

# Table des matières

<b>1</b>	<b>Introduction</b> . . . . .	1
1.1	Algorithmes aléatoires et méthodes de Monte-Carlo . . . . .	1
1.2	Prérequis et plan . . . . .	2
1.3	Pour aller plus loin . . . . .	4
<b>2</b>	<b>Tirages indépendants</b> . . . . .	9
2.1	Analyse des algorithmes de simulation . . . . .	9
2.1.1	Postulats . . . . .	9
2.1.2	Théorème central limite . . . . .	12
2.1.3	Intervalles de confiance . . . . .	14
2.1.4	Estimation d'une probabilité . . . . .	16
2.1.5	Conditionnement . . . . .	18
2.2	Inversion . . . . .	20
2.2.1	Principe . . . . .	20
2.2.2	Lois discrètes . . . . .	21
2.3	Méthodes de rejet . . . . .	23
2.3.1	Lois uniformes . . . . .	23
2.3.2	Lois à densité . . . . .	27
2.3.3	Lois discrètes . . . . .	30
2.4	Décomposition . . . . .	31
2.4.1	Principe . . . . .	31
2.4.2	Lois à densité . . . . .	32
2.4.3	Lois discrètes . . . . .	33
2.5	Permutations et échantillonnage . . . . .	35
2.5.1	Permutations aléatoires . . . . .	35
2.5.2	Echantillons aléatoires . . . . .	36
2.6	Simulation des lois normales . . . . .	38
2.6.1	Principe . . . . .	38
2.6.2	Algorithme polaire . . . . .	39
2.6.3	Algorithme de Box-Muller . . . . .	40
2.6.4	Conditionnement d'exponentielles . . . . .	40
2.6.5	Lois normales multidimensionnelles . . . . .	42
2.7	Calculs d'espérances . . . . .	43
2.7.1	Principe . . . . .	43

2.7.2	Variables négativement corrélées	45
2.7.3	Réduction de la variance	46
2.8	Générateurs pseudo-aléatoires	48
2.8.1	Suites uniformes	48
2.8.2	Implémentation	49
2.8.3	Complexité et hasard	51
2.8.4	Points régulièrement répartis	55
2.8.5	Suites de van der Corput	56
2.9	Exercices	57
<b>3</b>	<b>Méthodes markoviennes à temps fini</b>	<b>71</b>
3.1	Simulation des chaînes de Markov	71
3.1.1	Définition algorithmique	71
3.1.2	Espace d'états fini ou dénombrable	73
3.1.3	Relations algébriques	77
3.2	Résolution de systèmes linéaires	78
3.2.1	Puissances de matrices	78
3.2.2	Utilisation d'un état absorbant	81
3.3	Problèmes différentiels	82
3.3.1	Le problème de la chaleur	83
3.3.2	Simulation des processus de diffusion	86
3.3.3	Problèmes de Dirichlet	90
3.3.4	Equations de Fokker-Planck et Feynman-Kac	92
3.4	Méthodes particulières	97
3.4.1	Propagation du chaos	97
3.4.2	Equations de McKean-Vlasov	99
3.4.3	Equations de Boltzmann	101
3.5	Exercices	104
<b>4</b>	<b>Exploration markovienne</b>	<b>115</b>
4.1	Comportement asymptotique	115
4.1.1	Mesures stationnaires	116
4.1.2	Simulation exacte d'une mesure stationnaire	117
4.1.3	Mesures réversibles	121
4.1.4	Dénombrement par chaîne de Markov	124
4.1.5	Convergence vers une mesure réversible	126
4.1.6	Convergence abrupte d'un échantillon	132
4.2	Recuit simulé	137
4.2.1	Mesures de Gibbs	137
4.2.2	Schémas de température	139
4.2.3	Spectre à basse température	142
4.2.4	Implémentation	145
4.3	Algorithmes génétiques	147
4.3.1	Version classique	147
4.3.2	Théorie asymptotique	149

4.3.3	Implémentation .....	153
4.4	Algorithme MOSES .....	155
4.4.1	Définition et convergence .....	155
4.4.2	Implémentation .....	157
4.5	Exercices .....	160
<b>5</b>	<b>Processus markoviens de saut</b> .....	167
5.1	Algorithmes temporisés .....	167
5.1.1	Lois exponentielles et lois géométriques .....	167
5.1.2	Le processus de Poisson .....	170
5.1.3	Chaînes temporisées .....	174
5.1.4	Taux de transition et générateur .....	175
5.1.5	Chaîne incluse et chaîne harmonisée .....	179
5.1.6	Pratique de la simulation .....	181
5.1.7	Probabilités de transitions .....	187
5.2	Simulation de réseaux .....	190
5.2.1	Automates indépendants .....	190
5.2.2	Réseaux de Jackson .....	193
5.2.3	Réseaux de Petri .....	196
5.2.4	Systèmes de particules interactives .....	199
5.3	Convergence des systèmes de particules .....	203
5.3.1	L'heuristique du champ moyen .....	203
5.3.2	Mélange rapide et laplacien .....	206
5.3.3	Equations de réaction-diffusion .....	207
5.4	Exercices .....	211
<b>6</b>	<b>Simulation en Scilab</b> .....	221
6.1	Introduction à Scilab .....	221
6.1.1	Pourquoi Scilab ? .....	221
6.1.2	A savoir pour commencer .....	222
6.1.3	Types de fichiers .....	223
6.1.4	Style de programmation .....	225
6.2	Vecteurs aléatoires .....	226
6.2.1	Lois discrètes .....	227
6.2.2	Générateurs pseudo-aléatoires .....	228
6.2.3	Représentations graphiques .....	230
6.2.4	Calculs de moments .....	232
6.2.5	Calculs sur les lois usuelles .....	233
6.2.6	Calculs d'intégrales .....	236
6.3	Algorithmes de simulation .....	239
6.3.1	Simulation de diffusions .....	239
6.3.2	Recuit simulé .....	245
6.3.3	Processus markoviens de saut .....	248
6.4	Exercices .....	255

xii Table des matières

**Références** ..... 259

**Index** ..... 267

# 1 Introduction

## 1.1 Algorithmes aléatoires et méthodes de Monte-Carlo

Décrire l'évolution future d'un modèle par référence à son état présent est la fonction de la plupart des modèles temporels, qu'il s'agisse des systèmes dynamiques discrets ou des équations différentielles. La traduction informatique d'un tel modèle est un algorithme itératif, pour lequel à chaque pas les variables sont fonction du pas précédent.

