Devoir Apprentissage par Renforcement
Algorithme Ada-FP et applications

Lucas BIECHY
Professeur : Joon KWON

GitHub repository

Institut Mathématique d'Orsay(IMO)
Université de Paris-Saclay

23 janvier 2024

Table des matières

1 Étude théorique d'un nouvel algorithme de points fixes

Soit $d \geq 1$, $F : \mathbb{R}^d \to \mathbb{R}^d$ une application admettant un point fixe $x_* \in \mathbb{R}^d$, $\eta > 0$ et $x_0 \in \mathbb{R}^d$. On définit l'algorithme Ada-FP comme suit :

$$x_{k+1} = x_k + \eta \frac{Fx_k - x_k}{\sqrt{\sum_{l=0}^{k} ||Fx_l - x_l||_2^2}}$$

avec la convention $\frac{0}{0} = 0$. On utilise pour tout $k \geq 0$ les notations suivantes :

$$\begin{aligned} u_k &= Fx_k - x_k \\ \eta_k &= \frac{\eta}{\sqrt{\sum_{l=0}^{k} ||Fx_l - x_l||_2^2}} \\ D_k &= \max_{l \in [0,k]} \frac{1}{2} ||x_l - x_*||_2^2 \end{aligned}$$

Preuve de η_k décroissant On suppose $\exists l \in [0, k]$ telque $Fx_l \neq x_l$

$$\begin{aligned} 0 \leq ||Fx_{k+1} - x_{k+1}||_2^2 \Leftrightarrow \sum_{l=0}^{k} ||Fx_l - x_l||_2^2 \leq \sum_{l=0}^{k+1} ||Fx_l - x_l||_2^2 \Leftrightarrow \sqrt{\sum_{l=0}^{k} ||Fx_l - x_l||_2^2} \leq \sqrt{\sum_{l=0}^{k+1} ||Fx_l - x_l||_2^2} \\ \Leftrightarrow \frac{\eta}{\sqrt{\sum_{l=0}^{k} ||Fx_l - x_l||_2^2}} \geq \frac{\eta}{\sqrt{\sum_{l=0}^{k+1} ||Fx_l - x_l||_2^2}} \Leftrightarrow \eta_k \geq \eta_{k+1} \end{aligned}$$

Ainsi l'algorithme Ada-FP se réécrie :

$$x_{k+1} = x_k + \eta_k u_k$$

Question 1

Soit $k \geq 0$,

$$\begin{aligned} ||x_{k+1} - x_*||_2^2 &= ||x_k + \eta_k u_k - x_*||_2^2 && \text{par définition de } x_{k+1} \\ &= ||x_k - x_*||_2^2 + ||\eta_k u_k||_2^2 + 2 < x_k - x_*, \eta_k u_k > \\ &= ||x_k - x_*||_2^2 + \eta_k^2 ||u_k||_2^2 + 2\eta_k u_k^T (x_k - x_*) \end{aligned}$$

Donc on a bien en particulier,

$$\forall k \geq 0, ||x_{k+1} - x_*||_2^2 \leq ||x_k - x_*||_2^2 + 2\eta_k u_k^T (x_k - x_*) + \eta_k^2 ||u_k||_2^2$$

Question 2

Soit $k \geq 0$, on suppose $\eta_k > 0$,

$$\begin{aligned}
||x_{k+1} - x_*||_2^2 &\leq ||x_k - x_*||_2^2 + 2\eta_k u_k^T(x_k - x_*) + \eta_k^2 ||u_k||_2^2 \\
&\iff \\
u_k^T(x_* - x_k) &\leq \frac{1}{2\eta_k}\left(||x_k - x_*||_2^2 - ||x_{k+1} - x_*||_2^2 + \eta_k^2 ||u_k||_2^2\right) \\
&\leq \frac{1}{2\eta_k}\left(||x_k - x_*||_2^2 - ||x_{k+1} - x_*||_2^2\right) + \frac{\eta_k ||u_k||_2^2}{2}
\end{aligned}$$

