

Méthode numérique pour le contrôle optimal des processus Markoviens déterministes par morceaux

Benoîte de Saporta François Dufour

Université de Bordeaux et INRIA Bordeaux Sud-Ouest

[ANR-09-SEGI-004 Fautocoes](#)

Nancy, 17 mars 2011

Plan

- 1 Introduction
- 2 Arrêt optimal pour les EDS
- 3 Méthode numérique
- 4 Exemple
- 5 Conclusion et perspectives

Plan

- 1 Introduction
 - Motivation
 - Processus markoviens déterministes par morceaux
 - Objectif
- 2 Arrêt optimal pour les EDS
- 3 Méthode numérique
- 4 Exemple
- 5 Conclusion et perspectives

Maintenance préventive

Machine pouvant tomber en panne

Problème de maintenance

Trouver un **équilibre** optimal entre

- changer les pièces trop tôt/souvent
- ne rien faire jusqu'à la panne totale

Problème mathématique

- arrêt optimal
- contrôle impulsionnel

Maintenance préventive

Machine pouvant tomber en panne

Problème de maintenance

Trouver un **équilibre** optimal entre

- changer les pièces trop tôt/souvent
- ne rien faire jusqu'à la panne totale

Problème mathématique

- arrêt optimal
- contrôle impulsionnel

Collaboration avec Astrium

Structure de missile balistique stratégique soumis à corrosion



Collaboration avec Astrium

Structure de missile balistique stratégique soumis à corrosion

Profil d'emploi

Stockage dans 3 ambiances différentes

- 1 atelier
- 2 sous-marin nucléaire en mission
- 3 sous-marin en cale sèche

Exigence du sûreté très forte



Maîtriser l'évolution de l'épaisseur

Collaboration avec Astrium

Structure de missile balistique stratégique soumis à corrosion

Profil d'emploi

Stockage dans 3 ambiances différentes

- 1 atelier
- 2 sous-marin nucléaire en mission
- 3 sous-marin en cale sèche

Exigence du sûreté très forte



Maîtriser l'évolution de l'épaisseur

Collaboration avec Astrium

Structure de missile balistique stratégique soumis à corrosion

Profil d'emploi

Stockage dans 3 ambiances différentes

- 1 atelier
- 2 sous-marin nucléaire en mission
- 3 sous-marin en cale sèche

Exigence du sûreté très forte



Maîtriser l'évolution de l'épaisseur

Politique de Maintenance

Une seule intervention avant la rupture \Rightarrow remise à neuf de la structure

Optimisation de la maintenance : équilibre entre

- une maintenance trop précoce
- une maintenance trop tardive

Optimisation des marges

En phase de conception

- consolider les marges de dimensionnement par rapport aux spécifications
- assurer à 95% qu'aucune maintenance ne sera nécessaire avant la date objective contractuelle

Processus markoviens déterministes par morceaux

Davis (80's)

Classe générale de processus **hybrides** de type **non-diffusion** : trajectoire **déterministe** entre des sauts ponctuels **aléatoires**.

Applications

biologie, trafic internet, fiabilité dynamique, . . .

Dynamique

Processus hybride $X_t = (m_t, y_t)$

- mode **discret** $m_t \in \{1, 2, \dots, p\}$
- variable d'état **euclidienne** $y_t \in \mathbb{R}^d$

Caractéristiques locales dans le mode m

- E_m ouvert de \mathbb{R}^d , ∂E_m sa frontière \bar{E}_m sa fermeture
- **Flot** $\phi_m: \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ mouvement déterministe entre les sauts
- **Intensité** $\lambda_m: \bar{E}_m \rightarrow \mathbb{R}_+$ intensité des sauts aléatoires
- **Noyau markovien** Q_m sur $(\bar{E}_m, \mathcal{B}(\bar{E}_m))$ sélectionne la nouvelle position après les sauts

Deux types de sauts

- $t^*(m, y)$ temps de sortie déterministe de E_m

$$t^*(m, y) = \inf\{t > 0 : \phi_m(y, t) \in \partial E_m\}$$

- Loi du premier temps de saut T_1

$$\mathbb{P}_{(m,y)}(T_1 > t) = \begin{cases} e^{-\int_0^t \lambda_m(\phi_m(y,s)) ds} & \text{si } t < t^*(m, y) \\ 0 & \text{si } t \geq t^*(m, y) \end{cases}$$

Remarque importante

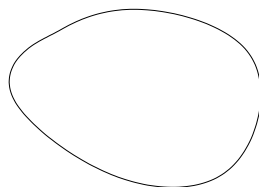
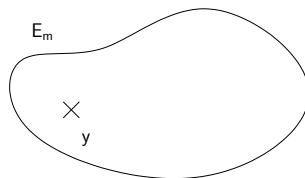
T_1 a une densité sur $[0, t^*(m, y)[$ mais a un **atome** en $t^*(m, y)$:

$$\mathbb{P}_{(m,y)}(T_1 = t^*(m, y)) > 0$$

Construction itérative

Point de départ

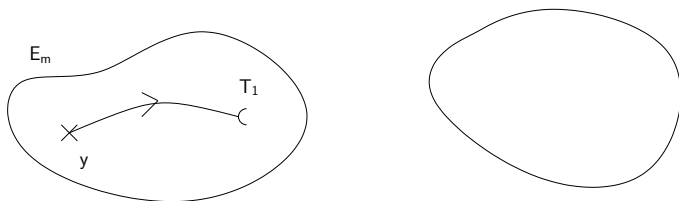
$$X_0 = Z_0 = (m, y)$$



Construction itérative

X_t suit le flot déterministe jusqu'au premier temps de saut $T_1 = S_1$

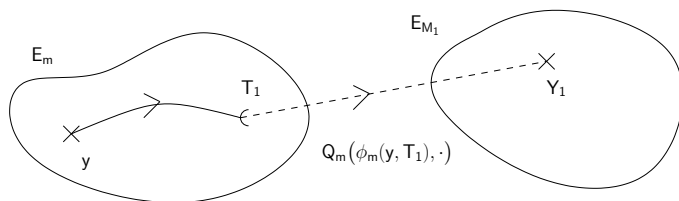
$$X_t = (m, \phi_m(y, t)), \quad t < T_1$$