Nous choisissons ici de présenter une chaîne de Markov comme la sortie d'un algorithme itératif auquel est greffée une source de hasard, introduisant à chaque pas des variables aléatoires indépendantes. Ces algorithmes itératifs ainsi "randomisés" sont l'objet de base dans ce livre. Certains sont utilisés pour résoudre des problèmes déterministes, d'optimisation ou de dénombrement par exemple, d'autres qui leur sont très proches sont plutôt des simulations de modèles aléatoires. Certains de ces modèles sont décrits en temps continu (processus de diffusion, files d'attente), d'autres en temps discret (marches aléatoires). La différence théorique entre le temps discret et le temps continu n'est pas très importante, et au moment de l'implémentation les pas d'itération seront toujours discrets.

Qu'ils soient vus comme des méthodes numériques ou des simulations, les algorithmes décrits ici partagent des caractéristiques communes. Ce sont en général des méthodes relativement peu précises, mais faciles à implémenter, robustes et destinées en priorité aux problèmes de très grande taille. Pour illustrer ceci, voici quelques éléments de comparaison avec les méthodes déterministes.

L'objectif à atteindre étant la solution approchée d'un certain problème avec une précision  $\varepsilon$  fixée, un critère de choix important dans le choix d'une méthode numérique est le temps d'exécution, mesuré en nombre d'opérations élémentaires (la complexité de l'algorithme). Intervient alors la notion de taille du problème, qui est un entier  $n$  dont la signification est habituellement claire en fonction du contexte. C'est la dimension de l'espace pour une intégrale multiple, la taille de la matrice pour la résolution d'un système, etc. . . . Les complexités des méthodes numériques déterministes dépendent habituellement de la taille de façon polynomiale, typiquement en  $O(n^2)$  ou  $O(n^3)$ . Tout aussi typiquement, une méthode de Monte-Carlo (utilisant le hasard) résoudra le même problème en  $O(n)$ . Cela semble miraculeux : où est

donc le piège? Il est dans la constante cachée par l'expression  $O(n)$ . Cette constante dépend de la précision  $\varepsilon$  à atteindre. Pour une dimension donnée, il est fréquent qu'une méthode numérique atteigne la précision  $\varepsilon$  en un temps  $O(\log(\varepsilon^{-1}))$  (en idéalisant, cela signifie que chaque nouveau pas d'itération fait passer la précision de  $10^{-k}$  à  $10^{-(k+1)}$ ). Pour une raison liée au théorème central limite, les méthodes de Monte-Carlo atteignent la précision  $\varepsilon$  en  $O(\varepsilon^{-2})$ , ce qui est très lent (typiquement  $10^6$  itérations pour une précision de  $10^{-3}$ ).

Déterministe	Monte-Carlo
$K \log(\varepsilon^{-1}) n^2$	$K \varepsilon^{-2} n$

C'est donc dans des cas où la taille  $n$  du problème est très grande et l'exigence sur la précision faible ( $\varepsilon$  grand) qu'il faut envisager les méthodes de Monte-Carlo. Même si, pour des raisons de clarté, nos exemples seront souvent en basses dimensions, on ne calcule pas en pratique une intégrale double ou triple, on ne résout pas un système  $2 \times 2$  (ni même  $100 \times 100$ ) avec une méthode de Monte-Carlo. L'utilisation du hasard ne devient concurrentielle qu'à partir de dizaines de dimensions pour une intégrale ou de milliers d'équations pour un système. Il peut même se faire que la taille du problème soit telle que les méthodes déterministes échouent pour des raisons de place en mémoire. Par exemple, la taille maximale des systèmes linéaires que l'on peut résoudre actuellement est de l'ordre de  $10^6$ . Dans certains cas, une méthode de Monte-Carlo ira bien au-delà. Remarquons également que les deux approches peuvent être complémentaires. De nombreuses méthodes numériques demandent à être initialisées par une valeur déjà assez proche de la solution (par exemple Newton-Raphson). Une méthode de Monte-Carlo pourra fournir à peu de frais cette initialisation.

## 1.2 Prérequis et plan

Ce livre est résolument pratique. Il se veut accessible à tout étudiant de second ou troisième cycle universitaire, ayant reçu une formation minimale en probabilités. Aucune connaissance particulière, autre qu'un peu de bon sens, ne sera supposée acquise. Les seules notions de probabilité qui seront utilisées sont les suites de variables indépendantes (fonctions d'appels de Random successifs), le théorème central limite, qui est l'outil de base pour déterminer la précision d'un algorithme de Monte-Carlo, et les chaînes de Markov, vues comme des algorithmes itératifs où chaque nouveau pas est calculé en fonction du précédent et d'un tirage aléatoire. Les connaissances correspondantes figurent dans tous les manuels classiques, qui vont en général bien au-delà (par exemple Berger [7], Bouleau [9, 10], Breiman [12], Feller [21, 22], Snell [59]...). Sur les chaînes de Markov plus particulièrement, on pourra se reporter au livre de Brémaud [13]. Les références de base restent Chung [14] et Kemeny et Snell [33]. Un point de vue plus appliqué est celui de Barucha-Reid [6], Bhat [8] et Karlin et Taylor [29, 30]. Çinlar [15] est particulièrement

clair. Anderson [2] traite plus spécialement du temps continu. Les livres de Neuts [128, 129] proposent une vision systématiquement tournée vers l’outil informatique. L’ouvrage à paraître d’Aldous et Fill [1] sera certainement une référence durable.

La deuxième partie traite de la génération d’échantillons aléatoires. Nous n’aborderons les générateurs pseudo-aléatoires et la simulation des lois de probabilité usuelles, que pour en donner quelques principes de base. La première application est le calcul d’intégrales, exprimées comme des espérances de variables aléatoires. Les calculs d’intégrales par Monte-Carlo sont traités de manière plus ou moins détaillée dans de nombreux manuels, comme ceux de Kennedy et Gentle [34], Hammersley et Handscomb [28], Gentle [27], Morgan [45], Rubinstein [54], Ripley [48], Kleijnen [35, 36]. Sans surprises sur le plan théorique, elles seront surtout l’occasion de rappeler un certain nombre de bases probabilistes et algorithmiques, l’objectif principal étant de développer l’état d’esprit assez particulier qui préside à une implémentation efficace des méthodes de Monte-Carlo. Il s’agit en effet de s’habituer à ce que la vitesse d’exécution de l’algorithme conditionne avant tout la précision du résultat. Nous illustrerons ce point de vue à l’aide de plusieurs astuces de programmation, habituellement regroupées sous l’appellation de “méthodes de réduction de la variance”.

La troisième partie traite des méthodes markoviennes qui, comme dans la partie précédente, utilisent la loi des grands nombres pour calculer une espérance. Cette espérance est celle d’une variable aléatoire, fonction de la trajectoire d’une chaîne de Markov sur un intervalle de temps borné. Elle est approchée par une moyenne des valeurs prises par la variable sur un grand nombre de trajectoires indépendantes. L’application aux systèmes linéaires nous servira surtout à introduire les méthodes de résolution d’équations aux dérivées partielles. Comme référence de base sur le sujet, nous utiliserons le livre de Lapeyre *et al.* [40]. Plus que l’aspect numérique, nous chercherons à développer les analogies entre modèles stochastiques et déterministes.