Ceci étant vrai pour tout $k \geq 0$, l'inégalité reste vrai pour la somme de chaque terme jusqu'à un k donné :

$$\begin{aligned}
\sum_{l=0}^{k} u_l^T(x_* - x_l) &\leq \sum_{l=0}^{k} \frac{1}{2\eta_l}\left(||x_l - x_*||_2^2 - ||x_{l+1} - x_*||_2^2\right) + \sum_{l=0}^{k} \frac{\eta_l ||u_l||_2^2}{2} \\
&\leq \frac{1}{\eta_0} \underbrace{\frac{1}{2}||x_0 - x_*||_2^2}_{\leq D_k} + \sum_{l=1}^{k} \left(\frac{1}{\eta_l} - \frac{1}{\eta_{l-1}}\right) \underbrace{\frac{1}{2}||x_l - x_*||_2^2}_{\leq D_k} \underbrace{- \frac{1}{2\eta_k}||x_{k+1} - x_*||_2^2}_{\leq 0} + \sum_{l=0}^{k} \frac{\eta_l ||u_l||_2^2}{2} \\
&\leq \frac{D_k}{\eta_0} + \sum_{l=1}^{k} \left(\frac{1}{\eta_l} - \frac{1}{\eta_{l-1}}\right) D_k + \sum_{l=0}^{k} \frac{\eta_l ||u_l||_2^2}{2} \\
&\leq \frac{D_k}{\eta_0} + \left(\frac{1}{\eta_k} - \frac{1}{\eta_0}\right) D_k + \sum_{l=0}^{k} \frac{\eta_l ||u_l||_2^2}{2} \\
&\leq \frac{D_k}{\eta_k} + \sum_{l=0}^{k} \frac{\eta_l ||u_l||_2^2}{2}
\end{aligned}$$

Question 3

a) Soit $(a_k)_{k\geq 0}$ une suite positive. Montrons que pour tout $k \geq 0$:

$$\sum_{l=0}^{k} \frac{a_l}{\sqrt{\sum_{m=0}^{l} a_m}} \leq 2\sqrt{\sum_{l=0}^{k} a_l}$$

Soit $(v_k)_{k\geq 0}$ une suite définit par $v_k := 2\sqrt{\sum_{l=0}^{k} a_l} - \sum_{l=0}^{k} \frac{a_l}{\sqrt{\sum_{m=0}^{l} a_m}}$ Montrer l'inégalité revient alors à montrer que $(v_k)_{k\geq 0}$ est une suite de terme positif.

Premier terme
$v_0 = 2\sqrt{a_0} - \frac{a_0}{\sqrt{a_0}} = \sqrt{a_0} \geq 0$ par définition de $(a_k)_{k\geq 0}$.

Croissance de la suite

$$\begin{aligned}
\frac{v_{k+1} - v_k}{\sqrt{\sum_{l=0}^{k+1} a_l}} &= \frac{1}{\sqrt{\sum_{l=0}^{k+1} a_l}} \left(2\sqrt{\sum_{l=0}^{k+1} a_l} - \sum_{l=0}^{k+1} \frac{a_l}{\sqrt{\sum_{m=0}^{l} a_m}} - 2\sqrt{\sum_{l=0}^{k} a_l} + \sum_{l=0}^{k} \frac{a_l}{\sqrt{\sum_{m=0}^{l} a_m}} \right) \\
&= \frac{1}{\sqrt{\sum_{l=0}^{k+1} a_l}} \left(2\sqrt{\sum_{l=0}^{k+1} a_l} - \frac{a_{k+1}}{\sqrt{\sum_{l=0}^{k+1} a_l}} - 2\sqrt{\sum_{l=0}^{k} a_l} \right) \\
&= \left(2 - \frac{a_{k+1}}{\sum_{l=0}^{k+1} a_l} - 2\frac{\sqrt{\sum_{l=0}^{k} a_l}}{\sqrt{\sum_{l=0}^{k+1} a_l}} \right) \\
&= \left(2 - 1 + \frac{\sum_{l=0}^{k} a_l}{\sum_{l=0}^{k+1} a_l} - 2\frac{\sqrt{\sum_{l=0}^{k} a_l}}{\sqrt{\sum_{l=0}^{k+1} a_l}} \right) \\
&= \left(1 - \frac{\sqrt{\sum_{l=0}^{k} a_l}}{\sqrt{\sum_{l=0}^{k+1} a_l}} \right)^2 \geq 0 \Rightarrow v_{k+1} - v_k \geq 0 \Rightarrow (v_k)_{k\geq 0} \text{est une suite croissante}
\end{aligned}$$

