



HAL
open science

GBNM: un algorithme d'optimisation par recherche directe. Application à la conception de monopalmes de nage

Marco Antonio Luersen

► **To cite this version:**

Marco Antonio Luersen. GBNM: un algorithme d'optimisation par recherche directe. Application à la conception de monopalmes de nage. Modélisation et simulation. INSA de Rouen, 1993. Français. NNT : 2004ISAM0014 . tel-00850658

HAL Id: tel-00850658

<https://theses.hal.science/tel-00850658>

Submitted on 7 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée le 1^{er} Décembre 2004 par

Marco Antônio LUERSEN

pour obtenir le grade de

**Docteur de l'Institut National des Sciences
Appliquées de Rouen**

Discipline : Mécanique

Ecole Doctorale SPMI

GBNM : Un Algorithme d'Optimisation par Recherche Directe - Application à la Conception de Monopalmes de Nage

Membres du jury :

M. Breitkopf P.	<i>Rapporteur</i>	Université de Technologie de Compiègne
M. Drapier S.		Ecole des Mines de Saint Etienne
M. Grédiac M.	<i>Rapporteur</i>	Université Blaise Pascal - Clermont-Ferrand
M. Le Maître O.		Université d'Evry Val d'Essonne
M. Le Riche R.	<i>Encadrant</i>	CNRS/Ecole des Mines de Saint Etienne
M. Souza de Cursi E.	<i>Directeur de thèse</i>	INSA de Rouen
M. Breier E.	<i>Invité</i>	Breier S.A.S.

Remerciements

En tout premier lieu, je tiens à remercier le CNPq (Conseil National pour le Développement Scientifique et Technologique) du Ministère de la Science et Technologie du Brésil, pour la bourse d'études.

Merci aussi aux membres du jury qui ont accepté de juger ce travail, en particulier les rapporteurs pour l'analyse attentive du manuscrit.

J'adresse mes remerciements au Laboratoire de Mécanique de Rouen (LMR) et au Département de Mécanique de l'INSA de Rouen pour les moyens matériels mis à ma disposition (bureau, ordinateur, logiciels, téléphone, fax, photocopieuse, etc.), et toutes les personnes qui m'ont apporté leur aide et sympathie, plus particulièrement Agnès Dalle-Quirion, Agnès Lepine, Didier Lemosse, Emmanuel Pagnacco, Fabrice Barbe, Leila Khalij, Franck Le Guillous et Sylviane Henocq.

Merci également à mes collègues doctorants du LMR pour la convivialité, l'encouragement et l'amitié, notamment Ramzi Rajhi, Aurélien Moreau, André Koscianski, Giovanni Gouveia et Amine Mohsine.

Merci à Eric Breier de m'avoir fourni les données sur les monopalmes de nage et à Olivier Le Maître pour les sources de son code de calcul fluide, et pour les discussions sur le couplage fluide-structure.

Merci à Eduardo Souza de Cursi pour m'avoir "ouvert les portes" pour que je puisse préparer une thèse en France et de m'avoir très bien accueilli lors de mon arrivée. Je lui en suis profondément reconnaissant.

Merci à Rodolphe Le Riche qui a encadré ma thèse avec soin et enthousiasme. Sa connaissance et sa créativité scientifique, ainsi que son esprit critique m'ont beaucoup apporté tout au long de cette période. Je lui témoigne mon admiration et aussi mon amitié.

Merci à Eliane, pour son soutien et sa tendresse.

Merci à ma famille pour l'encouragement qui venait de l'autre côté de l'Atlantique.

Merci à tous ceux que je n'ai pas cités, mais qui d'une façon ou d'une autre m'ont aidé pendant ce parcours.

Résumé

Dans ce travail, une méthode d'optimisation à coût fini, essentiellement locale, mais qui devient globale lorsque le nombre d'analyses croît est développée. La "globalisation" vient de ré-initialisations probabilisées de recherches locales prenant en compte les points de départ et de convergence passés. L'optimiseur local est une version améliorée de la méthode de Nelder-Mead, où les variables sont bornées, où les contraintes d'inégalité sont prises en compte par pénalisation adaptative, et où les dégénérescences du simplexe sont traitées par ré-initialisation. Cette méthode, appelée *Globalized and Bounded Nelder-Mead* (GBNM), est testée sur des fonctions multimodales et des problèmes de conception de stratifiés composites. Puis, des applications plus complexes sont traitées avec GBNM : l'optimisation de la raideur de flexion et l'identification des positions des sauts de plis de monopalmes de nage.

Abstract

In this work a fixed cost essentially local optimization method, which becomes global if the number of evaluations increases is developed. "Globalization" is achieved by probabilistic restarts based on past searches. An improved Nelder-Mead method is the local optimizer. Improvements concern the variables which are bounded, the nonlinear inequality constraints which are taken into account by adaptive penalization, and the failures by simplex degeneration which are detected and handled through restart. The resulting method is called *Globalized and Bounded Nelder-Mead* (GBNM). Numerical experiments on multimodal test functions and composite laminated design problems illustrate the performance of the method as compared to an evolutionary optimizer. More complex applications are dealt with GBNM : the flexural stiffness optimization and the identification of the ply drop positions of swimming monofins.

Table des matières

Principaux symboles et abréviations	1
Objectifs et organisation du mémoire	3
I La méthode d'optimisation GBNM	5
1 Introduction et étude bibliographique	6
1.1 Définitions	7
1.2 Méthodes d'optimisation globale	9
1.2.1 Méthodes énumératives	9
1.2.2 Recherches aléatoires	10
1.2.3 Méthodes de regroupement (<i>clustering</i>)	10
1.2.4 Méthodes de descente généralisées	11
1.2.5 Méthodes évolutionnaires	12
1.2.6 Généralités sur les méthodes d'optimisation globale	14
1.3 Coopération entre les recherches locale et globale	14
1.4 Algorithmes locaux d'ordre zéro	15
1.4.1 Méthodes de recherches par motifs généralisés (<i>generalized pattern search methods</i> , GPS)	16
1.4.2 Directions conjuguées (algorithme de Powell)	17
2 Composition de GBNM (Globalized and Bounded Nelder-Mead)	18
2.1 Ré-initialisation probabilisée d'une recherche locale	19

2.2	Un algorithme de Nelder-Mead amélioré	23
2.2.1	L'algorithme de Nelder-Mead	23
2.2.2	Détection et traitement des dégénérescences	28
2.2.3	Prise en compte des bornes	29
2.3	Assemblage des améliorations : GBNM	30
2.4	Analyse de l'algorithme globalisé	32
2.4.1	Contrôle de l'étalement des noyaux Gaussiens, α	32
2.4.2	Centrage des noyaux	34
2.4.3	Contrôle de l'erreur d'échantillonnage : nombre de points aléatoires, N_r	40
2.5	Prise en compte des contraintes par pénalisation linéaire et adaptative	41
3	Applications à des fonctions test analytiques	43
3.1	Minimisation de la fonction de Rosenbrock et dégénérescence sur les bornes	43
3.2	Minimisation de la fonction de Griewank en dimension 12	46
3.3	Minimisation de fonctions sous contraintes	48
4	Application aux plaques composites stratifiées	52
4.1	Problèmes sans contraintes	54
4.1.1	Maximisation de charge de flambement	54
4.1.2	Maximisation de module d'élasticité longitudinal	56
4.2	Problèmes avec contraintes	57
4.2.1	Maximisation du module d'élasticité longitudinal sous les contraintes du module de cisaillement et du coefficient de Poisson	57
4.2.2	Maximisation de charge de flambement sous les contraintes du critère de rupture de Hoffman et des coefficients de dilatation thermique	58
5	Conclusions de la Partie I	60

II	Optimisation de monopalmes de nage 2D par GBNM	62
6	Introduction et étude bibliographique	63
6.1	Nage d'animaux marins	65
6.2	Objectifs de la Partie II	67
7	Une représentation bidimensionnelle de la palme et du nageur	69
7.1	Calcul de l'écoulement	69
7.2	Grandeurs caractéristiques et normalisation	70
7.2.1	Grandeurs caractéristiques	70
7.2.2	Normalisation	71
7.3	Cinématiques de la palme et du nageur	72
7.3.1	Modèle de nageur	72
7.3.2	Modèle de palme et couplage fluide-structure	75
8	Optimisation	81
8.1	Critères d'optimisation	81
8.2	Etudes paramétriques	83
8.3	Résultats de l'optimisation et discussion	88
9	Conclusions de la Partie II	93
III	Identification de monopalmes 3D par GBNM	95
10	Introduction	96
11	Traduction de la monopalme 2D en une structure 3D	97
11.1	Méthodologie	97
11.2	Résultats de l'identification d'épaisseurs pour le passage 2D \rightarrow 3D	99
11.2.1	Identification en petits déplacements	102
11.2.2	Identification en grands déplacements	102

11.2.3	Identification en dynamique	105
11.2.4	Formulation mixte (statique-dynamique) de l'identification	105
12	Identification des positions des sauts de plis lors d'un changement de tissu dans la fabrication des monopalmes	109
12.1	Méthodologie	110
12.2	Résultats et discussion	113
12.2.1	Monopalme "souple"	114
12.2.2	Monopalme "dureté moyenne"	122
13	Conclusions de la Partie III	125
	Récapitulatif et perspectives	126
	Bibliographie	128
A	Adaptive linear penalty as generalized Lagrangian	136
B	Mise en œuvre de GBNM dans LAMKIT	140
C	Comparaison entre GBNM et la fonction <i>fmincon</i> de Matlab	143
C.1	Fonction de Rosenbrock	144
C.2	Fonction de Griewank en dimension n=12	144
C.3	Fonction de Fletcher et Powell de dimension n=4	145
C.4	Fonction Test 1 sous contraintes	147
C.5	Fonction Test 2 sous contraintes	148
D	Présence de minima locaux en conception de composites - Exemples particuliers	150
E	Un modèle d'écoulement parfait et instationnaire autour d'un corps mince	157
E.1	Écoulement potentiel	157
E.2	Sillage	159

E.2.1	Emission	159
E.2.2	Evolution du sillage	159
E.3	Résolution du problème fluide	160
E.4	Efforts exercés par le fluide	160
F	Equations des n barres liées élastiquement	162
G	Intégration des équations dynamiques de la palme	165
H	Influence de la discrétisation des barres, de l'incrément de temps et du temps total d'une simulation de la palme 2D	169
I	Flexural stiffness identifiability	172
J	Correction des raideurs 2D optimales	175
J.1	Obtention du coefficient de charge β	177
J.2	Correction de la répartition de raideurs 2D	178
K	Analyse modale du modèle de barres 2D	180
L	Dessins et propriétés matériaux de monopalmes	184

Principaux symboles et abréviations

f : fonction coût (Partie I) ; fréquence (Parties II et III)

g_i : contraintes d'inégalité

n : nombre de variables dans un problème d'optimisation

x^* : un des optima globaux

\hat{x}^* : une estimation de x^*

$p(x)$: densité de probabilité d'avoir exploré une région autour d'un point x

$\phi(x)$: densité de probabilité de ne pas avoir exploré une région autour d'un point x

N_r : nombre de points aléatoires utilisé pour déterminer celui qui a la plus faible probabilité p

α : paramètre qui contrôle l'étalement des noyaux Gaussiens

λ_i : paramètre de pénalisation pour traiter les contraintes d'inégalité

$\bar{\lambda}_i$: paramètre de pénalisation stabilisé moyen

E_x : module d'élasticité suivant la direction x

E_y : module d'élasticité suivant la direction y

G_{xy} : module de cisaillement, estimé par un essai de cisaillement dans les directions x et y

ν_{xy} : coefficient de Poisson, estimé par un essai de traction suivant la direction x

E_1 : module d'élasticité longitudinal (suivant la direction des fibres) d'un pli composite

E_2 : module d'élasticité transversal d'un pli composite

G_{12} : module de cisaillement d'un pli composite, estimé par un essai de cisaillement dans les directions 1 et 2

ν_{12} : coefficient de Poisson d'un pli composite, estimé par un essai de traction dans la direction 1

α_{t_1} , α_{t_2} : coefficients de dilatation thermique d'un pli composite suivant les directions 1 (longitudinal) et 2 (transversal), respectivement

α_{t_x} , α_{t_y} : coefficients de dilatation thermique d'un stratifié composite suivant les directions x et y , respectivement

ρ : masse volumique

ρ_L : masse linéique

θ_i : orientation des fibres d'un pli de composite unidirectionnel (Partie I) ; angle entre les segments qui représentent le modèle simplifié de nageur et de monopalme (Partie II)

$\dot{\theta}_i$: dérivée de θ_i par rapport au temps (vitesse angulaire) (Partie II)

$\ddot{\theta}_i$: dérivée de $\dot{\theta}_i$ par rapport au temps (accélération angulaire) (Partie II)

S_t : nombre de Strouhal

t : temps

T_c : temps caractéristique du problème (utilisé pour la normalisation dans le calcul fluide)

T_s, T_f : temps initial et final pour le calcul des forces et puissances moyennes

U_c : vitesse caractéristique

L_c : longueur caractéristique

L_s : longueur de monopalme

U_∞ : vitesse du nageur ou vitesse de l'écoulement

E_c : énergie cinétique

E_p : énergie potentielle

C_i : raideur des ressorts en flexion

F_{x1f}, F_{y1f}, M_{if} : efforts fluide sur le modèle de monopalme 2D

F_{x1}, F_{y1}, M_1 : efforts du pied du nageur sur la monopalme (modèle 2D)

\bar{P}_{fx} : puissance propulsive moyenne d'une monopalme

\bar{P}_f : puissance moyenne transmise par le fluide à la monopalme

\bar{P}^{min} : puissance limite fournie à la palme

ν_P : rendement en puissance d'une monopalme

J : fonction erreur

J_{static} : fonction erreur associée à la déformée statique

J_{freq} : fonction erreur associée à la fréquence propre

μ : facteur de pondération entre les critères d'identification statique et dynamique

ω : pulsation

[. ; .] : notation d'intervalle de validité des variables

$\| . \|$: norme euclidienne

GBNM : la méthode d'optimisation "Globalized and Bounded Nelder-Mead"

EA : algorithme évolutionnaire

EF : analyse par éléments finis

Objectifs et organisation du mémoire

De nombreux problèmes d'optimisation en mécanique sont caractérisés par l'existence d'optima locaux, des difficultés dans l'estimation des sensibilités (quand elles existent), par la présence de contraintes d'optimisation et de limites sur les variables et par un coût numérique d'évaluation du modèle non négligeable. Le propos, pragmatique, de ce mémoire est de considérer des problèmes d'optimisation qui présentent les caractéristiques mentionnées ci-dessus. A cet effet une méthode d'optimisation pour variables continues a été développée. La première Partie du mémoire (Chapitres 1 à 5) décrit cette méthode, nommée GBNM (Globalized and Bounded Nelder-Mead), puis l'applique à des fonctions tests analytiques et à des problèmes de conception de stratifiés composites.

L'optimisation d'un système plus complexe, un système couplé fluide-nageur-palme de nage, au moyen de GBNM est déclinée dans les Parties II et III.

Malgré les progrès conjoints de l'informatique et des méthodes numériques, il reste fréquent que l'optimisation de systèmes 3D ne soit pas envisageable à cause d'un coût numérique prohibitif, en particulier lors de couplages multiphysiques. Des simplifications sont alors réalisées afin de rendre le modèle numériquement "léger" et apte à entrer dans une boucle d'optimisation. En particulier, considérer un modèle en 2D plutôt qu'en 3D pour l'optimisation est une simplification courante. Ensuite la solution 2D optimisée doit être traduite en un homologue 3D. Les seconde et troisième Parties de ce mémoire traitent cette problématique autour de l'optimisation d'une monopalme de nage. Dans la seconde Partie (Chapitres 6 à 9) un modèle de monopalme bidimensionnelle, infiniment mince, plongée dans un écoulement instationnaire avec fluide non visqueux et incompressible est optimisé. Avant l'optimisation, les enjeux concernant la modélisation de la nage avec monopalmes sont présentés, les simplifications du modèle sont justifiées, et le problème d'optimisation est formulé en termes de puissance propulsive et puissance totale.

La troisième Partie (Chapitres 10 à 13) traduit les résultats de l'optimisation du modèle 2D en

une monopalme 3D. La procédure proposée est une identification où les comportements statique et dynamique du système 2D servent de cible à la détermination de la structure 3D. Une seconde application de la même démarche suit dans la Partie III : il s'agit de l'identification des positions des sauts de plis lors d'un changement de tissu composite dans la fabrication de monopalmes.

Le fil conducteur du mémoire est l'algorithme d'optimisation GBNM qui sert

- à optimiser des stratifiés composites en Partie I ;
- à la modélisation de la palme dans l'écoulement (Partie II) par la minimisation du résidu des équations d'équilibre ;
- à l'optimisation de la palme 2D ;
- et à la minimisation des fonctions erreurs des identifications de palmes 3D en Partie III.

Le mémoire a été écrit de façon à ce que les trois parties puissent être lues de manière indépendante.

Première partie

La méthode d'optimisation GBNM

Chapitre 1

Introduction et étude bibliographique

Cette première partie du mémoire présente le développement de la méthode d'optimisation GBNM (*Globalized and Bounded Nelder-Mead*). Avant de la présenter, une exposition de concepts de base est nécessaire. Ainsi, ce chapitre débute en Section 1.1, avec quelques définitions sur les problèmes d'optimisation. Ensuite, une étude bibliographique en trois sections des stratégies liées à GBNM est présentée. La Section 1.2 porte sur les méthodes d'optimisation globale, la Section 1.3 sur la coopération entre recherches locales et globales, et la Section 1.4 sur les algorithmes d'optimisation locale d'ordre zéro. Seule la méthode de Nelder-Mead, qui est une composante à part entière de GBNM, est présentée dans le Chapitre 2 qui décrit, en détails, la méthode GBNM. Les Chapitres 3 et 4 présentent, respectivement, des applications à des fonctions test analytiques et à des plaques composites stratifiées.

1.1 Définitions

Un problème d'optimisation est usuellement formulé comme un problème de minimisation¹, et écrit sous la forme

$$\left\{ \begin{array}{l} \min_x f(x) , \\ \text{tel que,} \\ g_i(x) \leq 0 , i = 1, \dots, m , \\ h_j(x) = 0 , j = 1, \dots, p , \\ x \in \mathcal{S} \subset \mathcal{R}^n , \end{array} \right. \quad (1.1)$$

où f est la fonction (scalaire) à minimiser, appelée fonction coût ou fonction objectif, x représente le vecteur des variables d'optimisation, g_i sont les contraintes d'inégalité et h_j les contraintes d'égalité, et \mathcal{S} est l'espace des variables (appelé aussi espace de recherche). \mathcal{S} indique quel type de variables sont considérées : réelles, entières, mixtes (réelles et entières dans un même problème), discrètes, continues, bornées, etc.

Un point x_A est appelé un point admissible si $x_A \in \mathcal{S}$ et si les contraintes d'optimisation sont satisfaites : $g_i(x_A) \leq 0 , i = 1, \dots, m$ et $h_j(x_A) = 0 , j = 1, \dots, p$. La solution de (1.1) est l'ensemble des optima $\{x^*\}$. Dans ce mémoire on abordera des problèmes d'optimisation en variables réelles et bornées, avec des contraintes d'inégalité.

x^* est un minimum global de f si et seulement si $f(x^*) \leq f(x) \forall x \in \mathcal{S}$, et x^* est un minimum local de f si et seulement si $f(x^*) \leq f(x) \forall x \in \mathcal{S} / \|x - x^*\| \leq \epsilon, \epsilon > 0$. La Figure 1.1 présente un exemple d'une fonction à une variable, avec des minima locaux et un minimum global. Parmi les minima locaux, celui qui possède la plus petite valeur de f est le minimum global.

Une fonction multimodale présente plusieurs minima (locaux et globaux), et une fonction unimodale n'a qu'un minimum, le minimum global. La Figure 1.2 montre une fonction multimodale à deux variables.

On appelle méthode (ou algorithme ou recherche) locale celle qui converge vers un minimum local. Les recherches locales partent usuellement d'un point initial x_0 avec un pas initial ρ_0 . Ces paramètres vont conditionner la descente d'une des vallées de la fonction (cf. schéma en Figure 1.1). De nombreuses méthodes locales existent. Les plus anciennes et les plus utilisées sont les

¹Un problème d'optimisation peut aussi se présenter comme une maximisation (par exemple, maximiser la résistance mécanique, le chargement de flambage, le profit etc.), mais on peut toujours le transformer en un problème de minimisation, en changeant, par exemple, f en $-f$.

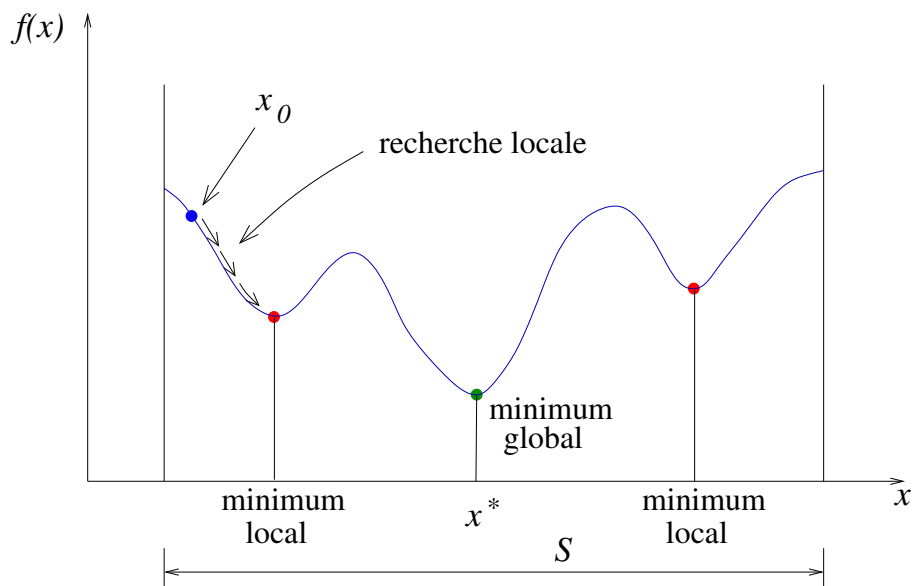


FIG. 1.1 – Minima locaux et minimum global d’une fonction à une variable.

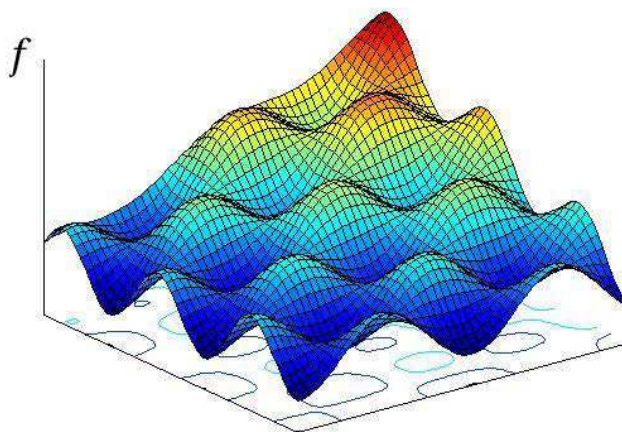


FIG. 1.2 – Une fonction multimodale à deux variables.

méthodes où la direction de descente est déduite des dérivées de la fonction (méthode de la plus forte pente, méthode de Newton, méthodes de gradient conjugué, méthodes quasi-Newtoniennes, cf. [66, 14]). Ici on s’intéressera aux méthodes locales sans calcul de dérivée (méthodes directes, par motifs, méthodes de directions conjuguées, cf. Section 1.4).

Les méthodes globales ont pour objectif d’atteindre un ou plusieurs optima globaux.

Typiquement, le coût machine d’une optimisation est conditionné par le nombre d’évaluations de la fonction objectif. Si l’on veut, par exemple, optimiser un système mécanique modélisé par

éléments finis, le temps de calcul sera principalement la somme des temps de simulations utilisées par l’optimiseur.

Il n’existe pas un algorithme optimal pour tous les problèmes (cf. “No Free Lunch Theorem” (NFL) ([92])), et la plupart des méthodes possèdent des paramètres à régler. Le NFL prouve que chaque méthode d’optimisation faisant des progrès sur une classe de fonctions régresse sur une autre classe. Le choix de la méthode à utiliser et le réglage des paramètres restent liés au problème à optimiser.

1.2 Méthodes d’optimisation globale

Différentes classifications des méthodes globales sont trouvées dans la littérature. Une classification, basée sur [47], est présentée dans cette section. Elle n’a pas pour objectif d’être complète, ni de montrer tous les détails des méthodes, mais plutôt de mettre en évidence les caractéristiques de chaque groupe.

1.2.1 Méthodes énumératives

Bien entendu, dans un espace de recherche fini, dont la taille est d’autant plus petit que le calcul de f est numériquement long, la fonction coût peut être évaluée en chaque point de l’espace pour localiser les optima.

Plus généralement, si la fonction f à minimiser possède une constante de Lipschitz L telle que,

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|, \forall x_1, x_2 \in \mathcal{S}, \quad (1.2)$$

les méthodes énumératives peuvent détecter des régions qui ne contiennent pas l’optimum (cf. Figure 1.3), et ainsi les exclure pour le reste de la recherche. De plus, l’erreur en f commise en approximant x^* (inconnu) par \hat{x}^* (estimé par l’optimiseur) est bornée,

$$|f(\hat{x}^*) - f(x^L)| \leq L\|\hat{x}^* - x^L\|, \quad (1.3)$$

où x^L est le point de \mathcal{S} non exclu et non exploré le plus éloigné de \hat{x}^* . Un exemple de méthode énumérative appliquée aux fonctions Lipschitziennes est l’algorithme DIRECT ([73]). DIRECT construit itérativement un pavage de \mathcal{S} . f est évaluée au centre de chaque pavé. Les pavés sont

progressivement divisés en considérant tous les compromis entre la valeur de f au centre de chaque pavé et la taille du pavé. Ceci équivaut à considérer (sans les expliciter) toutes les valeurs de L possibles.

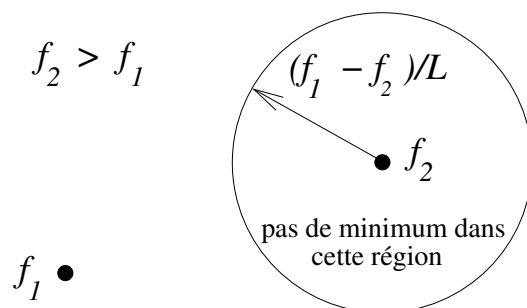


FIG. 1.3 – Détection de région qui ne contient pas le minimum global. f est une fonction Lipschitzienne de constante L .

1.2.2 Recherches aléatoires

Dans ce groupe de méthodes on peut avoir des recherches aléatoires pures, qui consistent à tirer un point, au hasard, à chaque itération. La fonction coût est évaluée en ce point, et s'il y a une amélioration, ce point et la fonction correspondante sont enregistrés, et le processus continue. Les recherches aléatoires peuvent aussi être associées aux recherches locales. Ainsi des points au hasard sont pris pour ré-initialiser des recherches locales. Ces ré-initialisations sont susceptibles de converger plusieurs fois vers les mêmes minima locaux. De plus, il n'y a pas de discrimination entre régions prometteuses ou non prometteuses.

1.2.3 Méthodes de regroupement (*clustering*)

Ces méthodes estiment la position du minimum global en trouvant des minima locaux ([86]). Elles sont composées de plusieurs étapes. Premièrement l'espace de recherche est échantillonné. Ensuite, au moyen d'analyses de proximité (*clustering techniques*), les points sont regroupés par régions prometteuses. Finalement des recherches locales sont exécutées et celles qui convergent vers des régions déjà explorées sont abandonnées. Les analyses de proximité divisent l'espace de recherche en régions, et utilisent, entre autres informations, les valeurs de la fonction ou de la matrice Hessienne ([89]) des points échantillonnés. Un désavantage de ces méthodes est qu'elles sont peu performantes pour les fonctions ayant de nombreux minima. Dans ces cas, en effet, un

grand nombre d'échantillons est nécessaire.

1.2.4 Méthodes de descente généralisées

Dans ces méthodes un algorithme d'optimisation local est employé itérativement, et la direction de descente est modifiée de manière à éviter la convergence vers un optimum local déjà trouvé. Une façon de réaliser ceci est de pénaliser la fonction f au fur et à mesure que l'optimiseur s'approche d'un minimum local x_i^* déjà trouvé. Par exemple,

$$f_P = f + P(x_1^*, x_2^*, \dots, x_c^*), \quad (1.4)$$

où

$$P = \sum_{i=1}^c \frac{K_i}{|x - x_i^*|^\alpha}, \quad (1.5)$$

et c est le nombre de minima locaux trouvés. Un désavantage de la pénalisation directe de f est que l'on peut introduire des optima locaux dans f_P qui n'existent pas dans f (cf. Figure 1.4).

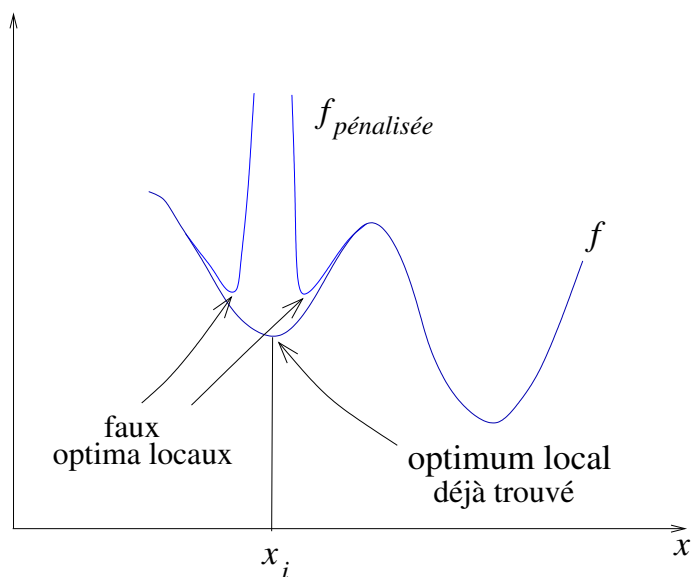


FIG. 1.4 – Illustration des faux optima locaux introduits par la pénalisation (Equations (1.4) et (1.5)).

Une autre manière d'explorer d'autres régions après avoir trouvé des minima locaux est d'utiliser les méthodes de tunnel ([55]). Elles cherchent des points de départ dans des vallées différentes. Cette méthode présente deux phases qui sont exécutées en alternance. Dans la première phase une recherche locale est effectuée sur la fonction f , à partir d'un point de départ x_0^i . Elle conver-

gera (on le suppose) vers un minimum (local) x_i^* . Ensuite la phase de “tunnel” se déroule. Un nouveau point de départ x_0^{i+1} est cherché dans une autre vallée, à l’aide d’une fonction tunnel T . x_0^{i+1} doit vérifier la condition $f(x_0^{i+1}) < f(x_i^*)$ pour garantir qu’il se situera dans une vallée plus basse, donc différente des précédentes. x_0^{i+1} doit satisfaire $T(x_0^{i+1}) < 0$, la fonction tunnel étant, par exemple,

$$T(x) = \frac{f(x) - \min_{i=1,c} f(x_i^*)}{\prod_{i=1}^c \|x - x_i^*\|^{2\alpha_i}}, \quad (1.6)$$

où α_i est un paramètre qui contrôle la “force” de répulsion autour de x_i^* .

La Figure 1.5 schématise la méthode de tunnel.

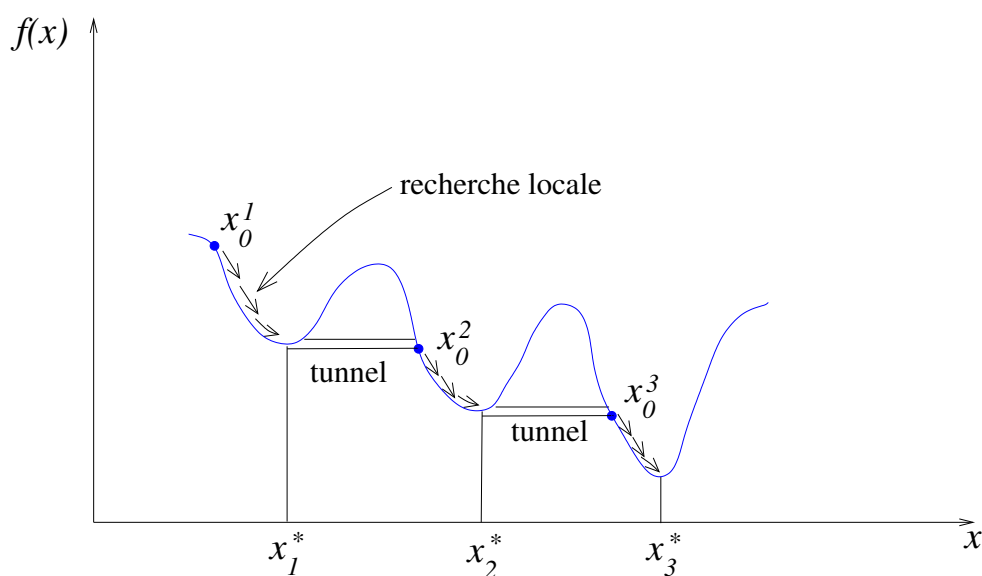


FIG. 1.5 – Schéma des méthodes de tunnel.