La quatrième partie traite d’un autre type de méthodes markoviennes, celles qui explorent un espace d’états soit de manière homogène pour approcher une mesure d’équilibre donnée (méthodes MCMC [52]), soit de manière dirigée à la recherche d’un point (extrémum d’une fonction [82]). Dans ce dernier cas, il s’agit de suivre une trajectoire d’une chaîne de Markov, qui visite avec une probabilité croissante un voisinage de la cible à atteindre. Parmi ces techniques, on peut ranger les méthodes neuronales, que nous n’aborderons pas (voir [37, 42, 50, 67]). Nous traiterons surtout le recuit simulé [70, 78, 84] et décrirons l’heuristique des algorithmes génétiques [4, 43, 44, 88, 89, 90, 106, 111], et de l’algorithme MOSES [104]. Ces algorithmes peuvent être vus comme des méthodes de descente de gradient, “bruitées” afin d’éviter les pièges d’éventuels minima locaux. Les questions

théoriques de convergence et de précision des méthodes d'exploration markovienne sont souvent très difficiles. Elles ont donné lieu à une intense activité de publication ces 15 dernières années (voir Saloff-Coste [136]). Nous n'aborderons ces questions que de manière assez superficielle dans le cadre des chaînes réversibles. Les deux livres de Duflo [19, 20] constituent une référence de base, d'un niveau sensiblement supérieur à celui de ce cours.

Les modèles à temps continu et espace d'états discrets (processus markoviens de saut) sont l'objet de la cinquième partie. Nous en proposerons une définition algorithmique qui permet de ramener leur étude mathématique aussi bien que leur simulation aux chaînes à temps discret. A titre d'exemple, nous étudierons plusieurs algorithmes de simulation sur des espaces de type produit comme les réseaux de files d'attente [133, 134] et les systèmes de particules interactives [100, 118, 119]. La convergence des systèmes de particules sera une nouvelle occasion de faire ressortir la cohérence entre les modèles stochastiques et déterministes.

L'objectif pratique de ce livre ne peut être atteint que si les algorithmes étudiés conduisent à des implémentations effectives et à une expérimentation numérique. Ceci fait l'objet de la sixième partie. Les impératifs de vitesse d'exécution imposent dans les applications le choix d'un langage compilé. Mais de plus en plus les praticiens utilisent aussi un environnement de calcul scientifique, qui permet de réaliser rapidement et facilement des tests d'algorithmes et des maquettes de logiciels. C'est ce que nous avons choisi avec le logiciel libre Scilab [46, 64, 143].

### 1.3 Pour aller plus loin

Introduire le hasard dans des modèles mathématiques et traiter ces modèles par la simulation est courant dans de nombreuses disciplines scientifiques. On englobe souvent dans les "méthodes de Monte-Carlo" tout ce qui a trait à l'utilisation du hasard dans des programmes informatiques (voir par exemple Fishman [23]). Cette dénomination reste assez imprécise, et varie selon les spécialités. L'adjectif "markovien" est lui aussi très général, puisqu'on l'associe à tous les modèles probabilistes à dépendance locale, qu'ils soient indicés par le temps (processus et chaînes) ou l'espace (champs).

La base de données bibliographiques du Zentralblatt recense plus de 2000 titres contenant "Monte-Carlo" et plus de 10000 contenant "Markov" : toute prétention de ce livre à l'exhaustivité est exclue.

Nous nous contenterons d'indiquer quelques pistes pour compléter ce qui est décrit ici, sous forme d'une liste de domaines plus ou moins interconnectés, liés à l'utilisation du hasard sur ordinateur, ou aux modèles markoviens.

- *La construction des générateurs pseudo-aléatoires*, ou comment coder efficacement une fonction Random. C'est un problème que nous considérons arbitrairement comme résolu par Marsaglia et Zaman [120, 121],

bien qu'une littérature importante continue à se développer sur la question. Quatre références de base sont les livres de Knuth [116], Dudewicz et Ralley [18], Fishman [23], Gentle [27]. L'article de Ripley [49] est une bonne introduction.

- *La simulation des variables aléatoires*, ou comment transformer un appel de `Random` (réalisation d'une variable aléatoire de loi uniforme sur  $[0, 1]$ ) en une variable aléatoire de loi donnée. Ce sujet est abordé à niveau élémentaire dans de nombreux manuels, par exemple les livres de Bouleau [9], Snell [59] ou Berger [7]. Il est traité à fond par Devroye dans [17]. Nous nous contenterons de quelques indications sur les trois principes généraux que sont l'inversion, le rejet et la décomposition.
- *La simulation des processus stochastiques* ou comment transformer une suite de variables aléatoires indépendantes et de même loi (suite d'appels de `Random`) en un processus quelconque (martingale, chaîne ou processus de Markov, champ aléatoire, processus de diffusion...). Plusieurs livres traitent de la simulation des processus, parmi lesquels celui de Bouleau et Lépingle [11]. Pour les processus de diffusion les livres de Kloeden et Platen [114, 115] sont la référence indispensable. Nous nous limiterons aux algorithmes de simulation les plus simples, ceux d'Euler-Maruyama et Heun-Milshtein.
- *L'analyse probabiliste d'algorithmes*. Etudier la complexité d'un algorithme (déterministe) dans le pire ou le meilleur des cas, reflète rarement son comportement sur des données courantes. On a donc souvent recours à une analyse "en moyenne" où les données d'entrée de l'algorithme sont tirées au hasard, en un sens qui dépend du type de problème étudié. Sur cette question, plusieurs références de la littérature informatique sont accessibles au mathématicien appliqué, parmi lesquelles Graham *et al.* [109], Sedgewick et Flajolet [56] ou Hofri [110], plus spécialisé.
- *Les algorithmes randomisés*. Ce sujet s'est développé ces dix dernières années sous l'impulsion d'informaticiens théoriciens. Parmi les problèmes dont la solution peut être programmée, on distingue ceux qui sont résolubles en un temps qui ne dépasse pas une certaine puissance de la taille du problème de ceux qui ne le sont pas. Pour certains de ces derniers, on a pu trouver des algorithmes de résolution approchée en temps polynômial, à base essentiellement de chaînes de Markov. Leur étude est devenue un domaine important de l'informatique théorique (voir Sinclair [58] ou Motwani et Raghavan [125]).
- *Les méthodes de Monte-Carlo en analyse numérique*. Calculs d'intégrales, résolution de systèmes linéaires ou d'équations différentielles, optimisation numérique, pour tous les problèmes classiques de l'analyse numérique des algorithmes utilisant les générateurs pseudo-aléatoires ont été proposés. Nous en traiterons quelques uns, sans chercher à raffiner systématiquement leurs performances ni à les comparer aux alter-

natives déterministes (voir par exemple [23, 54], ou le livre récent de Liu [41]).