Conclusion
Ainsi $(v_k)_{k\geq 0}$ est une suite croissante avec un premier terme positif, donc $v_k \geq 0$ pour tout $k \geq 0$. La suite $(v_k)_{k\geq 0}$ étant positive, l'inégalité est directement démontrer pour tout $k \geq 0$.

b) Soit $k \geq 0$,

$$\begin{aligned}
\sum_{l=0}^{k} u_l^T (x_* - x_l) &\leq \frac{D_k}{\eta_k} + \sum_{l=0}^{k} \frac{\eta_l ||u_l||_2^2}{2} && \text{d'après 2.} \\
&\leq \frac{D_k}{\eta} \sqrt{\sum_{l=0}^{k} ||u_l||_2^2} + \sum_{l=0}^{k} \frac{\eta}{2} \frac{||u_l||_2^2}{\sqrt{\sum_{m=0}^{l} ||u_m||_2^2}} && \text{par définition de } \eta_k \\
&\leq \frac{D_k}{\eta} \sqrt{\sum_{l=0}^{k} ||u_l||_2^2} + \eta \sqrt{\sum_{l=0}^{k} ||u_l||_2^2} && \text{d'après la question précédente avec } ||u_l||_2^2 = a_l \\
&\leq \left(\eta + \frac{D_k}{\eta}\right) \sqrt{\sum_{l=0}^{k} ||u_l||_2^2}
\end{aligned}$$

Question 4

On suppose F $\gamma_F - lipschitzienne$ avec $0 \leq \gamma_F < 1$

a) Soit $k \geq 0$,

$$\begin{aligned}
||Fx_k - Fx_*||_2^2 &= ||Fx_k - x_k + x_k - Fx_*||_2^2 \\
&= ||Fx_k - x_k||_2^2 + ||x_k - Fx_*||_2^2 + 2 < Fx_k - x_k, x_k - Fx_* > \\
&\leq \gamma_F^2 ||x_k - x_*||_2^2 \leq ||x_k - x_*||_2^2 \\
\Rightarrow ||Fx_k - x_k||_2^2 &\leq ||x_k - x_*||_2^2 - ||x_k - Fx_*||_2^2 - 2 < Fx_k - x_k, x_k - Fx_* > \\
&\leq -2 < Fx_k - x_k, x_k - x_* > && \text{car } x_* \text{ point stationnaire} \\
&\leq 2 < Fx_k - x_k, x_* - x_k >= 2(Fx_k - x_k)^T (x_* - x_k)
\end{aligned}$$

b) L'inégalité précédente étant vrai pour tout $k \geq 0$, on peut sommer chaque terme jusqu'à un $k \geq 0$ donné :

$$\begin{aligned}
\sum_{l=0}^{k} ||Fx_l - x_l||_2^2 &\leq \sum_{l=0}^{k} 2(Fx_l - x_l)^T (x_* - x_l) && \text{d'après la 4.a} \\
&\leq 2 \left(\eta + \frac{D_k}{\eta}\right) \sqrt{\sum_{l=0}^{k} ||Fx_l - x_l||_2^2} && \text{d'après la 3.b} \\
\Rightarrow \sqrt{\sum_{l=0}^{k} ||Fx_l - x_l||_2^2} &\leq 2 \left(\eta + \frac{D_k}{\eta}\right) \Rightarrow \sqrt{\sum_{\ell=0}^{k} \min_{l \in [0,k]} ||Fx_l - x_l||_2^2} \leq 2 \left(\eta + \frac{D_k}{\eta}\right) \\
\Rightarrow \sqrt{k \min_{l \in [0,k]} ||Fx_l - x_l||_2^2} &\leq 2 \left(\eta + \frac{D_k}{\eta}\right) \Rightarrow \min_{l \in [0,k]} ||Fx_l - x_l||_2 \leq \frac{2}{\sqrt{k}} \left(\eta + \frac{D_k}{\eta}\right)
\end{aligned}$$

