

Master GI
Parcours IN
Année : M2

Programmation en langage C++ : Programme principal & sous-programmes

Exercice 1 : pair_impair

Dans le but d'introduire la notion de sous-programme, créer un algorithme en langage de programmation C++ constitué d'un programme principale (main), lequel fait appel à un sous-programme (appelé aussi routine ou fonction) réalisant une tâche répétitive bien spécifique.

Exemple à traiter : création d'une routine **pair_impair** vérifiant la nature d'un entier (pair ou impair).

1. Appeler la routine **pair_impair** (nb: la routine est sans argument avec un return $\rightsquigarrow$ 0).
2. La routine **pair_impair** renvoie la nature de l'entier saisi à partir de l'invité de commandes : l'entier saisi est-il pair ou impair ?

Exercice 2 : pair_impair & while

Reprendre le même exercice en introduisant cette fois-ci la fonction **while**.

1. Dans un premier temps, la routine **doit dialoguer avec l'utilisateur** pour savoir s'il faut saisir un nouvel entier ou bien quitter le programme.
2. Dans un second temps, le programme doit afficher **le nombre total d'entiers testés** en quittant la routine **pair_impair** (nb: return $\neq$ 0 ; return $\rightsquigarrow N_{entiers}$)
3. Dans un troisième temps, le programme doit afficher **le nombre d'entiers pairs, impairs et total testés** après avoir quitter la routine **pair_impair** (nb: les sorties doivent être placées en tant qu'arguments de la routine **pair_impair**).

Exercice 3 : Random

Il s'agit dans le cadre de cet exercice de faire appel à la fonction **random**. Cette fonction permet d'effectuer un tirage aléatoire dans un intervalle donné. Suivre les étapes suivantes pour créer le programme appelant la fonction **random** :

1. Dans un premier temps, se renseigner sur le web concernant la syntaxe de la fonction **random** en langage de programmation C++.
2. Dans un second temps, appeler la fonction **random** depuis le programme principale (main) en affichant via le terminal la valeur du tirage aléatoire effectué.
3. Dans un troisième temps, calculer le **pourcentage de chaque occurrence** par rapport au nombre de tirages aléatoires réalisés.

Théoriquement, pour un grand nombre de tirages toutes les occurrences tendent vers le même pourcentage ($\% = 100 \times \frac{1}{Nb_{occure}}$). L'influence du nombre de tirages sur le pourcentage des occurrences est bien décrit par la figure 1.

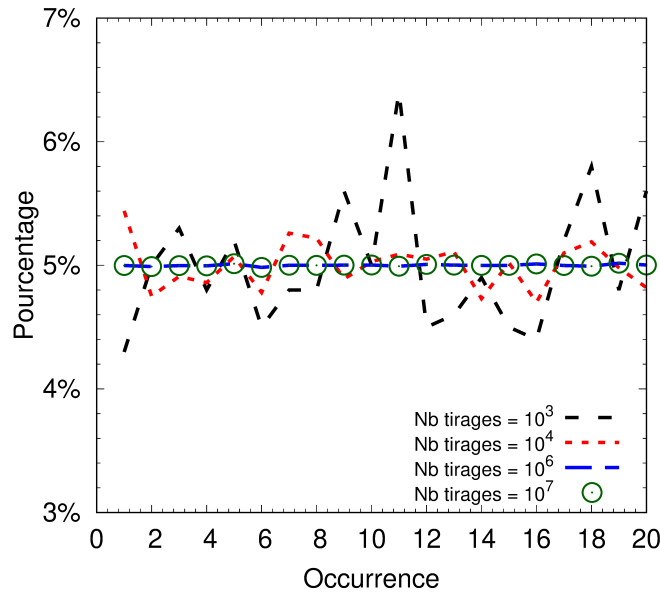


Figure 1: Influence du nombre de tirages

Exercice 4 : PFD (chute libre)

La résolution des équations du Principe Fondamental de la Dynamique (PFD) permet d'étudier le mouvement d'un solide soumis à l'action de forces et moments. Dans le cadre de cet exercice, on va s'intéresser à un cas simple, **celui d'une sphère de diamètre D , animée par une vitesse initiale, en chute libre**. Il s'agit donc de créer un programme en C++ qui permet de calculer l'accélération, la vitesse et la position de la sphère au cours de sa chute libre. La solution discrétisée (discretisation en temps) obtenue à l'aide de l'algorithme peut être confrontée à la solution analytique. Trois cas de figure seront étudiés :

1. Mouvement sans contact et **sans prise en compte de la résistance de l'air** (Fig. 2) : $h_0 = 1 \text{ m}$; $v_0 = [0 - 1, 5] \text{ m/s}$; $g = 9,81 \text{ m/s}^2$; $\Delta t = 10^{-3} \text{ s}$; $t_{total} = 1 \text{ s}$; $v(t) = -gt + v_0$; $h(t) = -g\frac{t^2}{2} + v_0t + h_0$.

