

Barrages et Risques d'Inondation

Zian Chen & Ruikai Chen

31 mai 2024

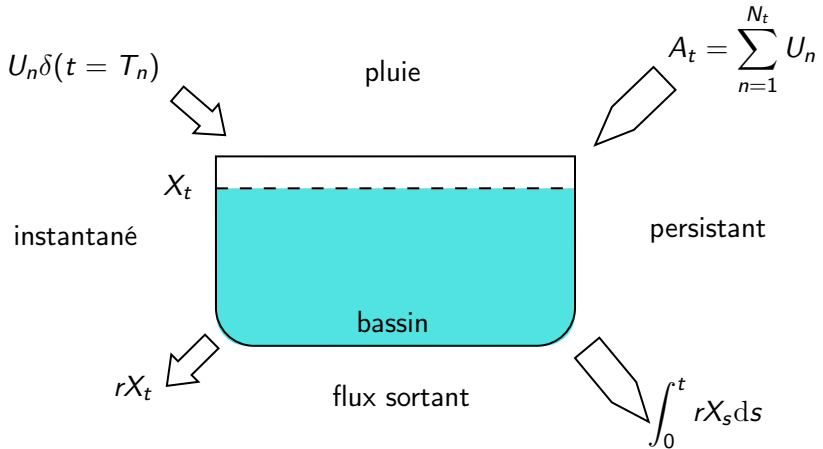
Sommaire

- 1 Introduction
- 2 Risque d'inondation
- 3 Simulation numérique

Introduction

Un barrage est un ouvrage artificiel ou naturel, établi en travers du lit d'un cours d'eau, retenant ou pouvant retenir de l'eau. Les barrages ont plusieurs fonctions, par exemple : la régulation de cours d'eau ; l'irrigation des cultures ; la lutte contre les incendies, etc. La rareté des accidents est le résultat d'efforts attentifs depuis un siècle.

Modélisation



Modélisation

- X_t : le volume de lac au temps t
- x_0 : le volume initial
- r : le taux de lâchers d'eau du lac
- A_t : le volume total de pluie au temps t

Modélisation

- X_t : le volume de lac au temps t
- x_0 : le volume initial
- r : le taux de lâchers d'eau du lac
- A_t : le volume total de pluie au temps t

On peut avoir l'équation d'évolution du volume d'eau

$$X_t = x_0 + A_t - \int_0^t rX_s ds.$$

Modélisation de A_t

On suppose que A_t est déterminé par des chutes de pluie intenses et de courte durée.

Modélisation de A_t

On suppose que A_t est déterminé par des chutes de pluie intenses et de courte durée.

Autrement dit, A_t est un processus de Poisson composé

$$A_t = \sum_{n=1}^{N_t} U_n,$$

Modélisation de A_t

On suppose que A_t est déterminé par des chutes de pluie intenses et de courte durée.

Autrement dit, A_t est un processus de Poisson composé

$$A_t = \sum_{n=1}^{N_t} U_n,$$

où

- N_t est un processus de Poisson de taux λ , qui décrit l'arrivée de pluie,
- U_n suit la loi ν de paramètres δ_1 et δ_2 , qui modélise séparément des pluies de grande et petite intensité :

$$\nu(u) = b\delta_1 \exp(-\delta_1 u)1_{u \geq 0} + (1 - b)\delta_2 \exp(-\delta_2 u)1_{u \geq 0}.$$

Solution de X_t

Si on représente le processus de Poisson composé par une série de sauts, i.e.

$$N_t = \begin{cases} 0 & \text{pour } t \in [0, T_1[, \\ 1 & \text{pour } t \in [T_1, T_2[, \\ \vdots & \vdots \end{cases}$$

Solution de X_t

Si on représente le processus de Poisson composé par une série de sauts, i.e.

$$N_t = \begin{cases} 0 & \text{pour } t \in [0, T_1[, \\ 1 & \text{pour } t \in [T_1, T_2[, \\ \vdots & \vdots \end{cases}$$

la solution est alors donnée par

$$X_t = \exp(-rt) \left(\int_0^t \exp(-rs) dA_s + x_0 \right),$$

où

$$dA_s = \begin{cases} U_n \delta(s = T_n) & \text{si } s \in \{T_1, T_2, \dots\} \\ 0 & \text{sinon} \end{cases}$$

au sens de distribution.