Ces méthodes souffrent typiquement de mauvais conditionnements numériques liés à $P(x)$ et $T(x)$.

1.2.5 Méthodes évolutionnaires

Les méthodes évolutionnaires sont des méthodes d’optimisation probabilistes qui sont souvent présentées par analogie avec la théorie de la sélection naturelle de Darwin. Les individus les plus performants d’une population ont une plus grande probabilité de survivre et de se reproduire, en donnant des descendants encore mieux adaptés. La correspondance avec la minimisation d’une fonction est la suivante : un “individu” est un point x , la “population” est un ensemble de points,

et “la performance d’un individu” x est évaluée par sa valeur de fonction coût, $f(x)$. En partant d’un échantillonnage de l’espace de recherche (une population initiale), ces méthodes réalisent des tirages de nouvelles populations (des “générations”) à l’aide d’opérations de sélection, croisement et mutation ([5]). Les différents algorithmes évolutionnaires se distinguent par la mise en œuvre de ces opérations. Historiquement, ils ont été développés indépendamment en trois variantes ([6]) : algorithmes génétiques, stratégies d’évolution et programmation évolutive.

Un exemple de structure d’algorithme évolutionnaire est la suivante :

1. initialisation de la population $P(t)$, $t = 0$ (au hasard, par exemple) ;
2. évaluation de la population courante $P(t)$: calcul de la fonction coût pour tous les points de la population ;
3. *sélection* de certains individus de $P(t)$ en fonction de leur fonction coût f (par exemple les meilleurs) : les “parents” $P_0(t)$;
4. *croisement* entre parents et création des “enfants” (des nouveaux points) $\Rightarrow P_1(t)$;
5. *mutation* (perturbation aléatoire) des individus de $P_1(t) \Rightarrow P_2(t)$;
6. évaluation de $P_2(t)$;
7. formation de la nouvelle population $P(t + 1)$ à partir des individus de $P(t)$ et de $P_2(t)$, par exemple en choisissant les meilleurs individus appartenant à $P(t)$ et à $P_2(t)$;
8. $t \leftarrow t + 1$;
9. si le critère d’arrêt n’est pas satisfait, retour en 3.

Une quatrième famille de méthodes évolutionnaires, plus distante de la métaphore biologique et au formalisme plus rigoureux, est la famille des algorithmes d’optimisation statistique. A partir des individus sélectionnés (étape 3), une densité de probabilité de présence de l’optimum est estimée. Cette densité servira à générer des nouveaux individus (étapes 4 et 5). Ces méthodes peuvent être considérées comme des algorithmes évolutionnaires sans opérateurs.

Les algorithmes évolutionnaires permettent de réaliser des recherches globales, formellement grâce à l’étape 5. Leur intérêt pratique réside dans leur applicabilité à une très large gamme de problèmes car ils n’utilisent que la fonction coût (pas de dérivées) et peuvent accommoder des variables mixtes. Cependant, pour trouver un optimum, ils ont besoin d’un grand nombre d’évaluations de la fonction coût. Ainsi, ils ne sont considérés directement utilisables que pour les problèmes où le calcul de la fonction ne nécessite pas de ressources importantes.

1.2.6 Généralités sur les méthodes d'optimisation globale

De manière synthétique, on notera que les méthodes d'optimisation globales présentent toujours deux composantes. Une composante *globale* ou *exploratrice*, qui permet d'explorer l'espace de recherche, et une composante *locale* ou *exploitatrice*, qui vise à exploiter les informations (relation entre x et f) recueillies.

Les méthodes d'optimisation globales sont, souvent, stochastiques, à travers leur composante exploratrice. Ainsi on ne peut pas les juger à partir d'une seule exécution. Pour les qualifier, il est nécessaire d'effectuer une analyse statistique (moyenne, écart-type, etc.) de plusieurs exécutions.

GBNM (cf. Chapitre 2) appartient en partie à la classe des méthodes locales directes, aux méthodes aléatoires, et aux méthodes de regroupement. Une étude bibliographique concernant plus particulièrement GBNM est présentée dans les deux sections ci-après.

1.3 Coopération entre les recherches locale et globale

L'existence de solutions locales dans les problèmes d'optimisation a engendré d'importants efforts pour le développement de méthodes d'optimisation globales ([86, 5]). Le coût numérique élevé des optimiseurs globaux est l'origine d'efforts subséquents pour augmenter la vitesse de la recherche, soit en ajoutant de la connaissance spécifique, soit en combinant des algorithmes locaux (efficaces) avec des algorithmes globaux. Il existe plusieurs manières d'associer les recherches locales et globales. Quelques cas ont été déjà présentés en Section 1.2 : recherches aléatoires combinées avec des recherches locales, méthodes de regroupement et méthodes de descente généralisées. La façon la plus simple de réaliser cette association est d'effectuer les recherches en série, c'est-à-dire qu'une optimisation globale à coût limité est d'abord exécutée, ensuite cette solution est raffinée par une recherche locale. Un exemple d'une méthode hybride en série est donné en [80], où la méthode du recuit simulé, qui est l'optimiseur global, est couplée avec une programmation quadratique séquentielle et l'algorithme de Nelder-Mead. De nombreuses recherches locales-globales en parallèle ont été proposées ([86, 21, 71]) et analysées ([85, 26]). Dans ces cas, les itérations des algorithmes locaux et globaux sont entrelacées. De plus, on peut diviser les algorithmes hybrides parallèles en deux catégories : ceux où la recherche locale converge, et ceux où la recherche locale est arrêtée prématurément. Des algorithmes génétiques mimétiques ([69]) et des méthodes avec des ré-initialisations multiples (e.g. ré-initialisation déterministe dans [3], ré-

initialisation aléatoire dans [35]) sont des exemples du premier cas. Les algorithmes du deuxième cas sont, en général, basés sur les méthodes de regroupement dans lesquelles les recherches locales qui arrivent à des régions déjà explorées sont abandonnées ([85, 34, 89]).

Lorsque l'on considère l'optimisation d'un problème réel de l'ingénieur, une situation courante est celle où le nombre d'analyses est limité, l'existence et le nombre de minima locaux sont inconnus, et la possibilité d'achever ne serait-ce que deux recherches locales est incertaine. Néanmoins, parvenir à des résultats globaux reste un des objectifs de l'optimiseur. Cela se produit typiquement quand on a affaire à une fonction inconnue et que l'on souhaite se limiter à environ 1000 évaluations de la fonction coût. Dans de tels cas, une méthode locale-globale basée sur des ré-initialisations est la stratégie la plus sûre car elle permet de finir au moins une recherche locale dans la durée impartie.

1.4 Algorithmes locaux d'ordre zéro

Les algorithmes d'optimisation d'ordre zéro sont ceux qui ne requièrent que l'évaluation de la fonction, mais pas celle des dérivées. Bien que les algorithmes évolutionnaires ou les recherches aléatoires pures, par exemple, soient des algorithmes d'ordre zéro, cette section n'abordera que les algorithmes locaux. Les méthodes d'ordre zéro sont importantes car, en pratique, un très grand nombre de fonctions à optimiser ne sont pas dérivables et fréquemment même pas continues. Les méthodes locales d'ordre zéro les plus répandues sont les recherches directes (*direct search*). Pour préciser le terme "recherche directe", la définition considérée dans ce travail est celle utilisée dans [93] : un algorithme de recherche directe i) utilise seulement les valeurs de la fonction, et ii) n'approxime pas de gradient. Le critère ii) exclue, par exemple, les méthodes qui utilisent les différences finies pour évaluer le gradient. Ainsi, les méthodes de recherche directe peuvent accepter des nouvelles itérations qui conduisent à une simple amélioration de la fonction coût, ce qui contraste avec les conditions de Armijo-Goldstein-Wolfe pour les méthodes quasi-Newtoniennes, qui requièrent qu'une condition de descente suffisante soit satisfaite ([58]).

Les méthodes de recherche directe ont été proposées pendant les années 50 et 60. Des exemples des premières méthodes sont les algorithmes de Box (1957), Hooke et Jeeves (*pattern search method*, 1961), Spendley, Hext et Himsworth (1962), Powell (1964), et l'algorithme de Nelder-Mead (1965) ([93]). Depuis leurs publications, les algorithmes de recherche directe sont utilisés avec succès dans plusieurs domaines, et sont populaires particulièrement en chimie, ingénierie des

procédés et médecine. Ils sont réputés pour être des algorithmes simples, robustes, et efficaces pour les problèmes d'optimisation en variables réelles, sans contraintes, où les fonctions sont bruitées ([63]). Malgré l'efficacité et la popularité auprès des praticiens, pendant plusieurs années la communauté scientifique d'optimisation ne s'est pas intéressée à ces méthodes car, à part quelques exceptions comme l'algorithme de Powell, des propriétés théoriques de convergence n'ont pas été prouvées. L'intérêt a ressurgi avec les publications [83] et [17], qui présentent une méthode appelée recherche multidirectionnelle, adaptée pour être efficace en calcul parallèle et qui possède des propriétés de convergence ([83]).

Les méthodes de recherche directe peuvent être divisées en trois groupes : les méthodes de recherches par motifs généralisés (*generalized pattern search methods*, GPS), les méthodes des directions conjuguées (algorithme de Powell et ses variantes), et les méthodes basées sur la figure géométrique d'un simplexe (méthode de Nelder-Mead et ses variantes).

Les méthodes de recherches par motifs généralisés et de Powell sont exposées ci-dessous. La méthode de Nelder-Mead, optimiseur local de GBNM, est discutée en Section 2.2.1.

1.4.1 Méthodes de recherches par motifs généralisés (*generalized pattern search methods*, GPS)

Les *generalized pattern search methods (GPS)* ([84]) sont une généralisation de la méthode de Hooke et Jeeves (1961). La spécificité de ces méthodes est que les directions de recherche ne changent pas avec les itérations. Les GPS sont caractérisées par une série de déplacements exploratoires autour du point courant. Ces déplacements forment des motifs qui présentent une disposition invariable (*patterns*). A chaque itération la fonction objectif est évaluée sur les points du motif. Si une amélioration est trouvée, le point associé est accepté comme nouveau point courant, et la taille du prochain motif est conservée ou augmentée. Sinon, la taille du nouveau motif, généré autour de l'ancien point courant, est réduite. Les GPS présentent des propriétés de convergence robustes pour des fonctions continues, différentiables et bornées ([84, 19]). Dans [56] et [57] les méthodes GPS sont appliquées à des problèmes avec bornes et avec des contraintes non-linéaires, respectivement.

Des applications intéressantes en mécanique peuvent être trouvées, comme par exemple celles données dans [10] où une méthode par motifs est associée à la technique de surface de réponse

([78]), la surface étant interpolée par des moindres carrés mobiles ([44]). L'intérêt d'employer les surfaces de réponse est de remplacer la fonction réelle coûteuse par une fonction construite à partir d'un nombre limité d'analyses et plus rapide à évaluer. La stratégie développée est testée dans la minimisation de la fonction de Rosenbrock et appliquée à des problèmes de mise en forme de métaux.

1.4.2 Directions conjuguées (algorithme de Powell)

L'algorithme de Powell ([74, 75]) réalise des minimisations unidimensionnelles suivant des directions conjuguées. Deux vecteurs (ou directions) s_1 et $s_2 \in \mathcal{R}^n$ sont *conjugués* vis à vis d'une matrice définie positive et symétrique A si $s_1^T A s_2 = 0$.

L'algorithme est motivé par la propriété selon laquelle le minimum d'une fonction quadratique² f à n variables est trouvé en n minimisations unidimensionnelles successives suivant n directions conjuguées, et par le théorème suivant (cf. preuve en [11]). De plus, des directions conjuguées peuvent être construites au moyen de ce théorème.

Théorème. *Soit f une fonction quadratique, s_1, \dots, s_p , p directions conjuguées, x^p et y^p les résultats de p minimisations unidimensionnelles suivant les directions s_i en partant de x^0 et y^0 , respectivement. Alors la direction $s_{p+1} = y^p - x^p$ est conjuguée par rapport aux autres directions s_i , $i = 1, p$.*

Le détail de la méthode de construction des directions conjuguées n'est pas donné ici. Le lecteur intéressé consultera les références [74], [66] ou [11].

Les minimisations unidimensionnelles peuvent être accomplies, sans calcul de dérivées, par dichotomie ou par section du nombre d'or ([14]).

Les méthodes de recherche directe sont des méthodes locales, ainsi elles peuvent s'arrêter quand un minimum local est trouvé. Afin de les utiliser dans l'optimisation globale de fonctions multimodales il devient nécessaire de les associer à une stratégie d'exploration de l'espace de recherche.

²Une fonction quadratique f est celle qui peut être écrite sous la forme,

$$f(x) = \frac{1}{2}x^T A x + b^T x + c, \quad (1.7)$$

où A est une matrice définie positive et symétrique, b est un vecteur et c un scalaire. f quadratique a la caractéristique d'être unimodale.

Chapitre 2

Composition de GBNM (Globalized and Bounded Nelder-Mead)

La sous-méthode locale utilisée dans GBNM est basée sur l'algorithme classique de Nelder-Mead (cf. Section 2.2.1). Un attribut de l'algorithme de Nelder-Mead est ne pas requérir le calcul de gradient. Or le calcul de sensibilités est souvent une étape laborieuse et délicate dans les modèles de systèmes physiques complexes. Le second avantage de Nelder-Mead est d'être une méthode rapide et précise par rapport aux recherches stochastiques. Par exemple, les algorithmes génétiques sont numériquement coûteux quand on a besoin de résultats précis ([76]). Ces avantages de Nelder-Mead sont ceux, plus généralement, des algorithmes de recherche directe ([58, 84]).

Cependant la méthode classique de Nelder-Mead présente des inconvénients : elle s'applique à des variables sans bornes et à problèmes sans contraintes, elle s'arrête quand un minimum local est trouvé, et la recherche peut échouer par stagnation sur un point non-stationnaire (cf. Section 2.2.1). C'est pourquoi des améliorations de la méthode de Nelder-Mead sont proposées. GBNM est donc un algorithme pragmatique car

- il résout des problèmes contraints en variables réelles et bornées ;
- il ne nécessite pas de calcul de gradient car il s'appuie sur la méthode de Nelder-Mead ;
- il est une méthode locale pouvant devenir globale si le coût des analyses le permet (cf. Section 2.1). La “globalisation” de GBNM vient d'une procédure de ré-initialisation, qui utilise une densité de probabilité adaptative en conservant une mémoire des recherches locales passées.

Dans ce Chapitre, les composantes de l’algorithme GBNM sont successivement introduites. La stratégie de ré-initialisation est décrite dans la Section 2.1. L’optimiseur local est présenté dans la Section 2.2. L’assemblage des améliorations et l’organigramme de GBNM sont exposés dans la Section 2.3. La Section 2.4 discute le choix des paramètres de l’algorithme globalisé. Enfin, en Section 2.5, une pénalisation adaptative est décrite pour la prise en compte de contraintes non-linéaires $g_i \leq 0$.

2.1 Ré-initialisation probabilisée d’une recherche locale

Les optimiseurs locaux peuvent exécuter une recherche globale lorsqu’ils sont ré-initialisés en différents points du domaine. Les méthodes de ré-initialisation les plus simples sont l’utilisation d’une grille régulière de points ou la ré-initialisation aléatoire. Dans le premier cas, il est nécessaire de connaître combien de ré-initialisations seront effectuées pour calculer la taille de la grille. De plus, la seule information propagée d’une recherche locale à une autre est son point initial. Dans le deuxième cas, aucune connaissance acquise lors des recherches précédentes n’est utilisée pour les ré-initialisations ultérieures. Ainsi, des recherches locales similaires, convergeant vers le même optimum local sont susceptibles d’être répétées, ce qui entraîne un surcoût numérique important. Une motivation essentielle dans le développement de GBNM est le pragmatisme. Ainsi, il est considéré que, lors d’une optimisation, le temps de calcul d’une analyse et le temps total dont on dispose sont connus a priori. Le quotient de ces deux temps donne le nombre d’analyses disponibles pour l’optimisation. Ce nombre total d’analyses fixé sert donc de critère d’arrêt à GBNM. Il résulte de ce critère d’arrêt que le nombre de ré-initialisations est inconnu car le coût de chaque recherche locale est inconnu. Ainsi, les ré-initialisations par grilles ne peuvent pas être utilisées. La démarche de GBNM pour piloter les initialisations de recherches locales cumule les avantages des grilles (points initiaux couvrant l’espace de recherche) et de l’aléatoire (pas de nombre de ré-initialisations nécessaire) tout en offrant un meilleur cadre pour la capitalisation des informations obtenues lors des recherches locales précédentes. Il s’agit de construire une densité de probabilité spatiale de redémarrage de recherche locale. Plus précisément, une densité de probabilité de ne pas avoir exploré une région autour d’un point¹ x , $\phi(x)$, est construite. GBNM privilégiera les régions où ϕ est la plus importante pour relancer une recherche locale.

¹La dénomination “région autour d’un point”, certes plus compliquée que “point” lui est préférée ici car $\phi(x_0) > 0$, et non $\phi(x_0) = 0$, même si x_0 a déjà été utilisé comme point initial. ϕ traduit donc une connaissance du voisinage de x_0 .

Nous détaillons maintenant l'estimation de ϕ .

Connaissant un certain nombre de points déjà explorés, une densité de probabilité $p(x)$ d'avoir exploré une région autour d'un point² x déjà échantillonné est estimée par une méthode de fenêtres de Parzen Gaussiennes ([20]). Cette méthode peut être considérée comme une version lissée des techniques d'histogrammes, les histogrammes étant centrés sur les points échantillonnés. $p(x)$ est écrite comme

$$p(x) = \frac{1}{N} \sum_{i=1}^N p_i(x) , \quad (2.1)$$

où N est le nombre de points déjà échantillonnés, et p_i est la fonction densité de probabilité multidimensionnelle,

$$p_i(x) = \frac{1}{(2\pi)^{\frac{n}{2}} (\det(\Sigma))^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - x_i)^T \Sigma^{-1} (x - x_i)\right) , \quad (2.2)$$

n est la dimension (nombre de variables), et Σ est la matrice de covariance,

$$\Sigma = \begin{bmatrix} (\sigma^1)^2 & 0 \\ & \ddots \\ 0 & (\sigma^n)^2 \end{bmatrix} . \quad (2.3)$$

Les variances $(\sigma^j)^2$ sont estimées par

$$(\sigma^j)^2 = \alpha (x^{j,max} - x^{j,min})^2 , \quad (2.4)$$

où α est un paramètre positif qui contrôle l'étalement des Gaussiennes, et $x^{j,max}$ et $x^{j,min}$ sont les bornes supérieure et inférieure dans la direction j . Pour des raisons de simplicité de la méthode les variances sont gardées constantes et α est fixé conformément au raisonnement de la Section 2.4.1. Une perspective de développement de GBNM serait d'ajuster les σ^j pour prendre en compte les densifications locales des points, ce qui pourrait être fait en utilisant validation croisée et maximum de vraisemblance comme, par exemple, dans [31], Annexe A.

La densité de probabilité p est telle que $\int_{-\infty}^{\infty} p(x) dx = 1$, mais comme un domaine borné Ω

²Le même commentaire (¹) s'applique à p car $p(x_0) < 1$. Ce choix de fonctions ϕ et p privilégie le lissage au détriment d'une validité ponctuelle des densités. En effet, contraindre $\phi(x_0) = 0$ ou $p(x_0) = 1$ pour tous les x_0 déjà explorés augmenterait inutilement l'aspect chahuté des fonctions.

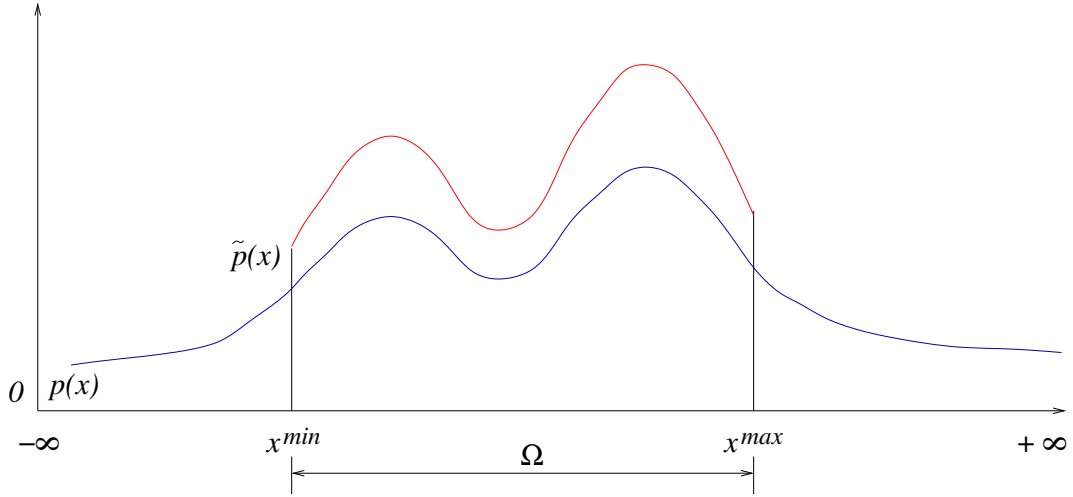


FIG. 2.1 – Fonctions densité de probabilité sans les bornes $p(x)$ et bornée $\tilde{p}(x)$.

est considéré, une probabilité bornée $\tilde{p}(x)$ est établie,

$$\tilde{p}(x) = \frac{p(x)}{M}, \quad M = \int_{\Omega} p(x) dx, \quad (2.5)$$

telle que $\int_{\Omega} \tilde{p}(x) dx = 1$. Ainsi, $\tilde{p}(x)$ est interprétée comme une densité de probabilité d'avoir exploré une région autour de x dans un domaine Ω . La Figure 2.1 montre, schématiquement, les fonctions $p(x)$ et $\tilde{p}(x)$.

Une densité de probabilité d'explorer une nouvelle région autour de x , $\phi(x)$, est une densité de probabilité de ne pas avoir exploré une région autour de x auparavant. $\phi(x)$ est construite de la manière suivante. Seul le point le plus haut x^H de $\tilde{p}(x)$ a une probabilité nulle de ne pas être échantillonné à l'itération suivante. La densité de probabilité ϕ est alors calculée comme

$$\phi(x) = \frac{H - \tilde{p}(x)}{\int_{\Omega} (H - \tilde{p}(x)) dx}, \quad H = \max_{x \in \Omega} \tilde{p}(x). \quad (2.6)$$

La Figure 2.2 illustre $p(x)$, $\tilde{p}(x)$ et $H - \tilde{p}(x)$, dans un domaine unidimensionnel. Rigoureusement, construire ϕ impliquerait les étapes suivantes :

- Calculer $p(x)$ en utilisant les points déjà échantillonnés (Equation (2.1)) ;
- Calculer $M = \int_{\Omega} p(x) dx$ puis $\tilde{p}(x) = \frac{p(x)}{M}$;
- Trouver $H = \max_{x \in \Omega} \tilde{p}(x)$;
- Calculer $\int_{\Omega} (H - \tilde{p}(x)) dx = H \int_{\Omega} dx - \int_{\Omega} \tilde{p}(x) dx = H \prod_{i=1,n} (x^{i,max} - x^{i,min}) - 1$, pour en déduire $\phi(x) = \frac{H - \frac{1}{M}p(x)}{H \prod_{i=1,n} (x^{i,max} - x^{i,min}) - 1}$;

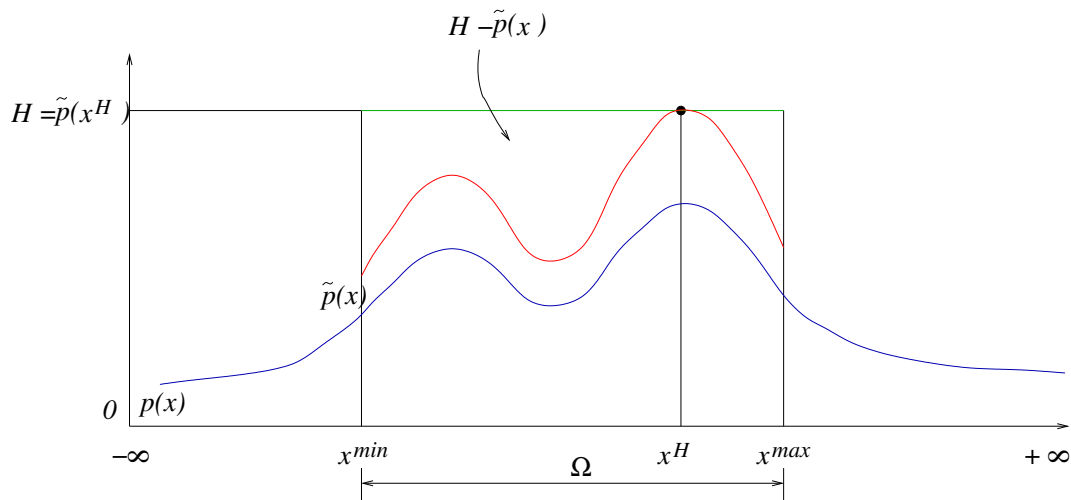


FIG. 2.2 – Illustration de la définition de la fonction densité de probabilité $\phi(x)$ pour la “Méthode 2”

– Trouver $\arg \max_{x \in \Omega} \phi(x)$.

La maximisation de $\phi(x)$ n’est pas effectuée de façon exacte non seulement à cause de son coût numérique mais aussi parce que, comme nous le verrons en Section 2.4.1, faire des erreurs d’échantillonnage de ϕ présente des avantages. Le maximum de $\phi(x)$ est approché par la stratégie suivante : N_r points sont tirés au hasard, et celui qui maximise $\phi(x)$ sera le point de départ de la prochaine recherche. Notons que pour maximiser $\phi(x)$, il n’est nécessaire de calculer ni M (2.5) ni H (2.6) : le maximum de ϕ est le minimum de p , ainsi seul p est calculé.

Trois paramètres influencent la maximisation de ϕ et, en conséquence, les points de départ : i) les points x_i qui sont gardés pour le calcul de la densité de probabilité p , ii) le nombre de points aléatoires utilisés pour maximiser ϕ , N_r , qui contrôle l’erreur d’échantillonnage et iii) le paramètre qui contrôle l’étalement des Gaussiennes, α . Ces paramètres sont discutés dans la Section 2.4.

La ré-initialisation probabilisée peut être appliquée à un optimiseur local quelconque. Dans ce travail, c’est une version améliorée de l’algorithme de Nelder-Mead qui est utilisée.

2.2 Un algorithme de Nelder-Mead amélioré

2.2.1 L'algorithme de Nelder-Mead

La méthode de Nelder-Mead ([70]) ou méthode du “polytope mouvant”, est fondée sur l'algorithme de Spendley, Hext et Himesworth ([81]), et utilise un arrangement de $(n+1)$ points x_i où la fonction coût est évaluée, n étant la dimension du domaine de recherche. Cet arrangement peut être vu comme les sommets d'un simplexe. Un simplexe régulier de taille initiale a est initialisé en x_0 , par exemple au moyen de la règle suivante ([33]),

$$x_i = x_0 + pe_i + \sum_{\substack{k=1 \\ k \neq i}}^n qe_k, \quad i = 1, n, \quad (2.7)$$

où

$$\begin{aligned} p &= \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} + n - 1 \right), \\ q &= \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} - 1 \right), \end{aligned} \quad (2.8)$$

et où e_i sont les vecteurs unitaires de la base. Chaque itération de la méthode commence avec les points d'un simplexe et les valeurs correspondantes de la fonction coût. Le simplexe est modifié à travers les opérations de *réflexion*, d'*expansion*, de *contraction*, ou par un *rétrécissement*, et un point est accepté ou rejeté en fonction de sa valeur de fonction coût. La Figure 2.3 montre les effets de ces opérations pour un simplexe dans un domaine bidimensionnel (un triangle). Une itération générique présente deux possibilités : i) un nouveau sommet au moins meilleur que le plus mauvais sommet lui est substitué, sinon, ii) un rétrécissement est effectué où un ensemble de n nouveaux points plus le meilleur des anciens points constituent le simplexe de la prochaine itération. L'organigramme de la méthode de Nelder-Mead est présenté dans la Figure 2.4. Les valeurs recommandées dans [70] pour les coefficients de réflexion (r), contraction (β) et expansion (γ) sont 1, $\frac{1}{2}$ et 2, respectivement. Ces valeurs seront utilisées dans ce mémoire.

Une interprétation intuitive de cet algorithme est qu'une direction de recherche est définie par le plus mauvais point (celui dont la fonction coût est la plus élevée) et le barycentre des sommets hormis le plus mauvais. Le simplexe peut accélérer (expansion) ou décélérer (contraction) dans cette direction pour localiser une région optimale et zoomer (rétrécissement) vers l'optimum. La

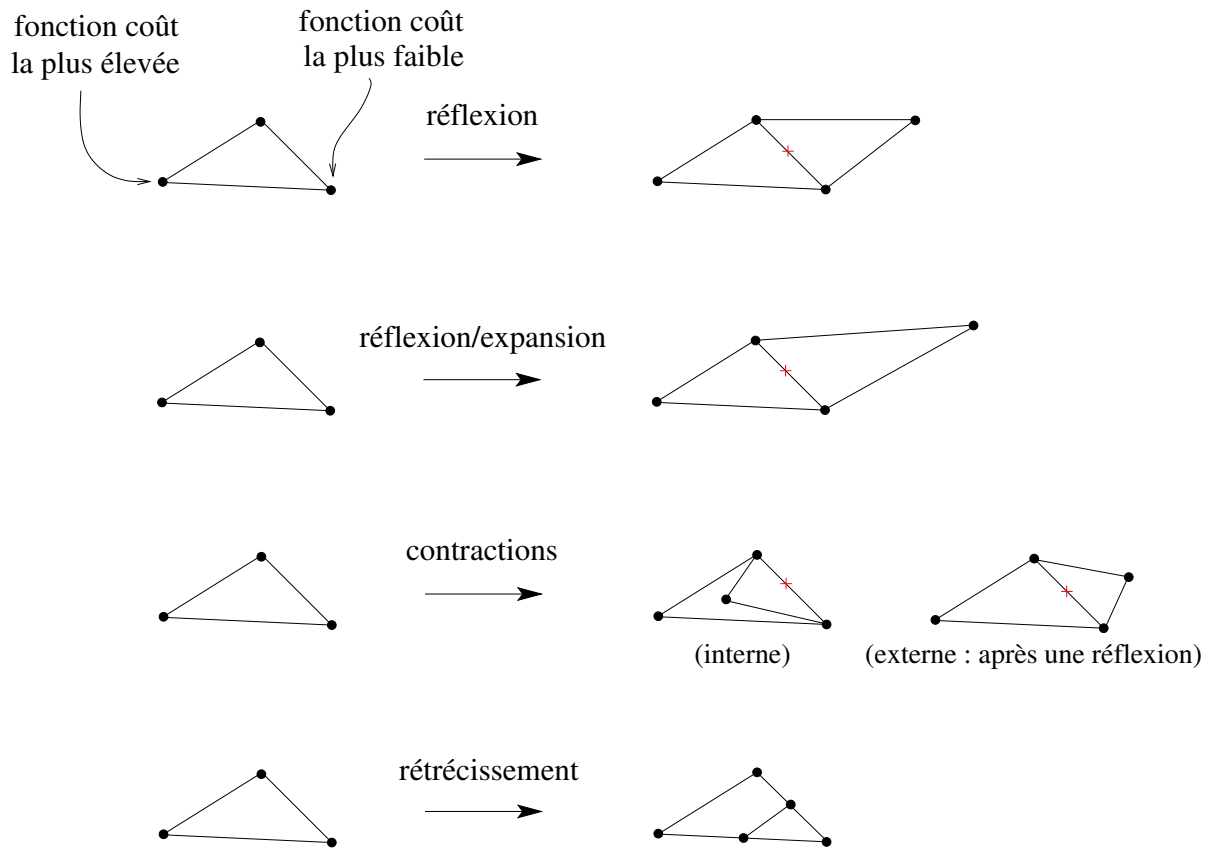


FIG. 2.3 – Opérations que le simplexe subit dans la méthode de Nelder-Mead. Le simplexe original est représenté par des traits discontinus, et + représente le barycentre des sommets hormis le plus mauvais.

