

TP DE PROCESSUS STOCHASTIQUES –  
SIMULATION STOCHASTIQUE ET MÉTHODE DE MONTE-CARLO

---

La formule de Feynman-Kac établit un lien entre les équations aux dérivées partielles paraboliques et les processus stochastiques. Plus précisément, la solution de l'équation aux dérivées partielles

$$\partial_t u = \frac{\sigma^2(t, x)}{2} \partial_{x,x} u + \mu(t, x) \partial_x u + V(t, x)u + f(x, t)$$

avec condition initiale  $u(0, x) = \psi(x)$  peut être représentée comme

$$u(t, x) = \mathbb{E}_x \left( \int_0^t e^{\int_0^s V(r, X_r) dr} f(s, X_s) ds + e^{\int_0^t V(s, X_s) ds} \psi(X_t) \right),$$

où le processus  $(X_t, t \geq 0)$  est la solution de l'équation différentielle stochastique

$$dX_t = \mu(t, X_t)dt + \sigma(t, X_t)dW_t,$$

avec  $X_0 = x$   $\mathbb{P}_x$ -p.s.

Cette représentation d'EDP sous forme d'espérance d'une variable aléatoire permet ainsi d'envisager l'utilisation de méthodes de Monte-Carlo pour résoudre numériquement ce type d'équations. Notons que si la formule ci-dessous est exprimée en dimension 1, elle reste valable en toute dimension. On se propose ici d'étudier quelques méthodes de résolution d'EDP basées sur cette observation.

**Exercice 1** (Exemple d'application). On considère l'équation aux dérivées partielles

$$\partial_t u = \frac{1}{2} \partial_{x,x} u - xu,$$

satisfaisant les conditions  $u(0, x) = 1$  et  $u(t, 0) = 0$ . Soit  $(B_t, t \geq 0)$  un mouvement Brownien standard, issu de  $x$  sous la loi  $\mathbb{P}_x$ .

1. Montrer que la solution de l'EDP  $\partial_t v = \frac{1}{2} \partial_{x,x} v - xv$  avec condition initiale  $v(0, x) = \mathbf{1}_{\{x > 0\}}$  est donnée par

$$v(t, x) = \mathbb{E}_x \left( e^{-\int_0^t B_s ds} \mathbf{1}_{\{B_t \geq 0\}} \right).$$

2. Grâce à la méthode de Monte-Carlo, calculez une approximation de  $v$ . Tracez  $x \mapsto v(1, x)$ ,  $x \mapsto v(2, x)$  et  $x \mapsto v(3, x)$ .

3. On observe que  $(B_t, \int_0^t B_s ds)$  est un vecteur gaussien dont on calculera la covariance. Grâce à cette expression, proposez une formule pour  $v$  faisant intervenir la fonction de répartition  $\Phi$  de la loi normale.

4. La solution  $u$  de l'EDP d'intérêt peut s'écrire

$$u(t, x) = \mathbb{E}_x \left( e^{-\int_0^t B_s ds} \mathbf{1}_{\{\forall s \leq t, B_s \geq 0\}} \right).$$

Modifiez votre programme pour déterminer une approximation de  $u$ .

**Exercice 2** (Un deuxième exemple). On considère la solution de l'équation différentielle stochastique

$$dX_t = -X_t dt + dW_t.$$

On pose  $u(t, x) = \mathbb{E} \left( e^{\int_0^t X_s ds} \right)$ .

1. Déterminer l'équation aux dérivées partielles satisfaites par  $u$ .
2. Grâce à la méthode de Monte-Carlo, construire une fonction permettant d'approximer  $(u(t, x), t \geq 0, x \in \mathbb{R})$ .
3. Utiliser cette fonction pour tracer  $v(1, x)$ ,  $v(2, x)$ ,  $v(3, x)$  pour  $x \in [-10, 10]$ .

**Exercice 3** (Une EDP non-linéaire). On considère l'équation aux dérivées partielles

$$\partial_t u = \frac{1}{2} \partial_{x,x} u + u(1 - u)$$

avec condition initiale  $u(0, x) = \mathbf{1}_{\{x < 0\}}$ .

1. Déterminer la solution de cette EDP issue de la condition initiale  $u(0, x) = 1/2$  sous la forme  $u(t, x) = f(t)$ .
2. La solution de cette EDP peut être représentée sous forme de Feynman-Kac comme

$$u(t, x) = \mathbb{E}_x \left( e^{\int_0^t (1 - u(s, B_s)) ds} \mathbf{1}_{\{B_t < 0\}} \right).$$

Proposer une méthode itérative pour déterminer une approximation de cette fonction  $u$  par méthode de Monte-Carlo, en utilisant des simulations successives de mouvements browniens pour obtenir une estimation de plus en plus précise de  $u$ . Implémenter cette méthode et tracer  $u(t, x)$  pour  $t = 1, 2, 3, 4, 5$  et  $x \in [-5, 5]$ .

3. On peut également construire  $u(t, x)$  comme  $\mathbb{P}(M_t \leq x)$ , où  $M_t$  est la position au temps  $t$  de la plus grande particule d'un système de particules se déplaçant indépendamment les unes des autres selon des mouvements browniens, et donnant naissance au bout d'un temps exponentiel à deux nouvelles particules indépendantes. Simulez ce processus et utilisez-le pour obtenir une autre approximation de  $u$ . Comparer ces deux algorithmes.