

Contrôle optimal des écoulements en temps quasi-réel par un algorithme génétique

M. Oulghelou ^a, C. Allery^b

LaSIE, UMR 7356 CNRS - La Rochelle Université,
Avenue Michel Crépeau, 17042 La Rochelle Cedex 1, France

^a mourad.oulghelou@univ-lr.fr, ^b cyrille.allery@univ-lr.fr

Résumé

Le contrôle des écoulements par des algorithmes génétiques (AGs) nécessite des capacités de stockage mémoire très importantes et est très coûteux en temps de calcul CPU. En effet, la recherche de l'optimum par les AGs s'effectue de façon itérative sur une population de chromosomes (paramètres de contrôle). Par conséquent, la fonctionnelle objectif doit être évaluée en résolvant le modèle complet pour chaque chromosome et pour plusieurs générations. Pour contourner ces difficultés un algorithme génétique réduit pour le contrôle optimal en temps quasi-réel est proposé dans cette communication. L'approche consiste à évaluer de façon rapide la fonctionnelle objectif en interpolant par la méthode Bi-CITSGM (Bi Calibrated Interpolation on the Tangent Space of the Grassmann Manifold) [1] des solutions précalculées, pour différentes valeurs du paramètre de contrôle du problème considéré, et stockées sous forme réduite par POD. Le potentiel de l'approche en termes de précision et de temps CPU est démontré sur le problème de contrôle optimal de l'écoulement autour d'un cylindre lorsque la valeur du nombre de Reynolds varie.

Abstract

Most genetic algorithms (GAs) used in literature to solve control problems are time consuming and involve important storage memory requirements. In fact, the search in GAs is iteratively performed on a population of chromosomes (control parameters). As a result, the cost functional needs to be evaluated through solving the high fidelity model or by performing the experimental protocol for each chromosome and for many generations. To overcome this issue, a non intrusive reduced real coded genetic algorithm (RGA) for near real time optimal control is designed. The proposed approach allows a very fast evaluation of the cost functional through the method of solutions interpolation Bi-CITSGM (Bi Calibrated Interpolation on the Tangent Space of the Grassmann Manifold). The potential of RGA in terms of accuracy and CPU time is demonstrated on control problem of the flow past a cylinder when the Reynolds number value varies.

Mots clefs : Algorithme génétique, Proper Orthogonal Decomposition, interpolation des sous-espaces vectoriels, contrôle optimal.

1 Introduction

En contrôle optimal des écoulements, les techniques d'optimisation les plus utilisées sont les méthodes de descente basées sur le gradient [2]. Ces approches étant susceptibles de générer des minima locaux [3], il est possible d'utiliser des stratégies d'optimisation plus puissantes telles que les algorithmes génétiques (AG) développés par John Holland [4]. Ces algorithmes sont basés sur une approche d'optimisation stochastique reconnue pour sa capacité à effectuer une optimisation globale [5]. C'est une forme de recherche évolutive qui utilise des opérateurs génétiques imitant le concept Darwinien de la sélection naturelle et de l'évolution. Il s'agit généralement de la sélection, du croisement et de la mutation. Les éléments de l'espace de recherche associés au problème de contrôle optimal sont codés sous forme d'un matériel génétique (ADN). Une population de solutions est générée au hasard et l'aptitude de chaque chromosome est évaluée de manière à ce que les chromosomes les plus forts aient plus de chance à se reproduire et à promouvoir leurs caractéristiques aux générations suivantes. Le croisement s'effectue par l'échange de parties de l'ADN de deux chromosomes et la mutation par altération aléatoire des gènes d'un chromosome. Étant donné que l'aptitude des chromosomes est directement évaluée à partir des paramètres d'optimisation, sans avoir besoin d'informations sur les dérivées locales, les AGs peuvent traiter des fonctions non lisses, non continues et non différentiables que l'on rencontre dans plusieurs situations d'optimisation. Dans le contexte de la mécanique des fluides, les AGs ont, entre autres, été appliqués avec succès pour minimiser la traînée de l'écoulement autour d'un cylindre [6], pour l'optimisation de forme [7], pour l'optimisation d'une éolienne [8]... Malgré leur robustesse en terme de précision, les AGs ont le désavantage de nécessiter des temps de calcul CPU très importants. En effet, le nombre d'évaluations de la fonctionnelle objectif requis par les AGs dépasse en général celui nécessaire à une optimisation basée sur le gradient [9, 10].

Une façon possible de réduire considérablement le coût de calcul de l'évaluation de la fonctionnelle est d'utiliser des techniques de réduction de modèles basées sur la POD (Proper Orthogonal Decomposition). Dans cette communication, nous proposons d'utiliser une approche de réduction non intrusive¹ qui ne requiert aucune connaissance préalable des équations du modèle. Ainsi, le contrôle optimal du problème physique se fait en utilisant uniquement les données provenant des solutions du modèle complet pour différentes valeurs des paramètres de contrôle. La méthode proposée est une version Hyper réduite de la méthode Bi-CITSGM (Bi-Calibrated Interpolation on the Tangent Space of the Grassmann Manifold) qui a été récemment développée et appliquée avec succès pour prédire rapidement l'écoulement autour d'un cylindre lorsque la valeur du nombre Reynolds varie [1]. En utilisant un ensemble de solutions paramétrées précalculées et stockées sous forme réduite, le processus d'interpolation par Bi-CITSGM s'effectue en trois étapes. Tout d'abord, les valeurs propres POD sont approximées en utilisant l'interpolation spline cubique. Ensuite, les bases spatiales et temporelles POD sont prédites en utilisant la méthode ITSGM (Interpolation on the Tangent Space of the Grassmann Manifold) [11]. Enfin, deux matrices de calibration orthogonales sont introduites afin d'assurer la meilleure correspondance entre les modes des bases interpolées et leurs valeurs propres correspondantes. Ces matrices sont obtenues comme solutions analytiques de deux problèmes d'optimisation sous contrainte. Dans cet article, nous proposons une version hyper réduite, appelée Hyper Bi-CITSGM, qui vise à accélérer le processus d'interpolation de la méthode Bi-CITSGM en le ramenant à une dimension nettement inférieure. Sur la base de ce résultat, nous concevons un algorithme génétique réduit (AGR) dans lequel le solveur des équations de Navier-Stokes est remplacé par la méthode Hyper Bi-CITSGM. Le potentiel de l'AGR en