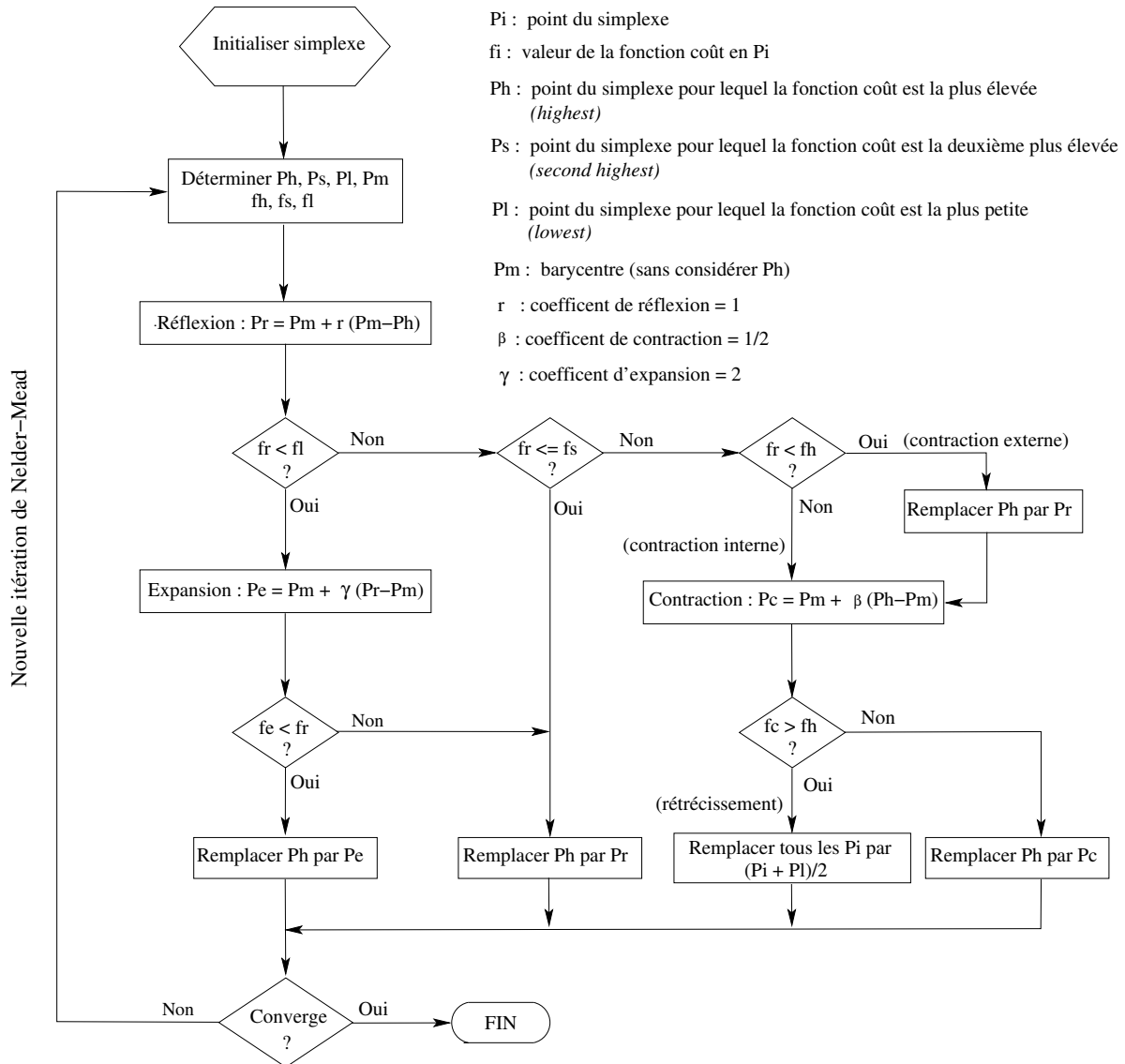


FIG. 2.4 – Organigramme de la méthode de Nelder-Mead.

recherche s’achève quand les valeurs des fonctions aux sommets sont proches,

$$\text{stop si } \sqrt{\sum_{i=1}^{n+1} (f_i - \bar{f})^2 / n} < \varepsilon, \quad \bar{f} = \frac{1}{n+1} \sum_{i=1}^{n+1} f_i, \quad (2.9)$$

où f_i est la valeur de la fonction coût en x_i , et ε est un petit scalaire positif.

L’algorithme de Nelder-Mead, qui parfois est appelé méthode du “simplexe de Nelder-Mead”, ne doit pas être confondu avec la méthode du simplexe mise au point par Dantzig et utilisée en programmation linéaire.

La convergence vers un point de minimum n’est pas garantie pour la méthode de Nelder-Mead ([93, 39, 40]). Des preuves de convergence pour des fonctions strictement convexes unidimensionnelles et des résultats restreints pour un ensemble de fonctions bidimensionnelles sont donnés dans [43]. Dans [63] le comportement de la méthode appliquée à une famille de fonctions bidimensionnelles strictement convexes et continues jusqu’à la troisième dérivée, où la convergence (ou stagnation) se produit sur un point non-stationnaire est analysé. L’explication de l’échec est la capacité du simplexe à se déformer pendant la recherche, où des déformations répétées peuvent amener à une dégénérescence du simplexe sur un sous-espace du domaine. Nonobstant ces inconvénients, l’algorithme de Nelder-Mead est probablement la méthode de recherche directe la plus utilisée et référencée ([43]) et est, en général, très efficace et rapide par rapport aux autres méthodes de recherche directe. Elle peut requérir, par exemple, moins d’évaluations de la fonction coût que la recherche multidirectionnelle ([93]). Par rapport aux méthodes de motifs (GPS), Nelder-Mead est capable de distordre le simplexe pour mieux s’adapter à la topologie de la fonction. Ceci lui confère à la fois une force et une faiblesse : Nelder-Mead gagne en vitesse de convergence, mais perd en robustesse (et en preuve formelle de convergence). Dans [70] les méthodes de Nelder-Mead et de Powell sont comparées pour la minimisation d’une fonction à deux variables (fonction test de Rosenbrock), d’une fonction à trois variables (fonction test de Fletcher et Powell), et d’une fonction à quatre variables (fonction test quadratique de Powell). Dans ces exemples, pour atteindre le minimum avec la même précision finale, la méthode de Nelder-Mead nécessite moins d’évaluations que la méthode de Powell. Comme les autres méthodes de recherche directe, elle est robuste pour les problèmes discontinus ou bruités [63]. Nelder-Mead est aussi la méthode de recherche directe la plus trouvée dans les codes de calcul numérique, et elle fait partie du *Matlab Optimization Toolbox* (fonction *fminsearch* pour l’optimisation non contrainte et sans

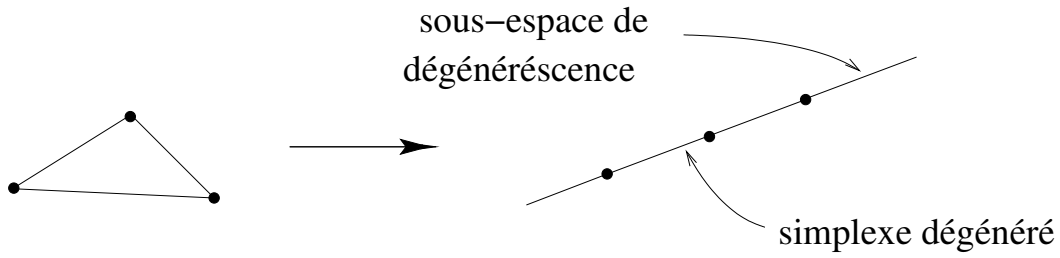


FIG. 2.5 – Dégénérescence d’un simplexe dans un espace bidimensionnel.

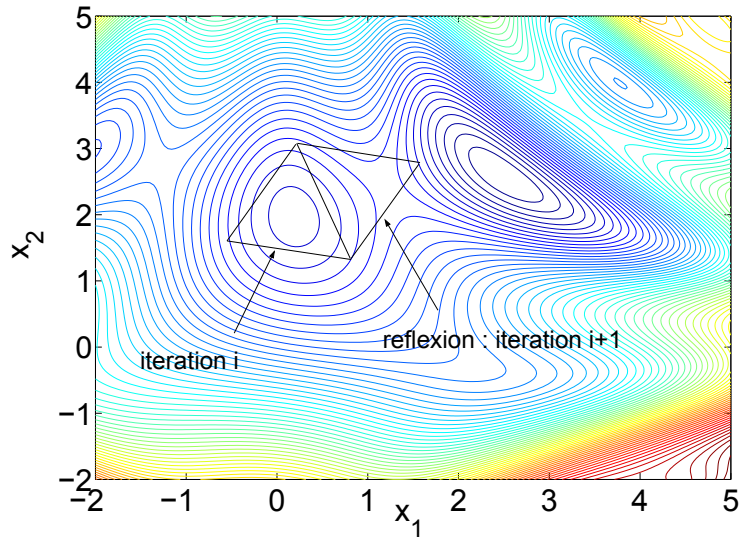
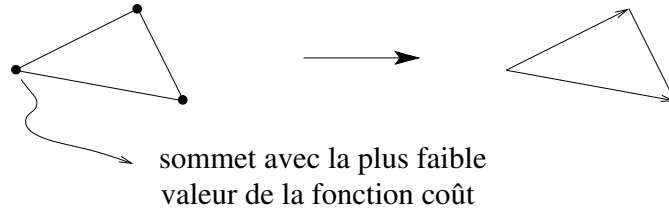


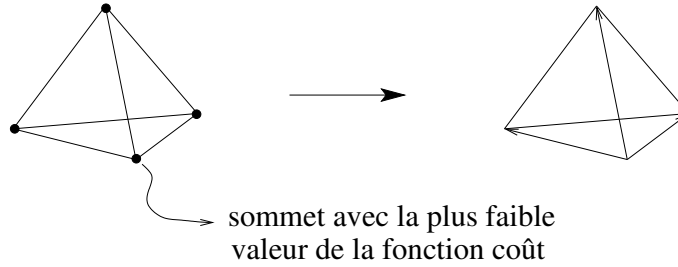
FIG. 2.6 – Exemple de simplexe qui traverse un bassin d’attraction.

limites sur les variables).

Deux propriétés de l’algorithme doivent être mentionnées. Premièrement, comme observé auparavant, la méthode peut ne pas converger vers un optimum local si le simplexe dégénère dans un sous-espace du domaine, c’est-à-dire si toutes les arêtes émanant d’un même sommet deviennent linéairement dépendantes. Le simplexe ne peut plus alors sortir du sous-espace couvert par ses arêtes (cf. Section 2.2.2 et Figure 2.5). Deuxièmement, la méthode peut traverser des bassins d’attractions si la taille du simplexe est suffisamment grande. La Figure 2.6 illustre un exemple de cela pour une fonction à deux variables, x_1 et x_2 . Néanmoins, à mesure que la taille du simplexe diminue, l’algorithme devient local.



(a) Simplexe bidimensionnel (n=2)



(a) Simplexe tridimensionnel (n=3)

FIG. 2.7 – Exemples de définition des arêtes du simplexe utilisées pour les tests de dégénérescence pour n=2 et n=3.

2.2.2 Détection et traitement des dégénérescences

La dégénérescence du simplexe est le symptôme le plus courant d'un échec de la recherche avec Nelder-Mead ([93]). Il est possible que lors des opérations sur le simplexe certaines arêtes attachées à un même sommet deviennent linéairement dépendantes. Dans ce cas, la méthode de Nelder-Mead n'est capable de rechercher un minimum que dans le sous-espace décrit par les arêtes. Afin d'éviter ce problème, une vérification de la dégénérescence a été mise en œuvre à chaque itération. Pour cette vérification, deux tests sont faits sur les arêtes du simplexe qui partent du sommet de plus faible fonction coût. La Figure 2.7 montre comment sont définies les arêtes dans les espaces bi et tridimensionnel. Le simplexe est considéré dégénéré s'il n'est pas petit (la définition de simplexe petit, utilisée comme critère de convergence, est donnée en Section 2.3 par l'Equation (2.12)), s'il ne touche pas les bornes (cf. Section 2.2.3), et si l'une des conditions suivantes est satisfaite,

$$\frac{\min_{k=1,n} \|\mathbf{e}^k\|}{\max_{k=1,n} \|\mathbf{e}^k\|} < \varepsilon_{deg1} \quad \text{or} \quad \frac{\det[\mathbf{e}]}{\prod_k \|\mathbf{e}^k\|} < \varepsilon_{deg2} , \quad (2.10)$$

où e^k est la k -ième arête, \mathbf{e} est la matrice formée par les composantes des arêtes, et ε_{deg1} et ε_{deg2} sont des petites constantes positives. Le premier test vérifie s'il y a une arête très petite par rapport aux autres, et le deuxième test analyse la dépendance linéaire des arêtes.

Si la dégénérescence est détectée pour un simplexe intérieur au domaine de définition des variables, il est ré-initialisé en utilisant comme point initial son meilleur sommet (celui qui possède la fonction coût la plus basse). Les autres sommets du simplexe sont définis suivant l'Equation (2.7). Le test de dégénérescence n'est pas exécuté lorsque des sommets touchent les bornes des variables car la dégénérescence peut être légitime, c'est-à-dire, le simplexe dégénéré peut continuer à être un simplexe non-dégénéré mais dans l'hyperplan des bornes actives (un "sous-simplexe" dans un "sous-espace") et la recherche peut continuer sur ces bornes. Pour un simplexe complètement sur les bornes, si le nombre d'arêtes linéairement indépendantes du simplexe ($\text{Rang}(\mathbf{e})$) est égal à la dimension du sous-espace (n - nombre de bornes actives), cette dégénérescence est légitime. Notons pourtant que si le simplexe perd plus de dimensions qu'il y a de bornes actives, il n'a plus la capacité d'explorer tout cet hyperplan : le "sous-simplexe" est dégénéré dans le "sous-espace". D'autres configurations de dégénérescence peuvent encore être envisagées quand une partie du simplexe est sur les bornes et l'autre est dans le domaine. Pour garder une mise en œuvre suffisamment simple, les tests de dégénérescence de sous-simplexes dans les sous-espaces ne sont pas réalisés. On se restreint au test présenté en Equation (2.10) (avec les conditions attenantes), pour que les vérifications de dégénérescence soient simples et que le simplexe ait la possibilité d'explorer les bornes. Enfin, si une convergence se produit sur les bornes, le test d'optimalité présenté en Section 2.2.3 est réalisé.

2.2.3 Prise en compte des bornes

L'algorithme original de Nelder-Mead a été proposé pour des problèmes sans bornes. Or une grande partie des problèmes d'optimisation, et en particulier la majorité des problèmes de conception en mécanique, ont des variables bornées. Pour leur appliquer la méthode de Nelder-Mead, une stratégie de prise en compte des bornes par projection est mise en œuvre :

$$\begin{cases} \text{si } (x^i < x^{i,min}), x^i = x^{i,min} , \\ \text{si } (x^i > x^{i,max}), x^i = x^{i,max} , \end{cases} \quad (2.11)$$

où x^i est la i -ème coordonnée du point x à analyser, et $x^{i,max}$ et $x^{i,min}$ sont les bornes dans la direction i . La projection peut intervenir après les étapes de réflexion ou d’expansion. Une conséquence de la projection sur les bornes est que le simplexe peut dégénérer dans l’hyperplan des bornes actives (cf. aussi section précédente). Si le simplexe a convergé avec des bornes actives, il peut soit avoir convergé vers un minimum local, soit avoir convergé vers un minimum dégénéré. En programmation mathématique, les fonctions et contraintes sont considérées comme différentiables et l’optimalité locale est vérifiée sous la forme des conditions de Karush, Kuhn et Tucker. Ici cependant, des fonctions pas nécessairement différentiables sont considérées. En guise de test d’optimalité le long des bornes, un redémarrage est réalisé à partir d’un petit simplexe au point de convergence. Si la recherche retourne au même point, il s’agit d’un minimum local sur les bornes. Remarquons que la taille du petit simplexe de redémarrage doit être supérieure à celle du critère de convergence (le critère de convergence utilisé par GBNM est présenté en Section 2.3), et suffisamment petite pour rester dans le même bassin d’attraction. Si le point de convergence avec bornes actives est un minimum dégénéré, la recherche continue vers un autre minimum. Dans ce cas, on remarque que cette recherche peut être coûteuse, puisque la taille initiale du simplexe est très petite. Un exemple de dégénérescence sur les bornes est présenté en Section 3.1.

2.3 Assemblage des améliorations : GBNM

La mise en œuvre des améliorations de l’algorithme de Nelder-Mead, qui composent GBNM, est basée sur un jeu d’options de ré-initialisations dont l’organigramme est en Figure 2.8. Les ré-initialisations ont deux objectifs.

Premièrement, la ré-initialisation “probabilisée” (cf. Section 2.1), qui est basée sur la densité p , vise à coordonner plusieurs recherches locales jusqu’à ce qu’un nombre total fixe d’analyses, C_{max} , soit atteint. La probabilité de trouver l’optimum global croît avec le nombre d’analyses. C’est l’aspect “globalisé” de la méthode. Dans l’implémentation des ré-initialisations probabilisées utilisée ici, la taille initial a du simplexe (voir Equation (2.7)) varie aléatoirement entre 2 à 10% de la plus petite dimension du domaine.

Deuxièmement, les ré-initialisations sont utilisées pour améliorer et vérifier les convergences locales de l’algorithme. Les deux schémas associés à la convergence initialisent un nouveau simplexe en utilisant comme point initial le meilleur point du simplexe courant. Les ré-initialisations nommées *small test* et *large test* utilisent un simplexe petit et grand, de tailles a_s et a_l , respecti-

vement. Ces ré-initialisations sont associées au test d'optimalité sur les bornes (cf. Section 2.2.3) et à la dégénérescence dans le domaine (cf. Section 2.2.2), respectivement.

La convergence des recherches locales de Nelder-Mead avec variables bornées est estimée à travers trois critères : *small*, *flat*, ou *degenerated simplex*, pour vérifier si le simplexe est petit, plat, ou dégénéré, respectivement.

Le simplexe est petit si

$$\max_{k=1, \dots, n+1} \left(\sum_{i=1}^n \left| \frac{e_i^k}{x^{i,max} - x^{i,min}} \right| \right) < \varepsilon_{small} , \quad (2.12)$$

où e_i^k est la i -ème coordonnée de la k -ième arête, $x^{i,max}$ et $x^{i,min}$ sont les bornes inférieure et supérieure dans la i -ème direction, et ε_{small} est une valeur de tolérance. Ici, contrairement au calcul de la dégénérescence, toutes les arêtes du simplexe sont prises en compte. On considère ainsi qu'un minimum local est trouvé si le simplexe courant est petit (et pas sur les bornes).

Le simplexe est plat si

$$|f_H - f_L| < \varepsilon_{flat} , \quad (2.13)$$

où f_H et f_L sont la valeur la plus haute et la plus faible de la fonction coût parmi les sommets du simplexe courant, respectivement, et ε_{flat} est une valeur de tolérance. Ce test permet d'échapper, par un redémarrage probabilisé (*T4*), aux éventuelles régions plates de la fonction coût.

Le critère pour déterminer si un simplexe est dégénéré dans le domaine a été présenté dans la Section 2.2.2, cf. Equation (2.10).

L'enchaînement des trois ré-initialisations (*probabilistic*, *small test* et *large test*) et des trois critères de convergence (simplexe *small*, *flat* et *degenerated*) est montré en Figure 2.8. Une mémoire des points de convergence des recherches locales passées est conservée, ainsi on prévient des calculs inutiles pour les points déjà analysés (voir test *T3* dans l'organigramme de la Figure 2.8). Un simplexe qui est dégénéré entraîne une ré-initialisation du type *large test* (*T8*). Quand l'optimalité des points de convergence est incertain, comme une convergence sur les bornes lorsque le simplexe est dégénéré (*T6*), un *small test* est exécuté en guise d'un test d'optimalité (cf. Section 2.2.3). Si le petit simplexe retourne au même point de convergence, ceci est considéré comme un minimum local.

Les tolérances pour détecter si un simplexe est petit ou dégénéré, ε_{small} et $[\varepsilon_{deg1}, \varepsilon_{deg2}]$, respectivement, peuvent être difficiles à régler. Par exemple, un simplexe qui devient petit pourrait

avant être jugé dégénéré. Ainsi, si une dégénérescence est détectée consécutivement deux fois au même point, ce point est gardé, puisqu'il est un possible optimum, et la ré-initialisation probabilisée est appelée. Similairement, si une dégénérescence est détectée après le test d'optimalité (*small test*), ce point est aussi gardé comme un possible optimum, et une nouvelle recherche avec redémarrage au meilleur point du simplexe courant (*large test*) est lancée. Une fois que le nombre d'évaluations atteint C_{max} , l'exécution se termine, et le résultat de la recherche est une liste d'optima locaux (éventuellement globaux) possibles. Cette liste doit ensuite être interprétée par l'utilisateur. En pratique, l'obtention de plusieurs optima locaux ou globaux est un avantage de la méthode par rapport aux optimiseurs globaux qui ne présentent qu'une seule solution.

GBNM a été programmé en C++ dans le logiciel LAMKIT© (cf. Annexe B), sous la plateforme Linux. Une version sous Matlab a aussi été programmée.

2.4 Analyse de l'algorithme globalisé

En utilisant la ré-initialisation probabilisée, on cherche à maximiser la probabilité de visiter tous les bassins d'attraction d'une fonction générique en un nombre total d'évaluations de la fonction objectif fixé a priori. Une densité de probabilité de ne pas avoir visité une région autour d'un point x , $\phi(x)$, est construite itérativement au moyen de noyaux Gaussiens. Trois paramètres influencent directement le calcul et l'échantillonnage de la densité de probabilité ϕ et, par conséquent, les points de départ :

- le paramètre qui contrôle l'étalement des Gaussiennes, α (voir Equation (2.4)) ;
- les points qui sont gardés comme centres des noyaux, et
- le nombre de points aléatoires (N_r) utilisé pour trouver lequel a la plus faible probabilité p .

Ces trois paramètres sont discutés ci-après.

2.4.1 Contrôle de l'étalement des noyaux Gaussiens, α

Les N noyaux Gaussiens p_i servant à construire les densités de probabilité p , \tilde{p} et ϕ (cf. Equations (2.1) et (2.2)), ont des écarts types σ contrôlés par le paramètre α à travers l'Equation (2.4), ou,

$$\sigma^j = \sqrt{\alpha}(x^{j,max} - x^{j,min}) . \quad (2.14)$$

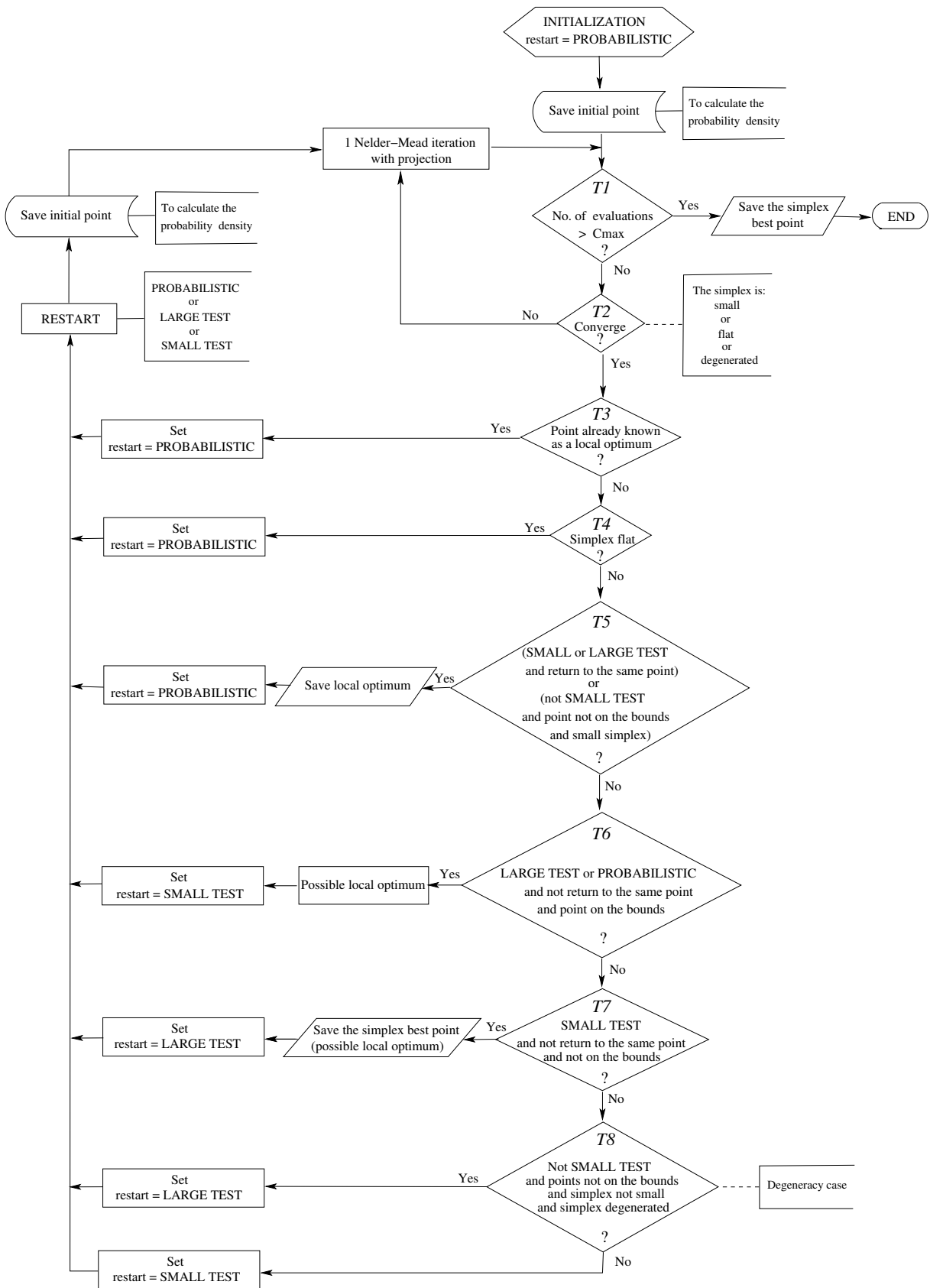


FIG. 2.8 – Enchaînement entre les ré-initialisations et les tests de convergence dans GBMM.

Le résultat élémentaire selon lequel 68.3% d'une infinité de tirages effectués suivant une loi normale de moyenne \bar{x} et d'écart-type σ se trouvent dans l'intervalle $[\bar{x}-\sigma; \bar{x}+\sigma]$ sert à interpréter l'effet de α . Sa déclinaison en plusieurs dimensions puis sur plusieurs noyaux est brièvement discutée.

Premièrement, chaque p_i est une loi multinormale, de dimension n non corrélée. Puisqu'il n'y a pas de corrélation, les résultats concernant les lois normales unidimensionnelles s'appliquent à chaque composante x^j indépendamment. Par exemple, si il n'y a qu'un noyau p_1 , la taille de l'intervalle sur chaque composante où se concentrent 68.3% des tirages suivant p_1 est $2\sigma_1^j$ qui est reliée à α comme illustré en Table 2.1. L'hypercube centré sur x_1 et de cotés $2\sigma_1^j$ concentre $((0.683)^n \times 100)\%$ des tirages. La Figure 2.9 illustre comment p_1 varie en fonction de α dans un domaine 2D, $[-90; 90]^2$, $x_1 = (0; 0)$. Les valeurs 90 sont liées aux bornes sur les angles de fibres utilisées en Chapitre 4 dans les applications aux composites. La valeur $\alpha = 0.01$, qui est graphiquement la plus satisfaisante (Figure 2.9 toujours) est adoptée pour le reste de ce mémoire.

Deuxièmement, les densités p , \tilde{p} ou ϕ sont des sommes de noyaux Gaussiens non-corrélés, p_i . $\frac{68.3}{N}\%$ des tirages effectués suivant p ont leur j -ième coordonnée dans $[x_i^j - \sigma_i^j; x_i^j + \sigma_i^j]$.

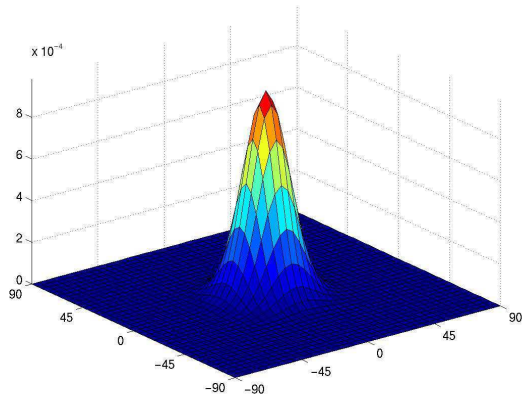
Signalons enfin que GBNM est, en général, moins sensible à α que ne le serait une procédure d'échantillonnage de p (ou \tilde{p} ou ϕ). En effet, GBNM réalise une minimisation approximative de p . Dans les cas, les plus usuels, où les minima ne changent pas radicalement de région, ils varient peu avec α .

α	$2\sigma^j$
	influence à 68.3%
0.002	$\approx \frac{1}{10}(x^{j,max} - x^{j,min})$
0.01	$\approx \frac{1}{5}(x^{j,max} - x^{j,min})$
0.1	$\approx \frac{2}{3}(x^{j,max} - x^{j,min})$
0.25	$(x^{j,max} - x^{j,min})$

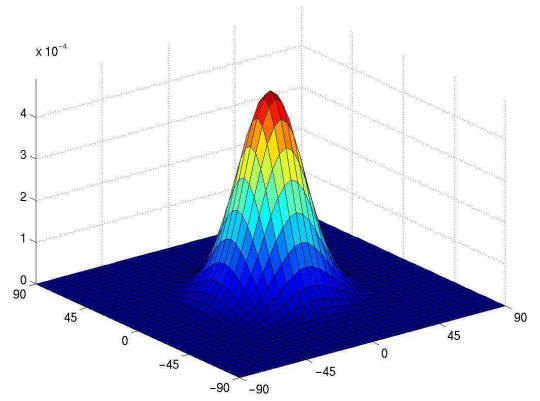
TAB. 2.1 – Rapport entre α et σ

2.4.2 Centrage des noyaux

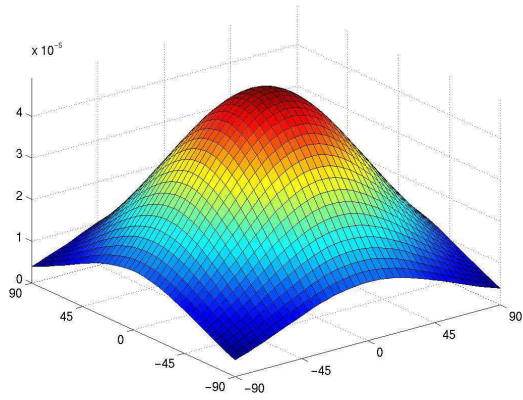
Trois stratégies sont comparées en termes de probabilité de ne pas trouver un des minima locaux, P_{ntm} : les x_i utilisés dans les Equations 2.1 et 2.2 sont i) les points de départ, ou ii) les points de départs et de convergence, ou iii) tous les points échantillonnés pendant la recherche.



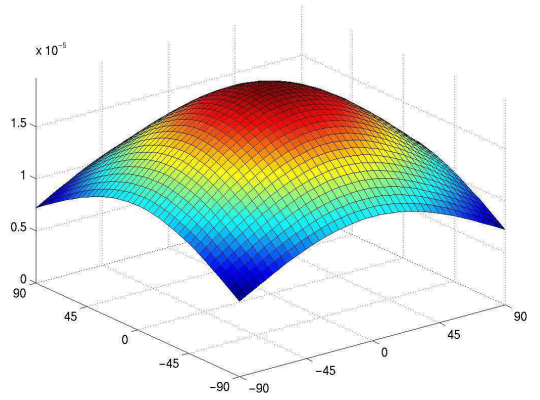
(a) $\alpha = 0.005$



(b) $\alpha = 0.01$



(c) $\alpha = 0.1$



(d) $\alpha = 0.25$

FIG. 2.9 – Représentation de l'estimation de la fonction de densité de probabilité p , pour différentes valeurs de α , et un noyau Gaussien, positionné au centre du domaine

On rappelle que les points de convergence ne sont jamais doublés (cf. test T_3 dans l'organigramme de la Figure 2.8). Les inconvénients de la stratégie iii) sont :

- l'utilisation de mémoire pour garder tous les points,
- le temps d'exécution pour calculer la densité de probabilité est augmenté,
- dans la région de convergence il y a une accumulation de points, ce qui pénalise trop cette région vis à vis des redémarrages. Ainsi, si au voisinage d'un point de convergence il y a d'autres minima, la difficulté de les trouver augmente.

C'est pour ces raisons que cette stratégie a été écartée dès les premiers tests, et n'est pas considérée dans les prochains essais numériques. La Figure 2.10(a) montre les points parcourus lors d'une convergence, et la Figure 2.10(b) la densité de probabilité p associée lorsque tous les points sont conservés. Cet exemple concerne la maximisation du module d'élasticité E_x pour un composite à 2 couches (cf. Chapitre 4 et Annexe D). Le point de départ est $[75/75]$, et le point de convergence $[0/0]$. On remarque que la région de convergence est sévèrement pénalisée vis à vis des prochains redémarrages.

