

TD 4

Algorithmes randomisés

On conseille ici d'aller lire l'excellente page Wikipédia sur le sujet :

https://fr.wikipedia.org/wiki/Algorithme_probabiliste

4.1 Marches Aléatoires

Énoncé, code, corrigé par S. Gonnord remis en forme par L. Gonnord, mai 2020.

Objectifs pédagogiques

- Réaliser et visualiser des marches aléatoires dans \mathbb{Z} , puis le plan, l'espace et même le cercle trigonométrique.
- Faire des statistiques sur de telles marches pour vérifier/tester des résultats plus ou moins standards.

On fournit du code "à trous". **Remarque : pour dessiner toutes les courbes, il y a un `pyp1.clf()` à commenter dans les fonction de dessin du code fourni.**

4.1.1 Marches dans \mathbb{Z}

On commence par des marches dans \mathbb{Z} avec une fonction réalisant un pas. Elle prend en entrée $p \in [0, 1]$, puis renvoie 1 avec probabilité p et -1 avec probabilité $1 - p$. On utilisera la fonction `random` qui, appelée sans argument, renvoie un flottant entre 0 et 1, avec une loi de répartition uniforme, de sorte que `random() <= p` est vrai avec probabilité p .

EXERCICE #1 ► Un pas, une marche

Écrire une telle fonction réalisant un pas, puis une marche.

```
>>> pas(0.5)
1
>>> pas(0.5)
-1
>>> pas(0.5)
-1
>>> pas(0.8)
1
>>> pas(0.2)
-1

>>> marche(10, 0.5)
[0, -1, -2, -1, 0, -1, 0, -1, -2, -3, -2]
>>> marche(10, 0.5)
[0, -1, 0, 1, 2, 1, 0, 1, 2, 3, 2]
>>> marche(10, 0.5)
[0, 1, 0, 1, 0, -1, -2, -3, -2, -1, -2]
>>> marche(10, 0.3)
[0, -1, -2, -3, -2, -1, 0, -1, -2, -3, -4]
```

EXERCICE #2 ► Dessins

Utiliser la fonction fournie pour dessiner des marches aléatoires sur un axe. On doit obtenir des dessins comme sur les Figures 4.1 et 4.2.

On va maintenant tenter d'observer expérimentalement quelques résultats portant sur une marche $S_n = X_1 + \dots + X_n$ (les X_i sont des pas indépendants) tels que (dans le cas non biaisé $p = 1/2$) :

- L'espérance de S_n est nulle.
- Sa variance est $Var(S_n) = E(S_n^2) = n$.
- L'espérance de sa valeur absolue est majorée par la racine de l'espérance de son carré : $E(|S_n|) \leq \sqrt{n}$.
- Avec probabilité 1, on revient à l'origine.

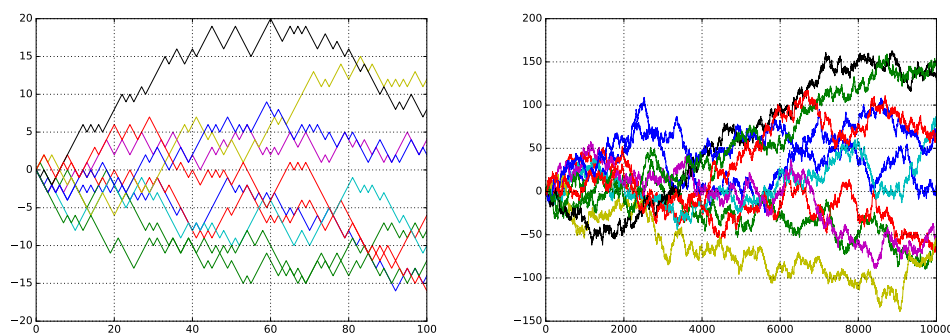
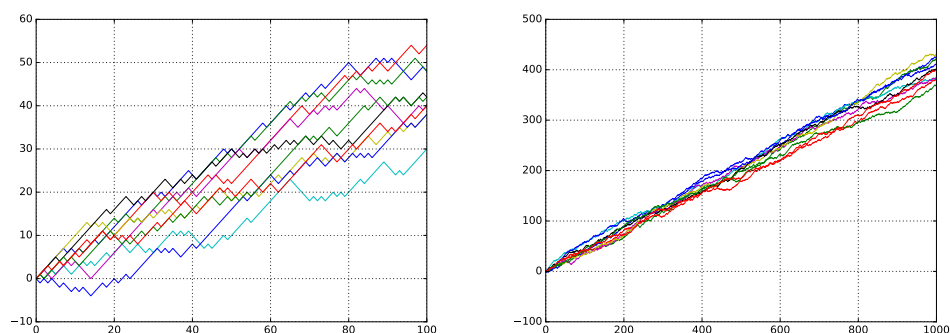