Solution de X_t

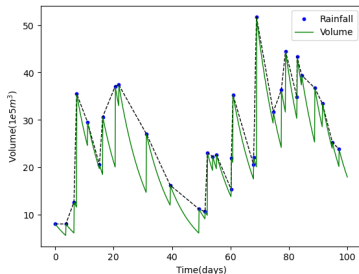
Autrement dit, on peut avoir

$$X_t = \begin{cases} x_0 \exp(-rt) & t \in [0, T_1[, \\ \exp(-rt)(x_0 + \exp(rT_1)U_1) & t \in [T_1, T_2[, \\ \exp(-rt)(x_0 + \exp(rT_1)U_1 + \exp(rT_2)U_2) & t \in [T_2, T_3[, \\ \vdots & \vdots \end{cases}$$

Solution de X_t

Autrement dit, on peut avoir

$$X_t = \begin{cases} x_0 \exp(-rt) & t \in [0, T_1[, \\ \exp(-rt)(x_0 + \exp(rT_1)U_1) & t \in [T_1, T_2[, \\ \exp(-rt)(x_0 + \exp(rT_1)U_1 + \exp(rT_2)U_2) & t \in [T_2, T_3[, \\ \vdots & \vdots \end{cases}$$



Risque d'inondation

Problème du contrôle de risque d'inondation : trouver x^* t.q.

$$\mathbb{P}(X > x^*) \leq \alpha, \quad \alpha = 10^{-6}.$$

- Pour un certain T : $X = X_T$
- Pour le volume maximal pendant $[0, T]$: $X = \max_{0 \leq s \leq T} X_s$

Risque d'inondation

Problème du contrôle de risque d'inondation : trouver x^* t.q.

$$\mathbb{P}(X > x^*) \leq \alpha, \alpha = 10^{-6}.$$

- Pour un certain T : $X = X_T$
- Pour le volume maximal pendant $[0, T]$: $X = \max_{0 \leq s \leq T} X_s$

Trois approches :

- L'algorithme de Monte-Carlo naïf
- Importance Sampling
- L'algorithme de la dernière particule

Monte-Carlo naïf

Idée : simuler N trajectoires (X_i) de $X \implies$ prendre le quantile empirique

$$\hat{q}_{1-\alpha} = \inf\{x \in \mathbb{R} : \hat{F}_N(x) \geq 1 - \alpha\}, \quad \hat{F}_N(x) = \frac{1}{N} \sum_{i=1}^N 1_{X_i \leq x}.$$

Si on trie les valeurs de X_i : $X_{(1,N)} \leq X_{(2,N)} \leq \dots \leq X_{(N,N)}$,

$$\hat{q}_{1-\alpha} = X_{(\lceil N(1-\alpha) \rceil, N)}.$$

Monte-Carlo naïf

Idée : simuler N trajectoires (X_i) de $X \implies$ prendre le quantile empirique

$$\hat{q}_{1-\alpha} = \inf\{x \in \mathbb{R} : \hat{F}_N(x) \geq 1 - \alpha\}, \quad \hat{F}_N(x) = \frac{1}{N} \sum_{i=1}^N 1_{X_i \leq x}.$$

Si on trie les valeurs de X_i : $X_{(1,N)} \leq X_{(2,N)} \leq \dots \leq X_{(N,N)}$,

$$\hat{q}_{1-\alpha} = X_{(\lceil N(1-\alpha) \rceil, N)}.$$

- Facile à implémenter : baseline pour les autres algorithmes
- Ne peut que traiter les cas où α n'est pas très petit

Importance Sampling

Idée : changer vers une loi qui est plus «dense» autour du $q_{1-\alpha}$.

Rappel : $A_t = \sum_{i=1}^{N_t} U_i$ est un processus de Poisson composé

Transformation de Esscher : Pour f mesurable, g fonction appliquée sur le trajectoire de A_t :

$$\mathbb{E}[g(A_t)] \approx \exp((\lambda_f - \lambda)T) \frac{1}{M} \sum_{i=1}^M g(A_t^{(i)}) \exp\left(-\sum_{i=1}^{N_t^{(i)}} f(U_i^{(i)})\right),$$

où $\lambda_f = \lambda \mathbb{E}[e^{f(U)}]$ et $\nu_f(du) = \frac{e^{f(u)} \nu(du)}{\mathbb{E}[e^{f(U)}]}$, $(A_t^{(i)})$ est une suite des processus de Poisson composé sous la mesure (λ_f, ν_f) .

