TP1 : Le Design Pattern Fonctor

Les *Fonctor* et le calcul d'intégrales

Implémentation du calcul de l'intégrale

Nous allons implémenter le calcul de l'intégrale pour une fonction f choisie. Pour cela, nous allons avoir plusieurs manières concernant l'implémentation du calcul de la fonction f : en tant que pointeur de fonction, d'objet contenant une méthode et enfin à l'aide d'un *Fonctor*.
Vous allez donc être amenés à coder 3 fonctions différentes :

compute_ptr Fonction prenant en argument un pointeur de fonction et retournant un double

compute_class Fonction prenant en argument un objet et retournant un double

compute_fonc Fonction prenant en argument un *Fonctor* et retournant un double

Vos fonctions auront donc comme prototype :
double `compute_X( < fonction` f `>, double X, double` δ_x`)`.
L'intervalle sur laquelle nous allons calculer l'intégrale sera $[X - \delta_x, X + \delta_x]$.
La fonction f, quant à elle, est définie comme prenant en paramètre un double et renvoyant un double.
Que remarquez-vous concernant les 3 manières de passer f en paramètre ?

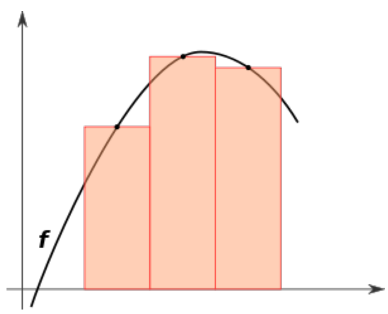
Temps d'exécution

Ensuite, vous allez calculer le temps d'exécution pour chacune des trois méthodes (pointeur, objet et *Fonctor*). Pour cela, vous utiliserez la fonction `gettimeofday` définie dans `#include <sys/time.h>`.
Pour son utilisation, se reporter au manuel (`man gettimeofday`).
Que remarquez-vous concernant les temps d'exécution ?

Rappel : Formule de l'intégration numérique

Vous trouverez à l'adresse suivante un rappel concernant le calcul d'intégrales par la méthode des rectangles : *http ://homeomath.imingo.net/methrect.htm.*
Vous pouvez très bien utiliser une autre méthode (trapèze, Simpson, ...).

Soit f la fonction à intégrer et a b les bornes. On a alors $I(f) = (b - a)f(\xi)$ avec f la fonction pour laquelle nous voulons calculer l'intégrale.



Source : Wikipedia