FIGURE 4.1 – Des marches aléatoires non biaisées...

FIGURE 4.2 – Et des marches biaisées (devinez la valeur de p).**EXERCICE #3 ► Moyenne**

Écrire une fonction permettant de calculer la moyenne des positions issues de marches successives : on réalise des marches, et on garde pour chacune la dernière valeur (la position à la fin de la marche), et on fait la somme de toutes ces dernières valeurs avant de diviser par le nombre de marches. On trouvera en testant des résultats du style :

```
>>> valeur_moyenne(100, 0.5, 10)
0.0
>>> valeur_moyenne(100, 0.5, 10)
1.0
>>> valeur_moyenne(100, 0.5, 10)
-5.4

>>> valeur_moyenne(100, 0.5, 10**4)
0.1628
>>> valeur_moyenne(100, 0.7, 10**3)
39.87
>>> valeur_moyenne(100, 0.2, 10**3)
-59.858
```

EXERCICE #4 ► Valeurs moyennes et espérance

En s'inspirant de ce qui précède, évaluer les valeurs moyennes de $|S_n|$ et de S_n^2 pour $n = 10$ puis $n = 100$ (à vous de déterminer le nombre de marches qui vous semble pertinent).

On s'intéresse maintenant à la question du passage par un point imposé : on fixe un objectif (un entier relatif, ici), et on réalise un certain nombre de marches pour savoir si on passe par cet objectif. La théorie dit qu'avec probabilité 1 on passe par cet objectif dans \mathbb{Z} et \mathbb{Z}^2 ... mais pas \mathbb{Z}^3 ! Bien entendu si on s'impose un nombre de pas maximum, cette probabilité est < 1 (elle augmente avec le nombre de pas maximum, et diminue avec la distance entre l'objectif et l'origine).

EXERCICE #5 ► Revenir au même point

Écrire une fonction réalisant une marche d'au plus N pas (N est un paramètre), retournant True si la marche est passée par l'objectif (paramètre de la fonction), et False sinon. On pourra prendre un troisième paramètre

$p \in [0,1]$ permettant de biaiser la marche (sans quoi, on fait des marches symétriques, ce qui est déjà intéressant). Par exemple on peut trouver :

```
>>> repasse_par(0, 10, 0.5)
True
>>> repasse_par(0, 10, 0.5)
True
>>> repasse_par(0, 10, 0.5)
False
>>> repasse_par(0, 10, 0.5)
True

>>> repasse_par(0, 10, 0.7)
False
>>> repasse_par(0, 10, 0.7)
True
>>> repasse_par(0, 10, 0.7)
False
>>> repasse_par(0, 10, 0.7)
False
>>> repasse_par(0, 10, 0.7)
False
```

EXERCICE #6 ► Probabilités - statistiques

Écrire une fonction réalisant des statistiques sur la question du passage par un objectif (on donne comme paramètre supplémentaire le nombre de marches).

```
>>> proba_passage(0, 2, 10**4, 0.5)
0.4978
>>> proba_passage(0, 3, 10**4, 0.5)
0.5021
>>> proba_passage(0, 4, 10**5, 0.5)
0.62563

>>> proba_passage(0, 10, 10**5, 0.5)
0.7553
>>> proba_passage(0, 100, 10**5, 0.5)
0.91979
>>> proba_passage(0, 10**4, 10**4, 0.5)
0.9913

>>> proba_passage(-5, 4, 10**5, 0.5)
0.0
>>> proba_passage(-5, 10, 10**5, 0.5)
0.10912

>>> proba_passage(-5, 100, 10**5, 0.5)
0.61873
>>> proba_passage(-5, 10**4, 10**4, 0.5)
0.9637

>>> proba_passage(-5, 10, 10**5, 0.8)
0.00071
>>> proba_passage(-5, 10, 10**5, 0.2)
0.75681
>>> proba_passage(-5, 100, 10**5, 0.8)
0.00107
>>> proba_passage(-5, 100, 10**5, 0.2)
1.0
>>> proba_passage(-5, 10**4, 10**3, 0.8)
0.001
>>> proba_passage(-5, 10**4, 10**3, 0.2)
1.0
```