Importance Sampling

But : estimer $q_{1-\alpha}$

$$\mathbb{E}[g(A_t)] \approx \exp((\lambda_f - \lambda)T) \frac{1}{M} \sum_{i=1}^M g(A_t^{(i)}) \exp \left(- \sum_{i=1}^{N_t^{(i)}} f(U_i^{(i)}) \right)$$

Soit $g(A_t) = 1_{X(A_t) > x^*}$, où

- $X(A_t) = \max_{0 \leq s \leq T} X_s(A_t)$ pour le volume maximal
- $X(A_t) = X_T(A_t)$ pour le volume en temps T

Il suffit de trouver x^* t.q. $\mathbb{E}[g(A_t)] = \alpha$.

Importance Sampling

$$\exp((\lambda_f - \lambda)T) \frac{1}{M} \sum_{i=1}^M 1_{X(A_t^{(i)}) > x^*} \exp \left(- \sum_{i=1}^{N_t^{(i)}} f(U_i^{(i)}) \right) \approx \alpha$$

Algorithm 1 Algorithme pour trouver x^*

- 1: $X_n \leftarrow$ list of $X(A_t^{(i)})$ for $i = 1, \dots, M$
 - 2: $V_n \leftarrow$ list of $\exp \left(- \sum_{i=1}^{N_t^{(i)}} f(U_i^{(i)}) \right)$ for $i = 1, \dots, M$
 - 3: $indices \leftarrow \text{argsort}(X_n)$; $idx \leftarrow 0$; $sum \leftarrow 0$
 - 4: **while** $sum < \alpha M \exp(-(\lambda_f - \lambda)T)$ & $idx < M$ **do**
 - 5: $sum \leftarrow sum + V_n[indices[idx]]$; $idx ++$
 - 6: $x^* \leftarrow V_n[indices[idx]]$
-

Importance Sampling

Question : comment choisir f ?

Choix naturel : $f(u) = \sigma u + \beta$

- $\sigma > 0$: augmenter à la fois la densité et la fréquence des pluies
- $\beta > 0$: augmenter la fréquence des pluies

On peut montrer qu'un choix candidat est $\beta = 0$, σ la solution de

$$-\sigma \phi'_A(\sigma) + \phi_A(\sigma) = \log(\alpha),$$

$$\phi_A(\sigma) = \lambda T(M_U(\sigma) - 1), \quad M_U(\sigma) = b \frac{\delta_1}{\delta_1 - \sigma} + (1 - b) \frac{\delta_2}{\delta_2 - \sigma}.$$

Algorithme de la dernière particule

Idée : renouveler la valeur minimale d'une suite de particules selon la loi de $(X|X > L)$ où $L = \min(X_i)$.

Algorithme de la dernière particule

Idee : renouveler la valeur minimale d'une suite de particules selon la loi de $(X|X > L)$ où $L = \min(X_i)$.

Initialisation : générer (X_1^1, \dots, X_n^1) i.i.d. selon la loi de X .

Par récurrence, étant donné (X_1^j, \dots, X_n^j) pour $j \geq 1$, on note $L_j = \min(X_1^j, \dots, X_n^j)$, et va construire $(X_1^{j+1}, \dots, X_n^{j+1})$.

Pour $1 \leq i \leq n$,

- si $X_i^j > L_j$, alors $X_i^{j+1} = X_i^j$,
- si $X_i^j = L_j$, alors on génère X_i^{j+1} selon la loi $(X|X > L_j)$ indépendamment.

Algorithme de la dernière particule

Règle d'arrêt : répéter $j_n = \left\lceil \frac{\log \alpha}{\log(1 - 1/n)} \right\rceil$ fois, le résultat est alors $\hat{q}_n = L_{j_n}$.

Algorithm 2 Algorithme pour obtenir \hat{q}_n

- 1: $X_N \leftarrow \text{list of } X(A_t^{(i)})_{1 \leq i \leq N}$
 - 2: **for** $_ \leftarrow 1$ **to** j_n **do**
 - 3: $L \leftarrow \min(X_N)$
 - 4: $\text{indices} \leftarrow \text{argwhere}(X_N == L)$
 - 5: **while** index **in** indices **do**
 - 6: $X_N[\text{index}] \leftarrow \text{get_a_new_one}(L)$
 - 7: $\hat{q}_n \leftarrow L$
-

Algorithme de la dernière particule

Question : comment générer un échantillon indépendant selon la loi $(X|X > L)$?

Algorithme de la dernière particule

Question : comment générer un échantillon indépendant selon la loi $(X|X > L)$?

Rappeler que X_t est obtenu du processus de Poisson composé A_t , il suffit alors de renouveler A_t par l'algorithme de coloriage :

Algorithme de la dernière particule

Question : comment générer un échantillon indépendant selon la loi $(X|X > L)$?