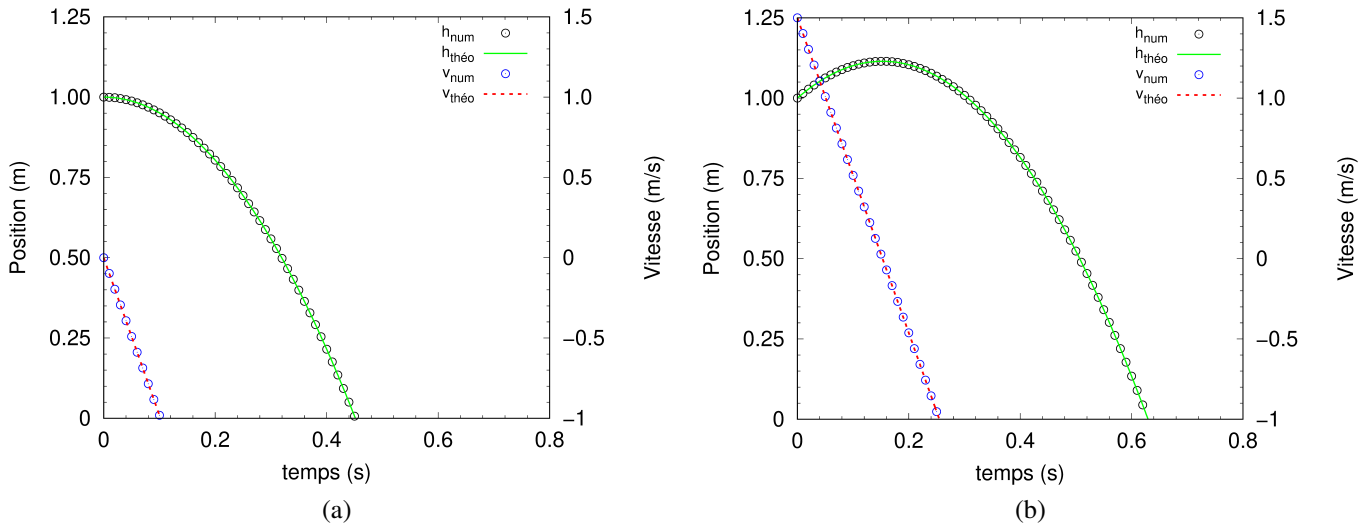


Figure 2: Mouvement de la sphère en chute libre : (a) $v_0 = 0$; (b) $v_0 = 1,5 \text{ m/s}$

2. Mouvement sans contact et **avec prise en compte de la résistance de l'air** (Fig. 3) : $h_0 = 1 \text{ m}$; $v_0 = 1,5 \text{ m/s}$; $g = 9,81 \text{ m/s}^2$; $\Delta t = 5 \cdot 10^{-4} \text{ s}$; $t_{total} = 1 \text{ s}$; force de frottement sphère/air $\vec{f} = -\alpha \times \text{sign}(\vec{v})$, avec $\alpha > 0$ (en Newton) ; $v(t) = (-g + \frac{f}{m})t + v_0$; $h(t) = (-g + \frac{f}{m})\frac{t^2}{2} + v_0t + h_0$.
3. Mouvement de la sphère avec **détection de contact (contact sphère/plan horizontal à $h = 0 \text{ m}$)** : $D_b = 10^{-1} \text{ m}$; $\rho = 2700 \text{ kg/m}^3$; $m = \rho V$; $F_n = -k_n \delta_n - c_n v_n$; F_n : interaction de contact ; $k_n =$