Construction itérative

Position après saut $Z_1 = (M_1, Y_1)$ tirée suivant la loi

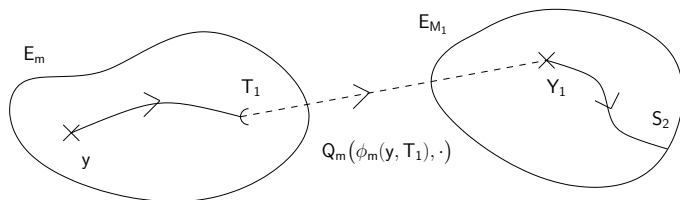
$$Q_m(\phi_m(y, T_1), \cdot)$$



Construction itérative

X_t suit le flot jusqu'au deuxième temps de saut $T_2 = T_1 + S_2$

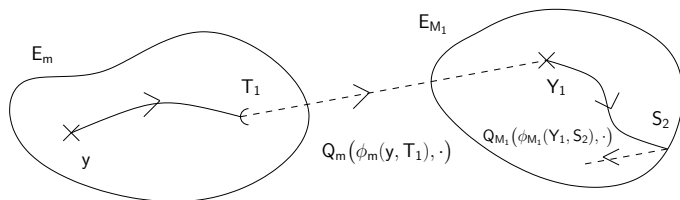
$$X_{T_1+t} = (M_1, \phi_{M_1}(Y_1, t)), \quad t < S_2$$



Construction itérative

Position après saut $Z_2 = (M_2, Y_2)$ tirée suivant la loi

$$Q_{M_1}(\phi_{M_1}(Y_1, S_2), \cdot) \dots$$



Chaîne de Markov sous-jacente

$\{X_t\}$ processus de Markov fort (M.H.A. Davis)

Chaîne de Markov sous-jacente

- Z_0 point de départ du processus, $S_0 = 0$, $S_1 = T_1$
- Z_n nouveaux mode et position après le n -ème saut,
 $S_n = T_n - T_{n-1}$, durée inter-saut

Proposition

(Z_n, S_n) est une chaîne de Markov en temps discret
Seule source d'aléa du PDP

Problème d'arrêt optimal

- Fonction de performance g
- Horizon aléatoire : N -ème temps de saut T_N du PDP
- \mathcal{M}_N ensemble des temps d'arrêt $\tau \leq T_N$

Problème d'arrêt optimal

- calculer la fonction valeur

$$V(x) = \sup_{\tau \in \mathcal{M}_N} \mathbb{E}_x[g(X_\tau)]$$

- trouver un temps d'arrêt (ε) -optimal τ^* qui atteint $V(x)(-\varepsilon)$

But de l'exposé

Objectif

Proposer une **méthode numérique** pour calculer

- la **fonction valeur**
- une **règle d'arrêt** ε -optimal

avec des **bornes de l'erreur** commise

Stratégie

Adapter la méthode numérique de quantification utilisée pour les **EDS**

Plan

- 1 Introduction
- 2 Arrêt optimal pour les EDS
 - Démarche
 - Quantification
 - Spécificités des PDP
- 3 Méthode numérique
- 4 Exemple
- 5 Conclusion et perspectives

Méthode numérique pour les diffusions

Y_t processus de diffusion à temps continu

- ① **discrétisation en temps** (schéma d'Euler) : $Y_k = Y_{k\Delta t}$ chaîne de Markov en temps discret à espace d'états continu
- ② **quantification** : remplacer Y_k par une variable aléatoire \hat{Y}_k à valeurs dans un espace d'état fini
- ③ replacer les **espérances conditionnelles** dans l'équation de programmation dynamique par des sommes finies

Hypothèses + régularité **Lipschitz** de la fonction de performance
 \implies **borne de l'erreur** d'approximation de la fonction valeur

Méthode numérique pour les diffusions

Y_t processus de diffusion à **temps continu**

- 1 **discrétisation en temps** (schéma d'Euler) : $Y_k = Y_{k\Delta t}$ chaîne de Markov en **temps discret** à espace d'états continu
- 2 **quantification** : remplacer Y_k par une variable aléatoire \hat{Y}_k à valeurs dans un espace d'état fini
- 3 replacer les **espérances conditionnelles** dans l'équation de programmation dynamique par des sommes finies

Hypothèses + régularité **Lipschitz** de la fonction de performance
 \implies **borne de l'erreur** d'approximation de la fonction valeur

Méthode numérique pour les diffusions

Y_t processus de diffusion à temps continu

- ① **discrétisation en temps** (schéma d'Euler) : $Y_k = Y_{k\Delta t}$ chaîne de Markov en temps discret à **espace d'états continu**
- ② **quantification** : remplacer Y_k par une variable aléatoire \hat{Y}_k à valeurs dans **un espace d'état fini**
- ③ replacer les **espérances conditionnelles** dans l'équation de programmation dynamique par des sommes finies

Hypothèses + régularité **Lipschitz** de la fonction de performance
 \implies **borne de l'erreur** d'approximation de la fonction valeur

Méthode numérique pour les diffusions

Y_t processus de diffusion à temps continu

- ① **discrétisation en temps** (schéma d'Euler) : $Y_k = Y_{k\Delta t}$ chaîne de Markov en temps discret à espace d'états continu
- ② **quantification** : remplacer Y_k par une variable aléatoire \hat{Y}_k à valeurs dans un espace d'état fini
- ③ replacer les **espérances conditionnelles** dans l'équation de programmation dynamique par des sommes finies

Hypothèses + régularité **Lipschitz** de la fonction de performance
 ⇒ **borne de l'erreur** d'approximation de la fonction valeur

Méthode numérique pour les diffusions

Y_t processus de diffusion à temps continu

- 1 **discrétisation en temps** (schéma d'Euler) : $Y_k = Y_{k\Delta t}$ chaîne de Markov en temps discret à espace d'états continu
- 2 **quantification** : remplacer Y_k par une variable aléatoire \hat{Y}_k à valeurs dans un espace d'état fini
- 3 replacer les **espérances conditionnelles** dans l'équation de programmation dynamique par des sommes finies

Hypothèses + régularité **Lipschitz** de la fonction de performance

⇒ **borne de l'erreur** d'approximation de la fonction valeur

Principe

Quantification d'une variable aléatoire $X \in L^p(\mathbb{R}^d)$

Approcher X par \hat{X} prenant un nombre **fini** de valeurs et telle que $\|X - \hat{X}\|_p$ soit **minimum**