- *Les méthodes de Monte-Carlo en statistique.* De nombreuses questions d'estimation, de tests ou de représentation de données se ramènent à des problèmes numériques d'optimisation ou de résolution de systèmes de grande taille. Les algorithmes qui seront présentés dans ce cours connaissent un grand succès auprès des statisticiens, qui en ont déduit des versions adaptées à leurs problèmes (échantillonneur de Gibbs, algorithmes EM, SEM... : voir Robert [52, 53] et McLachlan [122]). Il est très artificiel de couper, comme nous le ferons, les méthodes de résolution de problèmes déterministes de leurs applications naturelles en statistique. Dufflo [19, 20] traite d'ailleurs sans distinction les deux types d'applications. Robert [51] montre bien l'importance et l'intérêt des méthodes de Monte-Carlo en statistique, en particulier bayésienne (voir aussi [79, 139]).
- *Les algorithmes de filtrage.* Il existe de nombreuses méthodes adaptées aux cas où les données du problème à traiter ne sont pas connues exactement, soit qu'elles découlent d'un calcul numérique entaché d'erreurs importantes, soit qu'elles proviennent d'un échantillon statistique. Filtrage de Kalman, méthodes de fonctions splines [102] ou ondelettes [68], algorithmes de Robbins-Monro ou Kiefer-Wolfowitz, toute une panoplie de techniques permettent de traiter des données bruitées (voir Bénéaïm [76] ou Benvéniste *et al.* [77], et Winkler [145] pour le cas de l'analyse d'images). Là aussi notre séparation entre les méthodes où le hasard provient du modèle et celles où il est apporté par l'utilisation de la fonction `Random` est tout à fait artificielle. Les processus stochastiques sous-jacents sont essentiellement les mêmes, comme le montre bien Dufflo [20] (voir aussi [5, 38]).
- *Le traitement numérique des chaînes de Markov.* Une fois construit un modèle markovien, toute l'information sur l'évolution du modèle est en théorie accessible par des techniques d'algèbre linéaire. La limitation provient de la taille des matrices de transition, quand le nombre d'états explose, en particulier pour les modèles d'attente ou de fiabilité. Des techniques adaptées aux grandes matrices de transition ont été inventées. Ce sujet est traité par Stewart [140].
- *La parallélisation des algorithmes de Monte-Carlo.* Dans la mesure où il s'agit fréquemment de moyennner des trajectoires indépendantes, on comprend que les méthodes de Monte-Carlo soient bien adaptées à la parallélisation (voir par exemple [96]). Le problème se complique quand il s'agit de simuler des trajectoires dépendantes, comme par exemple pour un système de particules interactives. Pour les algorithmes d'optimisation stochastique une théorie complète a été développée : voir Catoni et Trouné [87], Shonkwiler et Van Vleck [138] et Trouné [142]).

- *La construction et la simulation de modèles stochastiques d'attente*, par exemple en recherche opérationnelle, automatique ou informatique : voir entre autres [3, 24, 55, 57, 60, 63, 133, 134, 135]). C'est un sujet suffisamment important pour avoir suscité le développement de langages de programmation spécialisés comme SIMULA ou plus récemment MODLINE et QNAP (sur les aspects algorithmiques voir aussi Watkins [62]). Nous nous limiterons à la simulation des réseaux de Jackson et de Petri markoviens. Les aspects théoriques de ces derniers sont développés par Baccelli [71] et [72].
- *La fiabilité des systèmes*. Des modèles markoviens peuvent décrire l'état des composants d'un système complexe, entre le bon fonctionnement et la panne (voir [16, 69, 74]). Il s'agit de processus de saut, proches des réseaux de Jackson ou des systèmes de particules de la cinquième partie.
- *L'utilisation du calcul stochastique en analyse financière*. Nous n'aborderons pas ce domaine en pleine expansion, sauf pour lui emprunter l'exemple de la diffusion de Black et Scholes. La simulation y joue pourtant un rôle croissant (voir [39, 113, 126, 137]).
- *Les processus markoviens de décision*. En économie, quand un modèle markovien est choisi pour les aléas du marché, le problème se pose de déterminer des stratégies de comportement optimales, en maximisant l'espérance de certaines fonctions des trajectoires. Puterman [47] est une bonne introduction à ce domaine.
- *Les chaînes à espace d'états continu*. A part pour les processus de diffusion dans la troisième partie, nous nous en tiendrons à des espaces d'états finis ou dénombrables. Les questions théoriques de convergence sont notablement plus difficiles dans le cas continu. La référence de base sur le sujet est le livre de Meyn et Tweedie [123]. Le point de vue fonctionnel est développé par Diaconis et Freedman dans [97].
- *La convergence vers les processus de diffusion*. De nombreux modèles discrets, en théorie des files d'attente, dynamique des populations ou chimie admettent des approximations continues, généralisant la convergence de la marche aléatoire simple vers le mouvement brownien. Ethier et Kurtz [101] donnent les principaux résultats, et traitent de nombreux exemples d'applications.
- *Les processus ponctuels spatiaux*. Pour modéliser des objets de formes variables dans l'espace (dunes, vagues, galaxies), on a recours à des extensions continues des modèles de particules. Baddeley et Møller [73] donnent une bonne introduction au sujet. La simulation de ces modèles fait l'objet de [124].

# Références

## Références de base

1. D.J. Aldous and J.A. Fill. Reversible Markov chains and random walks on graphs. [www.stat.berkeley.edu/~aldous/book.html](http://www.stat.berkeley.edu/~aldous/book.html), à paraître, 2002.
2. W.J. Anderson. *Continuous-time Markov chains. An applications-oriented approach*. Springer-Verlag, New York, 1991.
3. S. Asmussen. *Applied probability and queues*. Wiley, New York, 1987.
4. T. Bäck. *Evolutionary algorithms in theory and practice*. Oxford University Press, Oxford, 1996.
5. N. Bartoli et P. Del Moral. *Simulation et algorithmes stochastiques*. Cépaduès Editions, Toulouse, 2001.
6. A.T. Barucha-Reid. *Elements of the theory of Markov processes and their Applications*. McGraw-Hill, London, 1960.
7. M.A. Berger. *An introduction to probability and stochastic processes*. Springer-Verlag, New York, 1993.
8. U.N. Bhat. *Elements of applied stochastic processes*. Wiley, New York, 1984.
9. N. Bouleau. *Probabilités de l'ingénieur, variables aléatoires et simulation*. Hermann, Paris, 1985.
10. N. Bouleau. *Processus stochastiques et applications*. Hermann, Paris, 1988.
11. N. Bouleau and D. Lépingle. *Numerical methods for stochastic processes*. Wiley, New York, 1994.
12. L. Breiman. *Probability*. Addison-Wesley, Reading, 1968.
13. P. Bremaud. *Markov chains, Gibbs fields, Monte-Carlo simulation and queues*. Springer-Verlag, New York, 1999.
14. K.L. Chung. *Markov chains with stationary transition probabilities*. Springer-Verlag, New York, 1960.
15. E. Çinlar. *Introduction to stochastic processes*. Prentice Hall, New York, 1975.
16. Ch. Coccozza-Thivent. *Processus stochastiques et fiabilité des systèmes*. Mathématiques et applications 28. Springer-Verlag, Berlin, 1997.
17. L. Devroye. *Non-uniform random variate generation*. Springer-Verlag, New York, 1986.
18. E.J. Dudewicz and T.G. Ralley. *The handbook of random number generation and testing with TESTRAND computer code*. American Sciences Press Inc., Columbus., 1981.