1. Habituellement, le modèle d'ordre réduit décrivant la dynamique temporelle est construit par projection de Galerkin des équations sur la base POD. Cette approche est donc dite intrusive, étant donnée qu'elle nécessite l'accès au modèle complet.

termes de précision et de temps CPU est démontré sur les problèmes de contrôle de l'écoulement autour d'un cylindre lorsque la valeur du nombre de Reynolds varie.

Le papier est organisé de la manière suivante. Dans la section suivante, la méthode Hyper Bi-CITSGM est présentée. Dans la section 3, une brève introduction aux algorithmes génétiques est donnée. Les résultats sur le problème de contrôle optimal de l'écoulement autour d'un cylindre sont exposés dans la section 4. Enfin, le papier se termine par des conclusions et des perspectives.

2 Interpolation des solutions par la méthode Hyper Bi-CITSGM

Soit $\gamma_i \in \mathbb{R}^p$, $i = 1, \dots, N_p$, un ensemble de paramètres et S_{γ_i} les matrices de snapshots paramétrées associées. Ces matrices sont formées à partir de solutions à différents instants d'un problème physique non linéaire. S_{γ_i} sont donc des matrices de taille $N_x \times N_s$ où N_x est le nombre de degrés de liberté du problème et N_s le nombre d'instants. L'objectif de ce travail est de trouver une approximation $S_{\tilde{\gamma}}$ de la matrice des snapshots pour un nouveau paramètre $\tilde{\gamma} \neq \gamma_i$ en interpolant les matrices d'apprentissage S_{γ_i} . En général, les méthodes d'interpolation usuelles comme Lagrange ou RBF ne peuvent être utilisées à cause de la dépendance non linéaire des données vis à vis du paramètre de contrôle. Pour surmonter cette difficulté, nous allons introduire une méthode d'interpolation des solutions robuste et rapide, appelée Hyper Bi-CITSGM (Hyper Bi-Calibrated Interpolation on the Tangent Space of the Grassmann Manifold), qui est basée sur l'interpolation de bases POD par la méthode ITSGM (Interpolation on the Tangent Space of the Grassmann Manifold).

2.1 Interpolation de bases réduites par ITSGM

2.1.1 Variété de Grassmann, applications Exponentielle et Logarithme géodésiques

- La variété de Grassmann $\mathcal{G}(q, N)$ est définie comme l'ensemble de tous les sous espaces vectoriels de dimension q de \mathbb{R}^N , où $0 \leq q \leq N$. Un point $[\Phi] \in \mathcal{G}(q, N)$ peut être représenté par la classe d'équivalence

$$[\Phi] = \{\Phi Q \mid Q \in \mathcal{O}(q)\}$$

où Φ est une matrice de taille $N \times q$ dont les colonnes forment une base orthogonale, et $\mathcal{O}(q)$ le groupe orthogonal des matrices $q \times q$.

- En chaque point $[\Phi]$ de la variété $\mathcal{G}(q, N)$, on peut définir un espace tangent $\mathcal{T}_{[\Phi]} \mathcal{G}(q, N)$ de même dimension.
- Pour chaque direction $\mathcal{X} \in \mathcal{T}_{[\Phi]} \mathcal{G}(q, N)$, il existe un chemin géodésique unique² pouvant être défini par l'application Exponentielle $\text{Exp}_{[\Phi]} : \mathcal{T}_{[\Phi]} \mathcal{G}(q, N) \rightarrow \mathcal{G}(q, N)$. Soit $U \Sigma V^T$ la décomposition SVD tronquée de la vitesse initiale \mathcal{X} , l'exponentielle de \mathcal{X} est donnée par

$$[\Psi] = \text{span}\{\Phi V \cos(\Sigma) + U \sin(\Sigma)\}$$

2. Un chemin géodésique entre deux points $[\Phi]$ et $[\Psi]$ de la variété de Grassmann est le chemin qui minimise la distance géodésique définie par $\text{dist}_{\mathcal{G}}(\Phi, \Psi) = \sqrt{\sum_i \arccos^2(\sigma_i)}$ où σ_i sont les valeurs singulières de $\Phi^T \Psi$.

• Soit $\text{Log}_{[\Phi]}$ l'application inverse de l'application $\text{Exp}_{[\Phi]}$, définie uniquement dans un voisinage de $[\Phi]$. Si $\text{Exp}_{[\Phi]}(\mathcal{X}) = [\Psi]$, alors \mathcal{X} est le vecteur déterminé comme suit

$$\mathcal{X} = \text{Log}_{[\Phi]}([\Psi]) = U \arctan(\Sigma) V^T$$

où $U\Sigma V^T$ est la décomposition SVD tronquée de $(I - \Phi\Phi^T)\Psi(\Phi^T\Psi)^{-1}$ et $\text{Log}_{[\Phi]}([\Phi]) = 0$.

