et} \quad u_{n+1} = \frac{1}{2}\left(u_n + \frac{a}{u_n}\right) \text{ pour } n \geqslant 0.$$

On suppose que la suite (u_n) converge.

a. Expliquer rapidement pourquoi (u_n) converge vers $\sqrt{a}$.

b. Coder une fonction `double sqrt(double a,size_t n)` qui retourne u_n pour n'importe quel nombre $a > 0$.

c. Tester votre fonction pour différentes valeurs de a et de n. Par exemple $a = 2$ et $n = 4, 5, 6$.

2. Quadrature

Soit $I = [a,b]$ un intervalle de $\mathbb{R}$ et f une fonction continue de I dans $\mathbb{R}$. Nous souhaitons obtenir une valeur approchée de

$$A = \int_a^b f(t)\mathrm{d}t.$$

Exercice 4. Fixons un entier $n \geqslant 1$. Nous souhaitons découper l'intervalle I en n sous-intervalles $I_1, \ldots, I_n$ de même longueur.

a. Quelle est la longueur de l'intervalle I ? Et celle des intervalles I_k ?

Pour $k = 1, \ldots, n$, on pose $I_k = [x_k, x_{k+1}]$.

b. Déterminer les valeurs de x_k pour tout $k \in \{1, \ldots, n+1\}$.

c. Justifier l'égalité

$$\int_a^b f(t)\mathrm{d}t = \sum_{k=1}^{n} \int_{x_k}^{x_{k+1}} f(t)\mathrm{d}t.$$

Afin d'obtenir une valeur approchée de A nous considérons différentes méthodes donnant des valeurs approchées des intégrales

$$A_k = \int_{x_k}^{x_{k+1}} f(t)\mathrm{d}t$$

pour tout $k \in \{1, \ldots, n\}$.

Exercice 5 (Méthode des rectangles)**.** Fixons un entier $k \in \{1, \ldots, n\}$. La méthode des rectangles consiste à approcher f sur l'intervalle I_k par la fonction constante

$$f_k(x) = \frac{f(x_k) + f(x_{k+1})}{2}.$$

a. Faire un dessin

b. Quelle est la valeur approchée de A_k donnée par cette méthode ?

c. Ecrire (en pseudo langage) un algorithme permettant de calculer la valeur approchée de A en utilisant la méthode des rectangles.

Exercice 6 (Méthode de Simpson)**.** Fixons un entier $k \in \{1, \ldots, n\}$. La méthode de Simpson consiste à approcher f sur l'intervalle I_k par une fonction parabolique f_k passant par les points $(x_k, f(x_k))$ et $(m_k, f(m_k))$ et $(x_{k+1}, f(x_{k+1}))$ où $m_k = \frac{x_k + x_{k+1}}{2}$ est le milieu de l'intervalle I_k.

a. Faire un dessin

On note $f_k(x) = a_k \times x^2 + b_k \times x + c_k$ l'approximation de f ainsi obtenue sur l'intervalle I_k.

b. Déterminer a_k, b_k et c_k en fonction de f, a_k, b_k et c_k.

c. Quelle est la valeur approchée de A_k donnée par cette méthode ?

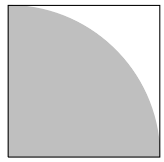
d. Ecrire (en pseudo langage) un algorithme permettant de calculer la valeur approchée de A en utilisant la méthode des trapèzes.

Exercice 7. Coder les deux méthodes d'approximation de A obtenues aux exercices précédents en `C++`.

3. Calcul approché de π.

Le but de cette section est de vous faire découvrir différentes méthodes retournant une valeur approchée de π.

Exercice 8. Supposons qu'on tire aléatoirement des fléchettes sur la cible suivante :



On suppose que le carré est de côté 1.

a. Quel est la surface du carré ? Du quart de disque ?

b. Quel est la propabilité qu'une fléchette touchant la cible soit dans la zone grise ?

c. En déduire une méthode probabiliste donnant une valeur approchée de π.

d. Ecrire une fonction `double monte_carlo(size_t n)` qui effectue n tires de fléchettes aléatoirement sur la cible et retourne la valeur approchée de π ainsi obtenue.

Exercice 9. On considère la fonction f de $[0,1]$ dans $\mathbb{R}$ donnée par $f(x) = \sqrt{1-x^2}$ et un entier n non nul quelconque. Conne à la section 2, on découpe l'intervalle $[0,1]$ en n sous-intervalles $I_k = [\alpha_k, \alpha_{k+1}]$ pour $k = 1, ..., n$. On suppose que les intervalles I_k sont de longueur $\frac{1}{n}$ et qu'on a $0 = x_1 < x_2 < ... < x_n < x_{n+1} = 1$. La courbe de la fonction est le quart de cercle de l'exercice précédent. On a donc

$$\frac{\pi}{4} = \int_0^1 f(t)\mathrm{d}t = \sum_{k=1}^{n} \int_{x_k}^{x_{k+1}} \sqrt{1-t^2}\mathrm{d}t.$$

a. Ecrire une fonction `double pi_rectangle(double n)` qui retourne une approximation de π en utilisant la méthode des rectangles pour approximer l'intégrale de f. On utilisera la fonction `sqrt` définie à l'exercice 3 (en prenant 20 pour le paramètre n.)

b. Ecrire une fonction `double pi_simpson(double n)` qui retourne une approximation de π en utilisant la méthode de Simpson.

c. Comparer ces deux méthodes.

Exercice 10. En 1706 le mathématicien John Machin donne la fomule suivante :

$$\frac{\pi}{4} = 4\arctan\frac{1}{5} - \arctan\frac{1}{239}$$

A l'aide de la fonction `double arctan_v1(double a,size_t n)` coder à l'exercice 2 écrire une fonction `double machin(size_t n)` retournant une valeur approchée de π à l'aide de la formule de Machin. (Le paramètre `n` sera celui utilisé dans l'appel de `arctan_v1`.)