Simulation de variables aléatoires.

1 Variables discrètes

Exercice 1

Donner des programmes permettant de simuler les lois discrètes usuelles : uniforme, Bernoulli, binomiale, géométrique.

Pour la loi de Poisson (et d'autres lois), on peut utiliser la méthode d'inversion discrète :

Soit X une variable aléatoire discrète, qui prend les valeurs $x_1 < x_2 < \cdots$. On note F sa fonction de répartition.

Pour simuler la variable X, on peut

- tirer aléatoirement un réel de $[0,1]$
- déterminer k tel que $t \in]F(x_{k-1}), F(X_k)]$ (si $t \leqslant F(x_1)$, on prendra $k = 1$).
- renvoyer x_k.

Exercice 2

Implémenter cette méthode pour simuler la loi de Poisson.

Exercice 3

Montrer que cette méthode permet bien de simuler la variable X.

2 Variables à densité

On rappelle que la fonction `random` du module `random` permet de simuler une loi uniforme sur $]0,1[$.

Exercice 4

Écrire une fonction `unif`, prenant en arguments deux réels `a` et `b`, et simulant une loi uniforme sur $]a,b[$.

En déduire des fonctions permettant de calculer expérimentalement les espérances et variances de ces lois.

Exercice 5

Comparer les résultats précédents avec la fonction `uniform` du module `numpy.random`.

Pour d'autres lois, on pourra utiliser la méthode de la fonction de répartition inverse :

Proposition
On suppose que la fonction de répartition F d'une variable X à densité est strictement croissante.

Alors, si $U \hookrightarrow \mathcal{U}(]0,1[)$, $F^{-1}(U)$ suit la même loi que X.

Exercice 6

Démontrer cette proposition.

En pratique, on utilisera souvent un résultat un peu plus partiel :

Proposition 2.1
Soit F la fonction de répartition d'une variable aléatoire X à densité. Soient $a < b \in \mathbb{R} \cup \{-\infty, \infty\}$. On suppose

- *F nulle sur $]-\infty, a[$*
- *F strictement croissante sur $]a, b[$*
- *F égale à 1 sur $]b, +\infty[$.*

Alors, si $U \hookrightarrow \mathcal{U}(]0,1[$ et G est la restriction de F à $]a, b[$, $G^{-1}(U)$ suit la même loi que X.

Exercice 7

À l'aide de cette méthode, simuler une loi exponentielle de paramètre λ en Python.

Comparer les résultats avec la fonction `exponential` du module `numpy.random`.

Pour simuler une loi normale, on pourra utiliser la commande `normal` du module `numpy.random`.

3 Exercice

Soit $\lambda > 0$. Pour tout entier $n \geqslant \lambda$, on définit une suite $(X_{n,i})_{i \in \mathbb{N}^*}$ de variables aléatoires indépendantes suivant toutes une loi de Bernoulli de paramètre $p_n = \frac{\lambda}{n}$.

On définit alors pour tout n :

$$N_n = \frac{1}{n} \min\{i \in \mathbb{N}^* \mid X_{n,i} = 1\}.$$

Exercice 8

1. Écrire une fonction Python permettant de simuler la variable N_n.
2. Écrire une fonction Python permettant de tracer une estimation de la fonction de répartition F_n de N_n.
3. Proposer un moyen de vérifier que pour tout $t > 0$, on a $F_n(t) \to F(t)$, où F est la fonction de répartition d'une loi exponentielle de paramètre λ.