Pour plus de détails sur ces notions le lecteur peut se référer à [12].

2.1.2 Méthode ITSGM

Soit $\Phi_{\gamma_1}, \Phi_{\gamma_2}, \dots, \Phi_{\gamma_{N_p}}$ un ensemble de bases POD et $[\Phi_{\gamma_1}], [\Phi_{\gamma_2}], \dots, [\Phi_{\gamma_{N_p}}]$ les sous-espaces vectoriels associés appartenant à la variété de Grassmann. En utilisant la définition du chemin géodésique, de l'application Exponentielle et de l'application Logarithmique, le but est d'approximer le sous-espace $[\Phi_{\tilde{\gamma}}]$ correspondant à un nouveau paramètre $\tilde{\gamma} \neq \gamma_i$. Les différentes étapes de la méthode ITSGM³ sont

- Choisir un point de référence $[\Phi_{\gamma_{i_0}}]$ où $i_0 \in \{1, \dots, N_p\}$.
- Pour $i \in \{1, \dots, N_p\}$, déterminer le vecteur $\mathcal{X}_i = \text{Log}_{[\Phi_{\gamma_{i_0}}]}(\Phi_{\gamma_i})$ par la relation

$$\mathcal{X}_i = U_i \arctan(\Sigma_i) V_i^T, i = 1, \dots, N_p$$

où $U_i \Sigma_i V_i^T$ est la décomposition SVD tronquée de $(I - \Phi_{\gamma_{i_0}} \Phi_{\gamma_{i_0}}^T) \Phi_{\gamma_i} (\Phi_{\gamma_{i_0}}^T \Phi_{\gamma_i})^{-1}$.

- Interpoler $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_{N_p}$ et obtenir la vitesse initiale $\mathcal{X}_{\tilde{\gamma}}$ associé au nouveau paramètre $\tilde{\gamma}$. Étant donné que l'espace tangent $\mathcal{T}_{[\Phi_{i_0}]} \mathcal{G}(q, N)$ est un espace vectoriel, il est possible d'utiliser les techniques standard d'interpolation telles que Lagrange, RBF...
- Déterminer le sous-espace vectoriel interpolé

$$\Phi_{\tilde{\gamma}} = \Phi_{\gamma_{i_0}} \tilde{V} \cos(\tilde{\Sigma}) + \tilde{U} \sin(\tilde{\Sigma})$$

où $\tilde{U} \tilde{\Sigma} \tilde{V}^T$ est la décomposition SVD tronquée de $\mathcal{X}_{\tilde{\gamma}}$.

2.2 Description de la méthode Hyper Bi-CITSGM

• Considérons la décomposition POD d'ordre q des matrices S_{γ_i} donnée comme suit

$$S_{\gamma_i} \approx U_{\gamma_i} \Sigma_{\gamma_i} V_{\gamma_i}^T$$

où $U_{\gamma_i} \in \mathbb{R}^{N_x \times q}$ et $V_{\gamma_i} \in \mathbb{R}^{N_s \times q}$ représentent respectivement les bases spatiales et temporelles, et $\Sigma_{\gamma_i} \in \mathbb{R}^{q \times q}$ la matrice des valeurs singulières. Une seconde décomposition POD d'ordre r et s , où $r, s \leq qN_p$, est ensuite effectuée sur les matrices par bloc suivantes

$$\begin{bmatrix} U_{\gamma_1} & U_{\gamma_2} & \cdots & U_{\gamma_{N_p}} \end{bmatrix} = \psi \varrho W^T \quad \text{et} \quad \begin{bmatrix} V_{\gamma_1} & V_{\gamma_2} & \cdots & V_{\gamma_{N_p}} \end{bmatrix} = \Pi \Theta Z^T$$

3. Pour plus de détails le lecteur peut se référer à [11].

où $\psi \in \mathbb{R}^{N_x \times r}$, $W \in \mathbb{R}^{qN_p \times r}$, $\mathbf{\Pi} \in \mathbb{R}^{N_s \times s}$, $Z \in \mathbb{R}^{qN_p \times s}$, $\varrho = \text{diag}(\rho_1, \rho_2, \dots, \rho_r)$ et $\Theta = \text{diag}(\theta_1, \theta_2, \dots, \theta_s)$. Soient $W_i \in \mathbb{R}^{q \times r}$ et $Z_i \in \mathbb{R}^{q \times s}$ les blocs des matrices W et Z définies par

$$W = \begin{bmatrix} W_{\gamma_1} \\ W_{\gamma_2} \\ \vdots \\ W_{\gamma_{N_p}} \end{bmatrix} \quad \text{et} \quad Z = \begin{bmatrix} Z_{\gamma_1} \\ Z_{\gamma_2} \\ \vdots \\ Z_{\gamma_{N_p}} \end{bmatrix}$$

Les matrices d'apprentissage S_{γ_i} peuvent alors être réécrites comme suit

$$S_{\gamma_i} \approx \psi \Phi_{\gamma_i} \Sigma_i \Lambda_{\gamma_i}^T \mathbf{\Pi}^T \quad (1)$$

où $\Phi_{\gamma_i} = \varrho W_{\gamma_i}^T$ et $\Lambda_{\gamma_i} = \Theta Z_{\gamma_i}^T$ sont deux bases orthogonales. D'après l'expression (1), il est évident que les matrices de grande taille ψ et $\mathbf{\Pi}$ ne dépendent pas du paramètre γ_i . L'avantage de cette écriture est que la matrice $\Phi_{\gamma_i} \Sigma_i \Lambda_{\gamma_i}^T$ qui varie en fonction du paramètre est de dimension $r \times s$ qui est nettement inférieure à la dimension initiale $N_x \times N_p$ des matrices de snapshot d'apprentissage. L'obtention des matrices ψ et $\mathbf{\Pi}$ se fait une fois pour toute et constitue la phase offline de la méthode Hyper Bi-CITSGM.