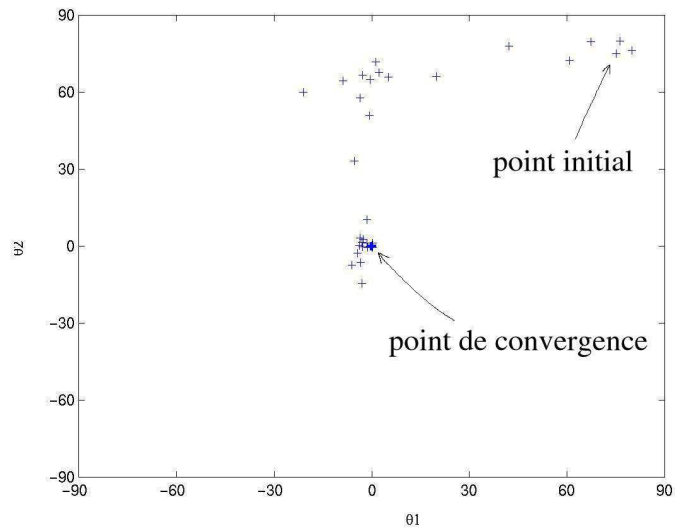
Les stratégies i) et ii) sont testées sur les trois fonctions multimodales bidimensionnelles données ci-après, en variant N_r de 1 à 1000,

$$\left\{ \begin{array}{l} f_1(x_1, x_2) = 2 + 0.01(x_2 - x_1^2)^2 + (1 - x_1)^2 + 2(2 - x_2)^2 \\ \quad + 7 \sin(0.5x_1) \sin(0.7x_2x_1) \quad , \quad x_1, x_2 \in [0; 5] \quad , \\ f_2(x_1, x_2) = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad , \quad x_1, x_2 \in [-3; 3] \quad , \\ f_3(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10 \quad , \\ \quad x_1 \in [-5; 10], \quad x_2 \in [0; 15] \quad . \end{array} \right. \quad (2.15)$$

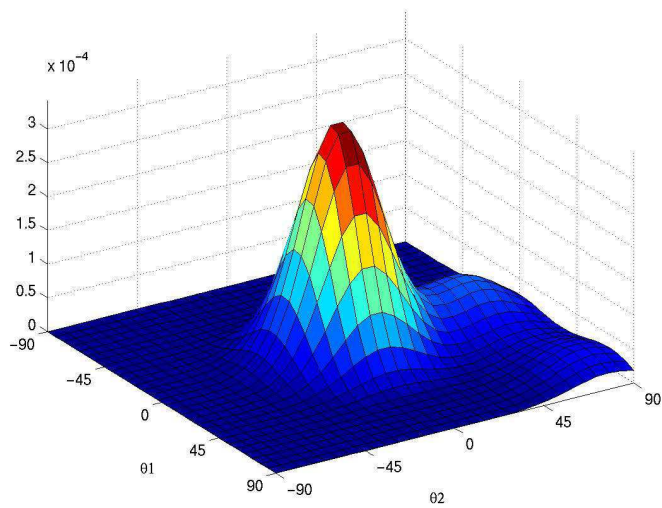
f_1 présente quatre minima locaux, f_2 six, et f_3 trois, comme le montre la Table 2.2 et la Figure 2.11. f_2 est connue comme la fonction *Six Humps Camel Back* et f_3 comme la fonction *Branin's rcos*. Ce sont des fonctions très utilisées pour tester les méthodes d'optimisation globales.

Les Tables 2.3 et 2.4 présentent, pour les deux premières stratégies respectivement, la probabilité de ne pas trouver un des minima locaux (P_{ntm}) après $C_{max} = 500$ analyses de la fonction coût, et $\alpha = 0.01$. L'estimation de la probabilité a été faite sur 1000 essais indépendants, et P_{ntm} est calculé comme :

$$P_{ntm} = 1 - P_1 P_2 \dots P_N \quad , \quad (2.16)$$

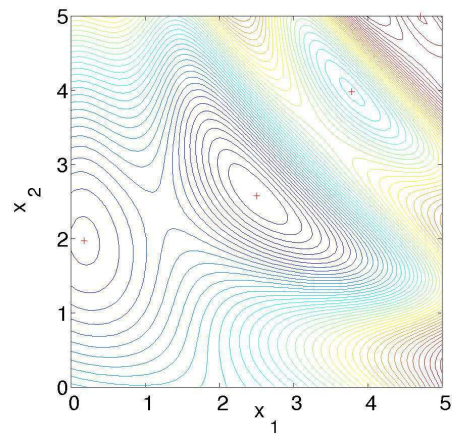
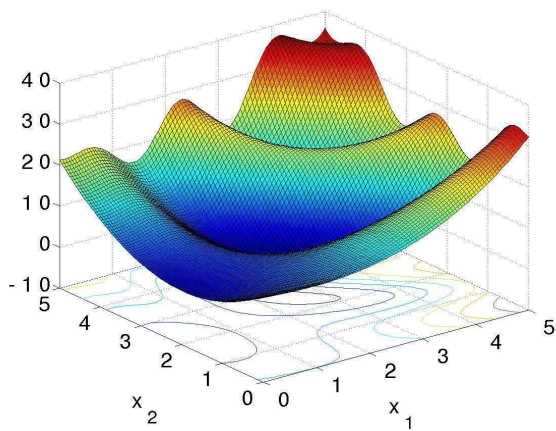


(a)

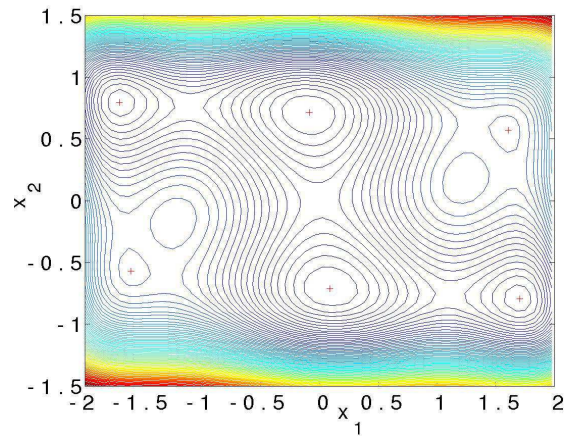
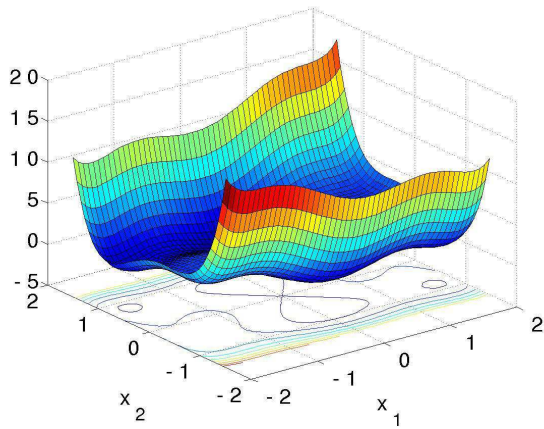


(b)

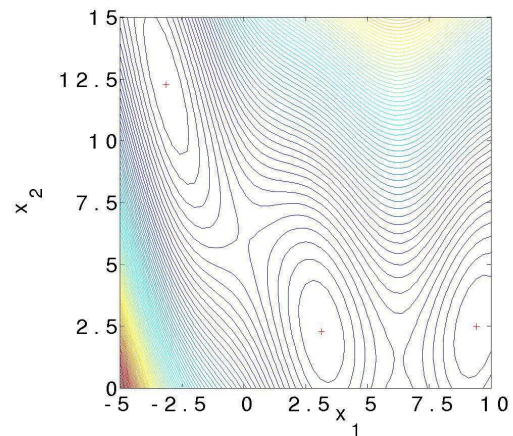
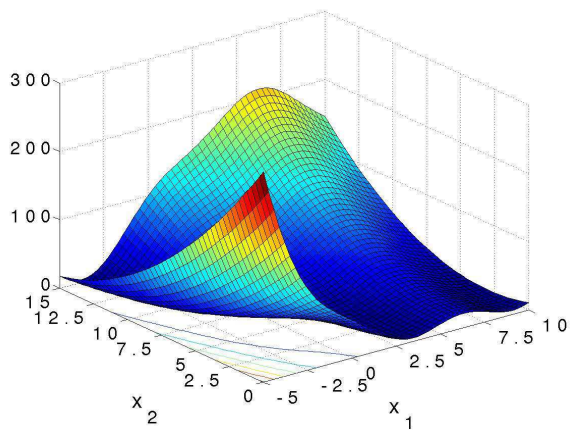
FIG. 2.10 – Points échantillonnés (a) et fonction densité de probabilité $p(x)$ calculée avec tous les points par lesquels le simplexe est passé (b).



(a)



(b)



(c)

FIG. 2.11 – Fonctions testées : f_1 (a), f_2 (b) et f_3 (c).

Minimum	f_1	f_2	f_3
1 ^{er}	$x_1 = 2.50443$ $x_2 = 2.57784$	$x_1 = 0.0898408$ $x_2 = -0.712656$	$x_1 = -\pi$ $x_2 = 12.275$
2 ^{ème}	$x_1 = 0.175882$ $x_2 = 1.97193$	$x_1 = -0.0898409$ $x_2 = 0.712656$	$x_1 = \pi$ $x_2 = 2.275$
3 ^{ème}	$x_1 = 3.78294$ $x_2 = 3.98083$	$x_1 = 1.70361$ $x_2 = -0.796082$	$x_1 = 9.42478$ $x_2 = 2.475$
4 ^{ème}	$x_1 = 4.7096$ $x_2 = 5$	$x_1 = -1.70361$ $x_2 = 0.796082$	
5 ^{ème}		$x_1 = 1.6071$ $x_2 = 0.568652$	
6 ^{ème}		$x_1 = -1.6071$ $x_2 = -0.568652$	

TAB. 2.2 – Positionnement des minima des fonctions.

N_r	f_1	f_2	f_3
1 (ré-initialisation aléatoire)	0.96904	0.99679	0.49420
2	0.93201	0.99597	0.33697
3	0.92247	0.99549	0.27986
10	0.87097	0.99752	0.15167
20	0.82177	0.99876	0.14210
50	0.77436	0.99960	0.11170
1000	0.79485	0.99974	0.08178

TAB. 2.3 – Probabilité P_{ntm} de ne pas trouver un des minima locaux ($C_{max}=500$ analyses, 1000 exécutions, $\alpha = 0.01$, les points de départs seuls sont gardés pour le calcul de la densité de probabilité). Les minima de P_{ntm} en fonction de N_r sont écrits en gras.

où N est le nombre de minima locaux, et P_i la probabilité de trouver le i -ème minimum.

Ces deux techniques appliquées aux exemples mentionnés ci-dessus ne présentent pas de différences remarquables concernant les temps d'exécution et la mémoire utilisée. De plus, on remarque que la stratégie consistant à garder simultanément les points de départ et les points de convergence permet de trouver plus de minima car le minimum de P_{ntm} en fonction de N_r est inférieur. Ces tables montrent que les points de départ et de convergence résument suffisamment la topologie des bassins d'attraction. C'est ce schéma qui est adopté pour mettre à jour p .

N_r	f_1	f_2	f_3
1 (ré-initialisation aléatoire)	0.96904	0.99679	0.49420
2	0.93323	0.99444	0.29337
3	0.91236	0.99860	0.23685
10	0.84521	0.99862	0.09115
20	0.79950	0.99945	0.07130
50	0.74687	0.99971	0.04735
1000	0.71452	0.99982	0.02681

TAB. 2.4 – Probabilité P_{ntm} de ne pas trouver un des minima locaux ($C_{max}=500$ analyses, 1000 exécutions, $\alpha = 0.01$, les points de départs et de convergence sont gardés pour le calcul de la densité de probabilité). Les minima de P_{ntm} en fonction de N_r sont écrits en gras.

2.4.3 Contrôle de l’erreur d’échantillonnage : nombre de points aléatoires,

N_r

Après la convergence d’une recherche locale, le prochain point initial est celui, parmi les N_r , qui a la plus faible probabilité p d’avoir été échantillonné. Si $N_r = 1$, la ré-initialisation est complètement aléatoire. Si N_r est grand, les points de départ dépendent strictement des points initiaux et de convergence des recherches passées, ce qui induit typiquement une certaine régularité géométrique des points de départ.

La Figure 2.12 présente des exemples de placements de points de départ pour $N_r = 1$, 10 et 1000 (les noyaux sont centrés sur les points de départ). Le premier point de départ est positionné au centre du domaine. On remarque que pour $N_r = 1000$ les points de départ forment un motif parfait (Figure 2.12(a)), pour $N_r = 1$ ils sont placés aléatoirement (Figure 2.12(c)), et pour $N_r = 10$ il y a un compromis entre les deux extrêmes précédents (Figure 2.12(b)).

Les tests rapportés en Tables 2.3 et 2.4 montrent que la distribution induite par N_r grand est intéressante pour les fonctions f_1 et f_3 qui ont des optima locaux près des coins du domaine. En général cependant, quand les optima locaux sont à l’intérieur du domaine, qui plus est si certains sont proches les uns de autres, il est plus efficace (cf. f_2 en Tables 2.3 et 2.4) de rompre cette régularité en introduisant des erreurs d’échantillonnages. Ainsi, en choisissant pour N_r un nombre petit, plus grand que 1, une caractéristique aléatoire-biaisée est donnée à la ré-initialisation, ce qui est un compromis entre une grille et un schéma avec des ré-initialisations aléatoires.

Désormais $N_r = 10$ est adopté, exception faite des cas où sa valeur est fournie explicitement.

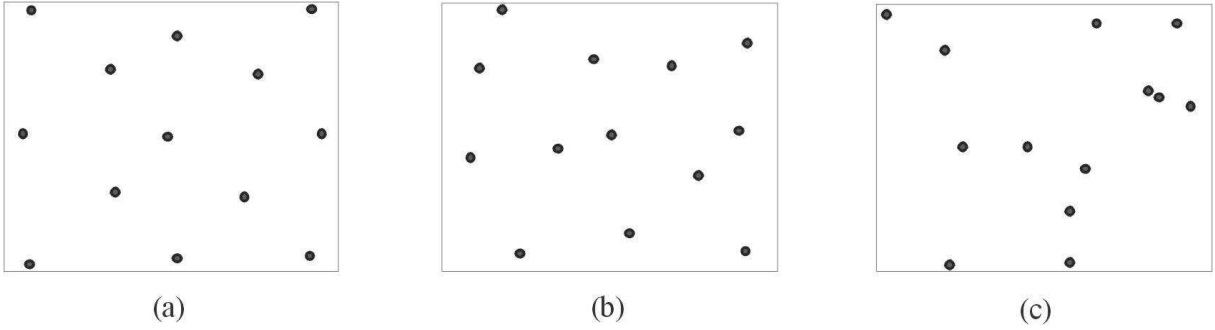


FIG. 2.12 – Exemples de points de départ dans un espace bidimensionnel. Le premier point de départ est au centre du domaine. Utilisation des points de départ seuls comme centre des noyaux, $N_r = 1000$ (a), $N_r = 10$ (b) et $N_r = 1$ (c).

2.5 Prise en compte des contraintes par pénalisation linéaire et adaptative

Une fonction de pénalisation linéaire et adaptative est utilisée pour la prise en compte des contraintes d'inégalité. Le problème primal,

$$(P) \quad \begin{cases} \min_{x \in \mathcal{S} \subset \mathbb{R}^n} f(x) , \\ \text{tel que } g_i(x) \leq 0 \quad , \quad i = 1, m , \end{cases} \quad (2.17)$$

est ré-écrit sous la forme pénalisée,

$$(PP) \quad \begin{cases} \min_{x \in \mathcal{S}} \bar{L}(x, \lambda) , \quad \text{où} \\ \bar{L}(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i \max(0, g_i(x)) . \end{cases} \quad (2.18)$$

Ce dernier problème est sans contraintes, mais les valeurs des paramètres de pénalisation λ_i doivent être estimées. Les approches les plus courantes en programmation mathématique sont basées sur les pénalisations quadratiques et les Lagrangiens ordinaires ou augmentés. La pénalisation adaptative \bar{L} présente les avantages suivants, dont les preuves sont données en Annexe 12 en utilisant la théorie du Lagrangien généralisé ([77, 66, 50]) :

- par rapport à la pénalisation quadratique, la convergence vers un optimum admissible peut être atteinte pour des valeurs finies des paramètres λ_i ;
- par rapport à un Lagrangien ordinaire, il existe des λ_i tels que résoudre (PP) est équivalent à résoudre (P) pour une classe plus ample de fonctions f et g_i ;

– enfin, les Lagrangiens augmentés ont plus de paramètres de pénalisation à régler que \bar{L} . Ces propriétés supplémentaires sont obtenues au prix de la différentiabilité de \bar{L} en $g(x) = 0$. Toutefois, ceci n'est pas un inconvénient décisif, puisque des fonctions non-différentiables sont considérées ici.

Les paramètres de pénalisation sont mis à jour par l'algorithme GBNM après chaque génération d'un point dans le domaine. La mise à jour est intuitive car elle consiste à accroître les paramètres de pénalisation des contraintes violées :

$$\begin{aligned}
 & \text{si } (\bar{L}(x^{\text{new}}, \lambda^k) \leq \bar{L}(x^{\text{best}}, \lambda^k)) , \\
 & \quad \lambda_i^{k+1} = \lambda_i^k + s \max(0, g_i(x^{\text{new}})) \quad , \quad i = 1, m , \\
 & \quad x^{\text{best}} = \arg \min_{x \in \{x^{\text{new}}, x^{\text{best}}, \text{simplex vertices}\}} \bar{L}(x, \lambda^{k+1}) , \\
 & \text{fin} ,
 \end{aligned} \tag{2.19}$$

où s est une taille positive de pas. Cette stratégie de mise à jour est interprétée comme une maximisation de la fonction duale généralisée par un gradient approximé et à pas fixe (voir Annexe 12).

Un exemple de convergence dans l'espace dual λ est maintenant présenté. La fonction test de Rosenbrock bidimensionnelle (voir Figure 3.1) est minimisée dans un domaine borné avec une contrainte additionnelle,

$$\left\{ \begin{array}{l} \min_{x_1, x_2 \in [0; 20]} 100(x_2 - x_1^2)^2 + (1 - x_1)^2 , \\ \text{tel que } 4 - x_1^2 \leq 0 . \end{array} \right. \tag{2.20}$$

100 optimisations indépendantes avec des points de départ aléatoires, $\lambda^0 = 0$, $\alpha = 0.01$ (cf. Equation (2.4)), et taille du pas $s = 0.001$ ont été réalisées. En moyenne, le paramètre de pénalisation se stabilise en $\bar{\lambda} = 0.5 \pm 10^{-6}$ au bout de 746 ± 178 analyses. Le minimum global, $x_1 = 2$, $x_2 = 4$, est toujours trouvé. On s'aperçoit que, dans ce cas, $\bar{\lambda} = 0.5$ est aussi le multiplicateur de Kuhn et Tucker à l'optimum. En général, dans des cas comme ceux qui seront présentés dans la Section 3.3, une sensibilité des valeurs convergées des λ_i aux points de départ x_0 se produit.

Chapitre 3

Applications à des fonctions test analytiques

Des essais avec la fonction analytique de Rosenbrock bidimensionnelle dans un domaine borné, sont montrés dans la Section 3.1. Lorsque des limites sur les variables sont considérées, ce problème peut engendrer une dégénérescence du simplexe sur les bornes. La Section 3.2 présente des résultats de minimisation d'une fonction multimodale en 12 dimensions (fonction de Griewank), et des comparaisons avec un algorithme évolutionnaire. Ces deux premiers problèmes, la minimisation des fonctions de Rosenbrock et de Griewank, sont sans contraintes mais avec des variables bornées. La Section 3.3 est consacrée à la minimisation de deux fonctions avec contraintes, et discute les paramètres de pénalisation. Le Chapitre 4 montrera des applications aux problèmes de plaques composites stratifiées, sans et avec restrictions.

3.1 Minimisation de la fonction de Rosenbrock et dégénérescence sur les bornes

On s'intéresse à minimiser la fonction de Rosenbrock bidimensionnelle,

$$y = 100(x_2 - x_1^2)^2 - (1 - x_1)^2. \quad (3.1)$$

C'est une fonction en forme de vallée étroite courbe, et à fond peu incliné. Elle est convexe, et son minimum $y^* = 0$ se trouve en $x_1^* = x_2^* = 1$. La Figure 3.1 la montre graphiquement. Malgré

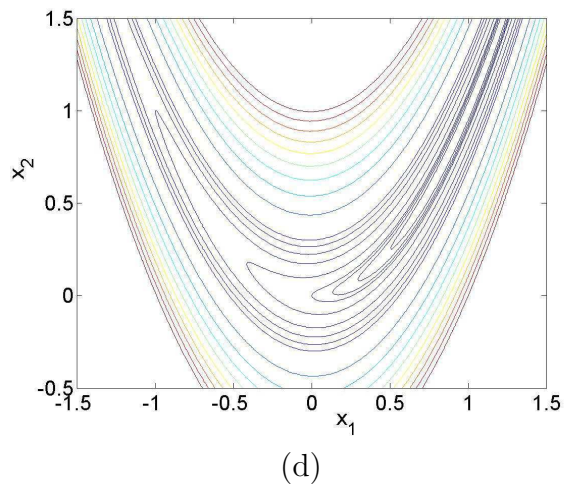
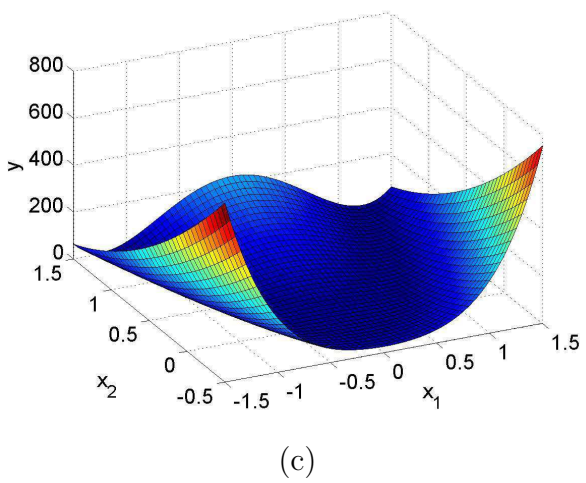
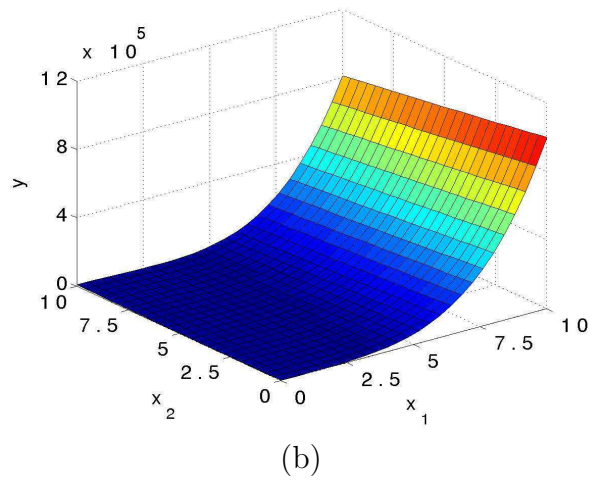
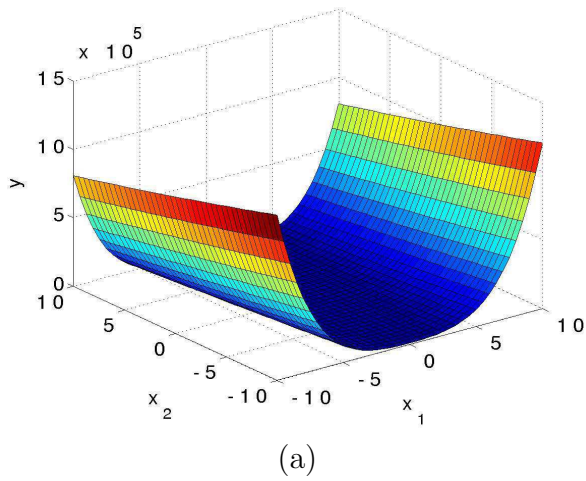


FIG. 3.1 – Fonction de Rosenbrock dans les régions $x_1, x_2 \in [-10; 10]$ (a) et $x_1, x_2 \in [0; 10]$ (b), au voisinage du minimum (1; 1) (c), et courbes de niveau (d).

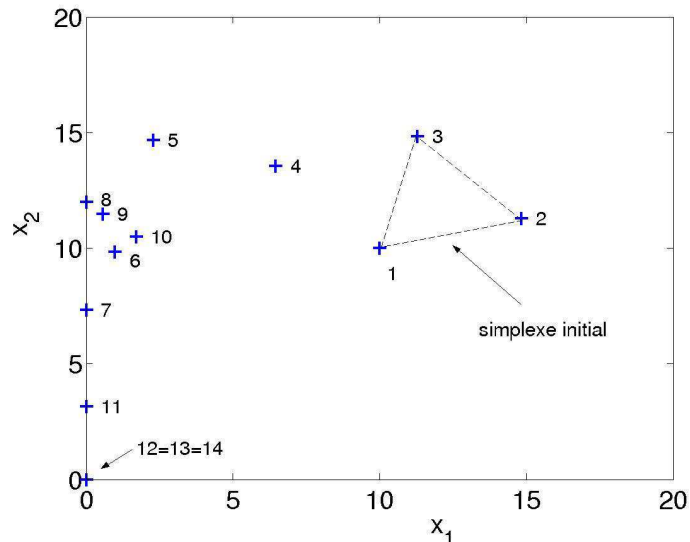


FIG. 3.2 – Cas de dégénérescence sur les bornes : minimisation de la fonction de Rosenbrock dans un domaine borné en utilisant la méthode de Nelder-Mead avec la stratégie de projection sur les bornes.

sa convexité, la fonction de Rosenbrock a un Hessien mal conditionné, ce qui rend les algorithmes de type plus forte pente inefficaces. Du fait de cette caractéristique, la fonction de Rosenbrock est très utilisée pour tester de nouvelles méthodes d'optimisation. Le domaine $(x_1; x_2) \in [0; 20]$ sera considéré. Il contient le minimum. L'utilité du test d'optimalité sur les bornes (cf. Sections 2.2.3 et 2.3) est illustré par l'exemple suivant.

Exemple de dégénérescence sur les bornes : La méthode de Nelder-Mead avec projection, mais dépourvue du test d'optimalité sur les bornes (*small test* : ré-initialisation avec le petit simplexe) ne réussit pas à trouver le minimum dans cet exemple. Le point initial est $(10; 10)$, et la taille initiale égale à 5. La Figure 3.2 montre les points par où le simplexe est passé. On observe qu'il dégénère rapidement sur la borne $x_1 = 0$, puis converge sur le point $(0; 0)$, qui n'est pas le minimum. La dégénérescence sur les bornes et la convergence sur le point $(0; 0)$ se produit également pour d'autres points de départ ou d'autres tailles de simplexe initiales. Si l'on fait le test d'optimalité, le minimum $(1; 1)$ est toujours trouvé, comme le montre la Table 3.1, qui présente le comportement de l'algorithme avec projection et test d'optimalité, en utilisant une tolérance d'arrêt $\varepsilon_{small} = 1 \times 10^{-6}$. Le nombre d'évaluations de la fonction coût varie selon le point initial et la taille initiale du simplexe.

Le test d'optimalité peut représenter un coût important sur les ressources disponibles. Dans la Table 3.2 sont montrés des résultats de minimisation de la fonction de Rosenbrock sans considérer

Point initial ($x_1; x_2$)	Taille initiale du simplexe	Point de convergence	Nombre d'analyses
(0;0)	1	(1;1)	230
(0;0)	5	(1;1)	231
(0;0)	15	(1;1)	233
(0;1)	5	(1;1)	233
(5;5)	5	(1;1)	233
(5;5)	15	(1;1)	233
(10;10)	5	(1;1)	238
(10;10)	15	(1;1)	236
(15;15)	5	(1;1)	241
(15;15)	15	(1;1)	247
(20;20)	1	(1;1)	243
(20;20)	5	(1;1)	236
(20;20)	15	(1;1)	232
(0;20)	15	(1;1)	242

TAB. 3.1 – Minimisation de la fonction de Rosenbrock avec bornes $[0; 20] \times [0; 20]$ et test d'optimalité (GBNM).

les limites sur les variables, en utilisant la méthode classique de Nelder-Mead. Dans ces cas, comme il n'y a pas de bornes, la convergence est toujours sur le minimum, sans la nécessité d'un test d'optimalité. Pour mesurer ce coût, on peut rapporter, pour les mêmes conditions initiales (point de départ et taille initial du simplexe), le nombre d'analyses de la Table 3.1 avec ceux de la Table 3.2. En moyenne la prise en compte des bornes par projection et le test d'optimalité engendrent un surcoût d'environ 25%.

3.2 Minimisation de la fonction de Griewank en dimension 12

Cet exemple consiste en la minimisation de la fonction test de Griewank en $n = 12$ dimensions,

$$F(x_1, \dots, x_n) = \frac{1}{400n} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad x_i \in [-1000; 1000] . \quad (3.2)$$

Cette fonction présente plusieurs minima locaux, et un seul minimum global, $F = -1$, qui se trouve en $x_i = 0.0$, $i = 1, n$. La Figure 3.3 illustre le cas unidimensionnel ($n = 1$), $x \in [-20; 20]$. La Table 3.3 compare GBNM et le meilleur point d'une population d'un algorithme évolutionnaire (EA), en 200, 1000, 5000 et 10000 appels de la fonction coût. L'algorithme évolutionnaire ([5]) a

Point initial (x_1, x_2)	Taille initiale du simplexe	Point de convergence	Nombre d'analyses
(0;0)	1	(1;1)	165
(0;0)	5	(1;1)	171
(0;0)	15	(1;1)	198
(5;5)	1	(1;1)	197
(5;5)	5	(1;1)	198
(5;5)	15	(1;1)	200
(10;10)	15	(1;1)	201
(10;10)	50	(1;1)	288
(10;10)	1000	(1;1)	480
(1000;1000)	1	(1;1)	613
(1000;1000)	15	(1;1)	628
(1000;1000)	50	(1;1)	604
(1000;1000)	1000	(1;1)	520
(-1000;-1000)	1	(1;1)	385
(-1000;-1000)	15	(1;1)	367
(-1000;-1000)	50	(1;1)	456
(-1000;-1000)	1000	(1;1)	312

TAB. 3.2 – Minimisation de la fonction de Rosenbrock, sans limites sur les variables et sans redémarrage (Nelder-Mead classique).

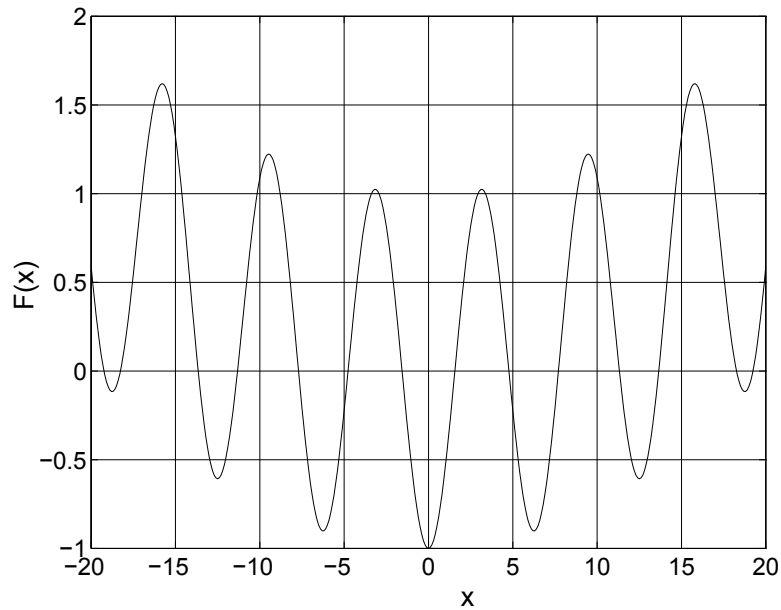


FIG. 3.3 – La fonction test de Griewank ($n = 1$).