4.1.2 Dans le plan, puis l'espace

On passe dans le plan! Cette fois, un pas est un couple (ou une liste de deux) entiers : il y a 4 déplacements équiprobables.

EXERCICE #7 ► Un pas

Écrire une fonction réalisant un pas dans le plan. On pourra choisir un pas dans la liste des quatre possibles, grâce à la fonction `randint` qui prend en entrée deux arguments entiers, disons a et b , et renvoie un entier n tel que $a \leq n < b$ (oui, une inégalité stricte à droite), chacun avec probabilité $\frac{1}{b-a}$.

```
>>> randint(0, 4)
0
3
>>> randint(0, 4)
2
>>> randint(0, 4)
3
>>> randint(0, 4)

>>> pas2()
[0, 1]
>>> pas2()
[-1, 0]
```

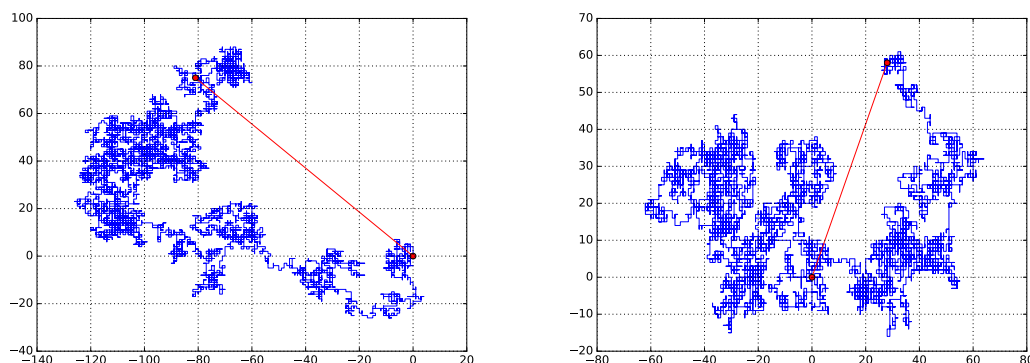
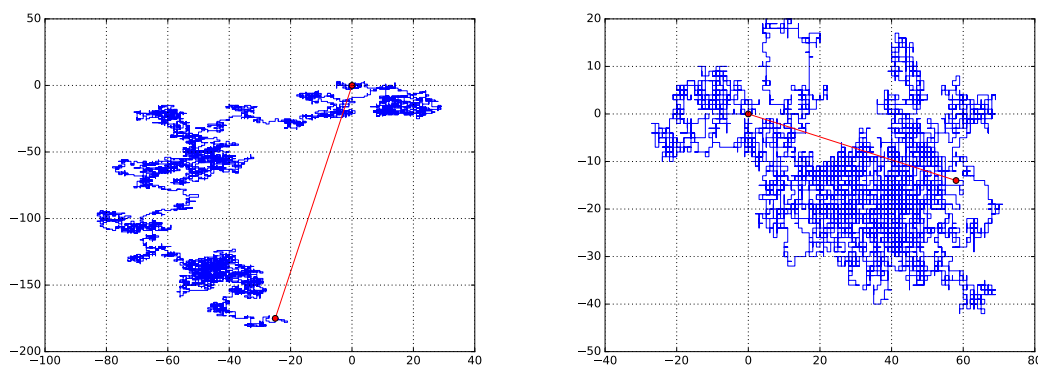
FIGURE 4.3 – Des marches (de 10^4 pas) dans le plan – j’ai relié l’origine à l’arrivée.

FIGURE 4.4 – Et deux de plus

EXERCICE #8 ► Marche, dessin de marche, etc.

Remplir les TODO du code fourni, pour obtenir des dessins ressemblants à ceux des Figures 4.3 et 4.4.