Rappeler que X_t est obtenu du processus de Poisson composé A_t , il suffit alors de renouveler A_t par l'algorithme de coloriage :

- garder chaque saut avec probabilité $p \in [0, 1]$ de manière indépendante des autres, obtenir et noter $N_t^{(p)}$,
- calculer $qU_i + (1 - q)V_i$ pour un point de saut gradé i , où $q \in [0, 1]$ et $V_i \sim \nu$ indépendant de U_i , obtenir $A_t^{(p)}$,
- simuler un autre processus composé indépendant de paramètre $(1 - p)\lambda$, noter $\bar{A}_t^{(1-p)}$, conclure en sommant $A_t^{(p)}$ et $\bar{A}_t^{(1-p)}$.

On veut répéter ce calcul plusieurs fois, en acceptant si le nouveau $\tilde{X} > L$, en refusant sinon.

Algorithme de la dernière particule

Algorithm 3 Algorithme pour obtenir un nouveau \tilde{A}_t

Require: N_t list of T_i , U list of U_i , p and q .

- 1: **for** $_ \leftarrow 1$ **to** M **do**
 - 2: $Unif \leftarrow$ Unif of length N_t ; $indices \leftarrow \text{argwhere}(Unif < p)$
 - 3: $N_t^{(p)}, U^{(p)} \leftarrow N_t[indices], U[indices]$
 - 4: $V \leftarrow$ list of ν of length $N_t^{(p)}$; $U^{(p)} \leftarrow qU^{(p)} + (1 - q)V$
 - 5: $\bar{N}_t^{(1-p)} \leftarrow$ Poisson distribution of parametre $(1 - p)\lambda$
 - 6: $\bar{U}^{(1-p)} \leftarrow$ list of ν of length $\bar{N}_t^{(1-p)}$
 - 7: $\tilde{N}_t, \tilde{U} \leftarrow N_t^{(p)} \cup \bar{N}_t^{(1-p)}, U^{(p)} \cup \bar{U}^{(1-p)}$
 - 8: **if** $X(\tilde{A}_t) > L$ **then**
 - 9: $N_t, U \leftarrow \tilde{N}_t, \tilde{U}$
-

Question : comment choisir p , q et M ?

Algorithme de la dernière particule

Point clé : devoir accepter un nombre suffisant de fois avant d'être considéré comme indépendant, sinon des erreurs se produiront, mais considérer aussi l'équilibre entre l'efficacité et l'exactitude.

Algorithme de la dernière particule

Point clé : devoir accepter un nombre suffisant de fois avant d'être considéré comme indépendant, sinon des erreurs se produiront, mais considérer aussi l'équilibre entre l'efficacité et l'exactitude.

Trois approches pour réaliser cet algorithme dans la pratique :

- fixer le nombre d'itérations M ,
- augmenter le nombre d'itérations M pendant l'exécution,
- fixer le nombre d'acceptations.

Algorithme de la dernière particule

Point clé : devoir accepter un nombre suffisant de fois avant d'être considéré comme indépendant, sinon des erreurs se produiront, mais considérer aussi l'équilibre entre l'efficacité et l'exactitude.

Trois approches pour réaliser cet algorithme dans la pratique :

- fixer le nombre d'itérations M ,
- augmenter le nombre d'itérations M pendant l'exécution,
- fixer le nombre d'acceptations.

Pour la sélection de p et q , qui dépend en fait de M et \tilde{M} , voir l'annexe du rapport.

Idée : montrer les intervalles de confiance à différents p et q lorsque α n'est pas trop petit, comparés aux intervalles de confiance de Monte-Carlo naïf.

Paramètres physiques

On met l'unité de temps comme 1 an et l'unité de volume comme $10^5 m^3$, et

- $r = 1$, $T = 1$, et il pleut $\lambda = 70$ fois par an,
- $\delta_1 = 0.07$ et $\delta_2 = 0.7$, où son unité est $(10^5 m^3)^{-1}$ et l'inverse de δ_i montre le volume moyen d'une seule pluie,
- la proportion $b = 0.1$, i.e. seule une sur dix est une grande pluie,
- le quantile $\alpha = 10^{-6}$.

Calcul de x_0

On veut que x_0 soit «l'état stationnaire» pour que ce soit compatible avec le processus physique, i.e. le flux d'eau entrant et sortant est presque équilibré pendant assez longtemps.

Calcul de x_0

On veut que x_0 soit «l'état stationnaire» pour que ce soit compatible avec le processus physique, i.e. le flux d'eau entrant et sortant est presque équilibré pendant assez longtemps.

