Université Pierre et Marie Curie 2014-2015
Master 2 IFMA

Examen

Durée : 3h ; les notes de cours sont autorisées.

Le but de ce sujet est de programmer une classe abstraite générique permettant de représenter informatiquement un système dynamique quelconque.

On définira les différentes classes dans un fichier `classes.cpp` avec un fichier d'en-tête `classes.hpp`. Les programmes demandés dans la deuxièmes parties sont à écrire dans des fichiers séparés. Les réponses aux questions ne demandant pas de programmation sont à rédiger dans un fichier texte. On demandera également les commandes à effectuer pour la compilation, idéalement regroupées dans un fichier `makefile` ou sinon écrites dans le fichier texte contenant les réponses aux questions.

Introduction

On désigne par *système dynamique* une suite $(X_n)_{n\in\mathbb{N}}$, déterministe ou aléatoire, d'éléments d'un certain ensemble E, définie par une relation de récurrence. Plus précisément, dans le cas déterministe, on aura

$$\forall n \in \mathbb{N},\ X_{n+1} = f(X_n),$$

où f est une fonction de E dans E. Dans le cas aléatoire, on aura

$$\forall n \in \mathbb{N},\ X_{n+1} = f(X_n, \theta_n),$$

où $(\theta_n)_{n\in\mathbb{N}}$ est une suite de variables indépendantes et de même loi (et indépendantes de X_0) à valeurs dans un ensemble F, et f est une fonction de $E \times F$ dans F.

Définition de la classe `SystDyn`

1. Écrire un modèle (*template*) de classe `SystDyn` qui va modéliser les systèmes dynamique dont les réalisation sont d'un type donné. La déclaration de la classe `SystDyn` va donc commencer par

   `template <typename T> class SystDyn`,

   où la variable `T` va désigner le type des éléments de l'ensemble E. La classe `SystDyn<T>` contiendra comme membres déclarés `protected` :

   - Deux variables `initial` et `courant` de même type `T`, correspondant respectivement à l'état du système avant la première itération (c'est-à-dire X_0) et à l'état du système à la $n^{\text{ème}}$ itération (c'est-à-dire X_n) ;
   - Une variable `n` de type `int` correspondant au nombre d'itérations de la relation de récurrence depuis l'état initial.

   La classe `SystDyn` contiendra comme membres déclaré `public` :

   - Un constructeur prenant en argument une variable `x` de type `T` et initialisant les variables `initial` et `courant` par la valeur de `x` et initialisant `n` à `0` ;
   - Une surcharge de l'opérateur `++` correspondant à l'itération du système dynamique, qui sera déclarée comme fonction virtuelle pure ;
   - Une méthode `affiche` qui affiche dans la sortie standard (`std::out`) la variable `courant` ;
   - Une méthode `reinitialise` qui ramène le système à son état initial (et donne à `n` la valeur `0`) ;
   - Trois accesseurs `nb_iterations`, `etat_courant` et `etat_initial` correspondant aux trois variables `n`, `courant` et `initial`.

2. Que va-t-il se passer si l'on tente de déclarer une variable de type `SystDyn<T>`, pour un type `T` donné. Pourquoi ?

3. Pourquoi les membres `n`, `initial` et `courant` sont-ils déclarés `protected` ?

4. Est-il nécessaire de redéfinir un constructeur de copie, un opérateur `=`, un destructeur ? Pourquoi ? Le faire si la réponse est oui.

Quelques exemples

Dans cette partie on présente quelques exemples d'utilisation du modèle de classe `SystDyn` défini précédemment.

Rotations

On considère le sytème dynamique à valeurs dans $[0, 2\pi[$ défini par

$$X_{n+1} = X_n + \alpha \mod 2\pi,$$

correspondant à des rotations d'angle $\alpha \in \mathbb{R}$.

1. Écrire une classe `rotation` qui hérite publiquement de la classe `SystDyn<double>` pour représenter cette suite. La classe `rotation` devra contenir :
   - un membre privé `alpha` correspondant à l'angle ;
   - un constructeur public prenant en argument une variable de type `double` qui donne la valeur de `alpha` et qui initialise la position du système à `0` ;
   - Un définition de l'opérateur `++`.
2. Écrire un programme qui calcule un histogramme à 20 barres des 1000 premières itérations du système pour $\alpha = 1$. On affichera les fréquence obtenues, mais on ne cherchera pas à afficher graphiquement l'histogramme.

Marches aléatoires

On appelle *marche aléatoire* une suite de variables aléatoires $(X_n)_{n\in\mathbb{N}}$ définie par

$$X_n = \sum_{k=1}^{n} Z_k,$$

où les $(Z_n)_{n\in\mathbb{N}}$ sont des variables aléatoires indépendantes et de même loi à valeurs dans $\mathbb{R}^d$.

