Simulation et Optimisation – TP1

# Introduction

Dans ce laboratoire, il nous est demandé d’implémenter plusieurs algorithmes afin de générer des réalisations de variables aléatoires en utilisant différents jeux de données.

Les trois algorithmes (ou méthodes) qui nous intéressent sont les suivants :

1. Acceptation-rejet (version « bête et méchante »),
2. Méthode des mélanges couplée à une approche géométrique,
3. Méthode des mélanges couplée à la méthode des fonctions inverses.

Nous allons observer dans un premier temps les résultats de ces méthodes en fonction de différents jeux de données. L’analyse des performances des différentes méthodes s’effectuera dans un deuxième temps, en comparant leurs temps d’exécution suivant les situations afin de se faire une idée de leur efficacité et de leurs cas d’utilisation.

# Résultats des tests

Voici ci-après les résultats des algorithmes avec 5000 simulations de 1000000 réalisations générées.

## Premier jeu de données – Uniforme (5,15)

*Espérance*: **10**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Méthode | Unité mesurée | Moyenne | Écart-type | IC | Largeur IC |
| 1 | Réalisations | 9. 99876 | 2.88805 | [9.9931,10.0044] | 0.0113212 |
| Temps [s] | 0.0564822 | 0.002313 | [0.0564181,0.0565463] | 0.000128226 |
| 2 | Réalisations | 10.0035 | 2.88826 | [9.9978,10.0091] | 0.011322 |
| Temps [s] | 0.0806404 | 0.00373474 | [0.0805369,0.0807439] | 0.000207043 |
| 3 | Réalisations | 9.99826 | 2.88586 | [9.9926,10.0039] | 0.0113126 |
| Temps [s] | 0.0538264 | 0.0022246 | [0.0537647,0.0538881] | 0.000123326 |

En regardant ce tableau, on remarque tout d’abord que les résultats des trois méthodes sont très proches les uns des autres (écarts-types et largeurs d’IC par exemple). On constate également que les moyennes obtenues sont relativement proches de l’espérance (calculée au préalable, avant les simulations). En jetant un œil aux 3 intervalles de confiance (à 95%), nous pouvons constater que l’espérance théorique s’y trouve (bien entendu, il aurait pu arriver qu’elle soit en dehors de l’IC, mais elle se retrouve statistiquement 95% du temps dedans).

Tout ceci peut nous laisser penser aux premiers abords que l’implémentation des algorithmes est correcte.

Nous reparlerons des temps en détail dans la section dédiée à l’analyse des performances.

## Deuxième jeu de données – Mélange de deux variables triangulaires

*Espérance*: **8.5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Méthode | Unité mesurée | Moyenne | Écart-type | IC | Largeur IC |
| 1 | Réalisations | 8.49921 | 4.62769 | [8.49014,8.50828] | 0.0181405 |
| Temps [s] | 0.167804 | 0.00859517 | [0.167566,0.168042] | 0.000476492 |
| 2 | Réalisations | 8.4945 | 4.62789 | [8.48543,8.50357] | 0.0181413 |
| Temps [s] | 0.0875782 | 0.00301226 | [0.0874947,0.0876617] | 0.000166991 |
| 3 | Réalisations | 8.49832 | 4.62784 | [8.48925,8.50739] | 0.0181411 |
| Temps [s] | 0.0682266 | 0.00263762 | [0.0681535,0.0682997] | 0.000146222 |

A nouveau, on peut remarquer que les résultats des trois méthodes sont similaires, que la moyenne est proche de l’espérance et que cette dernière se trouve dans les trois intervalles de confiance.

## Troisième jeu de données – Profil plutôt plat

*Espérance*: **10.758**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Méthode | Unité mesurée | Moyenne | Écart-type | IC | Largeur IC |
| 1 | Réalisations | 10.7559 | 5.22366 | [10.7456,10.7661] | 0.0204767 |
| Temps [s] | 0.0793284 | 0.00326109 | [0.079238,0.0794188] | 0.000180785 |
| 2 | Réalisations | 10.7412 | 5.20638 | [10.731,10.7514] | 0.020409 |
| Temps [s] | 0.0922574 | 0.0027182 | [0.0921821,0.0923327] | 0.000150689 |
| 3 | Réalisations | 10.7552 | 5.22255 | [10.745,10.7655] | 0.0204724 |
| Temps [s] | 0.0745272 | 0.00268177 | [0.0744529,0.0746015] | 0.00014867 |

On peut faire les mêmes commentaires que précédemment ici aussi, la seule exception étant l’espérance n’étant pas dans l’IC (relatif aux réalisations de la variable aléatoire de la méthode 2). On remarque alors un exemple où l’espérance n’est pas dans l’intervalle de confiance.

## Quatrième jeu de données – Profil accidenté

*Espérance*: **10.2616**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Méthode | Unité mesurée | Moyenne | Écart-type | IC | Largeur IC |
| 1 | Réalisations | 10.2598 | 5.70906 | [10.2486,10.2709] | 0.0223795 |
| Temps [s] | 0.257825 | 0.00595176 | [0.25766,0.25799] | 0.000329949 |
| 2 | Réalisations | 10.2584 | 5.70699 | [10.2472,10.2696] | 0.0223714 |
| Temps [s] |  |  |  |  |
| 3 | Réalisations |  |  |  |  |
| Temps [s] |  |  |  |  |

A nouveau, les mêmes remarques peuvent être faites au sujet de ce tableau.

# Analyse des performances

Observons à présent les différents temps d’exécution obtenus lors des simulations. Afin de comprendre le pourquoi derrière ces temps, il nous faut tout d’abord analyser la « forme » des différents jeux de données. Nous comparerons ensuite les différents algorithmes en utilisant leurs intervalles de confiance.

## Uniforme (5,15)

Ce cas est favorable pour l’acceptation-rejet comme il faut commencer par enfermer la fonction dans le rectangle [a, b] x [0, y\_max]. Comme chaque ordonnée des points de l’uniforme est égal à yMax, le point généré sera forcément accepté (il ne peut être généré au-dessus de la fonction). Il doit donc s’agir du cas dans lequel cette méthode est la plus efficace.

Les deux autres méthodes se basent sur la méthode des mélanges. Ici, il n’y a qu’un seul « morceau » de fonction. Le facteur déterminant sera donc plutôt l’approche géométrique et la méthode des fonctions inverses en elles-mêmes. Il faut générer un point puis potentiellement effectuer une symétrie dans l’approche géométrique, alors que la méthode des fonctions inverses renvoie un résultat très simple lorsque les deux ordonnées des points définissant ce morceau de fonction sont égales. La méthode des fonctions inverses est donc également appropriée ici. Les intervalles de confiance des temps sont les suivants :