

Modèles et techniques en programmation parallèle hybride et multi-cœurs

Foncteurs et fonctions lambda

Marc Tajchman

CEA - DEN/DM2S/STMF/LDEI

19/01/2026

Les foncteurs ou objets-fonctions sont des classes avec une fonction membre `operator()`.

Les variables (ou objets) de type foncteur peuvent s'utiliser comme des fonctions, mais avec des possibilités supplémentaires.

Les fonctions lambda sont des constructions syntaxiques qui simplifient la définition de foncteurs.

On illustrera la définition et l'utilisation des foncteurs et fonctions lambda par des exemples d'utilisation de plus en plus générale.

0. Utilisation d'une fonction dans une autre fonction.

Le fichier 0_Base/src/integrale.hxx contient une fonction C++ qui calcule une valeur approchée de

$$\int_0^1 f(x)dx$$

où f est une fonction C++ “utilisateur” contenue dans le programme principal main.hxx.

Inconvénients

- ▶ La fonction à intégrer doit s'appeler f (on ne peut pas l'utiliser pour intégrer la fonction système \sin par exemple)
- ▶ Impossible d'intégrer 2 fonctions différentes dans le même code

1. Utilisation de pointeur de fonction

On ajoute un paramètre supplémentaire à la fonction intégrale : un pointeur de fonction.

Exemple de déclaration et de définition d'un pointeur de fonction

```
double (*g) (double);  
g = &cos;  
g(0.5);
```

calcule $\cos(0.5)$ par l'intermédiaire du pointeur de fonction g.

Voir la définition modifiée de la fonction intégrale dans le fichier 1_PointeurFonction/src/integrale.hxx).

Les pointeurs de fonction existent depuis (très) longtemps et C et en C++.

Avantages :

- ▶ On peut intégrer toute fonction utilisateur qui prend un nombre (de type double) en paramètre et donne comme résultat un nombre de type double
- ▶ On peut intégrer les fonctions système (sin, cos, ...) qui ont les même nombre/type de paramètres/résultats.

Inconvénient:

- ▶ On ne peut pas (facilement) calculer des intégrales partielles, ou l'intégrale de fonctions qui dépendent de paramètres

Par exemple:

$$g(a) = \int_0^1 f(x; a, b) dx$$

on dit parfois que f est une fonction d'une variable (x) et qui a un état interne (a et b)

Pour définir une fonction d'une variable, et qui dépend d'un (ou de plusieurs) paramètres, C++ donne la possibilité de définir un type d'objet (une classe) appelé objet fonction ou foncteur (functor).

Par exemple

```
class Foncteur
{
public:
    Foncteur(double a, double b): p(a), q(b) {}
    double operator()(double x) const {
        return p*x+q;
    }
private:
    double p, q;
};
```

Une classe qui permet de définir une fonction $x \mapsto px + q$ où p et q sont des constantes et x est une variable.

Si on veut intégrer ce type de fonction, il faut définir une fonction intégrale comme suit:

```
double integrale(double a, double b, long n,
                  Foncteur & f)
{
    long i;
    double s = 0;
    double dx = (b-a)/(n-1);

    for (i=0; i<n; i++)
        s += f(a + i*dx);
    s *= dx;
    return s;
}
```

Pour intégrer $x \mapsto 2x + 1$ sur $[0, 5]$

```
Foncteur f(2, 1);
```

```
double result = integrale(0, 5, 100000, f);
```