c) Soit $l_* \in \underset{l\in[0,k]}{\text{argmin}} ||Fx_l - x_l||_2$

$$\begin{aligned}
||x_{l_*} - x_*||_2 &= ||x_{l_*} - x_* + Fx_{l_*} - Fx_{l_*}||_2 \\
&\leq ||x_{l_*} - Fx_{l_*}||_2 + ||Fx_{l_*} - x_*||_2 && \text{par inégalité triangulaire} \\
&\leq ||x_{l_*} - Fx_{l_*}||_2 + \gamma_F||x_{l_*} - x_*||_2 && \text{car } Fx_* = x_* \\
&\leq \frac{2}{\sqrt{k}}\left(\eta + \frac{D_k}{\eta}\right) + \gamma_F||x_{l_*} - x_*||_2 && \text{par 4.b} \\
&\leq \frac{2}{\sqrt{k}}\frac{1}{1-\gamma_F}\left(\eta + \frac{D_k}{\eta}\right) \\
\implies \min_{l\in[0,k]} ||x_l - x_*||_2 &\leq \frac{2}{\sqrt{k}}\frac{1}{1-\gamma_F}\left(\eta + \frac{D_k}{\eta}\right)
\end{aligned}$$

Ainsi dans le cas où $\frac{D_k}{\sqrt{k}} \sim o(\sqrt{k})$, l'algorithme passe au moins par un point qui tend vers le point fixe x_* quand k tend vers l'infini. Ce point fixe est unique, car F est contractante ce qui, dans un espace complet, suffit pour l'affirmer par le théorème de Banach.

Question 5

Comme vu en cours pour les algorithmes classiques, nous cherchons l'unique point fixe de B_π et de B_* qui sont $\gamma - lipschitzienne$ avec $0 \leq \gamma < 1$. Or nous venons de voir qu'en utilisant l'algorithme Ada-FP qu'il existe un point qui tend vers l'unique point fixe d'une fonction F avec comme seul condition qu'elle soit $\gamma_F - lipschitzienne$ avec $0 \leq \gamma_F < 1$. Par analogie, on peut donc définir les algorithmes suivant en utilisant $F = B_\pi$ et $F = B_*$:

$$v_{k+1} = v_k + \eta \frac{B_\pi v_k - v_k}{\sqrt{\sum_{l=0}^{k} ||B_\pi v_l - v_l||_2^2}} \qquad (\text{Ada-VI}_\pi^{(\text{V})})$$

et

$$v_{k+1} = v_k + \eta \frac{B_* v_k - v_k}{\sqrt{\sum_{l=0}^{k} ||B_* v_l - v_l||_2^2}} \qquad (\text{Ada-VI}_*^{(\text{V})})$$

2 Comparaison en pratique avec les algorithmes classiques

L'application est sur le labyrinthe vu en cours. Le MDP est donc le suivant :

- états : cases du labyrinthe (10x10)
- actions : haut, bas, gauche, droite
- gain : 1 si dernière case du labyrinthe, 0 sinon

La politique stationnaire implémentée est la politique gloutonne, qui est de plus déterministe, et v_0 est tiré aléatoirement suivant une loi uniforme centré réduite.
Sachant les algorithmes convergants, la valeur de v_π (resp. v_*) est approximé par la dernière valeur prise par l'algorithme Ada-$\text{VI}_\pi^{(\text{V})}$ (resp. Ada-$\text{VI}_*^{(\text{V})}$). On obtient alors après modélisation (code disponible ici) les graphiques suivants :