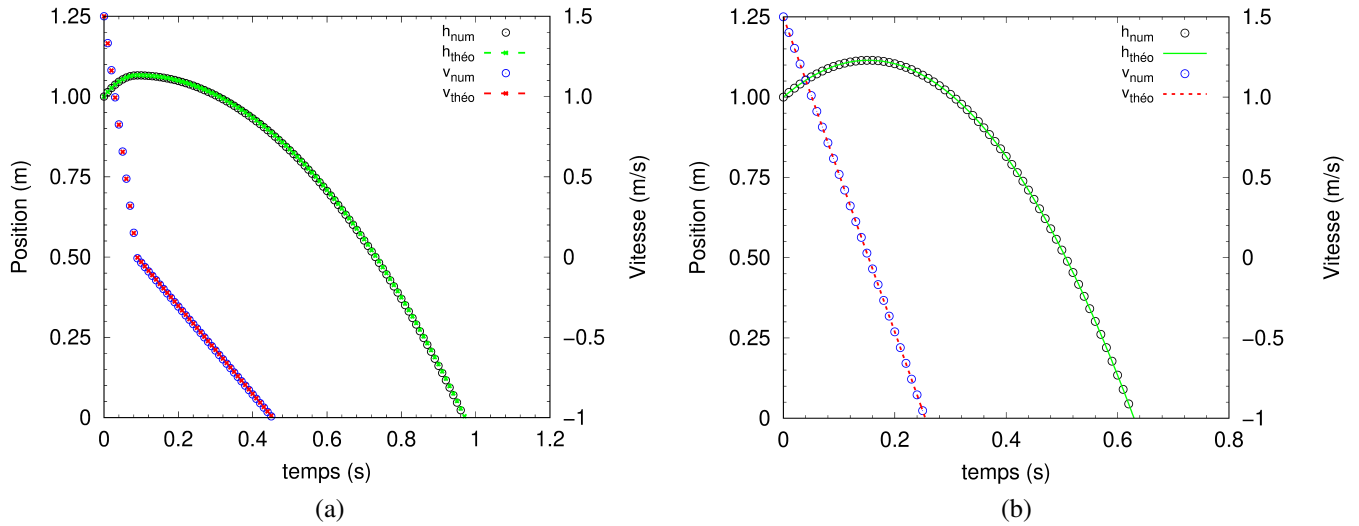


Figure 3: Mouvement de la sphère en chute libre avec/sans résistance de l'air : (a) $v_0 = 1.5 \text{ m/s}$ & $\alpha = 10$; (b) $v_0 = 1.5 \text{ m/s}$ & $\alpha = 0$

10^5 N/m ; k_n : raideur de contact ; $c_n = \sqrt{\frac{10}{3}}\psi\sqrt{k_n m}$; c_n : coefficient d'amortissement ;
 $\psi = \ln\alpha_e / \sqrt{\ln^2\alpha_e + \pi^2}$; $\alpha_e = 0,8$; α_e : coefficient de restitution.

Les figures 4(a-b) montrent les positions et vitesses de la sphère en fonction du temps dans la cas d'un contact sphère/plan avec et sans amortissement (amortissement de type visqueux). À noter que les coefficients de restitution α_e ont pour valeur 1 et 0,8, respectivement dans le cas non-amorti et amorti.

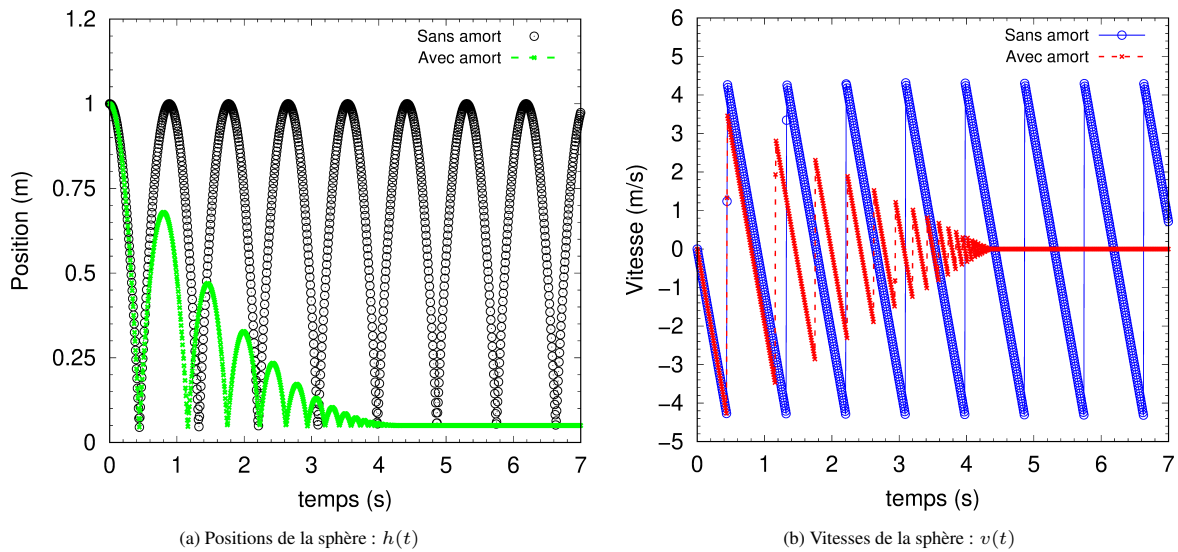


Figure 4: Grandeurs cinématiques de la sphère au cours du temps : (a) positions ; (b) vitesses