- Construire une grille Γ à $|\Gamma| = K$ points
- Poser $\hat{X} = p_\Gamma(X)$ le projeté au plus proche voisin

Propriétés

Propriétés asymptotiques

Si $E[|X|^{p+\eta}] < +\infty$ pour un $\eta > 0$, alors

$$\lim_{K \rightarrow \infty} K^{p/d} \min_{|\Gamma| \leq K} \|X - \widehat{X}^\Gamma\|_p^p = J_{p,d} \left(\int |h|^{d/(d+p)}(u) du \right),$$

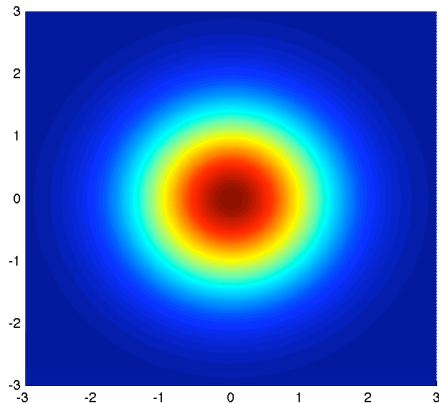
où $P_x(du) = h(u)\lambda_d(du) + \nu$ avec $\nu \perp \lambda_d$ et $J_{p,d}$ une constante.

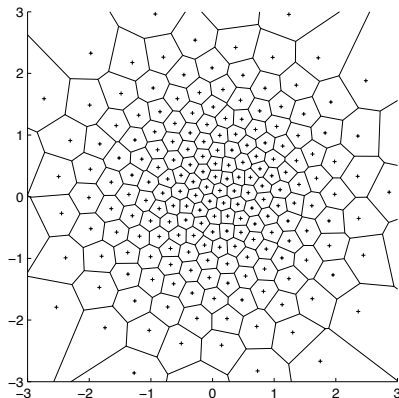
Calcul des grilles

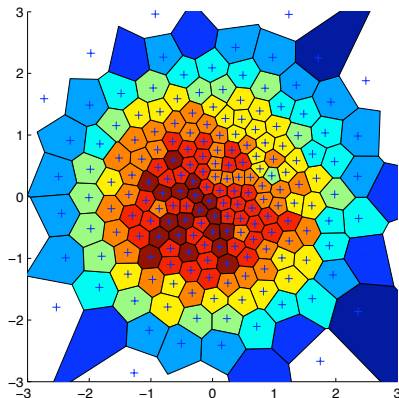
Algorithme de quantification

- **entrée** : simulateur de X
- **sortie** :
 - grille Γ
 - la **poids** de chaque point de Γ
 - les **probabilités de transition** pour la quantification de chaînes de Markov

Exemple $\mathcal{N}(0, I_2)$



Exemple $\mathcal{N}(0, I_2)$ 

Exemple $\mathcal{N}(0, I_2)$ 

Références

Quantification des EDS avec applications en

- [probabilités numériques](#) Pagès (98)
- [filtrage non linéaire](#) Pagès & Pham (05)
- [contrôle stochastique appliqué en finance](#) Bally & Pages (03); Pagès & Pham & Printemps (05); Bally & Pagès & Printemps (05)

Spécificités des PDP

- sauts à des instants aléatoires
- fonctions indicatrices dans l'équation de programmation dynamique

Solution

- utiliser la chaîne induite (Z_n, S_n)
- construire judicieusement les grilles en temps

Plan

- 1 Introduction
- 2 Arrêt optimal pour les EDS
- 3 Méthode numérique
 - Programmation dynamique
 - Approximation de la fonction valeur
 - Temps d'arrêt ϵ -optimal
- 4 Exemple
- 5 Conclusion et perspectives

Résultats théoriques

Equation de programmation dynamique rétrograde (U. Gugerli, 1986)

- $v_N = g$
- $v_n = L(v_{n+1}, g)$ pour $n \leq N - 1$

$$v_0(x) = \sup_{\tau \in \mathcal{M}_N} \mathbb{E}_x[g(X_\tau)] = V(x)$$

$L(w, g)(x)$

$$= \sup_{t \geq 0} \left[\int_0^{t \wedge t^*(x)} \lambda Qw(\phi(x, s)) e^{-\lambda(x, s)} ds + g(\phi(x, t \wedge t^*(x))) e^{-\lambda(x, t \wedge t^*(x))} \right]$$

$$= \vee \int_0^{t^*(x)} \lambda Qw(\phi(x, s)) e^{-\lambda(x, s)} ds + Qw(\phi(x, t^*(x))) e^{-\lambda(x, t^*(x))}$$

Interprétation probabiliste

Interpretation de l'opérateur L

$$L(w, g)(x)$$

$$= \sup_{t \geq 0} \left[\int_0^{t \wedge t^*(x)} \lambda Qw(\phi(x, s)) e^{-\lambda(x, s)} ds + g(\phi(x, t \wedge t^*(x))) e^{-\lambda(x, t \wedge t^*(x))} \right]$$

$$\vee \int_0^{t^*(x)} \lambda Qw(\phi(x, s)) e^{-\lambda(x, s)} ds + Qw(\phi(x, t^*(x))) e^{-\lambda(x, t^*(x))}$$

Interprétation probabiliste

Interpretation de l'opérateur L

$$\begin{aligned}
 L(w, g)(x) &= \sup_{u \leq t^*(x)} \left\{ \mathbb{E}_x \left[w(Z_1) \mathbf{1}_{\{S_1 < u\}} + g(\phi(x, u)) \mathbf{1}_{\{S_1 \geq u\}} \right] \right\} \\
 &\quad \vee \mathbb{E}_x [w(Z_1)]
 \end{aligned}$$

Interprétation probabiliste

Interpretation de l'opérateur L

$$\begin{aligned}
 L(w, g)(x) &= \sup_{u \leq t^*(Z_n)} \left\{ \mathbb{E} \left[w(Z_{n+1}) \mathbf{1}_{\{S_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{S_{n+1} \geq u\}} \mid Z_n = x \right] \right\} \\
 &\quad \vee \mathbb{E} [w(Z_{n+1}) \mid Z_n = x]
 \end{aligned}$$