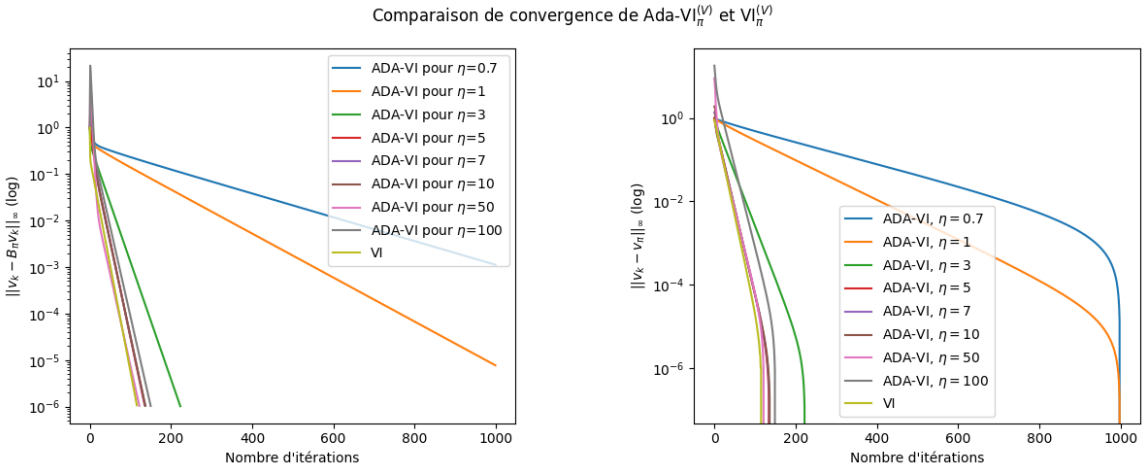

FIGURE 1 – Ada-VI$_\pi^{(V)}$ vs VI$_\pi^{(V)}$

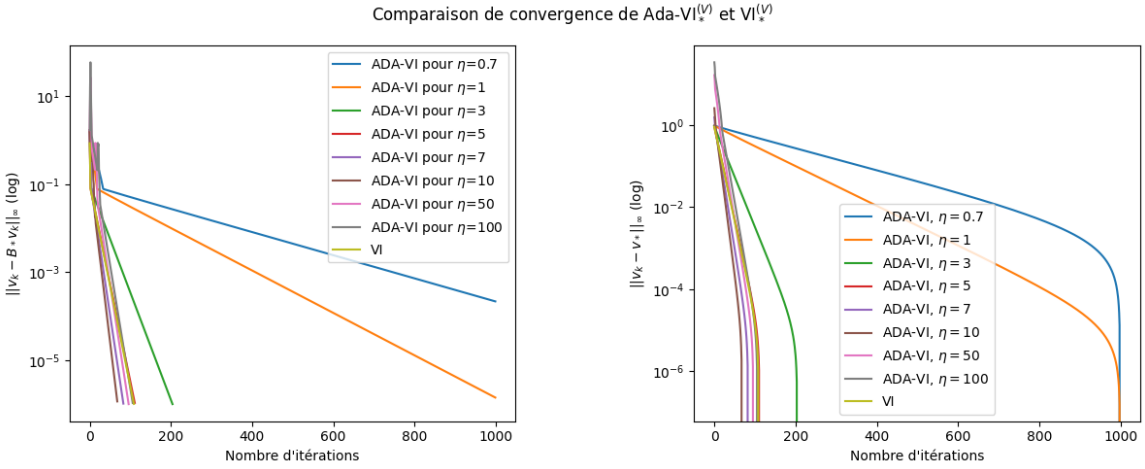

FIGURE 2 – Ada-VI$_*^{(V)}$ vs VI$_*^{(V)}$

À travers ces graphiques, on remarque principalement deux choses :

— Plus η est grand, plus les algorithmes Ada-VI$_\pi^{(V)}$ et Ada-VI$_*^{(V)}$ converge rapidement

— Le nouvelle algorithme Ada-VI$_\pi^{(V)}$ (resp. Ada-VI$_*^{(V)}$) est aussi efficace que l'algorithme classique VI$_\pi^{(V)}$ (resp.VI$_*^{(V)}$) seulement pour des η grand.

Ainsi, les algorithmes dérivés d'Ada-FP peuvent être intéressant à utiliser en posant un η assez grand. Si ce n'est pas le cas, les algorithmes classiques les surpasses.