1. Écrire une classe `MarcheAleatoire` qui hérite de la classe `SystDyn<int>` correspondant à la marche aléatoire en dimension 1 pour laquelle les Z_n vérifient

$$\mathbb{P}(Z_n = 1) = 1 - \mathbb{P}(Z_n = -1) = p \in]0, 1[.$$

2. Écrire un programme qui affiche les valeurs des 1000 premières itérations de la marche.[1]

3. Écrire une classe `point2D` contenant deux membres publics `x` et `y` de type `double`, pour représenter les coordonnées d'un point de $\mathbb{R}^2$. Écrire un constructeur par défaut pour cette classe prenant en argument deux variables de type `double` pour initialiser `x` et `y`. Surcharger l'opérateur `<<` pour cette classe pour afficher les valeurs des deux variables `x` et `y`.

4. Écrire une classe `MarcheAleatoire_2D` qui hérite de la classe `SystDyn<point2D>` correspondant à la marche aléatoire en dimension 2 pour laquelle les Z_n sont de loi uniforme sur le cercle de rayon r. Cette classe devra contenir un membre privé `r` correspondant au rayon du cercle.

5. Écrire un programme qui affiche les valeurs des 10000 premières itérations de la marche pour r=0.01.

[1] On rappelle que le programme gnuplot permet d'afficher le contenu d'un fichier. Pour cela, entrer `gnuplot` en ligne de commande, puis entrer `plot "nom_du_fichier" with lines`. Si le fichier est consitué de deux colonnes, le résultat est la deuxième colonne en fonction de la première. Si il n'y a qu'un colonne, le résultat est la colonne en fonction de $1, 2, ..., N$.

Urnes d'Ehrenfest

Le modèle d'Ehrenfest est le modèle stochastique suivant : on considère N particules pouvant chacunes se placer dans deux urnes, U_1 et U_2. On note X_n le nombre de particules présentes dans l'urne U_1 au temps n. La dynamique de la suite $(X_n)_{n\in\mathbb{N}}$ est la suivante : à chaque itération, on choisit une particule uniformément parmi les N et cette particule est déplacée d'une urne à l'autre. Autrement dit, X_n étant donné, on a

$$X_{n+1} = \begin{cases} X_n + 1 & \text{avec probabilité } \frac{N-X_n}{N}, \\ X_n - 1 & \text{avec probabilité } \frac{X_n}{N}. \end{cases}$$

1. Écrire une classe `Ehrenfest` qui hérite de la classe `SystDyn<int>` pour simuler cette suite. Cette classe devra contenir un membre privé `nb_particules` correspondant au nombre total de particules. Écrire un constructeur prenant en argument deux entiers qui correspondront aux valeurs de `nb_particules` et de X_0.

2. Écrire un programme qui affiche les valeurs des 3000 premières itérations de la marche pour laquelle $X_0 = 0$ et $N = 1000$.

Schéma de discrétisation pour une équation différentielle stochastique

Pour résoudre numériquement l'équation différentielle stochastique

$$\mathrm{d}X_t = b(X_t)\mathrm{d}t + \sigma(X_t)\mathrm{d}W_t,$$

on peut utiliser le schéma numérique suivant, dit *schéma d'Euler-Maruyama* :

$$\bar{X}_{n+1} = b(\bar{X}_n)\delta t + \sigma(\bar{X}_n)\sqrt{\delta t}G_n,$$

où $\delta t > 0$ est le pas de temps et $(G_n)_{n\in\mathbb{N}}$ est une suite de variables aléatoires indépendantes de loi $\mathcal{N}(0,1)$.

1. Écrire une fonction `gaussienne` prenant en argument deux variables de type `double` correspondant à la moyenne et à la variance et renvoyant une réalisation d'une variable aléatoire Gaussienne avec ces paramètres.

2. Écrire une classe `EDS` qui hérite de la classe `SystDyn<double>` et implémentant le schéma d'Euler explicite. Cette classe devra contenir comme membre privé une variable `dt` de type `double` correspondant au pas du schéma d'Euler, ainsi que deux variables `b` et `sigma2` de type `double (*)(double)` (c'est-à-dire des pointeurs-sur-fonction-qui-à-un-double-associe-un-double).

3. Écrire un programme affichant une approximation avec un pas de temps 0.01 de la solution de l'équation

$$\begin{cases} X_0 &= 2, \\ \mathrm{d}X_t &= -2X_t\mathrm{d}t + \mathrm{d}W_t, \end{cases}$$

sur l'intervalle $[0, 10]$.