	200 analyses		1000 analyses		5000 analyses		10000 analyses	
	F_{min}	Prob. de trouver le min. global	F_{min}	Prob. de trouver le min. global	F_{min}	Prob. de trouver le min. global	F_{min}	Prob. de trouver le min. global
GBNM	19.321 ± 26.709	0/100	-0.526 ± 0.499	0/100	-0.947 ± 0.074	15/100	-0.982 ± 0.024	30/100
EA	36.497 ± 15.537	0/100	4.861 ± 1.920	0/100	0.090 ± 0.096	2/100	-0.157 ± 0.221	29/100

TAB. 3.3 – Comparaison entre GBNM et le meilleur point d’une population d’un algorithme évolutionnaire pour la fonction de Griewank, $n = 12$, 100 exécutions (moyenne \pm écart-type).

une structure stationnaire (*steady-state*) ([82]) avec codage réel, croisement continu, et mutation gaussienne de variance $\sigma_i^{\text{mut}} = (x_i^{\text{max}} - x_i^{\text{min}})^2/16$. Pour une comparaison juste, les paramètres de l’EA choisis pour chaque test sont ceux qui présentent la meilleur performance en 100 exécutions indépendantes, parmi toutes les combinaisons de taille de population (20 ou 50), probabilité de mutation (0.15 ou 0.20), et probabilité de croisement (0.4 ou 0.5). Les résultats rapportés sont les moyennes (des meilleures variantes de l’EA), le point initial de GBNM étant choisi aléatoirement. On considère que l’utilisation type d’un EA est d’approcher l’optimum global, la détermination précise pouvant être effectuée a posteriori par un optimiseur local. Pour cela, dans les tests, on considère que l’EA trouve x^* s’il s’en est suffisamment approché, ce que l’on écrit,

$$\frac{1}{n} \|\hat{x}^* - x^*\| < 1, \quad (3.3)$$

où \hat{x}^* est le meilleur point trouvé, et x^* le minimum global. La définition de l’optimum global est plus sévère pour GBNM,

$$\|\hat{x}^* - x^*\| < 10^{-6}. \quad (3.4)$$

On constate que la méthode GBNM trouve, en moyenne, des meilleures valeurs pour la fonction objectif, avec une probabilité plus élevée de trouver le minimum global. L’avantage de GBNM sur EA est substantiel pour un nombre d’analyses faible, et diminue avec l’augmentation du coût numérique.

3.3 Minimisation de fonctions sous contraintes

Dans les problèmes contraints exposés dans cette section et dans la Section 4.2, pour une meilleur interprétation des résultats, les recherches primale et duale sont dissociées. 100 exécutions avec GBNM sont effectuées. Des valeurs nulles sont utilisées pour les paramètres de pénalisation

initiaux, $\lambda_i^0 = 0$. Les valeurs convergées sont moyennées, $\bar{\lambda}_i$, et employées comme paramètres de pénalisation constants ($s = 0$) pour la recherche primale. L'algorithme évolutionnaire employé dans les comparaisons adopte aussi la pénalisation linéaire adaptative pour la prise en compte des contraintes.

Dans cette Section, deux fonctions, présentées en [64], sont considérées pour tester l'algorithme GBNM. La première a deux variables et deux contraintes,

$$\left\{ \begin{array}{l} \min_{x_1, x_2 \in [0.001; 20]} -\frac{(\sin(2\pi x_1))^3 \sin(2\pi x_2)}{x_1^3(x_1+x_2)}, \\ \text{tel que,} \\ g_1(x_1, x_2) = x_1^2 - x_2 + 1 \leq 0, \\ g_2(x_1, x_2) = 1 - x_1 + (x_2 - 4)^2 \leq 0. \end{array} \right. \quad (3.5)$$

Le minimum global est -0.0958248 , et se trouve en $x_1 = 1.2280$, $x_2 = 4.2454$. Pour ce problème, en utilisant $s = 0.001$, les coefficients de pénalisation suivants ont été trouvés : $\bar{\lambda}_1 = 5.5$ et $\bar{\lambda}_2 = 98.4$, obtenus après 12204 à 22418 analyses.

Le deuxième test est une fonction à 7 variables, avec 4 contraintes,

$$\left\{ \begin{array}{l} \min_{x_i \in [-20, 20]} (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + \\ \quad 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7, \\ \text{tel que,} \\ -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0, \\ -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0, \\ -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0, \\ 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0. \end{array} \right. \quad (3.6)$$

Le minimum global est 680.6300573 , en $x^* = (2.330499 ; 1.951372 ; -0.4775414 ; 4.365726 ; -0.624487 ; 1.038131 ; 1.594227)$, avec la première et la dernière contraintes actives. Pour ce problème, les coefficients de pénalisation ont été : $\bar{\lambda}_1 = 68.5$, $\bar{\lambda}_2 = 26.0$, $\bar{\lambda}_3 = 5.2$, et $\bar{\lambda}_4 = 3.8$, trouvés après 52 à 620 analyses, avec $s = 0.0001$.

La Table 3.4 présente les résultats pour les Tests 1 et 2. La méthode GBNM ($\alpha = 0.01$, $N_r = 10$) est comparée avec le meilleur algorithme évolutionnaire (EA) de taille de population = 50 (Test 1) et 20 (Test 2), probabilité de mutation = 0.2, probabilité de croisement = 0.4, et les

		500 analyses		1000 analyses		2000 analyses	
		f du meilleur point admissible trouvé	Probabilité de trouver un minimum admissible	f du meilleur point admissible trouvé	Probabilité de trouver un minimum admissible	f du meilleur point admissible trouvé	Probabilité de trouver un minimum admissible
Test 1	GBNM	-0.093768 ± 0.011755	100/100	-0.095825(*) ± 0.00000	100/100	-0.095825(*) ± 0.00000	100/100
	EA	-0.059676 ± 0.029717	92/100	-0.064979 ± 0.032577	100/100	-0.087380 ± 0.019187	100/100
Test 2	GBNM	694.00 ± 15.36	69/100	685.18 ± 3.47	86/100	683.49 ± 2.40	99/100
	EA	904.14 ± 102.65	65/100	755.35 ± 43.45	86/100	718.74 ± 27.97	86/100

(*) probabilité de 100/100 de trouver le minimum global

TAB. 3.4 – Comparaison entre GBNM et EA pour les Tests 1 et 2 (moyenne \pm écart-type, 100 exécutions).

mêmes paramètres de pénalisation constants, $\bar{\lambda}_i$, comme pour la méthode GBNM. Trois valeurs différentes du nombre d'évaluations de la fonction coût ont été fixées : 500, 1000 et 2000. 100 exécutions indépendantes sont réalisées, le premier point d'initialisation pour la méthode GBNM est choisit aléatoirement.

A partir de l'observation de la Table 3.4, on constate que la méthode GBNM trouve, en moyenne, des meilleures valeurs de la fonction coût, avec une probabilité plus haute de trouver un minimum admissible, par rapport à l'algorithme évolutionnaire EA. Comme déjà observé dans la minimisation sans contraintes, l'avantage de GBNM est substantiel pour un faible nombre d'évaluations de la fonction objectif, et diminue avec l'augmentation des ressources numériques disponibles. On remarque que le nombre d'évaluations présenté dans la Table 3.4 ne prend pas en compte la procédure pour l'obtention des paramètres $\bar{\lambda}_i$. Effectivement, pour le Test 1, la majorité des ressources est dépensée pour trouver les paramètres de pénalisation $\bar{\lambda}_i$, ce que confirment les observations de [64] et [50] : il est difficile de trouver les paramètres de pénalisation appropriés pour le Test 1.

Le Test 1 est aussi un problème qui présente des minima locaux. Plusieurs minima possibles ont été trouvés par GBNM en une seule exécution de 2000 évaluations, et sont rapportés dans la Table 3.5.

Des tests additionnels sur les fonctions analytiques ont été réalisés et des comparaisons avec la fonction *fmincon* de Matlab sont résumées en Annexe C.

$(x_1 ; x_2)$	valeur de la fonction	contrainte g_1	contrainte g_2
$(1.22797 ; 4.24537)$ (*)	-0.095825	-1.73746	-0.167762
$(1.73414 ; 4.74608)$	-0.0291438	-0.738839	-0.177499
$(1.32441 ; 3.43043)$	-0.0272629	-0.67637	-8.6956×10^{-10}
$(1.67400 ; 3.80228)$	-0.0258123	-1.6623×10^{-9}	-0.634906
$(0.0386021 ; 4.22424)$	-55.7608	-3.22275	1.01168(**)

(*) optimum global

(**) contrainte non satisfaite

TAB. 3.5 – Exemples d’optima locaux possibles trouvés par GBNM pour le Test 1, en 1 exécution de 2000 analyses.

Chapitre 4

Application aux plaques composites stratifiées

Dans ce Chapitre sont présentés des résultats d'optimisation de plaques composites stratifiées. Les composites stratifiés sont composés de couches empilées, où chaque couche (ou pli) est caractérisée par des fibres orientées dans une certaine direction. L'ensemble des fibres est noyé dans une matrice en matériau isotrope de résistance mécanique beaucoup plus faible, comme par exemple l'époxyde (voir schéma en Figure 4.1). La matrice conserve la disposition géométrique des fibres et les protège. Elle transmet et distribue les sollicitations mécaniques ([36, 24]). La résistance et la rigidité d'un pli est la plus grande dans la direction où les fibres sont orientées, et la plus faible dans la direction perpendiculaire. Cependant, les propriétés mécaniques d'un assemblage de plis (un stratifié) ne peuvent pas, en général, être trivialement prédites, encore moins optimisées. Les problèmes résolus ici sont la détermination des orientations optimales des fibres pour chaque pli i , θ_i , qui donnent la meilleur performance sous certaines conditions. Depuis 20 ans environ, de nombreux travaux de recherche ont visé à optimiser la position des fibres des matériaux composites, ou, plus précisément, à optimiser des séquences d'empilement. Les premiers travaux sont attribués à Schmit et Farshi ([42]) qui ont conçu des stratifiés de masse minimale avec des contraintes sur le flambement en faisant varier les épaisseurs de plis. Depuis, Pedersen ([72]) a montré qu'en termes d'orientation des fibres, les problèmes d'optimisation de composites tendent à avoir plusieurs optima locaux ce qui les rend difficiles à approcher au moyen de méthodes d'optimisation locales basées sur les gradients. Dans certains cas néanmoins, en utilisant des variables de haut niveau qui résument la contribution des orientations de tous les plis,

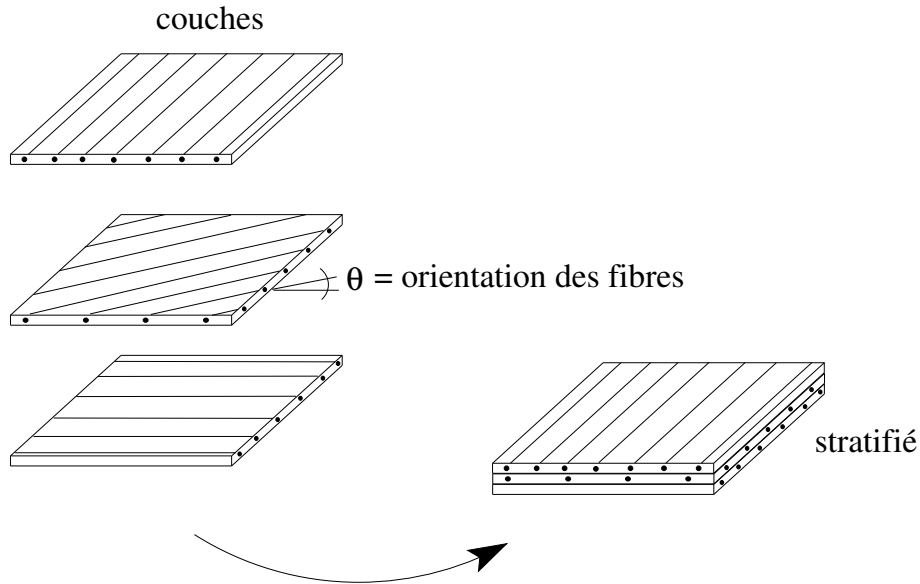


FIG. 4.1 – Schématisation de l’empilement de couches qui constitue un stratifié.

les paramètres de stratification ([33]), il est possible de résoudre le problème d’optimisation de stratifiés plus facilement par l’emploi d’une fonction continue et par une réduction du nombre de dimensions ([65, 23, 30, 31]). Cependant l’utilisation des paramètres de stratification n’est pas toujours possible (e.g. l’étude du phénomène de rupture au niveau des plis), et la traduction en une séquence d’empilement est un problème non trivial. D’autres auteurs enfin se sont tournés vers les méthodes d’optimisation stochastiques, en considérant les orientations des plis comme des quantités discrètes ([51]) ou continues ([48]). Un état de l’art détaillé de l’optimisation de stratifiés composites peut être trouvé dans [29].

Un problème de maximisation de la charge critique de flambement d’une plaque composite, puis la maximisation d’un module d’élasticité, sont discutés dans la Section 4.1. Ces deux problèmes d’optimisation sont sans contraintes et ont des solutions connues. Dans la Section 4.2, le module d’élasticité est maximisé sous les contraintes du module de cisaillement et du coefficient de Poisson. Enfin, un problème de maximisation de la charge critique de flambement, sous les contraintes du critère de rupture de Hoffman et des valeurs maximales pour les coefficients de dilatation thermique est résolu.

L’algorithme GBNM est comparé à un algorithme évolutionnaire (EA). L’algorithme évolutionnaire ([5]) a une structure stationnaire (*steady-state*) ([82]) avec codage réel, croisement continu, et mutation gaussienne de variance $\sigma_i^{\text{mut}} = (x_i^{\text{max}} - x_i^{\text{min}})^2/16$. Pour une comparaison juste, les paramètres de l’EA choisis pour chaque test sont ceux qui présentent la meilleur performance en

100 exécutions indépendantes, parmi toutes les combinaisons de taille de population (20 ou 50), de probabilité de mutation (0.15 ou 0.20), et de probabilité de croisement (0.4 ou 0.5).

4.1 Problèmes sans contraintes

4.1.1 Maximisation de charge de flambement

Considérons le problème de maximisation¹ de la charge critique de flambement d'une plaque carrée de carbone-époxyde, symétrique et équilibrée avec 16 couches. Chaque couche a 0.125 *mm* d'épaisseur, et les propriétés élastiques du matériau avec fibres dans un repère aligné sur les fibres sont : $E_1 = 115 \text{ GPa}$, $E_2 = 5 \text{ GPa}$, $G_{12} = 5 \text{ GPa}$ et $\nu_{12} = 0.35$. La plaque est simplement supportée sur ses 4 cotés, et soumise à un chargement de membrane $N_x = N_y$, comme montré dans la Figure 4.2. Les hypothèses utilisées pour la solution du problème de flambement de plaques sont celles présentés dans [7], où la théorie classique des stratifiés et un modèle de flambement élastique et linéaire sont employés. Le stratifié est symétrique par rapport à son plan moyen, il n'est donc nécessaire que de décrire la moitié des plis. Le stratifié est de plus équilibré, ce qui signifie qu'à chaque pli d'orientation $+\theta$ correspond un d'orientation $-\theta$. Ceci permet de diviser à nouveau par 2 le nombre de plis à considérer. Pour un stratifié de 16 plis, les variables se limitent à 4 orientations. Ces variables sont continues et comprises entre 0 et 90°. Ce problème a une solution unique selon laquelle toutes les couches sont orientées à $\pm 45^\circ$. On note aussi que, à cause de la raideur de flexion, les plis extérieurs ont plus d'influence sur le comportement de flambage que les plis intérieurs.

La Table 4.1 compare GBNM et un algorithme évolutionnaire (EA). Le point initial et la taille du simplexe utilisés pour les essais avec GBNM sont choisis aléatoirement (la taille, entre 2 et 10% de la dimension du domaine), et la tolérance de convergence $\epsilon_{small} = 1 \times 10^{-6}$. Les résultats rapportés dans la Table 4.1 sont les moyennes sur 100 exécutions. En ce qui concerne la méthode GBNM, la Table 4.1 indique de plus le nombre d'analyses pour trouver le minimum une fois (ce qui a un sens car il n'y a qu'un optimum). Cet exemple montre que la précision des résultats obtenus par GBNM est supérieure à celle d'un algorithme évolutionnaire, quand on a un nombre petit de variables. On note que l'algorithme évolutionnaire ne présente pas une précision

¹En forme normalisée, on minimise $1 - \frac{\lambda_{cri}}{\lambda_{ref}}$, où λ_{cri} est le facteur critique de flambement et λ_{ref} est une valeur de référence, typiquement égale à 1.

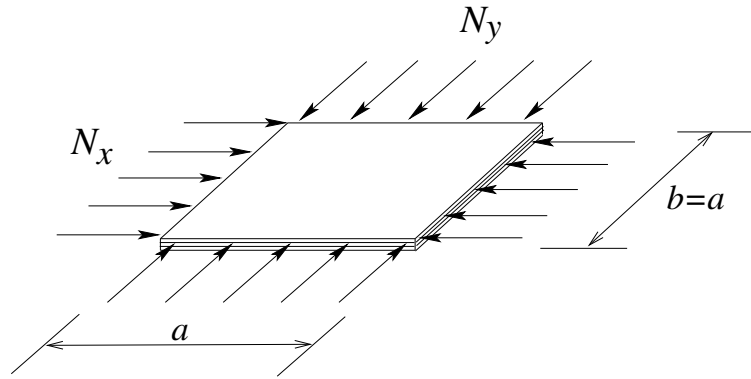


FIG. 4.2 – Plaque soumise à des forces compressives pour le problème de maximisation, sans contraintes, de la charge de flambement.

	Point de convergence				Nombre d'analyses
GBNM	[±45.0000] _{4s}				288.62
écart-type	[0.0000] ₄				±59.44
EA	±45.3583	±45.2355	±45.0125	±44.7363] _s	300
écart-type	±2.7235	±3.1970	±6.6032	±16.9491	-
EA	±45.0612	±44.8338	±44.9947	±44.2551] _s	500
écart-type	±1.5640	±2.1149	±4.0046	±10.2130	-
EA	±45.0690	±44.9371	±44.8874	±44.4456] _s	1000
écart-type	±0.5617	±0.8983	±1.3347	±3.3959	-

TAB. 4.1 – Maximisation de charge de flambement, sans contraintes, 4 variables, 100 exécutions.

élevée même quand le nombre d'évaluations de la fonction coût est beaucoup plus grand que celui nécessaire à la convergence de GBNM.

Pour analyser la performance de l'algorithme GBNM quand le nombre de variables est augmenté, le problème de flambement décrit ci-dessus est résolu, mais en utilisant 32 couches, c'est-à-dire 8 variables. La Table 4.2 compare GBNM et un algorithme évolutionnaire (EA). Elle présente les séquences d'empilement trouvées en 300, 500 et 1000 appels de la fonction objectif (coût fixe pour les deux algorithmes). Les résultats sont moyennés sur 100 exécutions indépendantes. En 300 évaluations, les valeurs des variables trouvées par GBNM et par EA sont globalement équivalentes. Cependant, pour 300 analyses toujours, GBNM, qui est par nature une recherche plus orientée, converge plus rapidement que EA pour les couches extérieures, et vice-versa pour les couches intérieures. A partir de 500 évaluations, GBNM converge plus précisément sur l'optimum que EA quelques soient les couches considérées.

Séquence d'empilement : moyenne \pm écart-type								
300 analyses								
GBNM	[± 45.04	± 44.97	± 45.02	± 45.38	± 45.38	± 44.97	± 43.28	± 49.65] _s
éc.-type	± 0.47	± 0.54	± 0.83	± 4.61	± 4.37	± 11.53	± 17.46	± 23.47
EA	[± 45.09	± 44.91	± 45.23	± 44.55	± 44.78	± 45.02	± 45.16	± 44.85] _s
éc.-type	± 1.75	± 1.96	± 2.67	± 3.00	± 3.66	± 5.19	± 8.80	± 15.69
500 analyses								
GBNM	[± 45.01	± 45.03	± 45.00	± 45.08	± 45.04	± 44.97	± 45.05	± 45.18] _s
éc.-type	± 0.17	± 0.22	± 0.39	± 0.40	± 0.29	± 0.46	± 0.92	± 4.22
EA	[± 45.13	± 44.95	± 44.99	± 44.95	± 44.80	± 45.00	± 44.70	± 46.45] _s
éc.-type	± 1.09	± 1.28	± 1.61	± 1.90	± 2.22	± 3.22	± 4.49	± 11.30
1000 analyses								
GBNM	[± 45.00	± 45.00	± 45.00	± 45.00	± 44.99	± 45.00	± 44.98	± 45.02] _s
éc.-type	± 0.02	± 0.02	± 0.03	± 0.05	± 0.04	± 0.06	± 0.15	± 0.44
EA	[± 44.96	± 44.98	± 44.96	± 45.07	± 44.99	± 44.92	± 45.13	± 45.27] _s
éc.-type	± 0.60	± 0.61	± 0.71	± 0.95	± 1.12	± 1.17	± 1.71	± 4.95

TAB. 4.2 – Maximisation de charge de flambement, sans contraintes, 8 variables, 100 exécutions.

4.1.2 Maximisation de module d'élasticité longitudinal

Le problème de la maximisation² du module d'élasticité E_x d'une plaque à 16 couches en verre-époxyde est étudié. Les propriétés des plis de verre-époxyde sont : $E_1 = 45 \text{ GPa}$, $E_2 = 10 \text{ GPa}$, $G_{12} = 4.5 \text{ GPa}$ et $\nu_{12} = 0.31$. La plaque est symétrique et équilibrée, donc il y a 4 orientations de fibres à trouver. Ce problème présente 16 optima locaux qui sont toutes les combinaisons des orientations 0° et 90° . L'optimum global est tel que toutes les couches sont à 0° . Cet exemple montre que, dans le domaine de la conception avec des composites, des optima locaux apparaissent même pour des problèmes simples a priori. Ce cas est similaire à ceux de l'Annexe D, où l'existence de solutions locales est mise en évidence graphiquement pour une et deux variables.

La Table 4.3 donne le nombre moyen d'optima locaux trouvés en 2000 analyses, en fonction de N_r , basé sur 100 exécutions indépendantes de GBNM. Le premier point initial et la taille initiale du simplexe sont choisis aléatoirement. On remarque que plus N_r est grand, plus il y a d'optima trouvés. Ceci est prévisible car les optima sont placés régulièrement sur le domaine. En outre, avec un budget de 2000 analyses, l'optimum global est trouvé pour toutes les exécutions, puisque son bassin d'attraction est le plus grand (voir Annexe D). Ce cas est particulier. Dans les cas généraux non Lipschitziens, on ne peut pas faire de relation directe entre l'optimum global et la

²En forme normalisée, on minimise $1 - \frac{E_x}{E_{ref}}$, où E_{ref} est une valeur de référence, qui dans le cas présent est $30 \times 10^9 \text{ Pa}$.

N_r	nombre moyen d'optima trouvés	écart-type
1	8.92	1.19
2	9.11	1.13
3	9.48	1.22
10	9.50	1.13
50	9.79	1.25
1000	10.03	1.41

TAB. 4.3 – Maximisation de E_x sans contraintes, 2000 analyses, 100 exécutions.

taille du bassin d'attraction.

4.2 Problèmes avec contraintes

4.2.1 Maximisation du module d'élasticité longitudinal sous les contraintes du module de cisaillement et du coefficient de Poisson

Une plaque symétrique, équilibrée, avec 16 plis, fabriquée en verre-époxyde, doit être conçue en maximisant son module d'élasticité longitudinal E_x , de telle sorte que le module de cisaillement G_{xy} soit au moins de 12 GPa , et le coefficient de Poisson ν_{xy} soit plus petit que 0.5. Les propriétés élastiques des plis de verre-époxyde sont : $E_1 = 45 GPa$, $E_2 = 10 GPa$, $G_{12} = 4.5 GPa$ et $\nu_{12} = 0.31$. La plaque est équilibrée et symétrique, ainsi il y a 4 orientations de fibres à trouver, θ_i , limitées entre 0° et 90° . Pour ce problème, les valeurs convergées moyennes des paramètres de pénalisation sont $\bar{\lambda}_1 = 6.24$ et $\bar{\lambda}_2 = 1.18$, obtenus après 44 à 435 analyses, avec $s = 1.0$. L'algorithme évolutionnaire le plus performant pour ce problème a une taille de population de 20, une probabilité de mutation de 0.2 et une probabilité de croisement de 0.4.

La Table 4.4 compare les algorithmes après 100, 200, et 500 évaluations de la fonction coût, basée sur 100 exécutions. Les meilleures solutions trouvées sont des permutations de la séquence d'empilement $[\pm 36.6 / \pm 43.1 / \pm 50.1 / \pm 54.9]_s$. Elles ont $E_x = 14.54 GPa$ et les deux contraintes sont actives.

D'autres séquences d'empilement quasi-optimales engendrent des convergences locales des simples et sont gardées comme des optima locaux possibles par GBNM (voir Figure 2.8). Ceux d'entre eux qui ont un écart plus petit que 0.2% de E_x optimal sont listés dans la Table 4.5.

	100 analyses		200 analyses		500 analyses	
	Plus grand E_x admissible [GPa]	Probabilité de trouver un optimum admissible	Plus grand E_x admissible [GPa]	Probabilité de trouver un optimum admissible	Plus grand E_x admissible [GPa]	Probabilité de trouver un optimum admissible
GBNM	14.4906±0.0862	47/100	14.5302±0.0059	97/100	14.5311±0.0000	99/100
EA	13.9060±0.3287	44/100	14.2502±0.2090	56/100	14.4550±0.0965	70/100

TAB. 4.4 – Maximisation de E_x sous contraintes, 100 exécutions (moyenne±écart-type).

Séquence d'empilement	E_x [GPa]	G_{xy} [GPa]	ν_{xy}
$[\pm 41.7 / \pm 57.7 / \pm 46.2 / \pm 39.5]_s$	14.53	12.00	0.50
$[\pm 43.0 / \pm 45.2 / \pm 39.1 / \pm 57.8]_s$	14.53	12.00	0.50
$[\pm 48.5 / \pm 34.9 / \pm 47.4 / \pm 53.9]_s$	14.53	12.00	0.50
$[\pm 49.7 / \pm 45.4 / \pm 35.4 / \pm 54.2]_s$	14.53	12.00	0.50
$[\pm 51.9 / \pm 52.1 / \pm 45.5 / \pm 35.1]_s$	14.53	12.00	0.50
$[\pm 57.7 / \pm 38.8 / \pm 45.0 / \pm 43.6]_s$	14.52	12.00	0.50

TAB. 4.5 – Exemples de solutions quasi-optimales trouvées pour le problème de la maximisation de E_x avec contraintes, 2000 analyses, 1 exécution.

4.2.2 Maximisation de charge de flambement sous les contraintes du critère de rupture de Hoffman et des coefficients de dilatation thermique

On considère une plaque rectangulaire simplement supportée sur ses 4 cotés, soumise à des charges dans son plan, $N_x = -2000$ N/mm et $N_y = 2000$ N/mm, schématisée en Figure 4.3. La plaque est équilibrée et symétrique, et a 48 couches de 0.125 mm d'épaisseur chacune. Les propriétés élastiques des couches sont $E_1 = 115$ GPa, $E_2 = 5$ GPa, $G_{12} = 5$ GPa, $\nu_{12} = 0.35$. Les coefficients de dilatation thermique sont $\alpha_{t_1} = -0.5 \times 10^{-6}/^\circ C$ et $\alpha_{t_2} = 20 \times 10^{-6}/^\circ C$. Les contraintes limites de rupture des plis dans le sens longitudinal (X_t et X_c), transversal (X_t et X_c) et en cisaillement (S) sont : $X_t = 964$ MPa, $X_c = 895$ MPa, $Y_t = 50$ MPa, $Y_c = 100$ MPa et $S = 94$ MPa. Le stratifié est conçu en maximisant le facteur de sécurité au flambement, f_{flamb} , sous les restrictions de ne pas avoir de défaillance selon le critère de Hoffman ([7]) et de garder les coefficients de dilatation thermique $|\alpha_{t_x}|$ et $|\alpha_{t_y}|$ inférieurs à $1 \times 10^{-6}/^\circ C$. Un modèle de flambement élastique linéaire est utilisé ([7]).

Puisque la plaque est équilibrée et symétrique, il y a 12 variables continues à trouver, les orientations des plis, qui sont limitées entre 0 et 90°.

Les valeurs convergées moyennes des paramètres de pénalisation sont $\bar{\lambda}_1 = 0.0062$, $\bar{\lambda}_2 = 0.0598$ et $\bar{\lambda}_3 = 0.4048$, où la convergence dans l'espace dual se produit en 116 à 978 analyses, avec

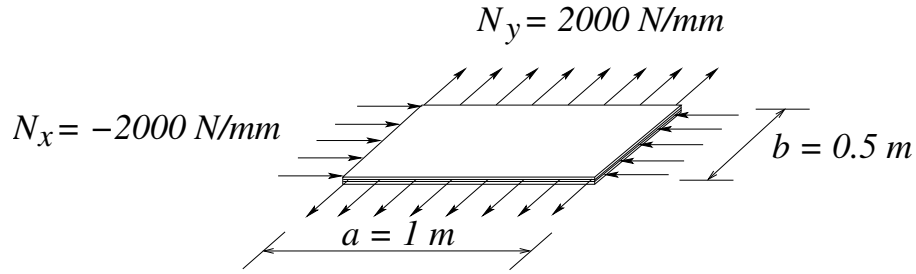


FIG. 4.3 – La géométrie et le chargement de la plaque pour le problème de maximisation, avec contraintes, de la charge de flambement.

	200 analyses		500 analyses		1000 analyses	
	f_{flamb} du meilleur point admissible	Probabilité de trouver un optimum admissible	f_{flamb} du meilleur point admissible	Probabilité de trouver un optimum admissible	f_{flamb} du meilleur point admissible	Probabilité de trouver un optimum admissible
GBNM	1.4260±0.0513	78/100	1.4883±0.0145	99/100	1.4959±0.0125	100/100
EA	1.4557±0.0134	69/100	1.4806±0.0073	83/100	1.4919±0.0035	94/100

TAB. 4.6 – Maximisation de charge de flambement sous contraintes, 100 exécutions (moyenne±écart-type).

$s = 0.001$. L'algorithme évolutionnaire le plus performant pour ce problème a une taille de population 50, une probabilité de mutation 0.15 et une probabilité de croisement 0.5. La Table 4.6 résume la comparaison entre les deux méthodes à 200, 500 et 1000 analyses, sur 100 essais indépendants. La meilleure solution trouvée par GBNM est $[\pm 27.5/\pm 27.5/\pm 28.0/\pm 28.7/\pm 29.7/\pm 30.7/\pm 32.2/\pm 35.8/\pm 43.9/\pm 70.7/\pm 89.9/\pm 89.9]_s$, où $f_{flamb} = 1.5002$, $\alpha_{t_x} = 3.04 \times 10^{-7}/^\circ C$, $\alpha_{t_y} = 1.00 \times 10^{-6}/^\circ C$, et le facteur de sécurité pour le critère de Hoffman est égal à 1.286.

Les résultats présentés dans ce Chapitre corroborent les tests sur les fonctions analytiques : la méthode GBNM est plus rapide et plus robuste pour trouver des minima admissibles que l'algorithme évolutionnaire. L'avantage, qui est important en dessous de 500 analyses, diminue progressivement à mesure que le nombre d'analyses est augmenté.

Chapitre 5

Conclusions de la Partie I

Les problèmes de conception de stratifiés composites sont révélateurs de difficultés rencontrées plus généralement lors de l'optimisation de systèmes mécaniques, à savoir,

- il y a plusieurs optima locaux ;
- les variables ont un sens physique, et sont donc bornées ;
- le calcul du gradient est souvent laborieux, numériquement coûteux (différences finies), ou bien il n'existe pas, ainsi on évite de le calculer ;
- le temps de calcul est limité à quelques milliers d'évaluations de la fonction coût.

Les principes qui ont guidé le développement de l'algorithme GBNM ont été :

- l'utilisation de l'algorithme de Nelder-Mead, car il ne nécessite pas le calcul du gradient et est assez rapide. Cependant, la méthode classique de Nelder-Mead est une méthode locale utilisée pour l'optimisation sans contraintes et sans limites sur les bornes, et la recherche peut échouer par dégénérescence du simplexe ;
- l'amélioration de Nelder-Mead par détection des cas de dégénérescence et la ré-initialisation du simplexe ;
- la prise en compte des bornes par projection ;
- un test d'optimalité, sur les bornes, sans calcul de gradient ;
- la prise en compte des contraintes par pénalisation adaptative ;
- la globalisation par ré-initialisation probabilisée, pour ne pas échantillonner des régions déjà explorées et pour travailler à coût d'optimisation donné a priori. Si le budget d'optimisation est faible par rapport à la taille de l'espace de recherche, GBNM ne réalise qu'une recherche

locale. Réciproquement, GBNM devient global si le nombre d'analyses le permet.

La ré-initialisation probabilisée avec N_r grand produit des points initiaux fortement biaisés par les points initiaux et par les points de convergence précédents, ce qui s'avère, en général, inefficace, sauf si les minima suivent un motif régulier. En comparant les ré-initialisations totalement aléatoire ($N_r = 1$), aléatoire-biaisée ($N_r = 3$ ou 10) et régulière ($N_r > 10$), sur une suite de problèmes tests, nous avons constaté qu'il était préférable d'opter pour la stratégie aléatoire-biaisée, qui trouve une variété plus grande de minima.

Nous avons observé que, pour un nombre petit de variables, l'algorithme GBNM est plus efficace que le calcul évolutionnaire. Quand le nombre de variables est grand ($n \geq 10$), GBNM peut présenter des difficultés pour trouver un minimum mais sa convergence demeure meilleure que celle de l'algorithme évolutionnaire.