• L'obtention de la solution $S_{\tilde{\gamma}}$ pour un nouveau paramètre $\tilde{\gamma} \neq \gamma_i$ se ramène à l'évaluation de la matrice $\Phi_{\tilde{\gamma}} Q_x \Sigma_{\tilde{\gamma}} Q_t^T \Lambda_{\tilde{\gamma}}^T$ où Q_x et Q_t sont deux matrices de calibration déterminer. Ces étapes qui constituent la phase online de l'interpolation Hyper Bi-CITSGM sont :

- Premièrement, la matrice des valeurs singulières $\Sigma_{\tilde{\gamma}}$ est approximée en utilisant l'interpolation spline cubique.
- Ensuite, les bases spatiales $\Phi_{\tilde{\gamma}}$ et temporelles $\Lambda_{\tilde{\gamma}}$ sont approximées en utilisant la méthode d'interpolation de bases réduites ITSGM.
- La troisième étape consiste à ajuster les signes des modes des bases d'apprentissage, et ce, en multipliant les $j^{\text{ème}}$ modes spatiaux $\Phi_{\gamma_k}^j$ et temporels $\Lambda_{\gamma_k}^j$ par -1 si la condition $\|\Phi_{\gamma_{k_0}}^j - \Phi_{\gamma_k}^j\|_2 > \|\Phi_{\gamma_{k_0}}^j + \Phi_{\gamma_k}^j\|_2$ est satisfaite, avec $k_0 = \underset{i \in \{1, \dots, N_p\}}{\text{argmin}} \text{dist}_{\mathcal{G}}(\Phi_{\tilde{\gamma}}, \Phi_{\gamma_i})$.
- Finalement, les bases interpolées sont calibrées en les multipliant par deux matrices orthogonales solutions des deux problèmes d'optimisation suivants

$$\underset{Q_x \in \mathcal{O}(q)}{\text{argmin}} \sum_{i=1}^{N_p} \omega_i \|\Phi_{\tilde{\gamma}} Q_x - \Phi_{\gamma_i}\|_F^2 \quad \underset{Q_t \in \mathcal{O}(q)}{\text{argmin}} \sum_{i=1}^{N_p} \kappa_i \|\Lambda_{\tilde{\gamma}} Q_t - \Lambda_{\gamma_i}\|_F^2 \quad (2)$$

où $\|\cdot\|_F$ est la norme de Frobenius et ω_i et κ_i sont les poids GIDW (Grassmann Inverse Distance weighting) donnés pour $m, l > 1$ par

$$\omega_i = \frac{\text{dist}_{\mathcal{G}}(\Phi_{\tilde{\gamma}}, \Phi_{\gamma_i})^{-m}}{\sum_{k=1}^{N_p} \text{dist}_{\mathcal{G}}(\Phi_{\tilde{\gamma}}, \Phi_{\gamma_k})^{-m}} \quad \kappa_i = \frac{\text{dist}_{\mathcal{G}}(\Lambda_{\tilde{\gamma}}, \Lambda_{\gamma_i})^{-l}}{\sum_{k=1}^{N_p} \text{dist}_{\mathcal{G}}(\Lambda_{\tilde{\gamma}}, \Lambda_{\gamma_k})^{-l}} \quad (3)$$

Les problèmes d'optimisation (2) admettent deux solutions analytiques qui sont

$$Q_x = \xi I_{Q_x}^+ \eta^T \quad Q_t = \zeta I_{Q_t}^+ \rho^T$$

où $I_{Q_x}^+$ (resp. $I_{Q_t}^+$) est la matrice diagonale dont les éléments sont égaux à 1 pour les valeurs singulières non nulles et 0 dans le cas contraire. ξ et η (resp. ζ et ρ) sont les vecteurs singuliers de gauche et droite de la matrice M_x (resp. M_t)

$$M_x = \Phi_{\bar{\gamma}}^T \sum_{i=1}^{N_p} \omega_i \Phi_{\gamma_i} \quad M_t = \Lambda_{\bar{\gamma}}^T \sum_{i=1}^{N_p} \kappa_i \Lambda_{\gamma_i}$$

M_x et M_t sont des matrices de taille $q \times q$.

e) La dernière étape est la reconstruction de la matrice de snapshots interpolée

$$S_{\bar{\gamma}} = \psi \Phi_{\bar{\gamma}} Q_x \Sigma_{\bar{\gamma}} Q_t^T \Lambda_{\bar{\gamma}}^T \Pi^T$$