Construction itérative

Equation de programmation dynamique rétrograde

- $v_N(Z_N) = g(Z_N)$
- $v_n(Z_n) = L(v_{n+1}, g)(Z_n)$ pour $n \leq N - 1$

$$v_0(Z_0) = \sup_{\tau \in \mathcal{M}_N} \mathbb{E}_x[g(X_\tau)]$$

$$v_n(Z_n) = L(v_{n+1}, g)(Z_n)$$

$$= \sup_{u \leq t^*(Z_n)} \left\{ \mathbb{E} \left[v_{n+1}(Z_{n+1}) \mathbf{1}_{\{S_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{S_{n+1} \geq u\}} \mid Z_n \right] \right\}$$

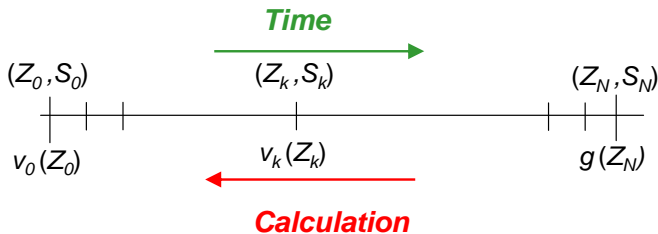
$$\vee \mathbb{E} [v_{n+1}(Z_{n+1}) \mid Z_n]$$

Calcul rétrograde

Equation de programmation dynamique rétrograde

- $v_N(Z_N) = g(Z_N)$
- $v_n(Z_n) = L(v_{n+1}, g)(Z_n)$ pour $n \leq N - 1$

$$v_0(Z_0) = \sup_{\tau \in \mathcal{M}_N} \mathbb{E}_x[g(X_\tau)]$$



Discretisation

Approximation de la fonction valeur

- $v_N(Z_N) = g(Z_N)$
- $v_n(Z_n) = L(v_{n+1}, g)(Z_n)$ for $n \leq N - 1$

$$\begin{aligned}
 & L(v_{n+1}, g)(Z_n) \\
 &= \sup_{u \leq t^*(Z_n)} \left\{ \mathbb{E} \left[v(Z_{n+1}) \mathbf{1}_{\{S_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{S_{n+1} \geq u\}} \mid Z_n \right] \right\} \\
 & \quad \vee \mathbb{E} [v(Z_{n+1}) \mid Z_n]
 \end{aligned}$$

Discrétisation

Discrétisation en temps

- $v_N(Z_N) = g(Z_N)$
- $v_n(Z_n) = L(v_{n+1}, g)(Z_n)$ for $n \leq N - 1$

$$\begin{aligned}
 & L_d(v_{n+1}, g)(Z_n) \\
 &= \max_{u \in G(Z_n)} \left\{ \mathbb{E} \left[v(Z_{n+1}) \mathbf{1}_{\{S_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{S_{n+1} \geq u\}} \mid Z_n \right] \right\} \\
 & \quad \vee \mathbb{E} [v(Z_{n+1}) \mid Z_n]
 \end{aligned}$$

Discretisation

Quantification

- $v_N(Z_N) = g(Z_N)$
- $v_n(Z_n) = L(v_{n+1}, g)(Z_n)$ for $n \leq N - 1$

$$\begin{aligned} & \widehat{L}_d(v_{n+1}, g)(Z_n) \\ &= \max_{u \in G(Z_n)} \left\{ \mathbb{E} \left[v(\widehat{Z}_{n+1}) \mathbf{1}_{\{\widehat{S}_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{\widehat{S}_{n+1} \geq u\}} \mid \widehat{Z}_n \right] \right\} \\ & \vee \mathbb{E} [v(Z_{n+1}) \mid \widehat{Z}_n] \end{aligned}$$

Discrétisation

Approximation de la fonction valeur

- $\hat{v}_N(\hat{Z}_N) = g(\hat{Z}_N)$
- $\hat{v}_n(\hat{Z}_n) = \hat{L}_d(\hat{v}_{n+1}, g)(\hat{Z}_n)$ for $n \leq N - 1$

$$\begin{aligned} & \hat{L}_d(v_{n+1}, g)(Z_n) \\ &= \max_{u \in G(Z_n)} \left\{ \mathbb{E} \left[v(\hat{Z}_{n+1}) \mathbf{1}_{\{\hat{S}_{n+1} < u\}} + g(\phi(Z_n, u)) \mathbf{1}_{\{\hat{S}_{n+1} \geq u\}} \mid \hat{Z}_n \right] \right\} \\ & \quad \vee \mathbb{E} [v(Z_{n+1}) \mid \hat{Z}_n] \end{aligned}$$

Vitesse de convergence

Théorème

Hypothèses de régularité Lipschitz sur ϕ , λ , Q , t^* and g

$$|v_0(x) - \hat{v}_0(x)| \leq C\sqrt{EQ}$$

C constante explicite,
 EQ erreur de quantification

$\sqrt{\cdot}$ due aux **indicatrices**

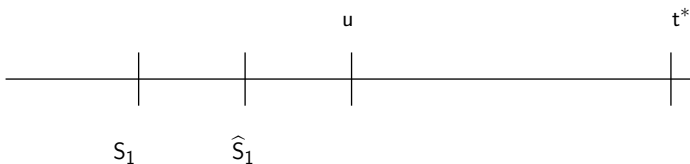
Indicatrices

Lemme

Soit η t.q. $\forall u \in G(\hat{Z}_0)$, $u + \eta < t^*(\hat{Z}_0)$

$$\left\| \max_{u \in G(\hat{Z}_0)} \mathbf{E}_{\hat{Z}_0} \left[\left| \mathbf{1}_{\{S_1 < u\}} - \mathbf{1}_{\{\hat{S}_1 < u\}} \right| \right] \right\|_2 \leq \frac{1}{\eta} \|S_1 - \hat{S}_1\|_2 + C\eta$$