La globalisation par ré-initialisation probabilisée n'est pas restrictive à l'algorithme de Nelder-Mead. Elle peut être appliquée à d'autres optimiseurs locaux. Par contre, la prise en compte des bornes par projection ne peut pas être utilisée pour un optimiseur quelconque et s'avérera souvent incompatible avec son principe de recherche (par exemple, recherches par motifs, par directions conjuguées).

Deuxième partie

Optimisation de monopalmes de nage 2D par GBNM

Chapitre 6

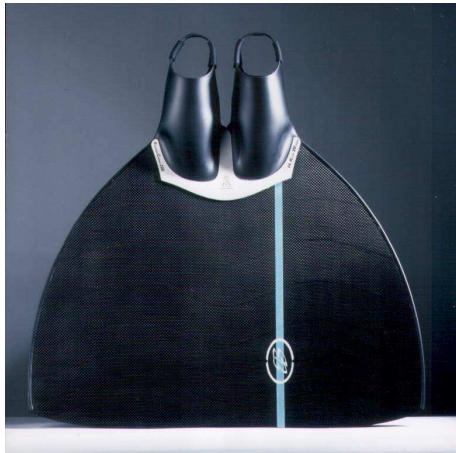
Introduction et étude bibliographique

La monopalme a été inventée dans les années 60 en Sibérie. Depuis, elle est devenue l'instrument des records de vitesse de nage et de plongée en apnée (Figure 6.1). Elle est une sorte de nageoire caudale, et permet au nageur d'imiter le déplacement des poissons et mammifères marins grâce à un mouvement ondulatoire du corps. Cette ondulation débute au niveau des épaules et continue jusqu'à l'extrémité de la palme. Une fois les techniques de base maîtrisées, un monopalmeur peut atteindre une vitesse de 8 à 10 km/h. A titre de comparaison, la vitesse moyenne du record mondial des 50 m nage libre est d'environ 8.4 km/h. Les monopalmes les plus sophistiquées sont fabriquées en matériaux composites.

Si les nageoires caudales des animaux marins ont été une inspiration, le dessin des monopalmes actuelles a été obtenu de façon empirique. A notre connaissance, il n'y a pas d'étude préalable concernant la conception de monopalme, et peu d'études ont été publiées sur la nage avec monopalmes. De nombreuses études ont, au contraire, été menées sur le mouvement des poissons et l'écoulement de fluide autour de leurs nageoires pour comprendre les mécanismes mis en jeu dans leur propulsion (e.g., [15, 79, 87]).

L'observation de certains poissons et mammifères marins, leurs performances, leurs capacités à se déplacer rapidement et sur de grandes distances, nous révèlent qu'ils utilisent un mode de locomotion efficace, adapté à leurs nécessités, résultat de millions d'années d'évolution. Bien que la monopalme soit un système passif alors que les nageoires de poissons sont actives, les liens entre les deux systèmes sont suffisamment étroits pour qu'en Section 6.1 soient résumés les travaux concernant la nage des poissons.

Les principes de la technique de nage avec monopalmes sont décrits dans [28]. Dans [18]



(a)



(b)

FIG. 6.1 – Monopalmes carbone Breier© (a). Utilisation par J.-M. Pradon (b).

et [94] des études comparant les dépenses énergétiques de la nage libre et de la nage avec bipalmes, respectivement, ont été réalisées. A notre connaissance, il n'existe pas de travaux publiés spécifique à la dépense énergétique en nage avec monopalmes. Cependant, dans [2], des études de physiologie du sport descriptives ont été consacrées à la cinématique de la nage avec monopalmes.

Comme la monopalme est flexible et déformable sous l'effet de l'écoulement fluide qui, à son tour, est perturbé par la nouvelle position de la structure, on est confronté à un phénomène de couplage fluide-structure. De nombreux travaux sur la modélisation du couplage entre le fluide et la structure et son impact sur la conception de la structure ont été réalisés (e.g.,[41]). Dans le domaine des voilures rigides ou peu déformables, beaucoup de travaux ont été faits en relation, en particulier, avec l'aéronautique. Les références sur les voilures flexibles sont beaucoup moins nombreuses bien que des études aient plus récemment été réalisées notamment pour la modélisation des voiles ([45, 16, 13, 67]). Le modèle de couplage fluide-structure adopté dans ce travail est décrit dans le Chapitre 7 et en Annexe E. La Section 6.2 présente les objectifs de cette deuxième Partie.

6.1 Nage d'animaux marins

Pour se déplacer, les poissons les plus rapides fléchissent leur corps dans un mouvement ondulatoire qui va jusqu'à la nageoire caudale. Ces mouvements se succèdent à un rythme tel qu'une nouvelle vague commence avant que la précédente n'ait atteint l'extrémité du corps. Cette façon de nager est classifiée comme locomotion par le corps et/ou par la nageoire caudale (*BCF : body and/or caudal fin locomotion*). D'autres poissons, plus lents, se propulsent par les nageoires médianes et pectorales, ce qui est appelé locomotion par nageoires médianes et/ou par nageoires paires (*MPPF : median and/or paired fin locomotion*).

Une autre classification trouvée dans la littérature est basée sur la caractéristique du mouvement : ondulatoire (une onde se propage par la structure propulsive) et oscillatoire (la structure propulsive bat en mouvement cyclique ayant comme pivot sa base, sans formation d'une onde) ([79]). Quatre modes de locomotion sont identifiés dans la locomotion ondulatoire des BCF. Ils dépendent de la longueur d'onde du corps (cf. Figure 6.2), et deux méthodes principales de propulsion sont identifiées : la méthode de la masse ajoutée et celle de la portance (ou vorticit ). Cette derni re est utilis e dans la nage des *thunniforms*. Les modes *anguiliform*, *subcarangiform* et *carangiform* sont associ s   la m thode de la masse ajout e. Selon [79], le mode de nage des *thunniforms* est le mode le plus efficace, o  la pouss e est cr e e par des tourbillons et permet des hautes vitesses de croisi re pendant de longues p riodes. Il est consid r e comme le plus avanc e du point de vue de l' volution et est trouv e chez les poissons teolost (thon, maquereau, etc.), requins et mammif res marins (dauphin, baleine, phoque, etc.) La nageoire caudale produit environ 90% de la pouss e. Elle est grande et raide, avec une forme en croissant de lune usuellement d sign e comme *lunate*. La forme des *thunniforms* est optimis e pour des hautes vitesses dans l'eau calme.

La nage avec monopalmes essaie d'imiter le mouvement des mammif res marins. Les mammif res marins nagent en fl chissant leur queue de haut en bas, contrairement   la majorit e des poissons, qui fl chissent leur corps lat ralement. La plupart des monopalmes sur le march  pr sentent la forme de la Figure 6.1(a), mais d'autres formes sont trouv es, comme par exemple celle pr sent e en Figure 6.3(a), qui ressemble plus   une queue de dauphin (Figure 6.3(b)).

Le battement de la nageoire caudale provoque des tourbillons successifs en sens oppos s (cf. Figure 6.4), responsables d'un effet de jet et en cons quence produit la pouss e. Un sillage, form e par l'ensemble des tourbillons discrets, est produit. Le param tre le plus utilis e pour caract riser

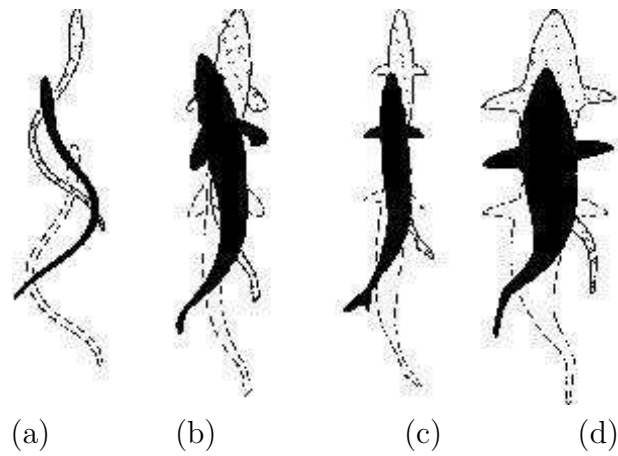


FIG. 6.2 – Mouvement de nage des BCF : modes anguilliform (a), subcarangiform (b), carangiform (c), et thunniform (d) (classification selon [59]).

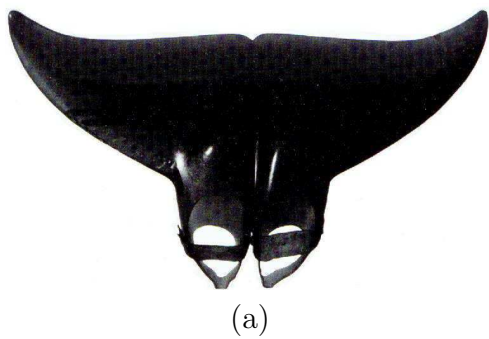


FIG. 6.3 – Monopalme en forme de queue de dauphin (Dolphin Tails, Inc©) (a), et vraies queues de dauphins (b).

la distribution de tourbillons générée par un aileron oscillant est le nombre de Strouhal S_t ,

$$S_t = \frac{fA}{U} \quad (6.1)$$

où f est la fréquence des battements, A la largeur du sillage (cf. Figure 6.4) et U la vitesse de nage. Comme la largeur du sillage est difficile à déterminer, A est, en général, approximé par deux fois l’amplitude des battements de la nageoire caudale (ou, dans notre cas, la hauteur parcourue par l’extrémité de la monopalme). Il a été constaté empiriquement que les poissons nageant vite ont $0.25 < S_t < 0.35$ ([87]). Des expériences avec des ailerons articulés montrent que l’efficacité (poussée moyenne multipliée par la vitesse moyenne du fluide divisée par la puissance d’entrée moyenne) est optimale pour ce même intervalle de nombre de Strouhal ([87]). Des discussions plus détaillées de l’hydrodynamique de la nage des poissons peuvent être trouvées dans les références [79] et [12].

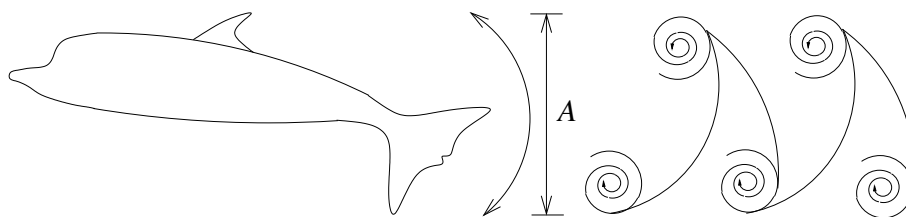


FIG. 6.4 – Schématisation des tourbillons créés par les oscillations de la nageoire caudale.

Les études de systèmes propulsifs sous-marins à base d’ailerons oscillants imitant la nage des poissons ont récemment retenu l’attention de plusieurs chercheurs ([88, 12, 62]). Une application notoire est le développement d’un robot sous-marin en forme de thon ([88]). Cette étude démontre que la nageoire caudale artificielle est un propulseur très efficace. La différence essentielle avec les monopalmes est que ces dernières sont très flexibles.

6.2 Objectifs de la Partie II

Cette seconde partie du mémoire a pour objectif la modélisation puis l’optimisation d’une monopalme. La modélisation d’une palme composite est très complexe car elle cumule de nombreuses difficultés : un écoulement fluide instationnaire, le couplage avec une structure en grands déplacements, et enfin l’usage de matériaux composites dans la palme.

La complexité des modèles usuels couplant fluide et structure rend la simulation du système

palme et nageur coûteuse en calculs. Ce coût interdit l'optimisation car elle implique la simulation de nombreux sous-systèmes. Ainsi, les concepteurs doivent prendre des décisions basées sur les résultats de modèles simplifiés. Cette étude propose donc des simplifications qui permettent d'optimiser la distribution de raideurs d'une monopalme de nage. Lorsque le fluide est pris en compte, on se limite à un problème bidimensionnel et à un fluide parfait (mais instationnaire). Le modèle présenté dans [45] et [46], initialement utilisé pour le calcul des voiles, servira à la description de l'écoulement autour de la palme. Ce modèle fluide peut reproduire les tourbillons créés lors du mouvement oscillatoire de la monopalme. Le nageur est représenté par des segments linéaires articulés dont la cinématique est imposée. La palme est modélisée par un système de barres rigides liées par des ressorts de torsion dont le mouvement en grands déplacements est couplé avec la dynamique du fluide. Le critère d'optimisation est défini en fonction des puissances propulsive et totale, moyennées dans le temps. Des études paramétriques avec des perturbations dans les raideurs sont réalisées, et finalement la monopalme est optimisée en utilisant l'algorithme GBNM. Les variables d'optimisation sont les raideurs des ressorts de torsion.

Chapitre 7

Une représentation bidimensionnelle de la palme et du nageur

7.1 Calcul de l'écoulement

Les palmes et nageurs sont représentés comme des corps infiniment minces plongés dans un liquide parfait (fluide incompressible et non visqueux) et instationnaire. Le nombre de Reynolds pour cette famille de problèmes est de l'ordre de 10^6 , ce qui justifie que l'on néglige la viscosité. L'écoulement est calculé au moyen du modèle proposé dans [46]. Il s'agit d'un modèle particulière, c'est-à-dire que le champ de vitesse du fluide est créé par la superposition du champ de vitesse à l'infini et des champs de vitesses générés par des tourbillons émis au bord de fuite. Les intensités des tourbillons sont calculées à chaque instant de manière à satisfaire : i) la condition de glissement du fluide le long de l'obstacle ; ii) la condition de Kutta, c'est-à-dire des vitesses d'écoulement égales sur les 2 cotés de l'obstacle au bord de fuite ; et iii) la conservation de la circulation totale. Ce modèle suppose que l'écoulement reste attaché tout le long de l'obstacle, ce qui est typiquement faux pour des grandes incidences de palme. Néanmoins, le modèle particulière est un bon compromis entre une description fine de la dynamique non linéaire d'un fluide instationnaire et le temps de calcul, ce qui est déterminant pour l'optimisation. Sur un PC standard, 8 s de simulation d'écoulement prennent de l'ordre de 17 *min* de calcul. Une présentation plus complète du modèle particulière est donnée en Annexe E.

7.2 Grandeurs caractéristiques et normalisation

7.2.1 Grandeurs caractéristiques

Dans l'optique de l'optimisation des propriétés mécaniques d'une palme, il convient au préalable de présenter les grandeurs caractéristiques du système étudié qui ont une influence fondamentale sur le rendement mécanique de la nage.

Le temps caractéristique du problème est défini comme l'inverse de la fréquence du battement de la nage. Notons T_c ce temps caractéristique :

$$T_c = \frac{1}{f} , \quad (7.1)$$

où f est la fréquence de battements.

Un choix naturel pour la longueur caractéristique du système serait la longueur de la palme, notée L_s , mais de plus amples réflexions conduisent plutôt à choisir comme caractéristique la longueur **hydrodynamique** du système en mouvement que nous noterons L_c . Cette longueur peut être soit la taille du nageur avec palmes, soit sa longueur totale avec les bras étendus en avant et les palmes, soit la longueur entre le centre de rotation de la palme (la hanche en première approximation) et l'extrémité de la palme. A partir de cette longueur et du temps caractéristique, on construit la vitesse caractéristique de la dynamique de nage, notée U_c , selon :

$$U_c = L_c f . \quad (7.2)$$

Cette vitesse caractérise la dynamique de nage uniquement, et non pas la vitesse d'avance du nageur (U_∞). Il peut paraître surprenant que la vitesse d'avance ne soit pas la vitesse caractéristique du problème, mais en vérité, ce que contrôle le nageur, c'est essentiellement sa cinématique de battement, en particulier la fréquence. Sa vitesse d'avance est un sous-produit de ses mouvements de nage qui dépend de nombreux paramètres (en particulier la résistance de vague et la résistance visqueuse du nageur). Autrement dit, dans notre analyse simplifiée, la vitesse d'avance n'est pas un paramètre, mais une donnée du problème. L'optimisation se fera après avoir choisi la vitesse d'avance, c'est-à-dire après avoir donné un point de fonctionnement.

La vitesse moyenne d'avance du nageur permet de définir un point de fonctionnement du système au travers d'un paramètre adimensionnel qui est le rapport entre la vitesse caractéristique

(du mouvement de nage) et la vitesse d'avance. Notons K_v ce rapport de vitesse :

$$K_v = \frac{U_c}{U_\infty} = \frac{L_c f}{U_\infty}. \quad (7.3)$$

K_v indique l'importance des effets instationnaires sur l'écoulement. $\sigma = 2\pi K_v$ est appelé la fréquence réduite ([79]). Il permet de distinguer plusieurs régimes de nage :

- Quand K_v est grand, la vitesse du battement est prépondérante devant la vitesse d'avance. Les efforts hydrodynamiques propulsifs générés par le battement peuvent être grands si la cinématique est efficace.
- A l'inverse, lorsque K_v est petit, la dynamique de la nage est lente vis à vis de la vitesse d'avance, et les efforts hydrodynamiques propulsifs induits plus faibles.

Ainsi, le **paramètre K_v reflète l'effort propulsif potentiellement disponible**. Nous illustrons cette interprétation par quelques cas types. Si la vitesse moyenne du nageur est très grande et la fréquence du battement finie, $K_v \approx 0$. Le nageur ne peut alors pas développer d'effort propulsif. Au mieux, il peut minimiser sa traînée, c'est-à-dire ne pas bouger. A l'opposé, si $U_\infty \approx 0$, le nageur peut développer un effort propulsif important. Considérons maintenant le cas d'une vitesse d'avance U_∞ constante. Un nageur dont le mouvement de palme est propulsif, augmente sa propulsion avec sa fréquence de battements. Enfin, K_v croît avec la longueur hydrodynamique L_c . On remarque que les poissons les plus rapides ont un allongement important et que les nageurs sont plus efficaces les bras tendus devant. Notamment, si on assimile le nageur à une plaque plane rigide en rotation, l'effort propulsif est d'autant plus important que le centre de rotation est avancé. Ceci est conforme à l'observation selon laquelle les nageurs avec palmes les plus performants ont une ondulation dont le centre de rotation est très avancé (au niveau des épaules). Ceci suggère aussi qu'une palme est un appendice artificiel qui permet au nageur d'augmenter artificiellement sa longueur hydrodynamique.

L'effort propulsif potentiellement disponible K_v et le nombre de Strouhal S_t (défini en Section 6.1) n'offrent cependant pas une caractérisation de la cinématique d'un nageur suffisamment fine pour l'optimisation. Une modélisation plus complète du système est proposée.

7.2.2 Normalisation

Au niveau du calcul fluide, on utilise les grandeurs caractéristiques pour normaliser les variables, ce qui évite les difficultés numériques liées aux différences d'ordres de grandeurs. Les

longueurs du système palme-nageur sont divisées par L_c . Dans ce travail, L_c est choisie comme étant la longueur de l'obstacle, si bien que la longueur d'obstacle normalisée est l'unité. Les temps sont divisés par T_c . La vitesse U_∞ est divisée par U_c . La fréquence de nage est divisée par la fréquence caractéristique, elle même en l'occurrence, donc la fréquence de nage normalisée est unitaire. Enfin, les forces sont divisées par $(\rho U_c^2 L_c)$ et sont exprimées par unités de largeur. Les relations de normalisation sont récapitulées dans la Table 7.1.

temps, t	$\underline{t} = t/T_c = tf$
longueur, l	$\underline{l} = l/L_c$
vitesse, u	$\underline{u} = u/U_c = u/(L_c f)$
force par unité de largeur, F	$\underline{F} = F/(\rho U_c^2 L_c)$
puissance par unité de largeur, p	$\underline{p} = p/(\rho U_c^2 L_c^2 f)$

TAB. 7.1 – Relations de normalisation. Les valeurs normalisées sont soulignées.

7.3 Cinématiques de la palme et du nageur

Le potentiel propulsif que représente K_v est réalisé ou non par le système nageur-palme en fonction de sa cinématique dans l'écoulement.

Ainsi, l'augmentation de longueur hydrodynamique n'est utile que si elle est perçue par l'écoulement : une palme qui ne possède aucune raideur de flexion et une inertie négligeable s'adapte à chaque instant à l'écoulement de manière à équilibrer les pressions exercées par le fluide de part et d'autre de sa surface. La palme est alors totalement passive. Elle suit les lignes de courant qui ne sont pas perturbées par sa présence. C'est pourquoi il est nécessaire que la raideur de la palme soit suffisante pour perturber l'écoulement et en extraire par réaction un effort propulsif.

Initialement la palme seule a été modélisée, mais après observations et tests numériques, on a constaté que la modélisation du nageur est importante, car il perturbe sensiblement l'écoulement. Ainsi, notre système est constitué du nageur et de la monopalme.

7.3.1 Modèle de nageur

Le nageur est représenté par 4 segments : les bras (étendus), le torse, les cuisses et les tibias. La monopalme est composée de n segments articulés par des ressorts en flexion (cf. Section 7.3.2).

Le premier segment de la palme représente aussi les pieds du nageur. La palme et le nageur sont schématisés en Figure 7.1. Contrairement à la palme (cf. Section 7.3.2), le mouvement du nageur

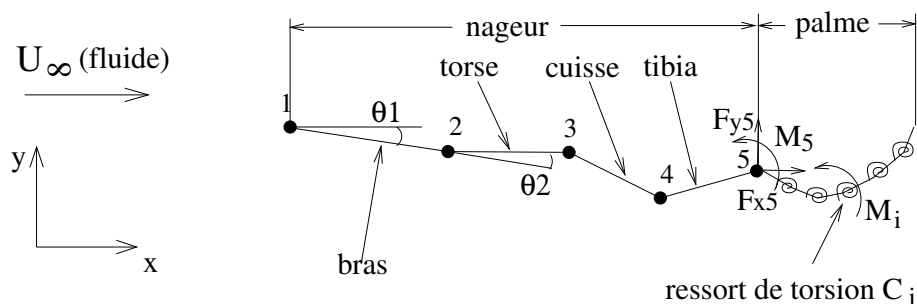


FIG. 7.1 – Schéma du nageur et de la palme.

est forcé est supposé suivre les équations suivantes :

$$y_1(t) = Y_1^c + Y_1 \sin(2\pi ft) , \quad (7.4)$$

$$\theta_i(t) = \Theta_i^c + \Theta_i \sin(2\pi ft - \phi_i) , \quad i = 1, 5 \quad (7.5)$$

où y_1 est le déplacement vertical de la main, θ_1 l'angle entre l'horizontale et le bras, ϕ_1 est un retard entre le mouvement vertical de la main et la rotation du bras, et θ_i et ϕ_i , $i = 2, 5$ sont l'angle et le retard entre les segments $(i - 1)$ et i , respectivement.

Les paramètres des équations (les amplitudes Y_1 et Θ_i , les valeurs moyennes Θ_i^c , les angles de phase ϕ_i et la fréquence f) sont identifiés à partir de données mesurées sur un nageur de sprint dont les mensurations sont : longueur des bras 0.5810 m, torse 0.5040 m, cuisses 0.42875 m, et tibia 0.4805 m. La longueur des bras, qui en fait est 1 m, est réduite pour prendre en compte l'espacement entre les bras qui laisse passer le fluide.

Les données ont été mesurées par B. Bideau, B. Colobert et G. Nicolas au Laboratoire de Physiologie et de Biomécanique de l'Université de Rennes 2. Les mouvements dans un plan vertical de certains points du nageur (doigts, coude, épaule, hanche, genou, cheville et orteil) sont enregistrés par une caméra à haute fréquence (100 Hz) et reconstruits en 3D par un algorithme de "direct linear transformation". La Figure 7.2 montre les déplacements verticaux obtenus.

Conformément à la Figure 7.1, nous avons négligé l'articulation du coude. Cette simplification est raisonnable car l'amplitude du mouvement relatif du coude est faible, par rapport à celles des autres points. Signalons par ailleurs que le point 5 de la Figure 7.1 est le talon du nageur

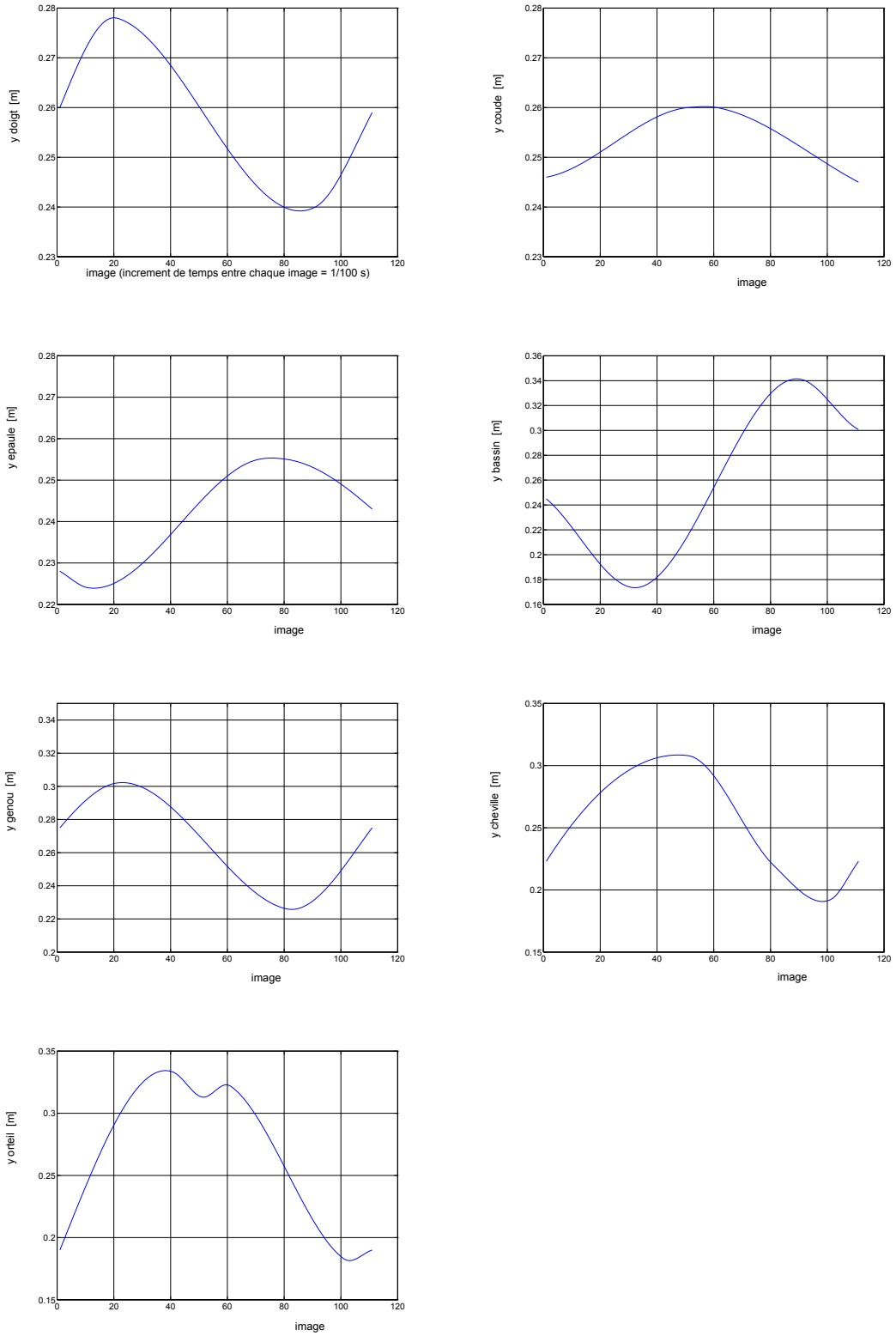


FIG. 7.2 – Déplacements verticaux en mètres, mesurés sur un nageur de sprint à 100 images / seconde.

alors que la position de la cheville est enregistrée, ce qui a nécessité une correction au moment de l'identification. On remarque sur la Figure 7.2 que les déplacements mesurés ressemblent à des fonctions sinusoidales de même pulsation, ce qui correspond aux hypothèses adoptées pour décrire le mouvement du nageur (Equations (7.4) et (7.5)). Du fait de la bonne adéquation entre la forme fonctionnelle de la cinématique du nageur et les enregistrements, une stratégie d'identification simple a été utilisée :

1. la fréquence est lue directement sur les courbes,
2. (Y_1^c, Θ_1^c) et (Y_1, Θ_1) sont calculés comme les moyennes et les demi-amplitudes des signaux enregistrés,
3. les déphasages ϕ_i sont les déphasages des minima des signaux enregistrés.

Les paramètres obtenus par cette identification sont :

Vitesse du nageur $U_\infty = 3.0 \text{ m/s}$

fréquence de nage $f = 1.0 \text{ Hz}$

$$\begin{aligned}
 Y_1 &= 0.07 \text{ m} & \Theta_1 &= 3.39^\circ & \Theta_2 &= 12.0^\circ & \Theta_3 &= 20.0^\circ & \Theta_4 &= 14.0^\circ & \Theta_5 &= 16.0^\circ \\
 Y_1^c &= 0.0 \text{ m} & \Theta_1^c &= -1.56^\circ & \Theta_2^c &= 1.56^\circ & \Theta_3^c &= -10.0^\circ & \Theta_4^c &= 14.0^\circ & \Theta_5^c &= -20.0^\circ \\
 & & \phi_1 &= -222.55^\circ & \phi_2 &= -152.8^\circ & \phi_3 &= 17.2^\circ & \phi_4 &= 17.2^\circ & \phi_5 &= 107.2^\circ
 \end{aligned}$$

7.3.2 Modèle de palme et couplage fluide-structure

La palme est constituée d'un assemblage de n barres rigides de masse linéique ρ_L , articulées par des raideurs de flexion C_i , comme le montre la Figure 7.3 ¹.

¹Dans cette Section, pour faciliter l'écriture des équations, le point 1 représente le premier point de la palme, et, pourtant, θ_1 est l'angle entre l'horizontal et la première barre de la palme (ne pas confondre avec le θ_1 utilisé dans la schématisation du nageur, qui représente le mouvement angulaire du bras). Dans le cas d'un nageur couplé à la palme, θ_1 de la palme sera la somme des θ_i , $i = 1, 5$, des points qui composent le nageur.

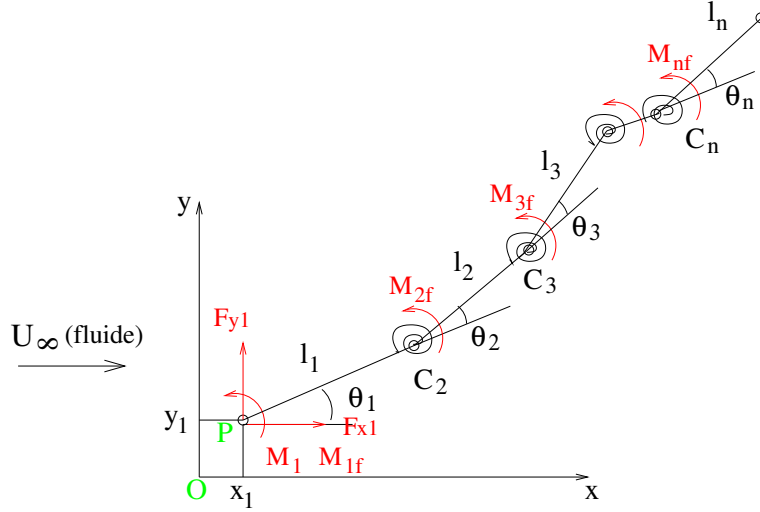


FIG. 7.3 – Représentation du modèle de palme à n pièces liées par des ressorts en flexion C_i .

Les équations du mouvement, calculées en Annexe F, sont ²

$$\rho_L \left(\sum_{i=1}^n l_i \right) \ddot{x}_1 = F_{x1f} + F_{x1} \quad (7.6)$$

$$\rho_L \left(\sum_{i=1}^n l_i \right) \ddot{y}_1 = F_{y1f} + F_{y1} \quad (7.7)$$

$$\frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}_1} - \frac{\partial E_c}{\partial \theta_1} = M_{1f} + M_1 \quad (7.8)$$

$$\frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}_i} - \frac{\partial E_c}{\partial \theta_i} + \frac{\partial E_p}{\partial \theta_i} = M_{if} \quad i = 2, n. \quad (7.9)$$

Dans les équations du mouvement, $x_1(t)$, $\dot{x}_1(t)$, $\ddot{x}_1(t)$, $y_1(t)$, $\dot{y}_1(t)$, $\ddot{y}_1(t)$, $\theta_1(t)$, $\dot{\theta}_1(t)$ et $\ddot{\theta}_1(t)$ sont connus (mouvement du nageur). Les efforts fluides F_{x1f} , F_{y1f} et M_{if} sont des fonctions de $x_1(t)$, $\dot{x}_1(t)$, $y_1(t)$, $\dot{y}_1(t)$, $\theta_i(t)$, et $\dot{\theta}_i(t)$, $i = 1, n$, qui définissent les positions et vitesses des points de la palme. Les inconnues des équations du mouvement sont $\theta_i(t)$, $\dot{\theta}_i(t)$, $\ddot{\theta}_i(t)$, $i = 2, n$, $F_{x1}(t)$, $F_{y1}(t)$ et $M_1(t)$.