3 Algorithme génétique réduit pour le contrôle des écoulements en temps quasi-réel

3.1 Position du problème de contrôle optimal

Considérons le problème d'optimisation sous contrainte non linéaire suivant

$$\min_{\gamma} \mathcal{J}(\mathbf{y}(\gamma), \gamma) \quad \text{tel que} \quad \mathcal{N}(\mathbf{y}(\gamma), \gamma) = 0 \quad (4)$$

où \mathcal{J} est la fonctionnelle coût et \mathbf{y} et γ les variables d'état et de contrôle liées l'une à l'autre par l'équation de contrainte non linéaire \mathcal{N} . En contrôle optimal des écoulements, \mathbf{y} peut correspondre par exemple à la vitesse, la pression ou la température du fluide, γ au nombre de Reynolds, nombre de Strouhal ou l'angle d'attaque, et l'application \mathcal{N} aux équations de Navier-Stokes ou à l'expérience. Dans ce qui suit, on s'intéresse à la résolution du problème de contrôle optimal (4) en utilisant les algorithmes génétiques sans avoir recours aux équations du modèle complet. Une brève introduction aux algorithmes génétiques est donnée dans le paragraphe suivant.

3.2 Algorithme génétique réduit

Les algorithmes génétiques démarrent d'un ensemble aléatoire (de taille N_{chrom}) de chromosomes $\gamma_1, \gamma_2, \dots, \gamma_{N_{\text{chrom}}}$. La robustesse de chaque chromosome γ_j est évaluée, via une fonction "fitness" $f(\gamma_j)$, en utilisant la fonctionnelle objectif \mathcal{J} et les contraintes \mathcal{N} . Ensuite, trois opérateurs génétiques principaux (sélection, croisement et mutation), inspirés des concepts Darwiniens de sélection naturelle, sont appliqués à la population afin de générer une "meilleure" population descendante. Dans l'étape de sélection, une nouvelle population de chromosomes est choisie pour survivre en se basant sur les valeurs de la fonction fitness de ces derniers. Quand la valeur de la fonction fitness associée à un chromosome est élevée, celui-ci a une probabilité élevée de se reproduire et par conséquent de transmettre ses gènes à la génération suivante. Dans l'opération de croisement, tous les chromosomes qui ont survécu sont aléatoirement couplés. Suivant une probabilité de croisement donnée P_c , les couples s'échangent les gènes entre eux. En pratique, une valeur aléatoire entre 0 et 1 est générée. Si cette valeur est inférieure à P_c , le croisement est effectué et deux nouveaux chromosomes sont créés pour remplacer les chromosomes originaux. Cependant, si la valeur aléatoire est supérieure à P_c , les deux chromosomes dans le couple original restent inchangés et passent à la génération suivante. Dans l'opération de mutation, les gènes dans chaque chromosome sont aléatoirement changés selon une probabilité P_m . Comme pour le croisement, la probabilité de mutation

est déterminée par un nombre aléatoire variant entre 0 et 1. Ces opérations génétiques sont répétées jusqu'à ce qu'une solution globale du problème d'optimisation est atteinte.

Malgré leur robustesse, les algorithmes génétiques sont très peu utilisés dans le cadre du contrôle optimal des écoulements car ils nécessitent des temps de calcul CPU et des capacités de stockages importantes. En effet, nous devons résoudre et stocker les solutions des équations de Navier-Stokes plusieurs fois pour chaque population, ce qui est numériquement très coûteux. Afin d'avoir un algorithme génétique réduit, capable de contrôler un écoulement en temps quasi-réel, nous remplaçons la résolution des équations de Navier-Stokes par la méthode Hyper Bi-CITSGM proposée dans la section 2. L'algorithme génétique réduit AGR utilisé dans ce cas est représenté dans la figure 1.

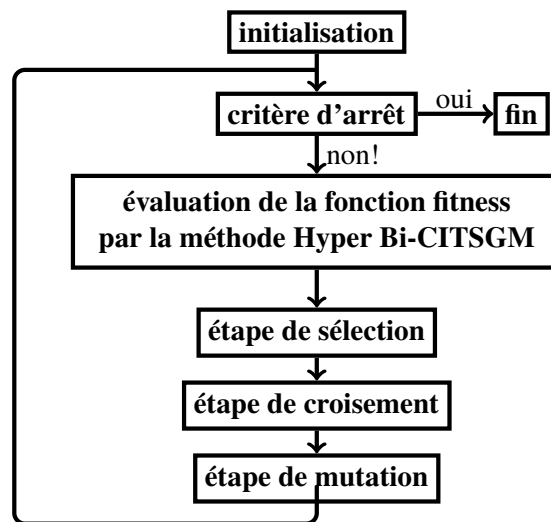


FIGURE 1 – Schéma de l'algorithme génétique réduit AGR.

4 Application au contrôle de l'écoulement autour d'un cylindre

4.1 Position du problème

Nous considérons l'écoulement d'un fluide dans un canal, autour d'un cylindre circulaire de diamètre D (Figure 2). Le domaine d'étude Ω est un domaine rectangulaire de hauteur $H = 30D$ et de longueur $45D$. Le centre du cylindre est situé à $L_1 = 10D$ de la frontière gauche et à $H/2$ de la paroi basse. Une vitesse d'intensité u_∞ est appliquée à l'entrée du canal et une condition de type outlet ($p\mathbf{n} - \frac{1}{Re}\nabla\mathbf{u} \cdot \mathbf{n} = 0$) à sa sortie. Sur les parois horizontales du canal, les conditions aux limites imposées sur la vitesse sont telles que $v = 0$ (composante verticale) et $\partial_y u = 0$ (composante horizontale). Le fluide est supposé

isotherme et incompressible et sa dynamique est régie par les équations de Navier-Stokes suivantes