Pour un petit temps Δt , le flux sortant est $-(rx_0)\Delta t$ alors que le flux entrant est $\mathbb{E}[\nu](\lambda\Delta t)$. Pour que x_0 soit l'état stationnaire, on a forcément que $-rx_0 + \lambda\mathbb{E}[\nu] = 0$, d'où

$$x_0 = \frac{\lambda}{r} \left(\frac{b}{\delta_1} + \frac{1-b}{\delta_2} \right) = 190.$$

Estimation du quantile $q_{0.99}$

But : Comparer les trois approches ; vérifier l'exactitude des implémentations

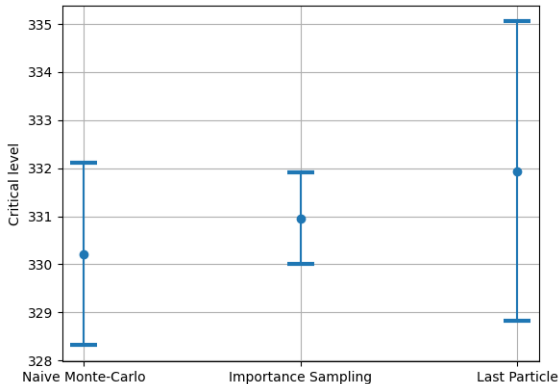


Figure – 10 expériences sont faits indépendamment, $\text{err} = \text{max} - \text{min}$

Estimation du quantile $q_{0.99}$

	time cost (s)	mean ($10^5 m^3$)	err
Monte-Carlo naïf	1.0	330.2	1.9
importance sampling	1.0	331.0	1.0
last particle	13.7	331.9	3.1

- les trois algorithmes donnent le même résultat
- importance sampling a une variance et un coût plus petits
- l'algorithme de la dernière particule nous permet d'obtenir une estimation pour la distribution après la volume critique x^* :
 $p(x|x > x^*)$

Estimation du quantile $q_{0.99}$

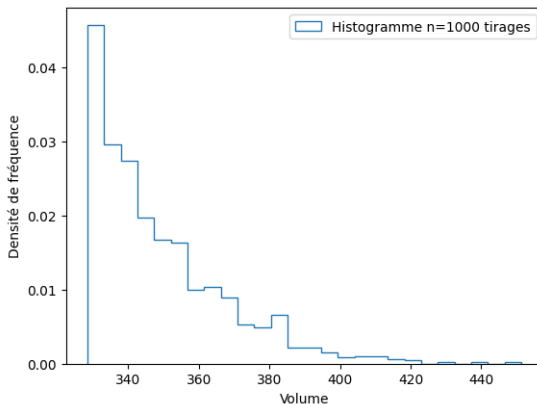
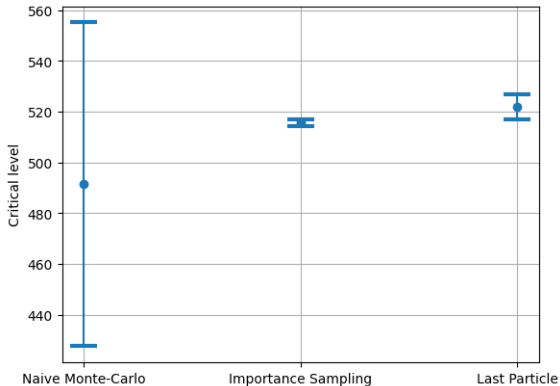


Figure – La distribution $p(x|x > q_{0.99})$ estimée par l'algorithme de la dernière particule.

Estimation du quantile $q_{1-10^{-6}}$

MC naïf, Importance sampling avec $N = 100,000$.

L'algorithme de la dernière particule avec $N = 1,000$, $M = 30$, on demande de plus qu'il y a au moins $M/2$ acceptations.



Conclusion

Nous avons réussi à simuler le risque d'inondation, qui est un effet un événement rare, avec deux approches différentes : l'importance sampling et l'algorithme de la dernière particule.

- l'importance sampling est très efficace pour ce problème et donne un résultat assez précis.
- l'algorithme de la dernière particule est plus flexible et nous permet aussi d'estimer la distribution dépassant le niveau critique.
- les choix des hyperparamètres dans ces deux algorithmes jouent un rôle crucial.

Travaux de future

- Etudier l'effet d'un $\beta \neq 0$. Nous avons utilisé une transformation de $f(x) = \sigma x + \beta$ avec $\beta = 0$ dans importance sampling. En fait, prendre un $\beta > 0$ a un impact similaire que prendre $\sigma > 0$.
- Comprendre plus l'impact de p, q dans l'algorithme de la dernière particule, surtout d'un point de vue théorique.