3 Extensions

3.1 Itérations de fonctions action-valeur

Par analogie, on a pour les fonctions action-valeur :

$$q_{k+1} = q_k + \eta \frac{B_\pi q_k - q_k}{\sqrt{\sum_{l=0}^{k} ||B_\pi q_l - q_l||_2^2}} \qquad \text{(Ada-VI}_\pi^{(\mathrm{Q})}\text{)}$$

et

$$q_{k+1} = q_k + \eta \frac{B_* q_k - q_k}{\sqrt{\sum_{l=0}^{k} ||B_* q_l - q_l||_2^2}} \qquad \text{(Ada-VI}_*^{(\mathrm{Q})}\text{)}$$

Ce qui après implémentation donne les graphiques suivants :

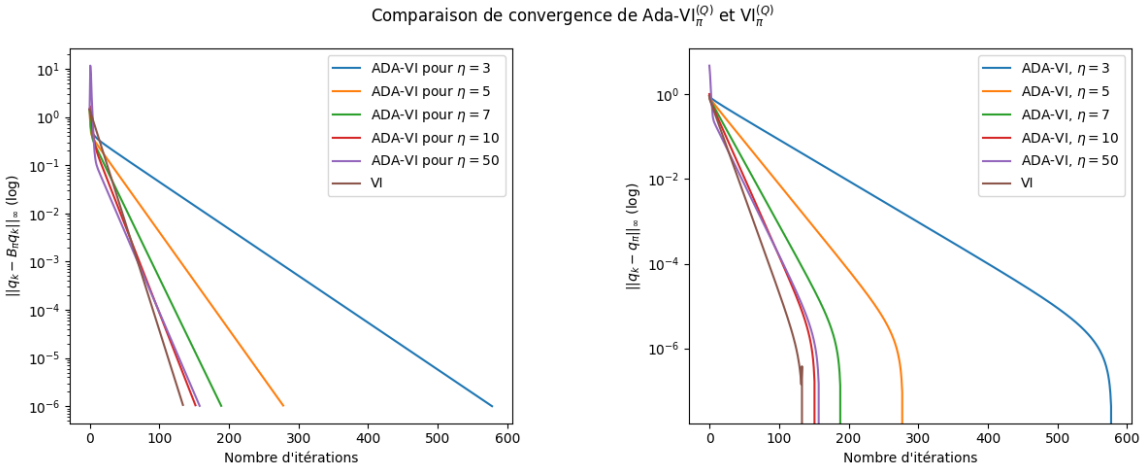

FIGURE 3 – Ada-VI$_\pi^{(\mathrm{Q})}$ vs VI$_\pi^{(\mathrm{Q})}$

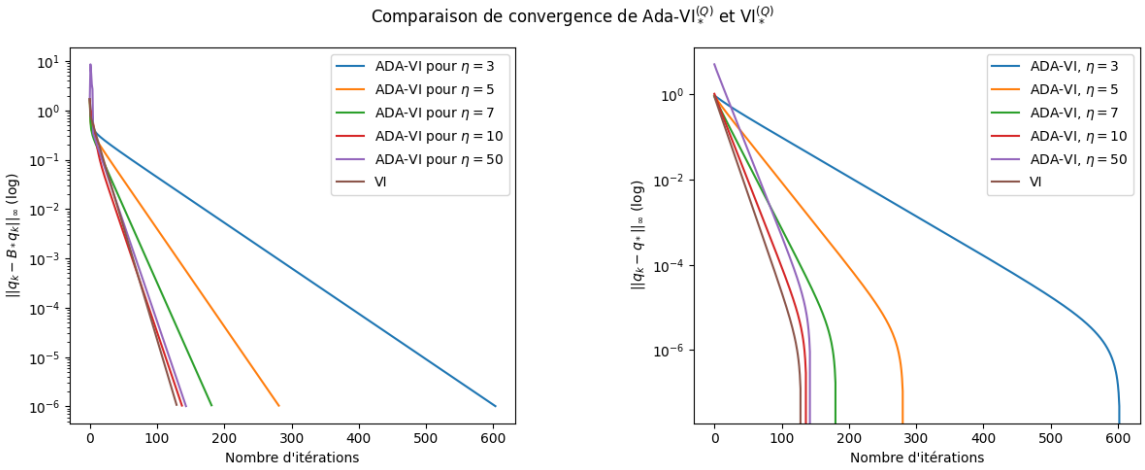

FIGURE 4 – Ada-VI$_*^{(\mathrm{Q})}$ vs VI$_*^{(\mathrm{Q})}$

On remarque encore une fois qu'un fort η est recommandé pour avoir une convergence plus rapide ainsi qu'une efficacité proche des algorithmes classiques. On note cependant que la convergence est globalement plus lente qu'avec les fonctions état-valeur.