$$\begin{cases} \partial_t \mathbf{u} - \frac{1}{Re} \Delta \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p = 0 & \text{dans } \Omega \times [0, T[\\ \nabla \cdot \mathbf{u} = 0 & \text{dans } \Omega \times [0, T[\\ \mathbf{u} = \mathbf{u}_\infty & \text{sur } \Gamma_{in} \times [0, T[\\ \mathbf{u} = 0 & \text{sur } \Gamma_{cylindre} \times [0, T[\\ u_2 = 0 & \text{sur } \Gamma_{noslip} \times [0, T[\\ \partial_n u_1 = 0 & \text{sur } \Gamma_{noslip} \times [0, T[\\ -\frac{1}{Re} \partial_n \mathbf{u} + p \mathbf{n} = 0 & \text{sur } \Gamma_{out} \times [0, T[\\ \mathbf{u}(0) = \mathbf{u}_0 & \text{dans } \Omega \end{cases}$$

Les valeurs du nombre de Reynolds $Re = \mathbf{u}_\infty D / \nu$ considérées dans cette étude varient entre 90 et 450. Les simulations numériques ont été effectuées avec Fenics [13] (éléments finis de Taylor-Hood $\mathbb{P}_2/\mathbb{P}_1$) en utilisant un maillage non uniforme, comprenant 85124 degrés de libertés pour la vitesse et 10694 pour la pression, raffiné autour du cylindre et le long de la zone du sillage. La condition initiale considérée pour toutes les valeurs du nombre de Reynolds correspond à la solution à un instant donné du régime périodique de l'écoulement à $Re = 100$. Le pas de temps est fixé à 0.01. Un échantillonnage

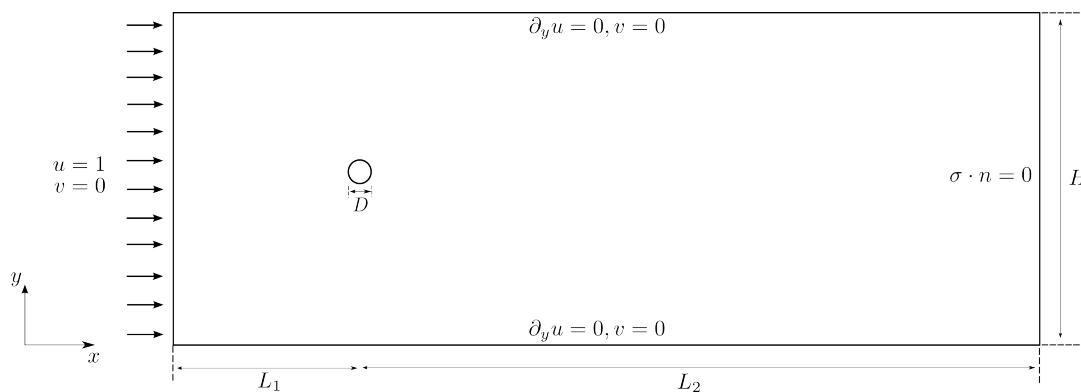


FIGURE 2 – Domaine spatial et conditions au limites pour le cas d'étude de l'écoulement autour d'un cylindre circulaire.

d'écoulements paramétrés⁴ est construit pour les valeurs du nombre de Reynolds comprises entre $Re = 90$ et $Re = 450$ en considérant un pas $\Delta Re = 30$, soit $Re = \{90, 120, 150, 180, \dots, 450\}$. Les bases de l'échantillonnage ont été calculées en considérant 500 snapshots ($N_s = 500$) régulièrement répartis entre les instants $t_i = 7$ et $t_f = 12$, ce qui représente environs 8 périodes du détachement tourbillonnaire de l'écoulement.

4.2 Contrôle optimal pour atteindre une vitesse cible

Soit la fonctionnelle objectif suivante

$$\mathcal{J}(\mathbf{u}, Re) = \int_{t_i + |\delta|}^{t_f - |\delta|} \int_{\Omega} |\mathbf{u}(t + \delta, x, Re) - \hat{\mathbf{u}}(t, x)|^2 dx dt$$

4. Les équations de Navier-Stokes sont résolues sous forme adimensionnelle. Lorsque le nombre de Reynolds varie, u_∞ n'est pas modifié, la variation se fait par l'intermédiaire de la viscosité cinématique ν .

où $\delta \in \mathbb{R}$ est le déphasage, \mathbf{u} est la vitesse solution des équations de Navier-Stokes, $\hat{\mathbf{u}}$ une vitesse cible donnée et $[t_i, t_f]$ l'intervalle temporel⁵ sur lequel les bases POD de l'échantillonnage ont été calculées. On rappelle que nous cherchons le nombre de Reynolds optimal qui minimise la fonctionnelle ci-dessus sous la contrainte des équations de Navier-Stokes régissant le problème d'écoulement autour d'un cylindre. Dans l'algorithme génétique, la méthode Hyper Bi-CITSGM est utilisée pour évaluer la fonction "fitness" qui correspond dans notre cas à l'inverse de la fonctionnelle objectif.