19. M. Duflo. *Méthodes récursives aléatoires*. Masson, Paris, 1990.
20. M. Duflo. *Algorithmes stochastiques*. Mathématiques et applications 23. Springer-Verlag, Berlin, 1996.
21. W. Feller. *An introduction to probability theory and its applications*, volume I. Wiley, London, 1968.
22. W. Feller. *An introduction to probability theory and its applications*, volume II. Wiley, London, 1971.
23. G.S. Fishman. *Monte-Carlo concepts algorithms and applications*. Springer-Verlag, New York, 1996.
24. G.S. Fishman. *Discrete-event simulation*. Springer-Verlag, New York, 2001.
25. T.C. Gard. *Introduction to stochastic differential equations*. Marcel Dekker, Inc., New York, 1988.
26. E. Gelenbe et G. Pujolle. *Introduction aux réseaux de files d'attente*. Eyrolles, Paris, 1985.
27. J.E. Gentle. *Random number generation and Monte-Carlo methods*. Springer-Verlag, New York, 1998.
28. J.M. Hammersley and D.C. Handscomb. *Monte-Carlo methods*. Methuen, London, 1964.
29. S. Karlin. *A first course in stochastic processes*. Academic Press, San Diego, 1966.
30. S. Karlin and H.M. Taylor. *A second course in stochastic processes*. Academic Press, San Diego, 1981.
31. J. Keilson. *Markov chain models - rarity and exponentiality*. Applied Mathematical Sciences 28. Springer-Verlag, New York, 1979.
32. F.P. Kelly. *Reversibility and Stochastic Networks*. Wiley, London, 1979.
33. J.G. Kemeny and J.L. Snell. *Finite Markov chains*. Van Nostrand, Princeton, 1960.
34. W.J. Kennedy and J.E. Gentle. *Statistical computing*. Marcel Dekker, Inc., New York, 1980.
35. J.P.C. Kleijnen. *Statistical techniques in simulation, Part I*. Marcel Dekker, Inc., New York, 1974.
36. J.P.C. Kleijnen and W. Van Groenendaal. *Simulation, a statistical perspective*. Wiley, New York, 1992.
37. J. Korst and E.H. Aarts. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization*. Wiley, New York, 1989.
38. H.J. Kushner and G.G. Yin. *Approximation algorithms and applications*. Springer-Verlag, New York, 1997.
39. D. Lamberton et B. Lapeyre. *Introduction au calcul stochastique appliqué à la finance*. Ellipses, Paris, 1991.
40. B. Lapeyre, E. Pardoux et R. Sentis. *Méthodes de Monte-Carlo pour les équations de transport et de diffusion*. Mathématiques et applications 29. Springer-Verlag, Berlin, 1997.

41. J.S. Liu. *Monte-Carlo strategies in scientific computing*. Springer-Verlag, New York, 2001.
42. T. Masters. *Practical neural network recipes in C++*. Academic Press, Boston, 1993.
43. Z. Michalewicz. *Genetic algorithms + Data structures = Evolution programs, 3rd ed.* Springer-Verlag, New York, 1996.
44. M. Mitchell. *An introduction to genetic algorithms*. MIT Press, Cambridge, MA, 1996.
45. B.J.T. Morgan. *Elements of simulation*. Chapman and Hall, London, 1984.
46. B. Pinçon. Introduction à Scilab. [www.iecn.u-nancy.fr/~pincon/scilab/scilab.html](http://www.iecn.u-nancy.fr/~pincon/scilab/scilab.html), 1996.
47. M. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley, New York, 1994.
48. B.D. Ripley. *Stochastic simulation*. Wiley, New York, 1987.
49. B.D. Ripley. Thoughts on pseudorandom numbers. *J. Comput. Appl. Math.*, 31:153–163, 1990.
50. B.D. Ripley. *Pattern recognition and neural networks*. Cambridge University Press, 1996.
51. C.P. Robert. *L'Analyse Statistique Bayésienne*. Economica, Paris, 1992.
52. C.P. Robert. *Méthodes de Monte-Carlo par chaînes de Markov*. Economica, Paris, 1996.
53. C.P. Robert and G. Casella. *Monte-Carlo statistical methods*. Springer-Verlag, New York, 1999.
54. R.Y. Rubinstein. *Simulation and the Monte-Carlo method*. Wiley, New York, 1981.
55. R.Y. Rubinstein and B. Melamed. *Efficient simulation and Monte-Carlo methods*. Wiley, New York, 1997.
56. R. Sedgewick et Ph. Flajolet. *Introduction à l'analyse des algorithmes*. Int. Thomson Publishing, France, 1996.
57. R. Serforzo. *Introduction to stochastic networks*. Springer-Verlag, New York, 1999.
58. A. Sinclair. *Algorithms for random generation and counting: a Markov chain approach*. Birkhäuser, Boston, 1993.
59. J.L. Snell. *Introduction to probability*. Random House, New York, 1988.
60. K.S. Trivedi. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. Prentice-Hall, New York, 1982.
61. J. Walrand. *Introduction to Queuing Networks*. Prentice-Hall, New York, 1989.
62. K. Watkins. *Discrete event simulation in C*. McGraw-Hill, London, 1993.
63. R.W. Wolff. *Stochastic Modelling and the Theory of Queues*. Prentice-Hall, Englewood Cliff, 1989.
64. B. Ycart. Démarrer en Scilab. [www.math-info.univ-paris5.fr/~ycart/polys/demarre\\_scilab/demarre\\_scilab.html](http://www.math-info.univ-paris5.fr/~ycart/polys/demarre_scilab/demarre_scilab.html), 2001.