Cas faciles



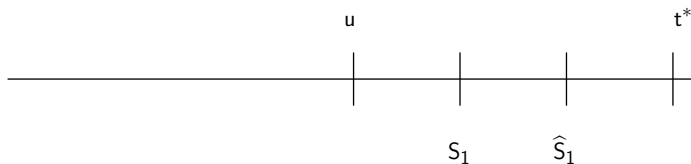
Indicatrices

Lemme

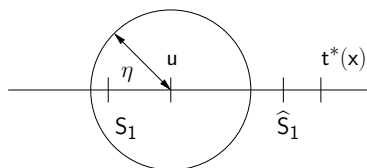
Soit η t.q. $\forall u \in G(\hat{Z}_0)$, $u + \eta < t^*(\hat{Z}_0)$

$$\left\| \max_{u \in G(\hat{Z}_0)} \mathbf{E}_{\hat{Z}_0} \left[\left| \mathbf{1}_{\{S_1 < u\}} - \mathbf{1}_{\{\hat{S}_1 < u\}} \right| \right] \right\|_2 \leq \frac{1}{\eta} \|S_1 - \hat{S}_1\|_2 + C\eta$$

Cas faciles



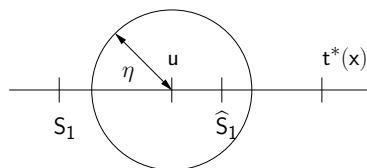
S_1 et \widehat{S}_1 sont de part et d'autre de u



$$|\mathbf{1}_{\{S_1 < u\}} - \mathbf{1}_{\{\widehat{S}_1 < u\}}| \leq \mathbf{1}_{\{|S_1 - u| \leq \eta\}} + \mathbf{1}_{\{|S_1 - \widehat{S}_1| > \eta\}}$$

$$\begin{aligned} \mathbf{E}_{\widehat{Z}_0} [\mathbf{1}_{\{u-\eta \leq S_1 \leq u+\eta\}}] &= \mathbf{E} \left[\mathbf{E} [\mathbf{1}_{\{u-\eta \leq S_1 \leq u+\eta\}} \mid Z_0] \mid \widehat{Z}_0 \right] \\ &= \int_{u-\eta}^{u+\eta} \lambda(\phi(z_0, u)) du \leq C\eta \end{aligned}$$

S_1 et \widehat{S}_1 sont de part et d'autre de u



$$|\mathbf{1}_{\{S_1 < u\}} - \mathbf{1}_{\{\widehat{S}_1 < u\}}| \leq \mathbf{1}_{\{|S_1 - u| \leq \eta\}} + \mathbf{1}_{\{|S_1 - \widehat{S}_1| > \eta\}}$$

$$\begin{aligned} \mathbf{E}_{\widehat{Z}_0} [\mathbf{1}_{\{u-\eta \leq S_1 \leq u+\eta\}}] &= \mathbf{E} \left[\mathbf{E} [\mathbf{1}_{\{u-\eta \leq S_1 \leq u+\eta\}} \mid Z_0] \mid \widehat{Z}_0 \right] \\ &= \int_{u-\eta}^{u+\eta} \lambda(\phi(z_0, u)) du \leq C\eta \end{aligned}$$

Temps d'arrêt optimal : τ^*

$$\mathbb{E}_x[g(X_{\tau^*})] = v_0(x) = \sup_{\tau \in \mathcal{M}_N} \mathbb{E}_x[g(X_\tau)]$$

Existence?

Temps d'arrêt ϵ -optimal : $\hat{\tau}$

$$v_0(x) - \epsilon \leq \mathbb{E}_x[g(X_{\hat{\tau}})] \leq v_0(x)$$

Temps d'arrêt optimal : τ^*

$$\mathbb{E}_x[g(X_{\tau^*})] = v_0(x) = \sup_{\tau \in \mathcal{M}_N} \mathbb{E}_x[g(X_\tau)]$$

Existence?

Temps d'arrêt ϵ -optimal : $\hat{\tau}$

$$v_0(x) - \epsilon \leq \mathbb{E}_x[g(X_{\hat{\tau}})] \leq v_0(x)$$

Règle d'arrêt calculable $\hat{\tau}$

- construction itérative **explicite**
- pas de calculs supplémentaires
- **vrai temps d'arrêt** pour la filtration du processus (X_t)

Règle d'arrêt calculable $\hat{\tau}$

- construction itérative **explicite**
- pas de calculs supplémentaires
- **vrai temps d'arrêt** pour la filtration du processus (X_t)

Optimalité

Théorème

Mêmes hypothèses

$$|v_0(x) - \mathbb{E}_x[g(X_{\hat{\tau}})]| \leq C_1 EV + C_2 \sqrt{EQ}$$

C_1, C_2 constantes explicites

EV erreur de la fonction valeur

EQ erreur de quantification

Autre approximation de la fonction valeur par [Monte Carlo](#)

Plan

- 1 Introduction
- 2 Arrêt optimal pour les EDS
- 3 Méthode numérique
- 4 Exemple**
 - Modélisation
 - Mise en oeuvre de la méthode
 - Résultats
- 5 Conclusion et perspectives

Dynamique du processus de dégradation

- Succession déterministe des ambiances : $1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \dots$
- Temps **aléatoire** passé dans l'ambiance i loi $Exp(\lambda_i)$
- Protection anti-corrosion initiale d'une durée **aléatoire** suivant une loi de Weibul
- Equation de la **perte d'épaisseur** dans l'ambiance i :

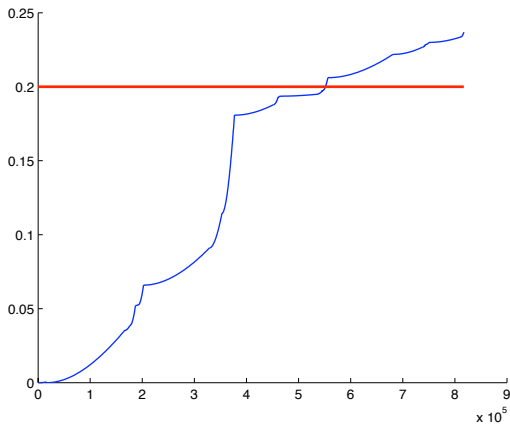
$$d_t = \rho_i \left(t - \eta_i + \eta_i \exp(-t/\eta_i) \right)$$

- ρ_i taux de corrosion stable **aléatoire** suivant une loi uniforme dépendant de l'ambiance i
- η_i durée de transition déterministe dans l'ambiance i .