Une population initiale de taille 100 a été aléatoirement générée telle que le déphasage δ et le nombre de Reynolds Re varient dans l'espace des contrôles admissibles K donné par

$$K = \{(Re, \delta) \in \mathbb{R}_+ \times \mathbb{R}, \quad 100 \leq Re \leq 300 \quad \text{et} \quad -0.3 \leq \delta \leq 0.3\}$$

L'évolution des populations par rapport aux itérations pour le nombre de Reynolds cible $Re_{cible} = 260$ est représentée dans la figure 3. Les points en bleu représentent la population initiale et les points en rouge sont les individus sélectionnés à chaque génération. On constate que le contrôle optimal réduit en utilisant l'algorithme génétique couplé avec la méthode Hyper Bi-CITSGM permet bien de retrouver une solution optimale. Cette approche permet de réduire drastiquement le temps de calcul CPU en comparaison avec l'approche du contrôle optimal usuelle. En effet, avec le même nombre de génération (9 générations) les équations de Navier-Stokes et leurs équations adjointes doivent être résolus 900 fois (temps estimé à plusieurs jours sur 1 CPU). Pour la même population, ce temps de calcul est réduit à une trentaine de secondes en utilisant la méthode Hyper Bi-CITSGM. Les résultats de la 9^{ème} génération de l'algorithme génétique couplé avec la méthode Hyper Bi-CITSGM sont répertoriés dans le tableau 1. Les paramètres de contrôle obtenus sont en bon accord avec les paramètres recherchés et les temps de calcul sont drastiquement réduits. Ces résultats sont satisfaisants et prouvent la robustesse de la méthode de réduction de modèle non intrusive Hyper Bi-CITSGM dans l'algorithme génétique.

Re_{cible}	Temps CPU	(Re, δ)	$\mathcal{J}(Re, \delta)$
130	38 sec	(130.89, 0.14)	8.9×10^{-4}
195	37 sec	(193.65, 0.12)	7.19×10^{-4}
260	38 sec	(262.38, -0.04)	1.02×10^{-4}

TABLE 1 – Temps CPU, paramètre de contrôle et valeur de la fonctionnelle associés à l'individu dominant de la génération 9 de l'algorithme génétique couplé avec la méthode Hyper Bi-CITSGM.

4.3 Contrôle optimal pour atteindre un coefficient $C_{L,rms}$ cible

La fonctionnelle objectif étudiée ici est donnée par

$$\mathcal{J}(\mathbf{u}, p, Re) = \left(C_{L,rms} - \hat{C}_{L,rms} \right)^2 \quad (5)$$

telle que

$$C_{L,rms} = \sqrt{\frac{1}{t_f - t_i} \int_{t_i}^{t_f} C_L dt}$$

On s'intéresse au problème de minimisation de la fonctionnelle (5) dont l'objectif est de trouver le nombre de Reynolds qui permet au coefficient $C_{L,rms}$ d'atteindre la valeur cible $\hat{C}_{L,rms}$. Une population initiale de taille 60 a été aléatoirement générée dans l'espace des contrôles admissibles $100 \leq Re \leq 300$.