**Pour en savoir plus**

65. D.J. Aldous. On the Markov chain simulation method for uniform combinatorial distributions and simulated annealing. *Probab. Eng. and Inf. Sciences*, 1:33–46, 1987.
66. D.J. Aldous. Approximate counting via Markov chains. *Statistical Science*, 8(1):16–19, 1993.
67. I. Aleksander and H.B. Morton. *An introduction to neural computing*. Chapman and Hall, London, 1990.
68. A. Antoniadis and G. Oppenheim, eds. *Wavelets and statistics*, L.N. in Stat 103. Springer-Verlag, New York, 1995.
69. T. Aven and U. Jensen. *Stochastic models in reliability*. Springer-Verlag, New York, 1999.
70. R. Azencott. Simulated annealing. *Séminaire Bourbaki*, 697:161–175, 1988.
71. F. Baccelli. Ergodic theory of stochastic Petri networks. *Ann. Probab.*, 20(1):375–396, 1992.
72. F. Baccelli, G. Cohen, G. Olsder, and J.P. Quadrat. *Synchronization and Linearity: an algebra for discrete event systems*. Wiley, Chichester, 1992.
73. A.J. Baddeley and J. Møller. Nearest-neighbour Markov point processes and random sets. *Int. Stat. Rev.*, 57:90–121, 1989.
74. R.E. Barlow and F. Proschan. *Mathematical theory of reliability*. SIAM, Philadelphia, 1996.
75. M. Béguin, L. Gray, and B. Ycart. The load transfer model. *Ann. Appl. Probab.*, 8(1):337–353, 1998.
76. M. Benaïm. Dynamics of stochastic algorithms. In J. Azéma *et al.*, eds, *Séminaire de Probabilité XXXIII*, L.N. in Math. 1709, pages 1–68. Springer-Verlag, New York, 1999.
77. A. Benveniste, M. Métivier et P. Priouret. *Algorithmes adaptatifs et approximations stochastiques*. Masson, Paris, 1987.
78. D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statistical Science*, 8:10–15, 1993.
79. J. Besag. Markov chain Monte Carlo for statistical inference. *Tech. Rep. 9, U. of Washington*, 2000.
80. N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1973.
81. G.W. Brams. *Réseaux de Petri : théorie et pratique*. Masson, Paris, 1983.
82. R. Cairoli and R.C. Dalang. *Sequential stochastic optimization*. Wiley, New York, 1996.
83. O. Catoni. Rough large deviation estimates for simulated annealing: Application to exponential schedules. *Ann. Probab.*, 20(3):1109–1146, 1992.
84. O. Catoni. Simulated annealing algorithms and Markov chains with rare transitions. In J. Azéma *et al.* eds, *Séminaire de Probabilité XXXIII*, L.N. in Math. 1709, pages 69–119. Springer-Verlag, New York, 1999.

85. O. Catoni. Rates of convergence for sequential annealing: a large deviation approach. In R. Azencott, ed., *Simulated annealing: parallelization techniques*, pages 25–35. Wiley, New York, 1992.
86. O. Catoni. Solving scheduling problems by simulated annealing. *SIAM J. on Control and Optim.*, 36(5):1539–1575, 1998.
87. O. Catoni and A. Trouvé. Parallel annealing by multiple trials : a mathematical study. In R. Azencott, ed., *Simulated annealing : parallelization techniques*, pages 129–143. Wiley, New York, 1992.
88. R. Cerf. *Une théorie asymptotique des algorithmes génétiques*. Thèse, Université Montpellier II, 1994.
89. R. Cerf. The dynamics of mutation-selection algorithms with large population sizes. *Ann. Inst. H. Poincaré, Probab. Stat.*, 32(4):455–508, 1996.
90. R. Cerf. A new genetic algorithm. *Ann. Appl. Probab.*, 6(3):778–817, 1996.
91. B. Chauvin. Branching processes, trees and the Boltzmann equation. *Math. and Comp. in Simulation*, 38:135–141, 1995.
92. B. Chauvin and A. Rouault. A stochastic simulation for a class of reaction-diffusion equations. *Adv. Appl. Probab.*, 22:88–100, 1990.
93. K.L. Chung. *Green, Brown, and probability and Brownian motion on the line*. World Scientific, London, 2001.
94. Y. Colin de Verdière, Y. Pan, and B. Ycart. Singular limits of Schrödinger operators and Markov processes. *J. of Operator Theory*, 41:151–173, 1999.
95. A. De Masi, P. Ferrari, and J. Lebowitz. Reaction diffusion equations for interacting particle systems. *J. Stat. Phys.*, 44:589–644, 1986.
96. G.M. Del Corso. Randomization and the parallel solution of linear algebra problems. *J. Comput. Math. Appl.*, 30(11):59–72, 1995.
97. P. Diaconis and D. Freedman. Iterated random functions. *SIAM Review*, 41(1):45–76, 1999.
98. P. Diaconis and L. Saloff-Coste. What do we know about the Metropolis algorithm? *J. Comp. Syst. Sci.*, 55(1):20–36, 1998.
99. P. Doyle and J. Snell. *Random walks and electric networks*. Math. Assoc. America, Washington, 1984.
100. R.T. Durrett. Ten lectures on particle systems. In P. Bernard, ed., *Ecole d'été de probabilité de Saint-Flour XXIII*, L.N. in Math. 1608, pages 97–201. Springer-Verlag, New York, 1995.
101. S.N. Ethier and T.G. Kurtz. *Markov processes: characterization and convergence*. Wiley, New York, 1986.
102. R.L. Eubank. *Spline smoothing and nonparametric regression*. Marcel Dekker, Inc., New York, 1988.
103. J.A. Fill. An interruptible algorithm for perfect sampling via Markov chains. *Ann. Appl. Probab.*, 8(1):131–162, 1998.
104. O. François. An evolutionary strategy for global minimization and its Markov chain analysis. *IEEE trans. on Evolutionary Computation*, 2(3):77–90, 1998.
105. M.I. Freidlin and A.D. Wentzell. *Random perturbations of dynamical systems*. Springer-Verlag, New York, 1984.

106. D. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, New York, 1989.
107. C. Graham, T.G. Kurtz, S. Méléard, P.E. Protter, M. Pulvirenti, and D. Talay. *Probabilistic Models for Nonlinear Partial Differential Equations*. L.N. in Math. 1627. Springer-Verlag, New York, 1996.
108. Forbes, F. and Ycart, B. Counting stable sets on Cartesian products of graphs. *Discrete Mathematics*, 186:105–116, 1998.
109. R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete mathematics: a foundation for computer science*. Addison-Wesley, Reading, 1989.
110. M. Hofri. *Probabilistic analysis of algorithms*. Springer-Verlag, New York, 1987.
111. J.H. Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, 1975.
112. I. Karatzas and S.E. Shreve. *Brownian motion and stochastic calculus*. Springer-Verlag, New York, 1991.
113. I. Karatzas and S.E. Shreve. *Methods of mathematical finance*. Springer-Verlag, New York, 1998.
114. P.E. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*. Springer-Verlag, New York, 1992.
115. P.E. Kloeden, E. Platen, and H. Schurz. *Numerical solution of SDE through computer experiment*. Springer-Verlag, New York, 1997.
116. D.E. Knuth. *The art of computer programming*, volume 2, seminumerical algorithms. Addison-Wesley, Reading, 1981.
117. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys. *The traveling salesman problem*. Wiley, New York, 1987.
118. T.M. Liggett. *Interacting Particle Systems*. Springer-Verlag, New York, 1985.
119. T.M. Liggett. *Stochastic interacting systems: contact, voter, and exclusion processes*. Springer-Verlag, New York, 1999.
120. G. Marsaglia and A. Zaman. Toward a universal random number generator. *Stat. Prob. Lett.*, 8:35–39, 1990.
121. G. Marsaglia and A. Zaman. A new class of random number generators. *Ann. Appl. Probab.*, 1:462–480, 1991.
122. G.J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, New York, 1997.
123. S.P. Meyn and R.L. Tweedie. *Markov chains and stochastic stability*. Springer-Verlag, London, 1993.
124. J. Møller. Markov Chain Monte-Carlo and spacial point processes. In W.S. Kendall *et al.*, eds, *Stochastic geometry. Likelihood and computation*, pages 141–172. Chapman and Hall, London, 1999.
125. R. Motwani and P. Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.
126. M. Musiela and M. Rutkowski. *Martingale methods in financial modelling*. Springer-Verlag, Berlin, 1997.