Vous pouvez voir qu’il y a une grande variété de scénarios possibles; en particulier sur la question du retour vers l’origine!

EXERCICE #9 ► Distance moyenne

Écrire des fonctions permettant d’évaluer la distance moyenne (resp. : au carré) après une marche de n pas. Expérimenter.

On s’intéresse enfin à la question du retour à l’origine :

EXERCICE #10 ► Retour au même point

Écrire des fonctions permettant d’évaluer la probabilité pour qu’une marche (d’une longueur imposée) atteigne un point donné.

Et maintenant, dans l’espace! Ici il y a un grand changement : la probabilité pour que l’on repasse par l’origine est strictement plus petite que 1.

EXERCICE #11 ► En autonomie

Reprendre les questions précédentes (distance (au carré) moyenne, probabilité de passage par l’origine), et tester!

4.1.3 Sur le cercle

On termine avec une marche aléatoire trigonométrique, avec un marcheur qui se déplace d’un angle $\pm\alpha$ à chaque étape – on part du point de coordonnées $(1,0)$. La théorie dit que la position moyenne (enfin, son espérance ...) après n pas est $(\cos^n \alpha, 0)$.

EXERCICE #12 ► Et sur le cercle?

Reprendre les questions précédentes dans ce cas.

Les maths en jeu Les 4 premiers points sont valables pour “le cas de l’axe”.

- par linéarité de l’espérance, $\mathbb{E}(S_n) = \sum_{i=1}^n \mathbb{E}(X_i) = n(p - (1 - p)) = n(2p - 1)$ (ce qui vaut 0 sans surprise dans le cas non biaisé $p = 1/2$).
- les variables aléatoires X_i **étant indépendantes** on a $\text{Var}(S_n) = \sum_{i=1}^n \text{Var}(X_i)$ et $\text{Var}(X_i) = \mathbb{E}(X_i^2) - \mathbb{E}(X_i)^2 = 1 - (2p - 1)^2 = 4p(1 - p)$ (X_i^2 est constante égale à 1, d’où son espérance!). Dans le cas $p = 1/2$, on a donc $\text{Var}(X_n) = n$ (et donc $\mathbb{E}(S_n^2) = n$).
- L’inégalité de Cauchy-Schwarz (par exemple, mais Jensen passe aussi!) donne $\mathbb{E}(|S_n|.1) \leq \sqrt{\mathbb{E}(S_n^2)}\sqrt{\mathbb{E}(1^2)}$, majorant qui vaut \sqrt{n} dans le cas $p = 1/2$.
- Le fait que la marche repasse par l’origine (et en fait par tout point) avec probabilité 1 est à la fois accessible avec des maths de licence ... et tout de même non évident (exercice de niveau L2/L3).
- dans le plan, on a cette fois un couple de variables aléatoires $Z_n = (S_n, T_n) = (X_1 + \dots + X_n, Y_1 + \dots + Y_n)$. La linéarité de l’espérance nous donne encore $\mathbb{E}(S_n) = 0$ (les X_i et les Y_i sont d’espérance nulle, et de variance égale à $\mathbb{E}(X_i^2) = 2/4 = 1/2$). On obtient de même (par indépendance) $\mathbb{E}(\|S_n\|^2) = \sum_{i=1}^n \mathbb{E}(X_i^2 + Y_i^2) = n$, puis $\mathbb{E}(\|S_n\|) \leq \sqrt{n}$. Le passage presque sûr (avec probabilité 1) par l’origine ou tout autre point est plus difficile que dans le plan.
- sur le cercle trigonométrique, tout se comprend bien avec peu de calcul si on note que cette fois $Z_n = e^{i\alpha Z_n} = e^{i\alpha X_1} e^{i\alpha X_2} \dots e^{i\alpha X_n}$, et l’espérance du produit est le produit des espérances car les $e^{i\alpha X_k}$ sont indépendantes!

4.2 Miller - Rabin

Un des algorithmes randomisés classiques est le test de primalité de Miller-Rabin, qui n’est pas difficile à implémenter. Comme d’habitude, c’est l’analyse qui est plus compliquée. https://fr.wikipedia.org/wiki/Test_de_primalit%C3%A9_de_Miller-Rabin