5. On rappelle que l'intervalle correspond à 8 périodes du détachement tourbillonnaire de l'écoulement.

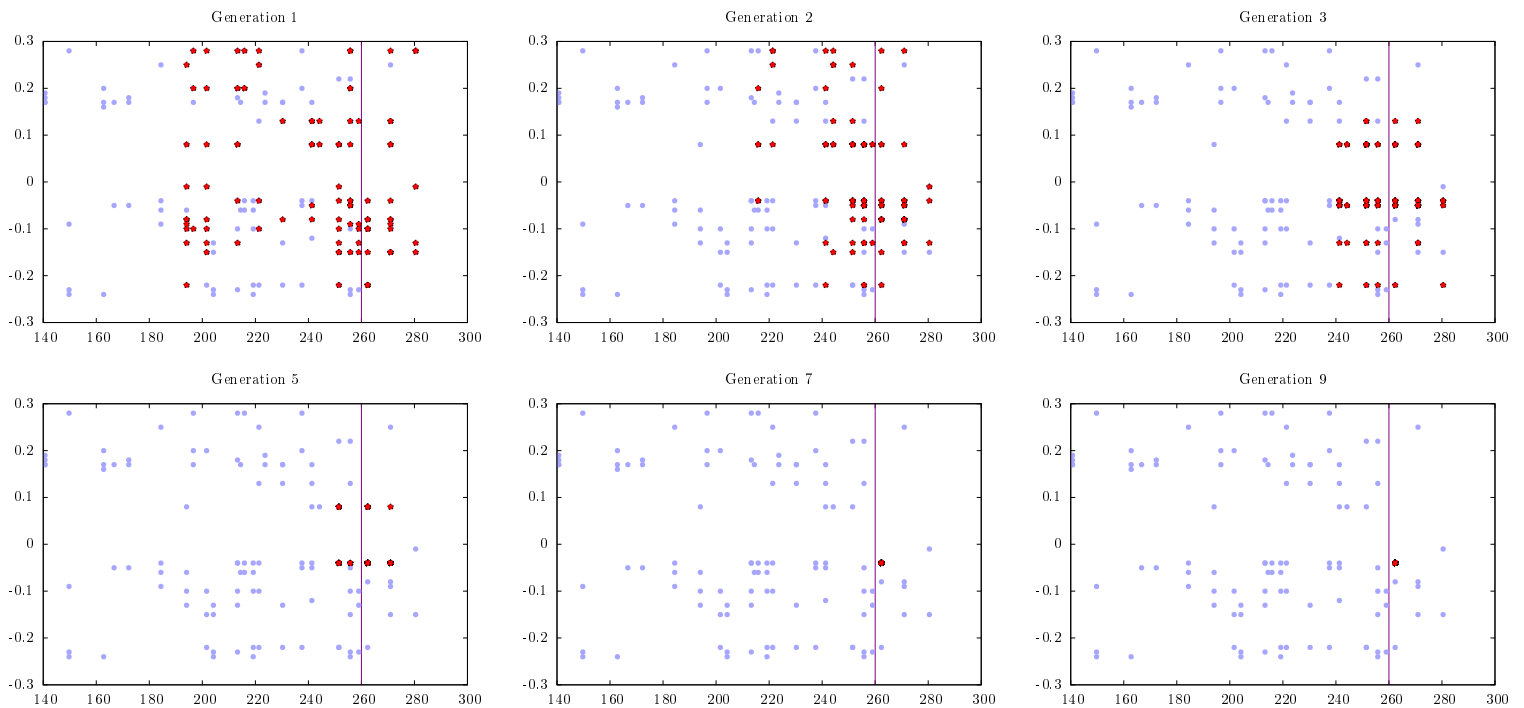


FIGURE 3 – Évolution des générations pour $Re_{cible} = 260$. Les points en bleu représentent la population initiale et les points en rouge les populations à chaque étape de l’algorithme génétique. Chaque point représente un chromosome ayant deux gènes, la valeur du nombre de Reynolds Re sur l’axe des abscisses, et la valeur du déphasage δ sur l’axe des ordonnées.

Par la suite, trois coefficients de portance cibles $\hat{C}_{L,rms}$ correspondant à $Re_{cible} = 135$, $Re_{cible} = 165$ et 225 ont été considérés. Les résultats issus de la 7^{ème} génération de l’algorithme génétique couplé avec la méthode Hyper Bi-CITSGM sont répertoriés dans le tableau 2. On constate que l’approche proposée permet de retrouver une bonne approximation du nombre de Reynolds recherché en 17 secondes.

Re_{cible}	Temps CPU	Re	$\mathcal{J}(Re)$
135	17 sec	135.78	1.3×10^{-3}
165	17 sec	162.44	2.2×10^{-3}
225	17 sec	222.23	1.9×10^{-3}

TABLE 2 – Temps CPU, paramètre de contrôle et valeur de la fonctionnelle associés à l’individu dominant de la génération 7 de l’algorithme génétique.

5 Conclusions

Dans cette communication, un algorithme génétique couplé à la méthode de réduction de modèle Hyper Bi-CITSGM pour le contrôle optimal des écoulements en temps quasi-réel a été présenté. L’efficacité de l’approche a été testée numériquement sur le problème de contrôle de l’écoulement autour d’un cylindre. On a constaté qu’en utilisant un ensemble de solutions pré-calculées du problème d’écoulement, l’algorithme a permis de fournir de bonnes approximations des solutions optimales recherchées en une trentaine de secondes. Cela montre que la stratégie proposée est robuste en termes de précision et de temps CPU pour des problèmes de contrôle optimal. Enfin, il convient de noter que le temps de l’algorithme peut être davantage réduit car il est possible de calculer en parallèle les fonctionnelles objectifs

associées aux différentes combinaisons du paramètre de contrôle. La parallélisation permettrait d'obtenir des solutions de contrôle optimales en temps réel.

Références

- [1] M. Oulghelou and C. Allery, “Non intrusive method for parametric model order reduction using a bi-calibrated interpolation on the grassmann manifold,” *arXiv :1901.03177*, 2018.
- [2] M. Gunzburger, “Adjoint equation-based methods for control problems in incompressible, viscous flows,” *Flow, Turbulence and Combustion*, vol. 65, 12 2000.
- [3] D. W. Zingg, M. Nemec, and T. H. Pulliam, “A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization,” *European Journal of Computational Mechanics*, vol. 17, no. 1-2, pp. 103–126, 2008.
- [4] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI : University of Michigan Press, 1975. second edition, 1992.
- [5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA : Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.
- [6] T. K. Sengupta, K. Deb, and S. B. Talla, “Control of flow using genetic algorithm for a circular cylinder executing rotary oscillation,” *Computers & Fluids*, vol. 36, no. 3, pp. 578 – 600, 2007.
- [7] A. Shahrokhi and A. Jahangirian, “Airfoil shape parameterization for optimum navier–stokes design with genetic algorithm,” *Aerospace Science and Technology*, vol. 11, no. 6, pp. 443 – 450, 2007.
- [8] L. Daróczy, G. Janiga, and D. Thévenin, “Computational fluid dynamics based shape optimization of airfoil geometry for an h-rotor using a genetic algorithm,” *Engineering Optimization*, vol. 50, no. 9, pp. 1483–1499, 2018.
- [9] Terry, L. Hoist, Thomas, and H. R. Pulliam, “Aerodynamic shape optimization using a real-number-encoded genetic algorithm,”
- [10] S. Obayashi and T. Tsukahara, “Comparison of optimization algorithms for aerodynamic shape design,” *AIAA Journal*, vol. 35, no. 8, pp. 1413–1415, 1997.
- [11] D. Amsallem and C. Farhat, “An interpolation method for adapting reduced-order models and application to aeroelasticity,” *AIAA Journal*, pp. 1803–1813, 2008.
- [12] A. Absil, R. Mahony, and R. Sepulchre, “Riemann geometry of Grassmann manifolds with a view on algorithmic computation,” *Acta Applicandae Mathematicae*, vol. 80, Issue 2, p. 199–220, 2004.
- [13] H. P. Langtangen and A. Logg, *Solving PDEs in Python : The FEniCS Tutorial I*. Springer Publishing Company, Incorporated, 1st ed., 2017.