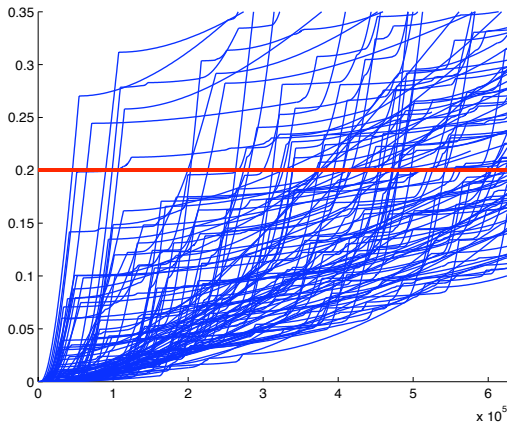
On dispose de valeurs numériques pour tous les paramètres.

Structure inutilisable si $d_t \geq 0.2mm$

Exemples de trajectoires simulées



Exemples de trajectoires simulées



Modélisation par un PDMP

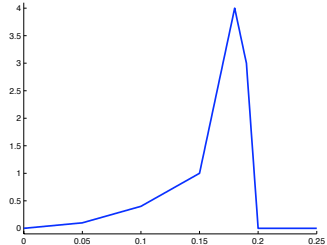
Processus Markovien déterministe
par morceaux

$$X_t = (m_t, d_t, \gamma_t, \rho_t)$$

mode m_t : ambiance à l'instant t

γ_t : reste de la protection
anti-corrosion à l'instant t

Fonction de performance g



Problème pratique

Echelles des différents paramètres

- taux de corrosion stable $\rho \sim 10^{-6}$
- temps moyen de séjour dans l'ambiance $2 \lambda_2^{-1} = 131\,400$ h

Loi uniforme sur
 $[0, 1] \times [0, 5000]$

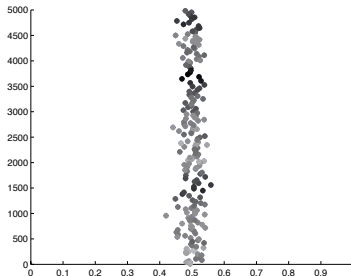
Problème pratique

Echelles des différents paramètres

- taux de corrosion stable $\rho \sim 10^{-6}$
- temps moyen de séjour dans l'ambiance $2 \lambda_2^{-1} = 131\,400$ h

Loi uniforme sur
 $[0, 1] \times [0, 5000]$

algorithme standard



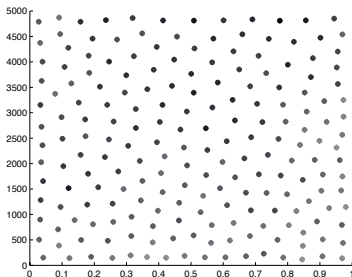
Problème pratique

Echelles des différents paramètres

- taux de corrosion stable $\rho \sim 10^{-6}$
- temps moyen de séjour dans l'ambiance $2 \lambda_2^{-1} = 131\,400 \text{ h}$

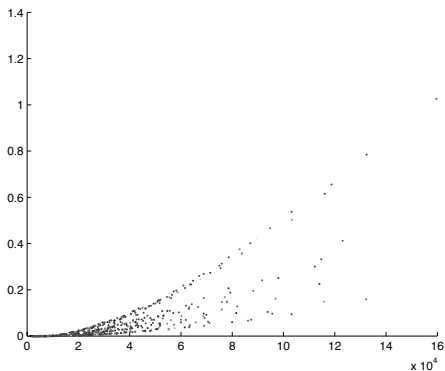
Loi uniforme sur
 $[0, 1] \times [0, 5000]$

algorithme pondéré



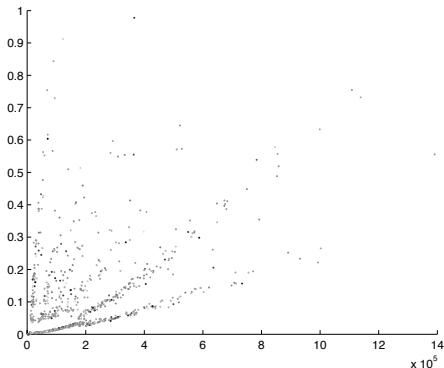
Grilles pour le processus de dégradation

Dans l'ambiance 2 après le 1er saut



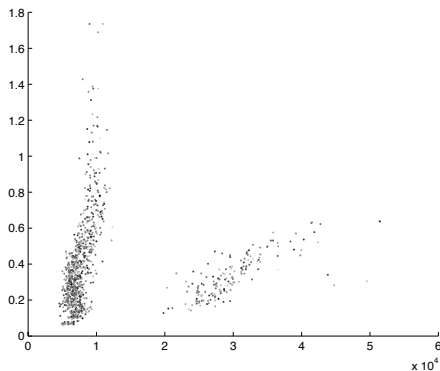
Grilles pour le processus de dégradation