3.2 Itérations asynchrones

Comme vu en cours, la mise à jours ne ce fait que sur un sous ensemble d'états à chaque itération, ce qui donne :

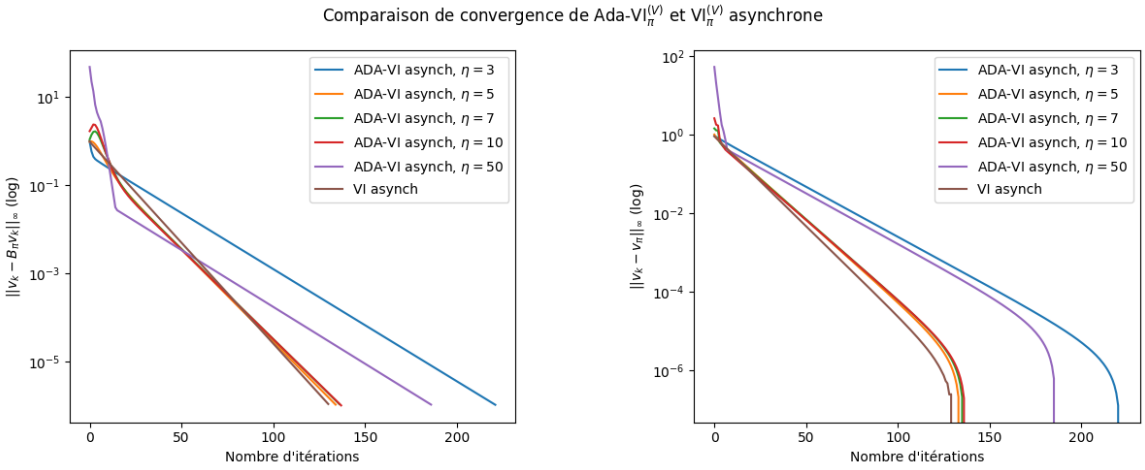

FIGURE 5 – Ada-VI$_\pi^{(\mathrm{V})}$ vs VI$_\pi^{(\mathrm{V})}$ asynchrone

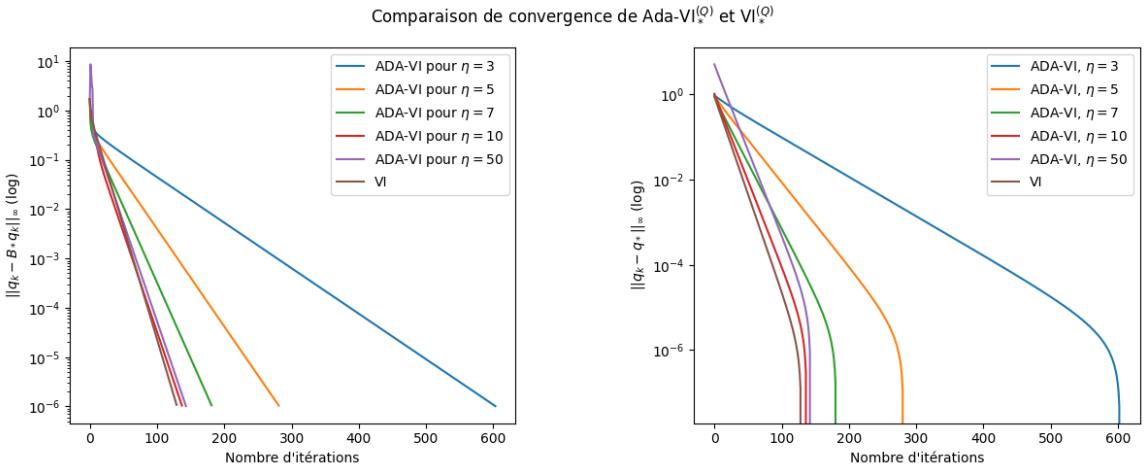

FIGURE 6 – Ada-VI$_*^{(\mathrm{V})}$ vs VI$_*^{(\mathrm{V})}$ asynchrone

Attention, dans ces graphiques une itération représente une mise à jour sur tous les états. De ce fait, pour avoir le vrai nombre d'itérations, il faut multiplier par le nombre de partition de nos états initiaux (dans notre code 5). Cependant cela n'a pas été fait car j'ai estimé que cela facilitait la comparaison entre les graphiques.
Les convergences sont équivalentes à celles non asynchrones.

3.3 Variante définie composante par composante

Les algorithmes deviennent alors d'après le nouvel algorithme de convergence donné :