En conséquence, à chaque pas de temps, on résout les $(n-1)$ Equations (7.9), puis on obtient les efforts F_{x1} , F_{y1} et M_1 , à partir des Equations (7.6), (7.7) et (7.8), respectivement. Pour résoudre ces équations non-linéaires, on utilise le schéma temporel implicite de Newmark couplé à la méthode itérative de Newton-Raphson et à la méthode de minimisation GBNM (cf. Annexe G). GBNM est utilisée quand les corrections de Newton-Raphson sont infructueuses et a ici pour but

²Les Equations (7.8) et (7.9) sont simplifiées, en n'explicitant pas les énergies cinétique E_c et potentielle E_p , dont les expressions sont données en Annexe F.

de minimiser le résidu (voir Figure G.1).

Les Figures 7.4, 7.5 et 7.6 présentent le comportement d'une palme à 6 barres/5 ressorts, avec nageur de sprint, distribution constante de raideurs $C_i = 1000 \text{ Nm/rad}$, $i = 2, 6$, et masse linéique $\rho_L = 2 \text{ kg/m}$. Cette valeur de ρ_L est utilisée en toutes les simulations effectuées ici.

Les efforts $F_{x_1}(t)$, $F_{y_1}(t)$ et $M_1(t)$ provoqués par le pied du nageur sur la palme sont montrés en Figure 7.4. $F_{x_1}(t)$ est interprétée comme le contraire de la force motrice instantanée provoquée par la palme. On aperçoit que la fréquence de F_{x_1} est le double de la fréquence de F_{y_1} , ce qui est cohérent car la fréquence de x_1 est aussi le double de la fréquence de y_1 . En d'autres termes, à chaque montée ou descente de la palme correspond à la fois une période propulsive et une période de freinage. Le mouvement horizontal et vertical du pied du nageur, x_1 et y_1 , sont présentés en Figure 7.5. La Figure 7.6 illustre $\theta_1(t)$ et $\theta_2(t)$, θ_1 étant l'angle imposé par le pied du nageur et θ_2 l'angle relatif entre les deux premières barres de la palme. On remarque que le mouvement imposé n'est pas symétrique et le nageur privilégie la phase descendante, conformément à l'asymétrie du corps humain ([28]). Une visualisation de l'écoulement, du nageur et de la palme est donnée en Figure 7.7.

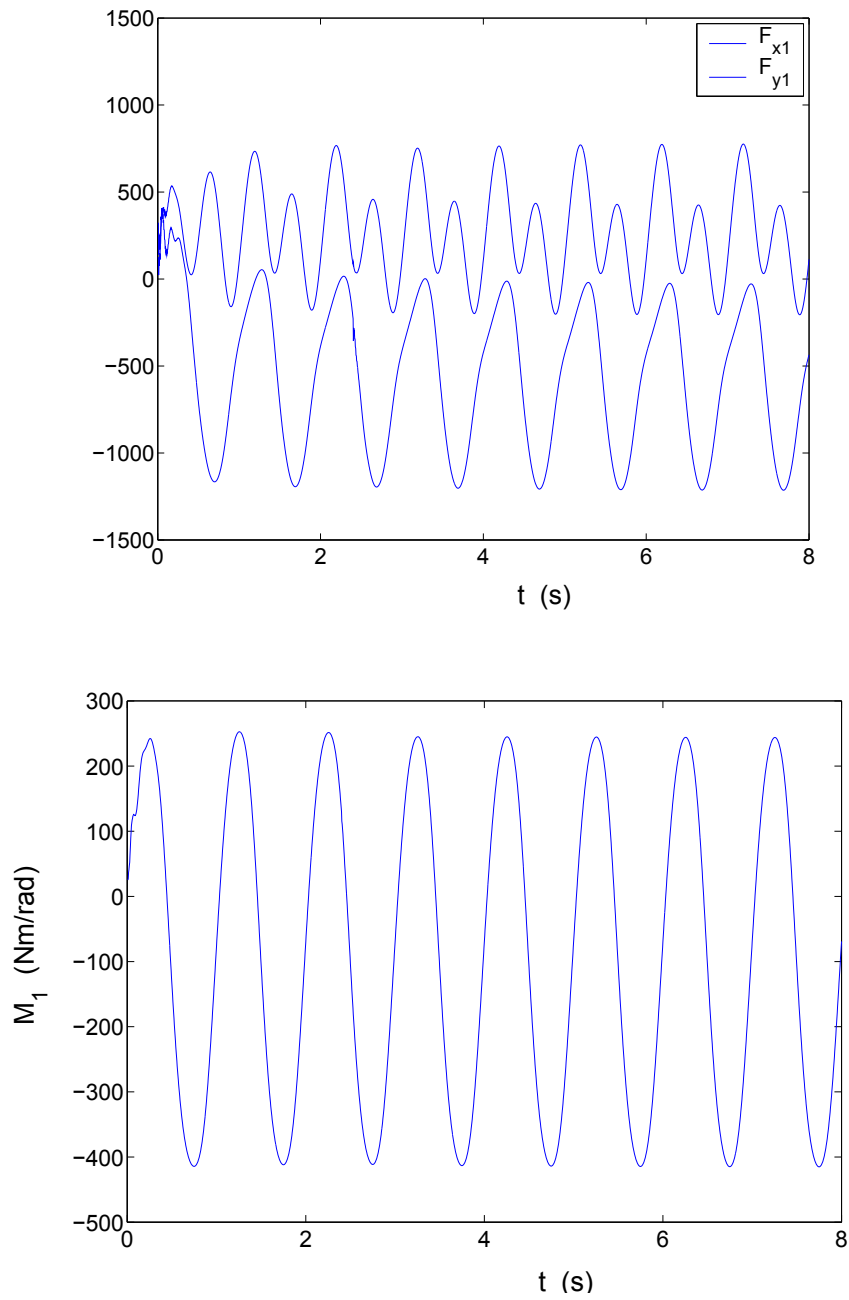


FIG. 7.4 – Efforts du pied du nageur sur la palme pour $C_i = 1000 \text{ Nm/rad}$, $i = 2, 6$. En haut les forces (en Newtons), en bas le moment.

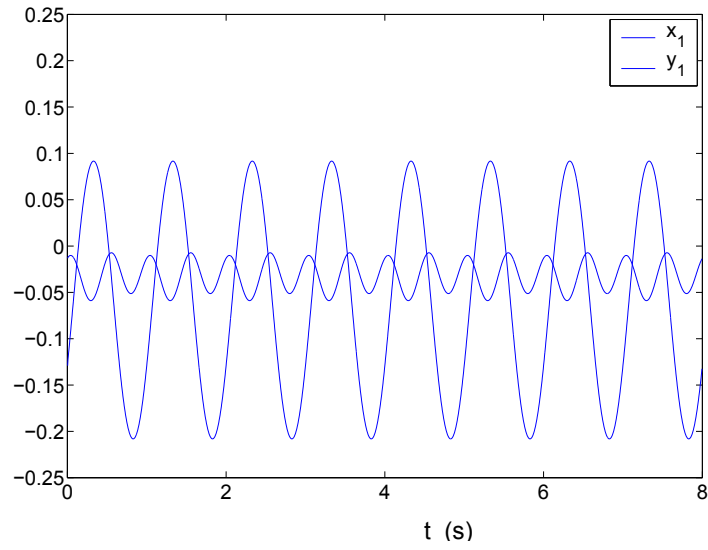


FIG. 7.5 – Mouvement du pied du nageur, en mètres, pour $C_i = 1000 \text{ Nm/rad}$, $i = 2, 6$.

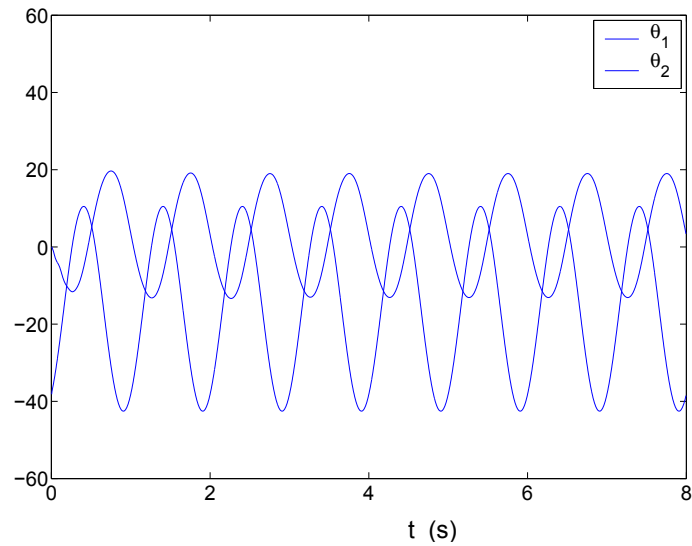


FIG. 7.6 – Mouvement angulaire, en degrés, des deux premières barres de la palme, pour $C_i = 1000 \text{ Nm/rad}$, $i = 2, 6$.

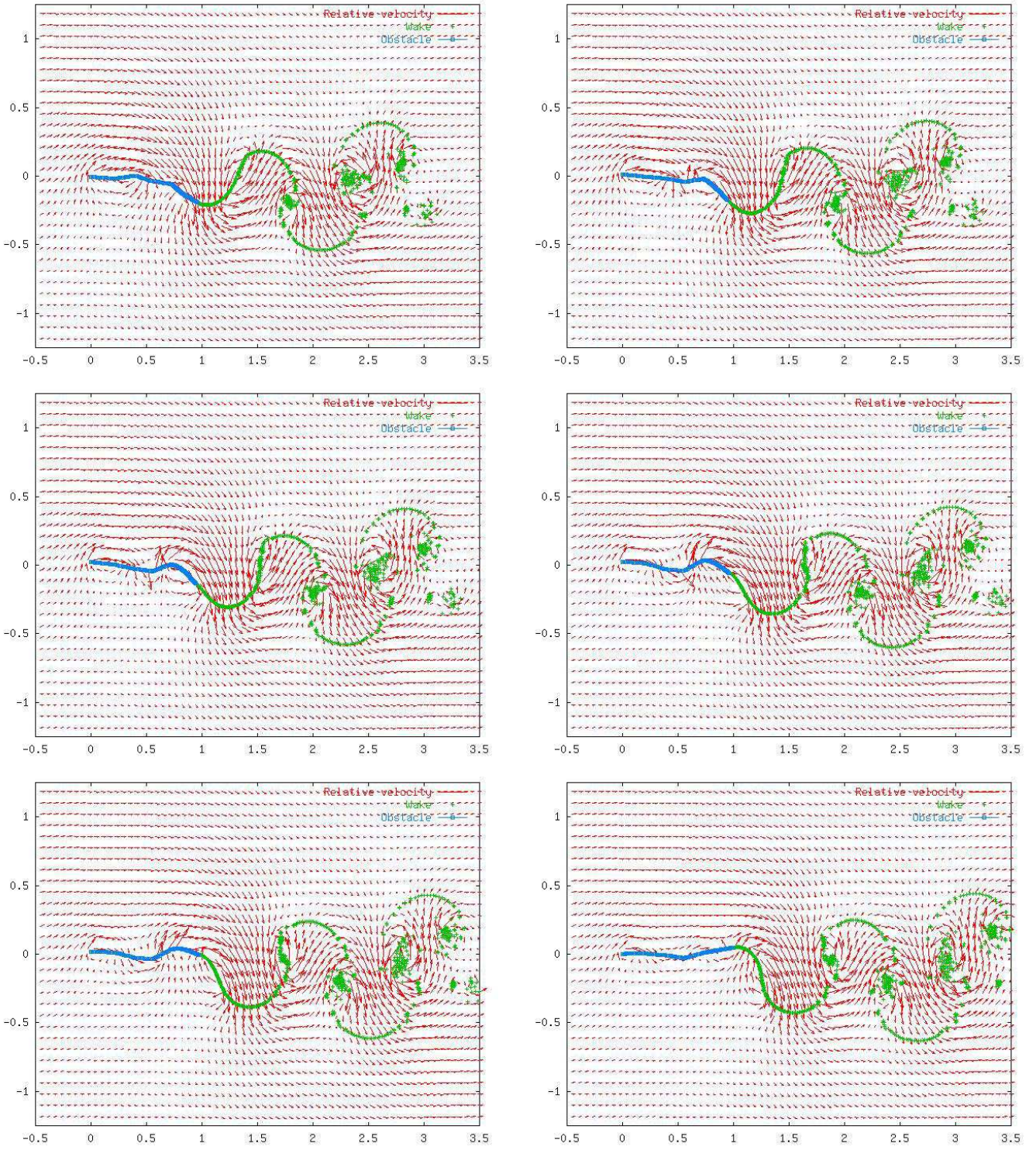


FIG. 7.7 – Cinématique du nageur et de la palme. Trait continu, en bleu, l'obstacle (la palme et le nageur); \times , en vert, les particules; \rightarrow , en rouge, vitesse relative du fluide (en valeurs normalisées).

Chapitre 8

Optimisation

8.1 Critères d'optimisation

Le système considéré est composé de trois entités, le nageur, la palme, et le fluide. Les échanges énergétiques dans le système sont schématisés sur la Figure 8.1, en négligeant les échanges directs entre fluide et nageur ainsi que tous les phénomènes liés à la viscosité du fluide. L'énergie potentielle élastique de la palme étant un stockage temporaire et l'énergie cinétique de la palme négligeable, l'énergie musculaire du nageur se transforme en énergie cinétique du fluide et en énergie cinétique du nageur par l'intermédiaire de la palme. L'objectif final de l'optimisation est, pour une dépense d'énergie musculaire donnée dans un intervalle de temps, de maximiser l'accroissement d'énergie cinétique (suivant x) du nageur en modifiant la palme.

Il est cependant nécessaire de formuler l'optimisation de la palme au moyen d'autres critères

- pour obtenir une solution réaliste (on introduira par exemple des limitations sur la puissance fournie par le nageur),
- pour respecter le modèle fluide qui est défini à vitesse de nage U_∞ (i.e., énergie cinétique de

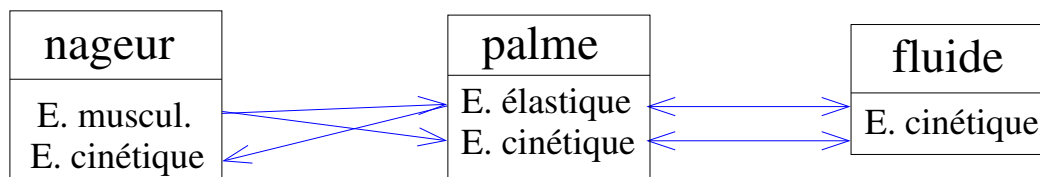


FIG. 8.1 – Bilan des échanges énergétiques en négligeant les échanges directs entre fluide et nageur ainsi que tous les phénomènes dissipatifs visqueux.

nageur) constante,

– et pour permettre différents niveaux de simplification.

Des critères d'optimisation en forces et en énergies sont ainsi définis.

Comme la majorité de la poussée est produite par la monopalme, les critères sont formulés à partir de quantités calculées sur la monopalme. La force propulsive moyenne exercée par le fluide sur la palme est

$$\bar{F}_x = \frac{1}{T_f - T_s} \int_{T_s}^{T_f} \int_0^{L_s} f_x(s) ds dt, \quad (8.1)$$

où L_s est la longueur de la palme et $f_x(s)$ la force linéique suivant x exercée par le fluide sur la palme en s . On note que \bar{F}_x négatif correspond à un effort propulsif vers l'avant du nageur du fait du sens des x (cf. Figures 7.1 et 7.3). Les valeurs par défaut de l'intervalle de temps considéré pour la mesure sont $T_s = 4 s$, ce qui correspond à un temps de mise en place de la simulation (émission des premiers tourbillons) et $T_f = 8 s$ (stabilisation des valeurs moyennes, comme on le verra plus tard) soit un intervalle d'une durée de 4 s. Contrairement à \bar{F}_x , une mesure moyenne des efforts verticaux exercés par le fluide sur la palme, \bar{F}_y , n'est pas utile dans notre cas, car il caractérise la symétrie du mouvement. Si le mouvement du nageur est parfaitement symétrique par rapport à la verticale, \bar{F}_y est nul.

On associe à \bar{F}_x la puissance propulsive moyenne

$$\bar{P}_{fx} = \frac{1}{T_f - T_s} \int_{T_s}^{T_f} \int_0^{L_s} f_x(s) U_\infty ds dt. \quad (8.2)$$

La puissance moyenne transmise par le fluide à la palme est

$$\bar{P}_f = \frac{1}{T_f - T_s} \int_{T_s}^{T_f} \int_0^{L_s} (f_x(s)(\dot{x}(s) + U_\infty) + f_y(s)\dot{y}(s)) ds dt, \quad (8.3)$$

où $\dot{x}(s)$ et $\dot{y}(s)$ sont les deux composantes de la vitesse de la palme en s dans un repère fixe par rapport au fluide à l'infini, et $f_y(s)$ est la force linéique suivant y exercée par le fluide sur la palme en s .

Les hypothèses de la simulation de l'écoulement imposent de travailler à une vitesse de nage U_∞ constante. La fonction objectif sera typiquement la minimisation de \bar{F}_x ou \bar{P}_{fx} (on rappelle que \bar{F}_x et \bar{P}_{fx} sont négatifs pour un nageur qui avance et accélère). Ces fonctions objectifs s'interprètent comme le potentiel qu'a le système nageur-palme à accélérer. En Figures 8.2 et 8.3 on représente l'évolution de l'estimation de \bar{F}_x , \bar{F}_y , \bar{P}_{fx} et \bar{P}_f pour T_f allant de 4 à 12 s pour

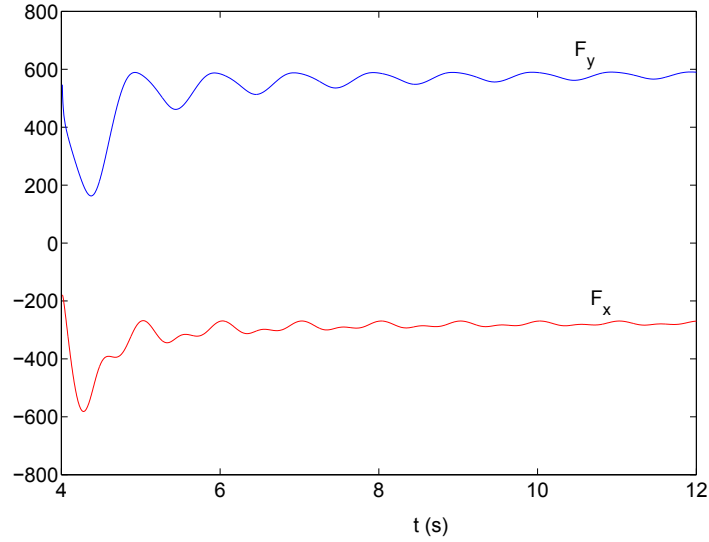


FIG. 8.2 – Evolution des forces moyennes \bar{F}_x et \bar{F}_y . Les valeurs des forces sont données en Newtons.

la cinématique du nageur présentée en Section 7.3.1, modèle de palme à 5 ressorts, longueur de palme $L_s = 0.72 \text{ m}$, et $C_i = 1000 \text{ Nm/rad}$, $i = 2, 6$. On remarque la convergence à des valeurs stabilisées après environ 8 s de simulation.

Dans ce travail, on a choisi la puissance \bar{P}_{fx} comme fonction objectif. La minimisation est réalisée pour une puissance moyenne de nageur limitée, ce qui se traduit comme une borne inférieure sur \bar{P}_f , c'est-à-dire une valeur maximale de l'énergie transmise par la palme au fluide. Les variables d'optimisation sont les raideurs C_i .

L'optimisation est finalement formulée comme,

$$\left\{ \begin{array}{l} \min_{C_i} \bar{P}_{fx} , \\ \text{tel que,} \\ \bar{P}^{\min} \leq \bar{P}_f , \\ C_i^{\min} \leq C_i \leq C_i^{\max} , \quad i = 2, n . \end{array} \right. \quad (8.4)$$

8.2 Etudes paramétriques

Dans un premier temps, afin d'avoir une compréhension intuitive du système, des études paramétriques sont réalisées.

Comme première étude, nous faisons varier la longueur L_s de la palme, pour des raideurs

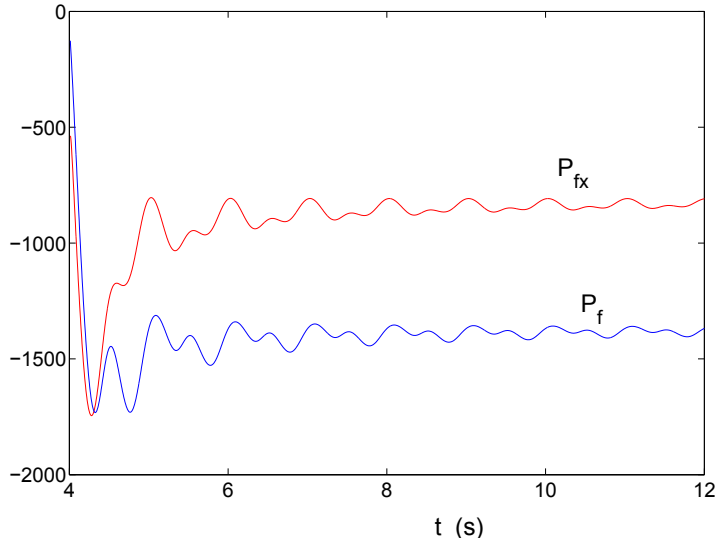


FIG. 8.3 – Evolution des puissances moyennes propulsive \bar{P}_{fx} et totale \bar{P}_f (en *Watts*).

constantes $C_i = 1000 \text{ Nm/rad}$, et nous regardons l'effet sur \bar{P}_{fx} , \bar{P}_f et le rendement en puissance $\nu_P = \bar{P}_{fx}/\bar{P}_f$. Les résultats de cette étude sont présentés en Table 8.1. On observe que l'augmentation de la longueur L_s accroît toujours \bar{P}_{fx} et ν_P . C'est donc pour augmenter la puissance propulsive et le rendement que les nageurs de sprint de haut niveau utilisent plutôt des palmes longues. Une conclusion similaire par rapport à l'augmentation de l'effort propulsif potentiellement disponible K_v avec l'augmentation de la longueur hydrodynamique a été déjà présentée en Section 7.2. En revanche, la longueur est contrainte par la puissance maximale fournie par le nageur ainsi que par la manœuvrabilité, et ne peut pas être augmentée de manière illimitée. Ainsi, par la suite dans cette Section, une longueur de palme $L_s = 0.72 \text{ m}$ est adoptée. Cette longueur correspond à celle de la palme du nageur sur lequel la cinématique a été identifiée (cf. Section 7.3.1)

$L_s \text{ (m)}$	0.5	0.72	1.0
$\bar{P}_{fx} \text{ (W)}$	-527.62	-809.67	-1147.28
$\bar{P}_f \text{ (W)}$	-1153.30	-1369.88	-1747.60
ν_P	0.457	0.591	0.656

TAB. 8.1 – Effets de la variation de longueur de palme L_s sur les puissances.

Dans les études suivantes l'effet des raideurs C_i sur \bar{P}_{fx} , \bar{P}_f et ν_P est étudié. Les résultats sont rapportés en Tables 8.2 et 8.3. Ces tables montrent aussi le nombre de Strouhal S_t associé à chaque conception. La vitesse du nageur et la fréquence des battements étant constantes, la

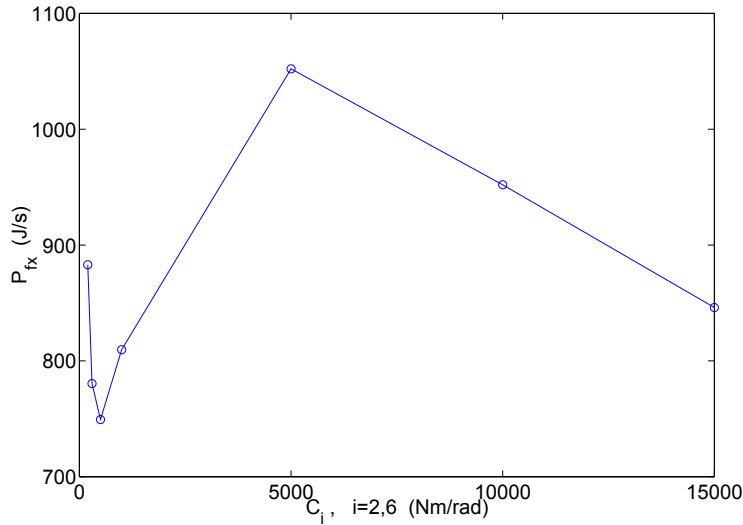


FIG. 8.4 – Puissance propulsive $|\bar{P}_{fx}|$ en fonction de la raideur de la palme.

variation de S_t (Equation (6.1)) est la conséquence de la variation de l'amplitude du mouvement du bout de la palme.

En Table 8.2 les raideurs C_i sont constantes tout au long de la palme et à chaque nouvelle simulation on les modifie toutes à la fois. En Table 8.3 on prend un cas de référence, $C_i = 1000 \text{ Nm/rad}$, $i = 2, 6$, et on ne fait varier qu'une raideur à chaque nouvelle simulation, en remplaçant une des raideurs à 1000 Nm/rad par une autre à 2000 Nm/rad .

Afin de mieux les visualiser, les résultats de la Table 8.2, en valeurs absolues, sont également présentés graphiquement en Figures 8.4, 8.5 et 8.6. Les variations de la puissance propulsive \bar{P}_{fx} , de la puissance totale \bar{P}_f et du rendement ν_P , respectivement, sont montrées en fonction de la raideur de la palme.

Le comportement des courbes de \bar{P}_f et \bar{P}_{fx} nous incite à penser que pour $C_i < 300 \text{ Nm/rad}$ le modèle ne reproduit plus la réalité car une palme très souple ($C_i = 200 \text{ Nm/rad}$) dépense plus de puissance qu'une palme plus raide ($C_i = 300 \text{ Nm/rad}$ ou même $C_i = 500 \text{ Nm/rad}$).

On observe que, parmi les paramètres testés et valables, la puissance propulsive maximale est obtenue pour $C_i = 5000 \text{ Nm/rad}$, et le rendement optimal pour $C_i = 300 \text{ Nm/rad}$. Les tendances suivantes ressortent de la Table 8.2 ou des Figures 8.4, 8.5 et 8.6 :

- Augmenter les raideurs C_i accroît toujours la puissance totale \bar{P}_f transmise au fluide.
- La puissance motrice \bar{P}_{fx} présente une valeur maximale pour $C_i = 5000 \text{ Nm/rad}$. Ceci

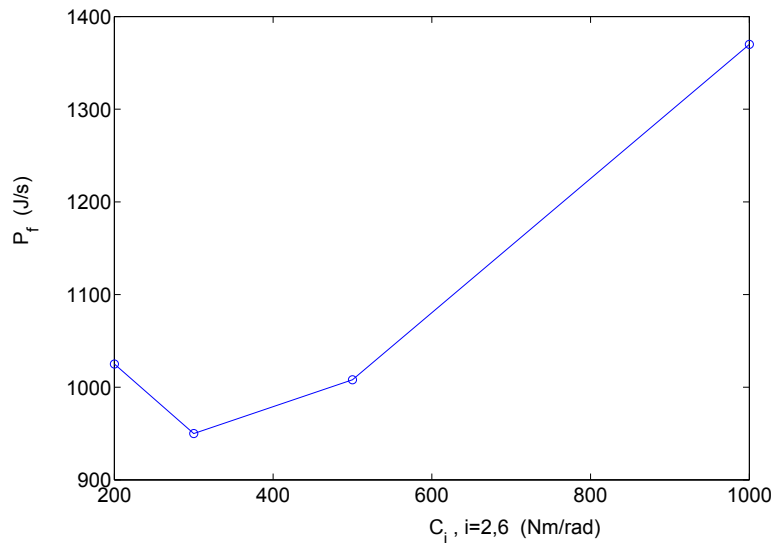
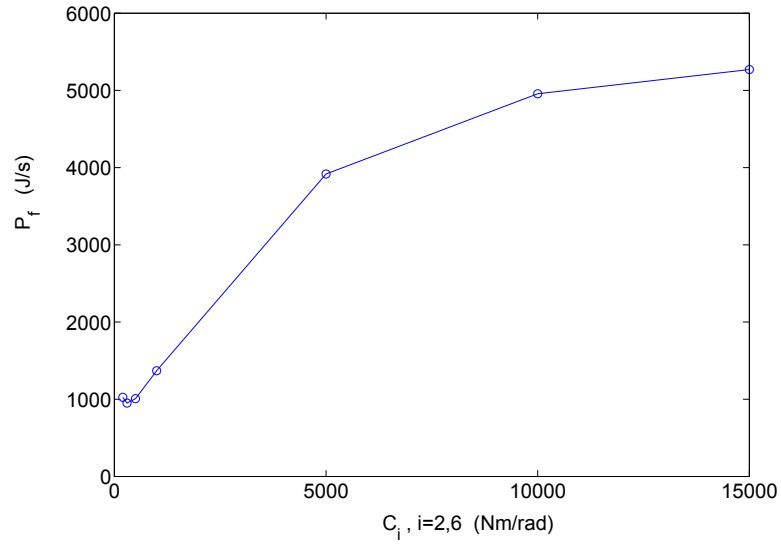


FIG. 8.5 – Puissance totale $|\bar{P}_f|$ en fonction de la raideur de la palme. En bas un agrandissement de la courbe dans l'intervalle $C_i = 200$ à $C_i = 1000 \text{ Nm/rad}$.

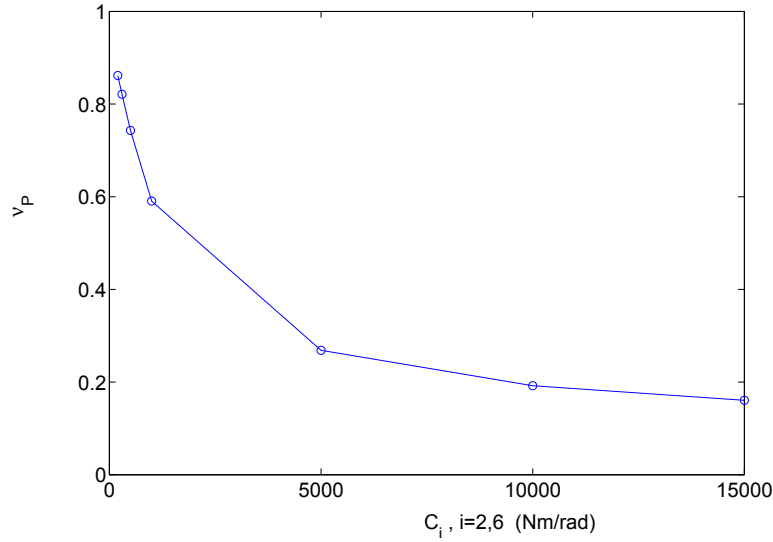


FIG. 8.6 – Rendement ν_P en fonction de la raideur de la palme.

confirme que, même sans limite sur \bar{P}_f , la palme rigide (les C_i grands) n'est pas optimale.

- Le rendement augmente avec la diminution de la raideur. Ce comportement a été aussi constaté dans [38], où un aileron bidimensionnel flexible soumis à des grands déplacements a été étudié.

La Table 8.3 montre que :

- Augmenter les raideurs C_i proches du pied du nageur accroît la puissance totale \bar{P}_f et la puissance propulsive \bar{P}_{fx} , avec une diminution du rendement. Le système est très sensible à ces raideurs.
- Augmenter la raideur centrale diminue la puissance propulsive \bar{P}_{fx} et augmente la puissance totale \bar{P}_f . L'augmentation relative de \bar{P}_f est inférieure à la diminution relative de \bar{P}_{fx} .
- l'augmentation des raideurs au bout de la palme provoque une très faible diminution de \bar{P}_{fx} et de \bar{P}_f , le rendement restant inchangé. Le système est peu sensible à la raideur au bout de la palme.

$C_i, i = 2, 6$	200	300	500	1000	5000	10000	15000
\bar{P}_{fx} (W)	-883.00	-780.38	-749.39	-809.67	-1052.15	-952.02	-846.08
\bar{P}_f (W)	-1025.23	-950.34	-1008.32	-1369.88	-3917.57	-4955.79	-5270.33
ν_P	0.861	0.821	0.743	0.591	0.269	0.192	0.161
S_t	0.206	0.197	0.201	0.222	0.295	0.308	0.309

TAB. 8.2 – Effets de la variation de toutes les valeurs des raideurs sur les puissances, sur le rendement et sur le nombre de Strouhal.