Dans l'ambiance 3 après le 2ème saut

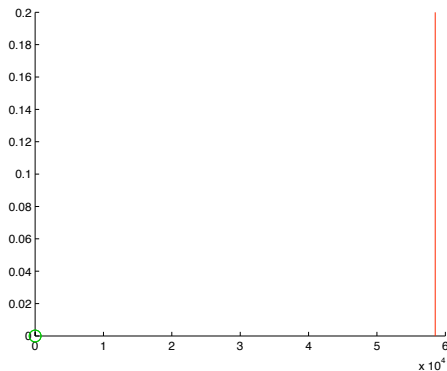


Grilles pour le processus de dégradation

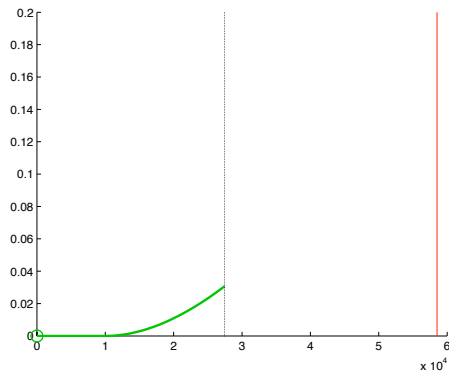
Dans l'ambiance 1 après le 15ème saut



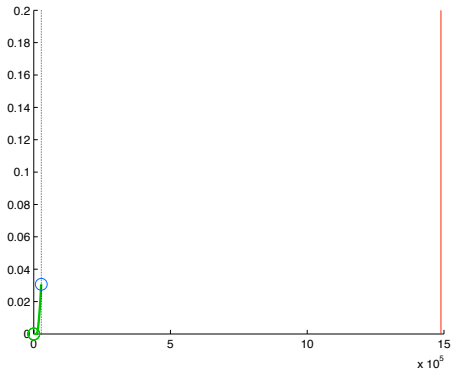
Règle d'arrêt itérative



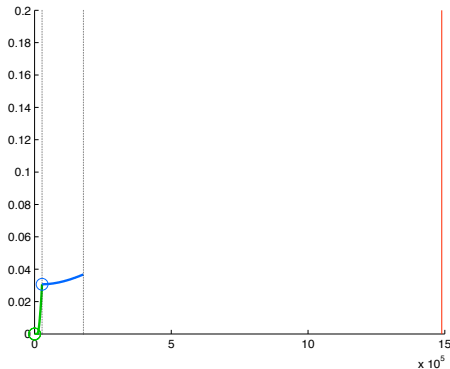
Règle d'arrêt itérative



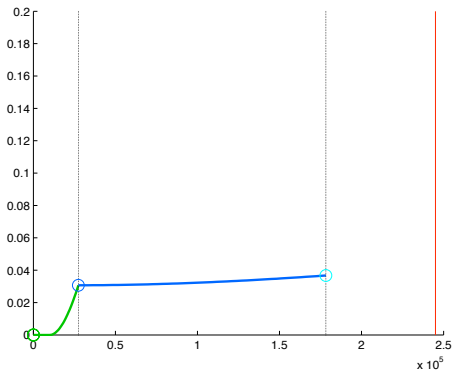
Règle d'arrêt itérative



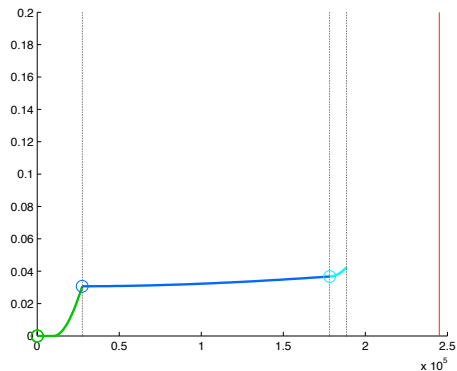
Règle d'arrêt itérative



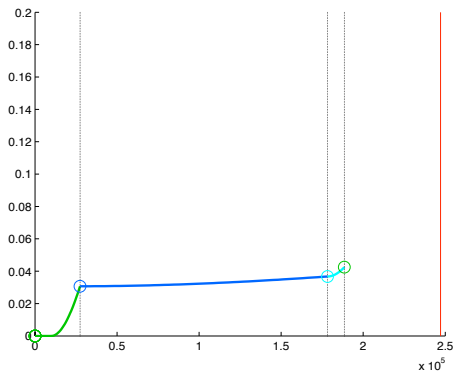
Règle d'arrêt itérative



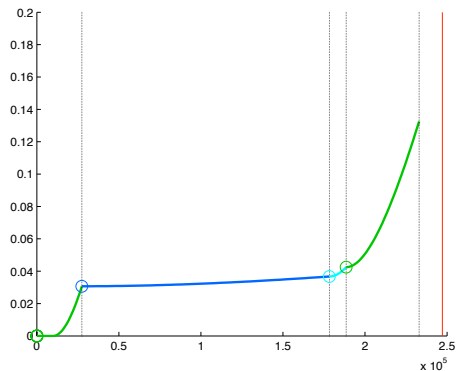
Règle d'arrêt itérative



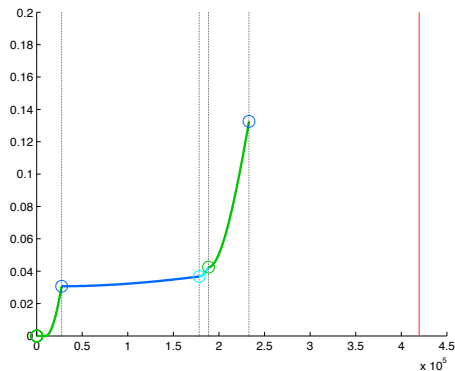
Règle d'arrêt itérative



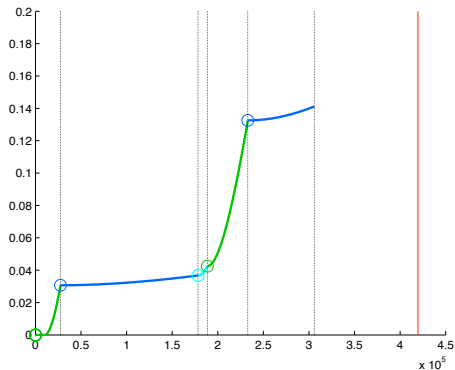
Règle d'arrêt itérative



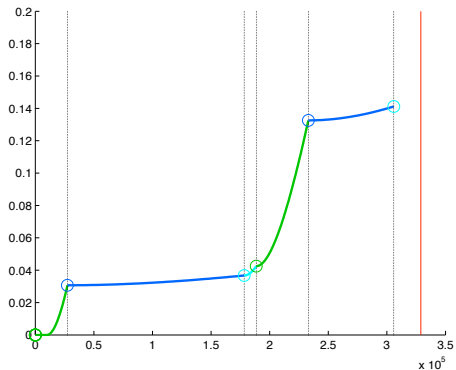
Règle d'arrêt itérative



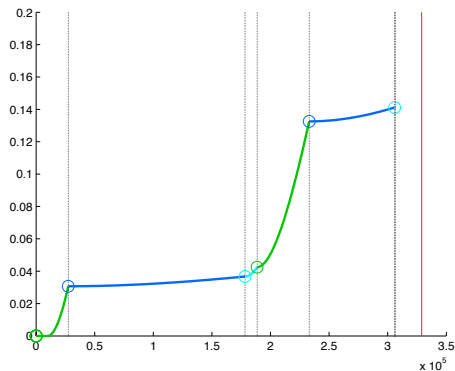
Règle d'arrêt itérative



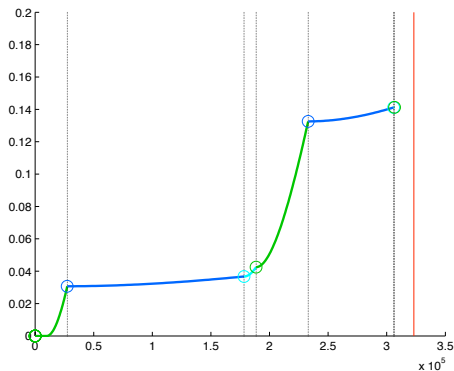
Règle d'arrêt itérative



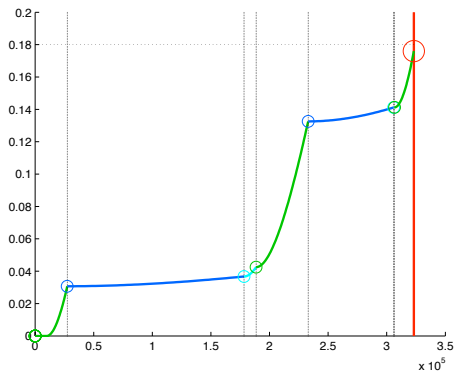
Règle d'arrêt itérative



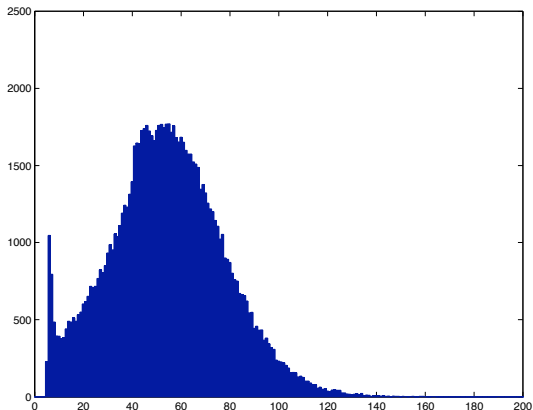
Règle d'arrêt itérative



Règle d'arrêt itérative



Optimisation des marges



Seuil	Probabilité
5 ans	0.0002
10 ans	0.0304
15 ans	0.0524
20 ans	0.0793
40 ans	0.2647
60 ans	0.6048
80 ans	0.8670
100 ans	0.9691
150 ans	0.9997

Calcul de la fonction valeur

Résultats numériques (vraie valeur : 4)

Nombre de points dans les grilles de quantification	Fonction valeur approchée	Fonction valeur par Monte Carlo
10	2.48	0.94
50	2.70	1.84
100	2.94	2.10
200	3.09	2.63
500	3.39	3.15
1000	3.56	3.43
2000	3.70	3.60
5000	3.82	3.73
8000	3.86	3.75

Plan

- 1 Introduction
- 2 Arrêt optimal pour les EDS
- 3 Méthode numérique
- 4 Exemple
- 5 Conclusion et perspectives
 - Points forts
 - Prochaine étape

Conclusion

- méthode numérique performante
- calculs lourds *off line* indépendants de la fonction coût
- calcul de la règle arrêt en temps réel
- règle d'arrêt adaptée à chaque trajectoire
- pas besoin de mesure en continu : seulement aux changements d'ambiance
- contexte mathématique rigoureux