127. V.N. Nair and A.E. Freeny. Methods for assessing distributional assumptions in one and two sample problems. In J. Stanford and S. Vardeman, eds., *Statistical methods for physical science*. Academic Press, New York, 1992.
128. M.F. Neuts. *Matrix-geometric solutions in stochastic models*. The John Hopkins University Press, London, 1981.
129. M.F. Neuts. *Algorithmic Probability: a collection of problems*. Chapman and Hall, London, 1995.
130. P. Propp and D.B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Rand. Struct. Algo.*, 9(2):223–252, 1996.
131. G. Reinelt. *The traveling salesman: computational solutions for TSP applications*. L.N. in Computer Science 840. Springer-Verlag, New York, 1994.
132. Ph. Robert. *Réseaux et files d'attente : méthodes probabilistes*. Mathématiques et applications 35. Springer-Verlag, Berlin, 2000.
133. T.G. Robertazzi. *Computer networks and systems: queuing theory and performance evaluation*. Springer-Verlag, New York, 1990.
134. R.Y. Rubinstein. *Monte-Carlo optimization, simulation and sensitivity of queuing networks*. Wiley, New York, 1986.
135. R.Y. Rubinstein and A. Shapiro. *Discrete event systems: sensitivity analysis and stochastic optimization by the score function method*. Wiley, New York, 1993.
136. L. Saloff-Coste. Lectures on finite Markov chains. In P. Bernard, ed., *Ecole d'été de probabilité de Saint-Flour XXVI*, L.N. in Math. 1665, pages 301–413. Springer-Verlag, New York, 1997.
137. A.N. Shiryaev. *Essentials of stochastic finance*. World Scientific, London, 1999.
138. R. Shonkwiler and E. van Vleck. Parallel speed-up of Monte-Carlo methods for global optimization. *J. Complexity*, 10(1):64–95, 1994.
139. A.F.M. Smith and G.O. Roberts. Bayesian computation via the Gibbs sampler and related Markov chain Monte-Carlo methods. *J. R. Statist. Soc.*, 55:3–23, 1993.
140. W.J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1995.
141. D. Talay. Simulation and numerical analysis of stochastic differential systems : a review. In P. Krée and W. Wedig, eds., *Probabilistic Methods in Applied Physics*, L.N. in Physics 451, chap. 3, pages 54–96. Springer-Verlag, New York, 1995.
142. A. Trouvé. *Parallélisation massive du recuit simulé*. Thèse, Université Paris XI, 1993.
143. L.E. van Dijk and C.L. Spiel. Scilab bag of tricks. [www.hammersmith-consulting.com/scilab/sci-bot/sci-bot.html](http://www.hammersmith-consulting.com/scilab/sci-bot/sci-bot.html), 2000.
144. D.B. Wilson. Annotated bibliography of perfectly random sampling with Markov chains. In D. Aldous and J. Propp, eds., *Microsurveys in Discrete Probability*, volume 41 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 209–220. American Mathematical Society, 1998.

145. G. Winkler. *Image analysis, random fields and dynamic Monte-Carlo methods*. Springer-Verlag, Berlin, 1995.
146. W. Woess. *Random walks on infinite graphs and groups*. Cambridge University Press, 2000.
147. B. Ycart. The philosophers' process : an ergodic reversible nearest particle system. *Ann. Appl. Probab.*, 3(2):356–363, 1993.
148. B. Ycart. Cutoff for samples of Markov chains. *ESAIM Probability-Statistics*, 3:89–107, 1999.
149. B. Ycart. Stopping tests for Monte-Carlo Markov chain methods. *Meth. and Comp. in Appl. Probab.*, 2(1):23–36, 2000.
150. B. Ycart. Cutoff for Markov chains: some examples and applications. In E. Goles and S. Martínez, eds., *Complex Systems*, pages 261–300. Kluwer, Dordrecht, 2001.

# Index

- aide en ligne, 223
- algorithme
  - analyse d', 5
  - brutal, 54
  - d'Euler-Maruyama, 5
  - d'inversion, 20
  - de Box-Muller, 40
  - de décomposition, 31, 192
  - de Heun, 5
  - de Metropolis, 122, 126, 138, 139, 201
  - de Propp et Wilson, 120
  - de recuit simulé, 137, 139, 142, 145, 245, 246
  - de rejet, 23, 28, 30, 126
  - génétique, 147, 153, 154
  - itératif, 2, 72
  - markovien, 71
  - MOSES, 155
  - polaire, 39, 89
  - randomisé, 5
- ans, 223
- apropos, 223
- argn, 226
  
- Black et Scholes, 7, 88, 239, 240, 243, 245
- Boltzmann, 101, 138
- Box-Muller, 40
- break, 226
- brownien, 73, 85, 87, 94, 101, 175
  
- case, 225
- cdf\*, 233
- chaîne
  - de Markov, 1, 2, 71, 87, 89, 132, 167, 174
  - harmonisée, 180, 182, 248
  - incluse, 176, 182, 248
  
- champ moyen, 203, 205
- Chapmann-Kolmogorov, 189
- chdir, 223
- chromosome, 148
- clear, 223, 225
- commentaire, 222
- complexité, 53
- corr, 232
- corrélation, 232
- covariance, 232
- croisement, 148, 150
  
- définition d'une fonction, 224
- diagramme de transition, 73
- Dirichlet, 90
- discrétisation, 83
- disp, 226
- distance
  - du khi-deux, 129
  - en variation totale, 129
  