$$v_{k+1,j} = v_{k,j} + \eta \frac{(B_\pi v_k)_j - v_{k,j}}{\sqrt{\sum_{l=0}^{k} \left(B_\pi (v_l)_j - v_{l,j}\right)^2}} \qquad \text{(Ada-VI}_\pi^{(\mathrm{V})}\text{-component)}$$

et

$$v_{k+1,j} = v_{k,j} + \eta \frac{(B_* v_k)_j - v_{k,j}}{\sqrt{\sum_{l=0}^{k} \left(B_* (v_l)_j - v_{l,j}\right)^2}} \qquad \text{(Ada-VI}_*^{(\mathrm{V})}\text{-component)}$$

Cela donne après implémentation :

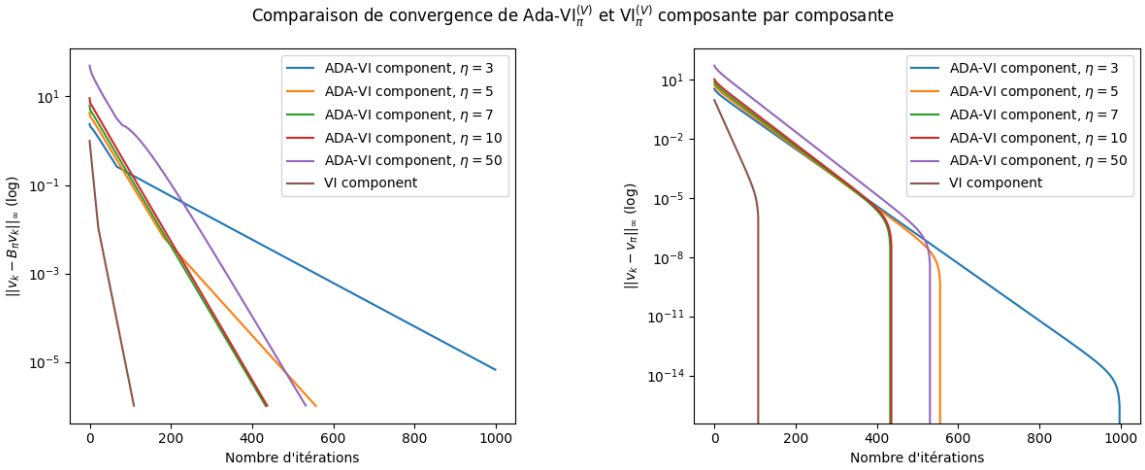

FIGURE 7 – Ada-VI$_\pi^{(\mathrm{V})}$ vs VI$_\pi^{(\mathrm{V})}$ composante par composante

Attention, même raisonnement que précédemment : il faut multiplier par 100 (car labyrinthe de taille 10x10) le nombre d'itérations pour avoir sa vrai valeur. De plus, le temps de calcul étant très long pour Ada-VI$_*^{(\mathrm{V})}$-component, le graphique n'est ici pas représenté.
Pour chaque η, l'algorithme converge plus lentement que l'algorithme Ada-VI$_\pi^{(\mathrm{V})}$ mais la convergence est plus linéaire. Cela fait sens car la modification des valeurs se fait une par une.