 - algorithme général pour ce type de processus
 - preuve de convergence avec vitesse
- **EADS-Astrium** étude exploratoire en vue d'optimiser et de justifier les marges de conception

Contrôle impulsif

Maintenance avec réparations éventuellement partielles

⇒ **contrôle impulsif**

- choisir les instants d'intervention
- **choisir le nouveau point de départ du processus après intervention**

Définition du problème de contrôle impulsif

Stratégie $\mathcal{S} = (\tau_n, R_n)_{n \geq 1}$

- τ_n dates d'intervention
- R_n nouvelles positions après intervention à valeurs dans \mathbb{U} fini

Fonction valeur

$$\mathcal{J}^{\mathcal{S}}(x) = E_x^{\mathcal{S}} \left[\int_0^{\infty} e^{-\alpha s} f(Y_s) ds + \sum_{i=1}^{\infty} e^{-\alpha \tau_i} c(Y_{\tau_i}, Y_{\tau_i}^+) \right],$$

$$\mathcal{V}(x) = \inf_{\mathcal{S} \in \mathcal{S}} \mathcal{J}^{\mathcal{S}}(x).$$

Programmation dynamique

Costa, Davis, 1988

pour N assez grand

- $v_N = g$
- $v_n = L(Mv_{n+1}, v_{n+1})$ pour $n \leq N - 1$

$$v_0(x) \simeq \mathcal{V}(x)$$

Opérateur à itérer

$$\begin{aligned}
& L(Mw, w)(Z_n) \\
&= \left(\inf_{t \in \mathbb{R}_+} \mathbb{E} \left[F(Z_n, t) + e^{-\alpha S_{n+1}} w(Z_{n+1}) \mathbf{1}_{\{S_{n+1} < t \wedge t^*(Z_n)\}} \right. \right. \\
&\quad \left. \left. + e^{-\alpha t \wedge t^*(Z_n)} Mw(\phi(Z_n, t \wedge t^*(Z_n))) \mathbf{1}_{\{S_{n+1} \geq t \wedge t^*(Z_n)\}} \mid Z_n \right] \right) \\
&\quad \wedge \mathbb{E} \left[F(Z_n, t^*(Z_n)) + e^{-\alpha S_{n+1}} w(Z_{n+1}) \mid Z_n \right].
\end{aligned}$$

avec

$$\begin{aligned}
F(x, t) &= \int_0^{t \wedge t^*(x)} e^{-\alpha s - \Lambda(x, s)} f(\phi(x, s)) ds \\
Mw(x) &= \inf_{y \in \mathbb{U}} \{c(x, y) + w(y)\}
\end{aligned}$$

Solution

Plus de relation de récurrence sur les $v_n(Z_n)$

- calculer Mv_n à part
- utiliser **deux** séries de grilles de quantification

Théorème

Convergence du schéma numérique vers $\mathcal{V}(x)$ avec vitesse

Stratégie optimale?

aucun résultat de stratégie ϵ optimale

⇒ Sujet de post doc financé par l'ANR