- échantillon, 36
- échantillonnage par importance, 29, 44
- échelle de temps, 103
- else, 225
- elseif, 225
- end, 225
- énergie, 138, 144
- entropie, 137
- équation
  - de Black et Scholes, 88, 239, 245
  - de Boltzmann, 101
  - de Feynman-Kac, 92, 93, 96
  - de Fokker-Planck, 92, 93, 96, 100
  - de Kolmogorov, 94, 189, 205
  - de McKean-Vlasov, 99
  - différentielle stochastique, 86, 100
- error, 226

- état absorbant, 81, 175
- Euler-Maruyama, 5, 87, 239, 245
- `exec`, 224
- exponentielle de matrice, 187
- Feynman-Kac, 92, 93, 96
- fichiers
  - de commandes, 224
  - de fonctions, 224
  - de sauvegarde, 223
- file
  - d'attente, 7
  - G/G/1, 250
  - M/M/ $\infty$ , 185
  - M/M/s, 186
  - M/M/1, 184, 192, 195, 197
  - M/M/1/R, 249
- Fisher-Wright, 241
- flot, 241
- Fokker-Planck, 92, 93, 96, 100
- fonction
  - de répartition, 233
  - quantile, 233
- `for`, 225
- fréquences, 227, 231
- générateur, 91, 93, 102, 175, 179, 204, 206, 208
- générateur pseudo-aléatoire, 4, 9, 12, 50, 227, 228, 230
- `getf`, 224
- graine, 51
- `grand`, 230
- hauteur de communication, 140
- `help`, 223
- Heun-Milshtein, 5, 89, 245
- Hille-Yosida, 93
- `hist3d`, 231
- histogramme, 23, 230
  - en dimension trois, 231
- `histplot`, 230
- horloge interne, 180
- `if`, 225
- importance
  - échantillonnage par, 29, 44
- intégrale, 27, 43
- intervalle de confiance, 14, 16, 132
- laplacien, 84, 207
- loi
  - binomiale, 16, 19, 158, 172
  - conditionnelle, 18
  - de Bernoulli, 16
  - de Boltzmann, 138
  - de Poisson, 21
  - exponentielle, 20, 40, 167, 177
  - géométrique, 18, 25, 29, 37, 167, 172, 177
  - log-normale, 244
  - normale, 14, 38, 89
  - normale multidimensionnelle, 42
  - uniforme, 23
- marche aléatoire
  - sur un groupe, 72
  - symétrique, 74, 122, 139
- Markov, 4, 71, 87, 89, 91, 132, 167
- matrice
  - de sélection, 122, 139
  - de transition, 73, 79, 126, 150, 151
  - exponentielle de, 187
- McKean-Vlasov, 99
- `mean`, 232
- `median`, 232
- mesure
  - de Gibbs, 138, 142, 200
  - réversible, 121, 201
  - stationnaire, 116, 149
- méthode
  - d'Euler-Maruyama, 5, 87, 239, 245
  - d'inversion, 5, 20
  - de décomposition, 5, 31, 192, 237
  - de Heun-Milshtein, 5, 89, 245
  - de Monte-Carlo, 1, 3, 4, 27, 43, 236
  - de Propp et Wilson, 120
  - de rejet, 5, 23, 28, 30, 126, 236
  - MCMC, 115, 124
  - particulière, 97, 102
  - polaire, 38, 39
- Monte-Carlo, 1, 3, 4, 27
- mouvement brownien, 73, 85, 87, 94, 101, 175
- mutation, 148, 150
- nuage de points, 231
- `ones`, 222

- opérateur différentiel, 91
- Ornstein-Uhlenbeck, 241, 243
  
- palier, 141, 152
- param3d1, 232
- pause, 226
- permutation aléatoire, 35
- plot2d, 231
- polaire, 39
- population, 148, 153, 157
- précision, 1, 16, 27, 44, 88, 238
- problème
  - de Cauchy, 86
  - de Dirichlet, 83, 85, 90, 91
  - de la chaleur, 83
  - différentiel, 82
  - du voyageur de commerce, 145, 154, 158, 245, 246
- processus
  - à accroissements indépendants, 171
  - d'élection, 200, 253, 254
  - d'exclusion, 206
  - d'Ornstein-Uhlenbeck, 241, 243
  - de contact, 200, 252, 253
  - de diffusion, 85, 91, 98
  - de Fisher-Wright, 241
  - de Ising, 200
  - de Markov, 91, 175
  - de naissance et de mort, 184, 185
  - de Poisson, 170, 173, 180
  - de saut, 101, 102, 167, 174
  - des philosophes, 200
- programmation, 225
- propagation du chaos, 97
- pwd, 223
  
- Random, 9, 12
- recuit simulé, 137, 245, 246
- réduction de la variance, 3, 15, 46, 90
- répertoire, 223
- réseau
  - de files d'attente, 175
  - de Jackson, 193
  - de Petri, 175, 196
- resume, 226
- return, 226
  
- schéma de température, 139, 152, 157
- select, 225
  
- sélection, 148, 150
- semi-groupe, 92
- simulation, 12, 16, 17, 51
  - d'un automate binaire, 182
  - d'un processus de contact, 202
  - d'un processus de diffusion, 86, 101, 239
  - d'un processus de saut, 174, 248
  - d'un processus produit, 191
  - d'un réseau de Jackson, 194
  - d'un réseau de Petri, 197
  - d'un système de spin, 201, 252
  - d'une chaîne de Markov, 71, 76, 87, 89
  - d'une file G/G/1, 250
  - d'une file M/M/∞, 185
  - d'une file M/M/s, 186
  - d'une file M/M/1, 184, 192, 195, 197, 251
  - d'une file M/M/1/R, 249
  - d'une loi binomiale, 158
  - d'une loi de Poisson, 21
  - d'une loi discrète, 21
  - d'une loi exponentielle, 20
  - d'une loi normale, 14, 38, 89
  - d'une loi normale multidimensionnelle, 42
  - d'une loi uniforme, 23, 25, 125
  - de lois conditionnelles, 18
  - des lois classiques, 230
  - exacte d'une mesure stationnaire, 117
- simuler, 12
- spectre, 127, 141
- stables d'un graphe, 123, 135–137, 202
- st\_deviation, 232
- suite
  - aléatoire, 52, 54
  - complexité d'une, 53
  - déterministe, 55
  - de van der Corput, 56
  - uniforme, 48, 49, 52
- suppression de l'affichage, 222
- système
  - de particules interactives, 199
  - de spin, 199, 203, 252
  - linéaire, 78
  
- taux
  - de sortie, 177

– de transition, 177, 179, 199  
température, 138, 149, 200  
– schéma de, 139, 152, 157  
temps d'atteinte, 133  
**then**, 225  
théorème  
– central limite, 2, 12, 131, 181  
– de Hille-Yosida, 93  
**type**, 226  
**typeof**, 226

variables  
– globales, 225  
– locales, 225  
variables négativement corrélées, 45  
voyageur de commerce, 145, 154, 158,  
245, 246  
**warning**, 226  
**while**, 225  
**who**, 223  
**whos**, 223