C_2	1000	2000	1000	1000	1000	1000
C_3	1000	1000	2000	1000	1000	1000
C_4	1000	1000	1000	2000	1000	1000
C_5	1000	1000	1000	1000	2000	1000
C_6	1000	1000	1000	1000	1000	2000
$P_{fx} (W)$	-809.67	-927.97	-832.36	-789.65	-790.545	-804.12
$\bar{P}_f (W)$	-1369.88	-1748.94	-1512.46	-1382.90	-1354.35	-1362.15
ν_P	0.591	0.531	0.550	0.571	0.584	0.590
S_t	0.222	0.243	0.229	0.222	0.220	0.222

TAB. 8.3 – Effets du changement de la valeur d’une des raideurs sur les puissances, sur le rendement et sur le nombre de Strouhal, à partir d’un cas de référence (première colonne de gauche).

8.3 Résultats de l’optimisation et discussion

Le problème du choix optimal des raideurs C_i est maintenant traité dans sa forme (8.4) au moyen de l’algorithme GBNM. On choisit comme bornes des raideurs $C_i^{min} = 300 \text{ Nm/rad}$ et $C_i^{max} = 15000 \text{ Nm/rad}$, $i = 2, 6$. Le nombre maximal d’évaluations est fixé à 200. Chaque évaluation correspond à 8 s de simulation d’écoulement et prend de l’ordre de 17 min sur un PC standard. Ainsi, les 200 itérations de l’optimisation durent 56 h environ.

Le problème d’optimisation est résolu pour les puissances de nageur $\bar{P}^{min} = -1400 \text{ W}$, -2000 W et -3000 W . A titre de comparaison, la puissance mesurée dans [18] sur des distances moyennes parcourues par des nageurs est d’environ 1400 W. Les limites supérieures pour le deuxième et troisième cas prennent en considération la distance parcourue par le nageur, qui est plus petite, et la modélisation bidimensionnelle, qui surestime les efforts (le fluide ne peut pas passer par les cotés et il n’y a pas de séparation entre le fluide et l’obstacle).

Le problème d’optimisation présente les caractéristiques suivants : peu de variables (5), problème sous contraintes, le gradient n’est pas disponible et doit être calculé par une méthode numérique (différences finies) relativement coûteuse et imprécise. C’est pourquoi GBNM a été choisi comme algorithme d’optimisation. Ses paramètres sont (cf. Annexe B) :

```
length    1000.0
reflection 1.0
contraction 0.5
expansion 2.0
max_nb_iteration 200
size_stop 0.0001
```

```

nb_random_points 10
coef_size_gauss 0.01
lag_mult_step 0.0

```

Le coefficient de pénalisation des contraintes est égal à 200 et le point initial est $C_i = 1000$, $i = 2, 6$.

La Table 8.4 donne les meilleurs points obtenus pour les trois conditions de puissance limite.

C_i (Nm/rad)	\bar{P}^{min} (W)		
	-1400	-2000	-3000
C_2^*	1598.26	5707.80	9707.63
C_3^*	557.44	4060.70	5859.33
C_4^*	300.00	342.74	1869.28
C_5^*	300.00	300.00	300.00
C_6^*	300.00	300.00	300.00

TAB. 8.4 – Les raideurs optimales pour les trois conditions de puissance limite.

La Figure 8.7 illustre les répartitions optimales de raideurs le long de la palme, pour les trois conditions de \bar{P}^{min} . La Table 8.5 présente les puissances, le rendement et le nombre de Strouhal associés à ces répartitions.

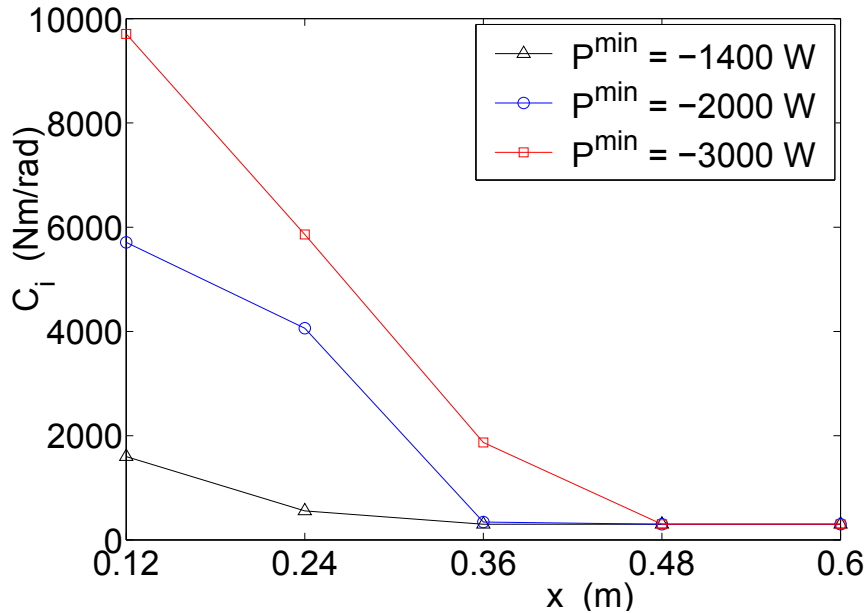


FIG. 8.7 – Répartitions optimales de raideurs.

Ces résultats montrent d'une part que, pour les trois limites de puissance totale simulées, la distribution optimale de rigidité possède des raideurs élevées près des pieds du nageur qui

$\bar{P}^{min} (W)$	-1400	-2000	-3000
$\bar{P}_{fx} (W)$	-973.15	-1129.01	-1208.37
$\bar{P}_f (W)$	-1399.99	-1999.54	-2988.16
ν_P	0.695	0.565	0.404
S_t	0.234	0.266	0.197

TAB. 8.5 – Les puissances, le rendement et le nombre de Strouhal pour les répartitions optimales de raideurs.

décroissent progressivement vers le bord de fuite. On observe d'autre part que plus la dépense énergétique est grande plus la palme est raide. Ces deux résultats sont conformes à l'usage et à l'intuition. On remarque aussi que, pour la condition la plus énergétique, $\bar{P}^{min} = -3000 W$, le gain en puissance motrice est faible par rapport à l'incrément de puissance totale consommée, ce qui va de pair avec un nombre de Strouhal s'éloignant de l'intervalle d'optimalité ($S_t \in [0.25; 0.35]$).

Les efforts $F_{y1}(t)$ et $M_1(t)$ pour la palme optimale, sous la contrainte $\bar{P}^{min} = -2000 W$, sont présentés en Figure 8.8, ainsi que $\theta_1(t)$ et $\theta_2(t)$ en Figure 8.9. On observe que les efforts maximaux sont plus grands que ceux trouvés pour le cas de référence montré en Section 7.3.2. On constate de plus que l'amplitude de $\theta_2(t)$ est faible, ce qui s'explique par la raideur de la palme optimale dans la région proche du pied du nageur. La Figure 8.10 montre le mouvement vertical du pied du nageur et de l'extrémité de la palme.

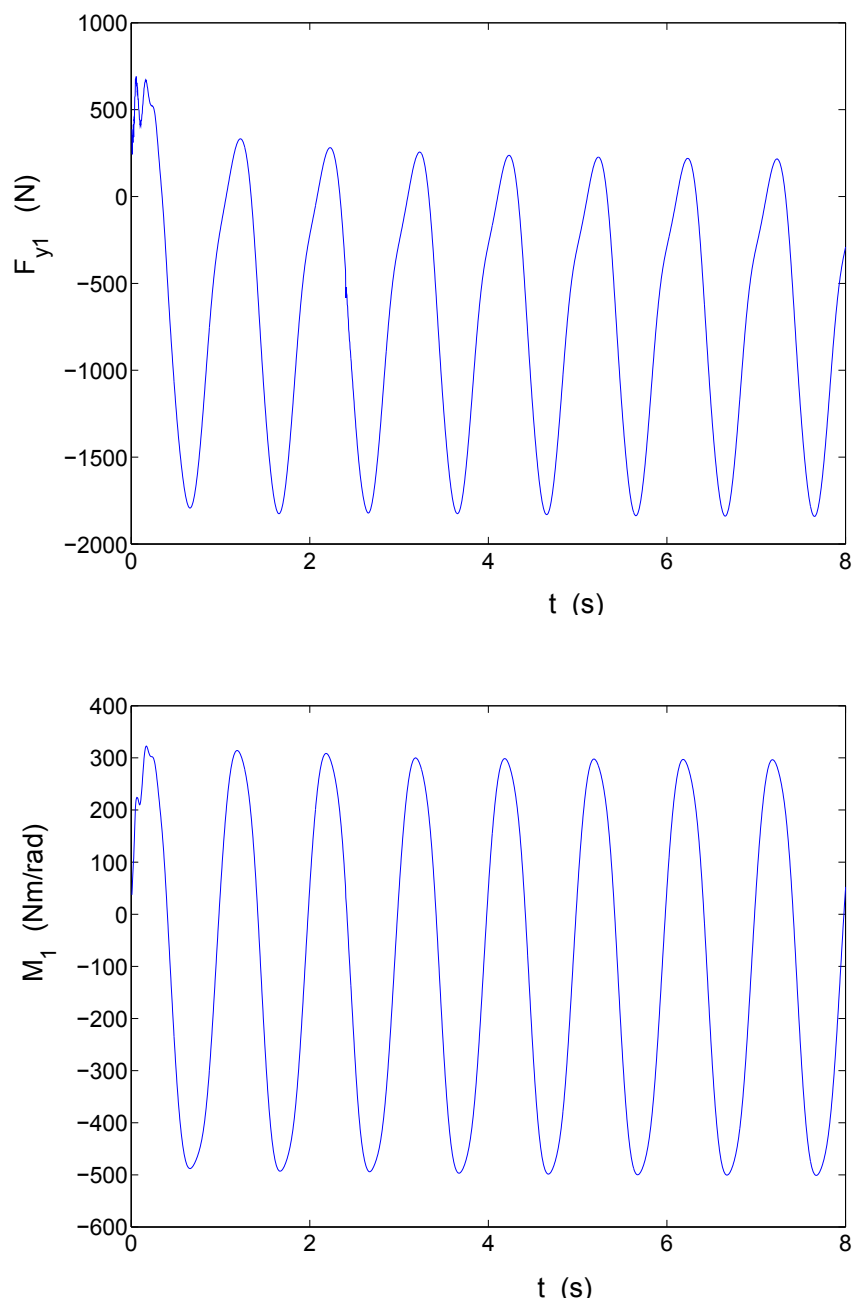


FIG. 8.8 – Efforts du pied du nageur sur la palme pour la palme optimale, $\bar{P}^{min} = -2000 \text{ W}$. En haut la force suivant y , en bas le moment.

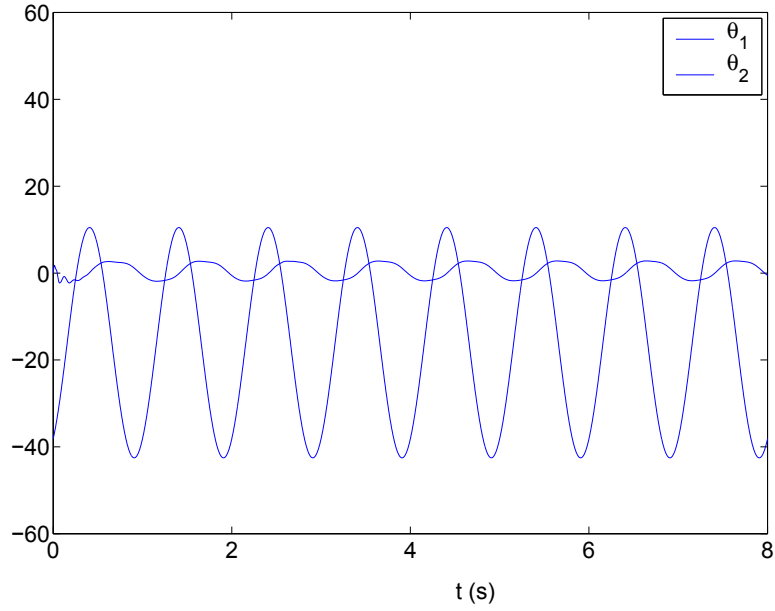


FIG. 8.9 – Mouvement angulaire, en degrés, des deux premières barres de la palme optimale, $\bar{P}^{min} = -2000 W$.

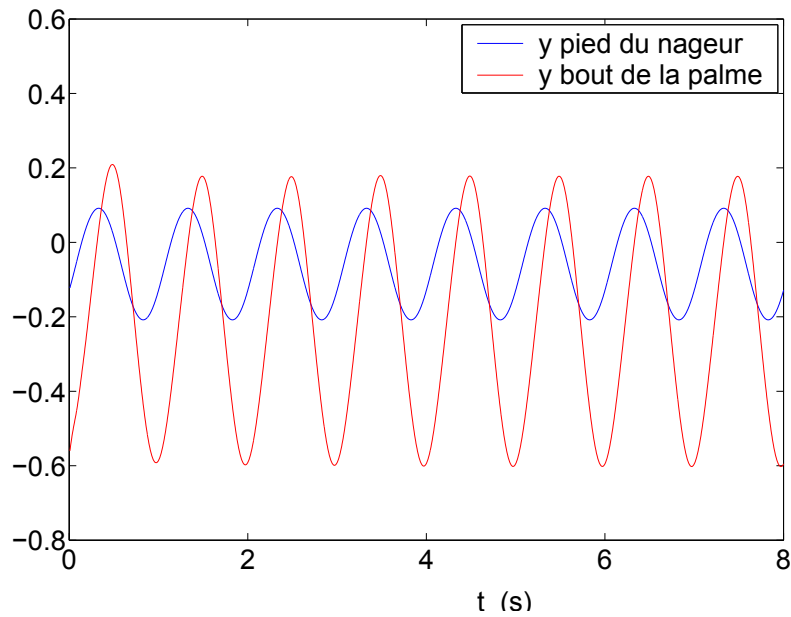


FIG. 8.10 – Mouvement vertical, en mètres, du pied du nageur et de l'extrémité de la monopalme optimale, $\bar{P}^{min} = -2000 W$.

Chapitre 9

Conclusions de la Partie II

Un modèle simplifié de nageur avec monopalme a été proposé. Il permet, grâce à son faible coût numérique, d’optimiser la monopalme en maximisant la puissance propulsive pour une puissance totale limitée. Ce modèle utilise des hypothèses de fluide incompressible, non visqueux à écoulement bidimensionnel et instationnaire autour d’un corps mince. Le nageur est représenté par des segments linéaires dont la cinématique a été d’abord identifiée puis imposée. La monopalme est composée de barres rigides liées par des ressorts en flexion. Elle est en équilibre dynamique avec le fluide.

Les influences de la distribution de rigidité de la palme sur les puissances et le rendement ont été décrites par des études paramétriques.

Il ressort de ces études, en première approximation, qu’un accroissement global de rigidité augmente la puissance totale dissipée alors que la puissance propulsive augmente puis décroît à partir d’un certain seuil. De plus, la sensibilité des puissances propulsive et totale aux raideurs diminue des pieds du nageur vers le bord de fuite.

Les raideurs ont été optimisées pour trois puissances limites différentes au moyen de l’algorithme GBNM. Les résultats des optimisations indiquent des raideurs optimales décroissantes depuis la base vers le bord de fuite de la palme (cf. Figure 8.7). Pour une dépense énergétique plus importante, la rigidité globale de la monopalme croît et son rendement diminue (cf. Tables 8.4 et 8.5). Cette dernière tendance a été déjà observée dans les études paramétriques (cf. Figure 8.6).

Enfin, on remarque que les monopalmes optimales mises en évidence dans cette partie ont des nombres de Strouhal proches de l’intervalle optimal ($S_t \in [0.25; 0.35]$, déterminé expérimentalement dans [87]). Cependant S_t ne caractérise pas suffisamment le mouvement de la palme pour l’opti-

miser selon les critères de puissance et de rendement ici présentés, car on trouve des monopalmes à bas rendement qui ont aussi S_t dans cet intervalle.

Dans cette application, GBNM a été utilisé à deux reprises : une fois comme algorithme d'optimisation pour résoudre le problème (8.4), et une seconde fois pour aider à minimiser le résidu des équations dynamiques du couplage fluide-structure (cf. Annexe G et Figure G.1). Le problème (8.4) présente des caractéristiques auxquelles GBNM est bien adapté : le nombre de variables est raisonnable, c'est un problème contraint, et le calcul du gradient n'est pas disponible.

Pour que les résultats de l'optimisation des raideurs 2D puissent être utilisés pour la fabrication d'une monopalme, une traduction vers une structure 3D s'impose. C'est le sujet de la prochaine partie du mémoire.

Troisième partie

Identification de monopalmes 3D par GBNM

Chapitre 10

Introduction

Dans cette partie, la méthode GBNM est appliquée à des problèmes d'identification de monopalmes 3D. Deux applications sont exposées. La première, présentée dans le Chapitre 11, a pour but de déterminer les épaisseurs d'une monopalme 3D à partir des résultats obtenus en Partie II de l'optimisation 2D des raideurs de flexion. Cette identification est basée sur la correspondance du modèle 2D avec un modèle 3D en éléments finis. La seconde application, détaillée dans le Chapitre 12, trouve les placements des sauts de plis de manière à avoir deux monopalmes équivalentes fabriquées avec des tissus différents. Cela est obtenu avec une correspondance entre les modèles 3D en éléments finis des deux monopalmes, la monopalme originale et la monopalme à identifier. Les identifications sont formulées comme des problèmes d'optimisation, lesquels sont résolus par l'algorithme GBNM.

Chapitre 11

Traduction de la monopalme 2D en une structure 3D

11.1 Méthodologie

La répartition des raideurs qui constitue la solution du problème d'optimisation bidimensionnel (Partie II) est maintenant traduite en une structure tridimensionnelle. Dans sa formulation la plus générale, le problème d'identification associé à cette traduction est mal posé car, la famille des formes 3D étant plus riche que celle des formes 2D, plusieurs formes 3D correspondent à l'optimum 2D. Cependant, dans la réalité, la forme du plan de la monopalme est choisie en fonction de considérations marketing puis fixée une fois qu'un moule a été fabriqué. La forme plane de la palme étant connue, les raideurs 2D ne sont plus associées qu'à la répartition des épaisseurs en 3D, qui plus est, la palme est fabriquée à partir de tissus préimprégnés empilés, l'épaisseur est constante le long de l'envergure et ne varie qu'aux sauts de plis le long de la corde. Il s'agit donc de traduire les raideurs du système 2D en épaisseurs suivant la corde de la palme 3D. Ce problème est moins pathologiquement mal posé que la formulation générale.

La traduction est la recherche d'une palme équivalente au modèle 2D qui, dans une perspective d'identification, est vu comme la "cible" ou les "expériences". L'équivalence mécanique entre les deux modèles peut être requise dans les comportements statique, dynamique¹ ou un mélange des deux. L'avantage de l'équivalence statique est la disponibilité d'analyses en grands déplacements. Cependant l'inertie de la palme est négligée devant l'inertie de l'eau et la raideur de flexion de la

¹Dans cette partie du manuscrit le terme "dynamique" fait référence aux fréquences et/ou aux modes propres d'un modèle. Ainsi, des fois, dans ce même sens, le terme "modale" est utilisé au lieu de "dynamique".

palme. L'équivalence dynamique modale prend en compte l'inertie et la raideur de la monopalme, mais elle est, par essence, une analyse en petits déplacements. En outre, les modes de vibrations qui ne sont pas de flexion n'ont pas de correspondance dans le modèle bidimensionnel. Pour cette raison, seul le premier mode est considéré car il a été constaté empiriquement qu'il est toujours de flexion. La répartition d'épaisseur 3D est obtenue à partir de la minimisation d'une fonction erreur J , qui mesure l'écart entre le comportement des deux modèles, et est donnée par,

$$J = \mu J_{static} + (1 - \mu) J_{freq} , \quad (11.1)$$

où,

$$J_{static} = \frac{\sum_{i=1}^{NCP} ((u_x^i - \tilde{u}_x^i)^2 + (u_y^i - \tilde{u}_y^i)^2)}{\sum_{i=1}^{NCP} ((u_x^i)^2 + (u_y^i)^2)} \quad \text{et} \quad J_{freq} = \frac{|\omega_1^2 - \tilde{\omega}_1^2|}{\omega_1^2} . \quad (11.2)$$

μ est un facteur de pondération entre les critères statique et dynamique, (u_x^i, u_y^i) sont les déplacements cibles des articulations dans le modèle des barres 2D, $(\tilde{u}_x^i, \tilde{u}_y^i)$ sont les déplacements des NCP points de contrôle dans le modèle 3D (cf. Figure 11.1), ω_1 est la première fréquence propre du modèle 2D, et $\tilde{\omega}_1$ est la première fréquence propre du modèle 3D. Le chargement sous lequel les déplacements $u_x^i, u_y^i, \tilde{u}_x^i$ et \tilde{u}_y^i sont calculés est arbitraire. Il existe des conditions peu restrictives sur ce chargement qui induisent l'identifiabilité de la palme (cf. Section 11.2 et Annexe 14). La monopalme 3D est analysée avec un modèle en éléments finis volumiques améliorés pour réduire les blocages en cisaillement transverse, en membrane et en courbure ([53, 54]). Ces éléments sont capables de modéliser efficacement des structures minces sous petits ou grands déplacements en utilisant un seul élément suivant l'épaisseur. Une loi matériau orthotrope a été ajoutée au code de calcul original développé dans [53].

L'identification de la palme est formulée comme le problème d'optimisation suivant,

$$\left\{ \begin{array}{l} \min_{h_i} J , \\ \text{tel que,} \\ h_i^{min} \leq h_i \leq h_i^{max} , \quad i = 1, 6 , \quad \text{et} \\ h_i \leq h_{i-1} , \quad i = 2, 6 , \end{array} \right. \quad (11.3)$$

où les épaisseurs h_i sont bornées par $h_i^{min} = 1 \times 10^{-5} \text{ m}$ et $h_i^{max} = 2 \times 10^{-2} \text{ m}$, $i = 1, 6$. Pour résoudre ce problème d'optimisation, l'algorithme GBNM est employé. Les dernières contraintes,

$h_i \leq h_{i-1}$, $i = 2, 6$, sont un moyen d'injecter une connaissance obtenue lors de l'optimisation 2D dans l'identification : les palmes optimales sont effilées, i.e., leur épaisseur décroît. Ces contraintes sont traitées par la remise en ordre des variables h_i avant l'analyse par éléments finis (EF), tout en les conservant désordonnées dans l'optimisation. Une telle opération peut être interprétée comme une projection dans le domaine faisable permettant d'en atteindre tous les points. Lorsqu'un tel opérateur existe il est plus efficace pour traiter la contrainte qu'une pénalisation adaptative car il n'y a pas à apprendre les $\bar{\lambda}_i$ avant de résoudre le problème. L'ordonnement des h_i a donc été utilisé à la place de la pénalisation.

Les points de contrôle sur le maillage 3D sont dans le plan inférieur ($y = 0$) (cf. Figure 11.1(b)). Ainsi, un changement d'épaisseur correspond à la variation des coordonnées des nœuds du plan supérieur. Cela nous donne des positions initiales inchangées pour les points de contrôle au début de chaque évaluation, tout au long du processus d'identification. On remarque qu'un placement différent des points de contrôle le long de l'épaisseur n'a pas d'influence sur les résultats finaux de l'identification. La palme étant une structure très mince, les écarts entre les déplacements (et par conséquent entre les J_{static}) mesurés dans les plans inférieur, moyen et supérieur sont négligeables. Cela a été confirmé par des essais numériques.

11.2 Résultats de l'identification d'épaisseurs pour le passage 2D \rightarrow 3D

Une étude de l'effet des formulations de l'identification sur la répartition finale d'épaisseur (problème (11.1)) est réalisée. Dans un premier temps, $J = J_{static}$ ($\mu = 1$), les influences des cas de chargement et des analyses en petits ou grands déplacements sont examinées (Sections 11.2.1 et 11.2.2). Trois cas de chargement vertical sont testés, comme schématisés en Figure 11.2. Notons qu'il y a deux conditions nécessaires pour que $h^* = \arg \min J_{static}$ soit unique : i) les angles d'équilibre relatifs (les θ_i dans le système 2D) doivent être non nuls ii) le nombre de points de contrôle NCP doit être égal ou supérieur au nombre de variables (6 ici) sinon plusieurs combinaisons de variables peuvent produire des déplacements identiques de ces points de contrôle. L'Annexe 14 discute ces deux conditions plus en détails.

Dans un second temps, la formulation dynamique $J = J_{freq}$ ($\mu = 0$) est utilisée et les résultats sont exposés en Section 11.2.3.

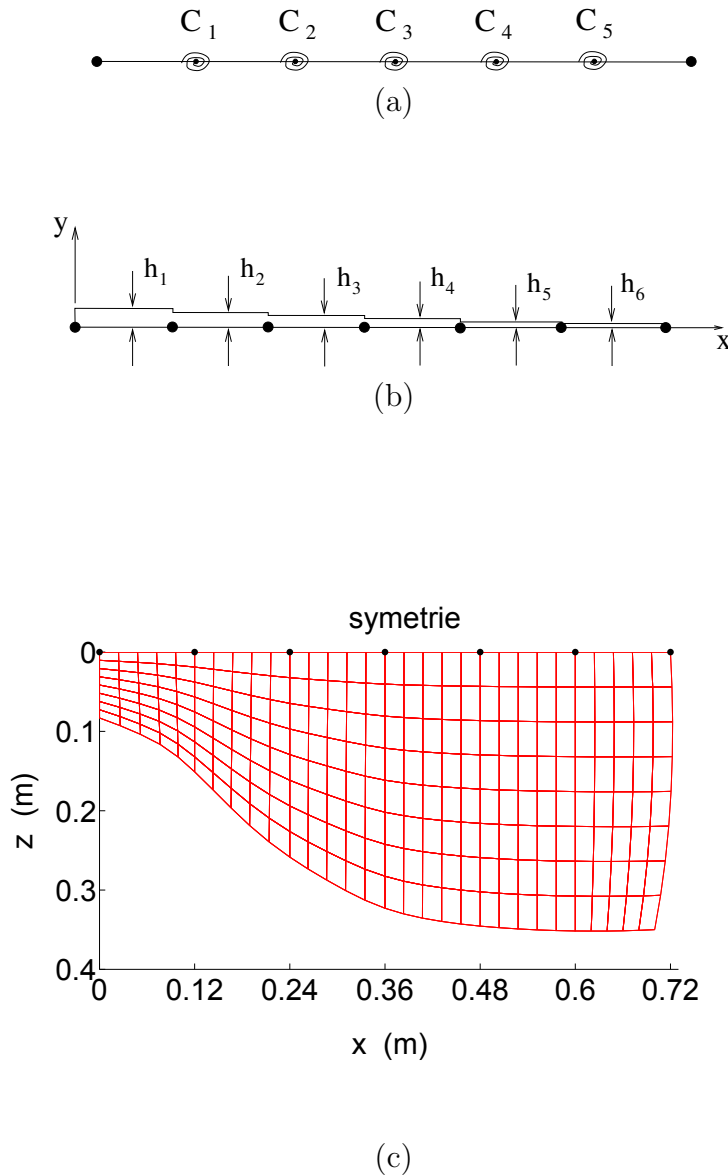


FIG. 11.1 – Correspondance 2D \rightarrow 3D : le modèle 2D (a), paramétrisation du profil d'épaisseur de la monopalme 3D (b), et vue de dessus avec maillage en éléments finis (c). Le modèle éléments finis a 1 élément dans l'épaisseur. En (b) et (c) les points sont les $NCP = 7$ points de contrôle où J est évalué.

Finalement, le problème est résolu en utilisant un critère mixte entre les analyses statique et modale ($\mu = 0.5$, Section 11.2.4).

Puisque le plan de la monopalme est symétrique, une seule moitié de la palme 3D est modélisée, l'autre moitié ayant un comportement symétrique (cf. Figure 11.1(c)). Cette modélisation ne permet pas d'obtenir les fréquences propres associées aux modes propres non-symétriques par rapport au plan de symétrie (modes de torsion). Cependant, ces modes ne présentent pas d'équivalence dans le modèle 2D et n'ont donc pas d'influence sur les résultats de l'identification.

La monopalme est fabriquée en tissu composite, les coefficients élastiques du stratifié suivant les directions x et z (cf. repère xyz donné en Figure 11.1) étant $E_x = 40 \text{ GPa}$, $E_z = 40.0 \text{ GPa}$, $G_{xz} = 4.9 \text{ GPa}$, $\nu_{xz} = 0.11$, et la densité $\rho = 1840 \text{ kg/m}^3$. L'identification des épaisseurs est réalisée pour la répartition optimale de raideurs sous la condition de dépense énergétique du nageur $\bar{P}^{min} = -2000 \text{ W}$.

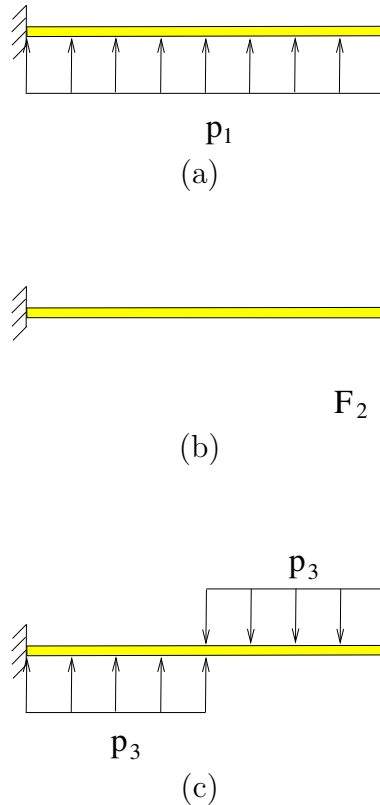


FIG. 11.2 – Cas de chargement statique et conditions aux limites considérés pour l'identification des épaisseurs.

Les résultats préliminaires de l'identification ont fourni des valeurs d'épaisseurs trop importantes. Cela est dû à la surestimation des efforts dans le modèle 2D. Ainsi, une correction des efforts puis une correction des raideurs de flexion 2D est réalisée. Ces corrections exploitent la connaissance de profils d'épaisseurs réalistes, déjà utilisés dans la fabrication des monopalmes, et sont détaillées dans l'Annexe J. La répartition de raideurs 2D optimale qui sert de cible pour l'étude est celle de la Figure J.4.

11.2.1 Identification en petits déplacements

Tout d'abord une procédure d'identification en analyses statiques seules ($\mu = 1$), sous petits déplacements, est réalisée. Les trois cas de chargement, présentés en Figure 11.2, sont testés avec $p_1 = 4.0875 \text{ N/m}$, $F_2 = 0.73575 \text{ N}$ et $p_3 = 4.0875 \text{ N/m}$. Les répartitions initiale et finales d'épaisseur sont montrées en Figure 11.3. Pour les cas de chargement considérés, on remarque que les répartitions finales sont proches. La différence la plus importante est de 2.5% de variation sur h_2 . La Figure 11.4 compare les déformées du modèle de barres 2D (déformée cible) et de la ligne $z = 0$ du plan inférieur du modèle 3D éléments finis (EF), sous le cas de chargement 1.

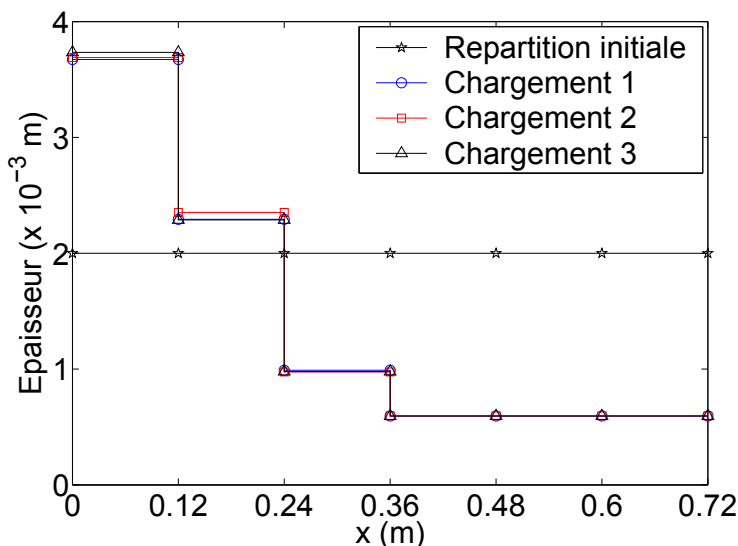


FIG. 11.3 – Profils d'épaisseur : identification en petits déplacements.

11.2.2 Identification en grands déplacements

Les trois cas de chargement utilisés pour l'identification en petits déplacements sont ici repris mais avec des intensités six fois plus grandes, pour caractériser un problème en grands déplacements : $p_1 = 24.525 \text{ N/m}$, $F_2 = 4.4145 \text{ N}$ et $p_3 = 24.525 \text{ N/m}$.

Les répartitions initiale et finales d'épaisseur sont montrées en Figure 11.5. Pour les cas de chargement considérés, on remarque que les répartitions sont proches, la différence la plus importante étant de 9% pour h_2 . La Figure 11.6 compare les déformées des modèles de barres 2D et par éléments finis 3D.

On observe que les formulations statiques en petits déplacements sont très peu sensibles au cas

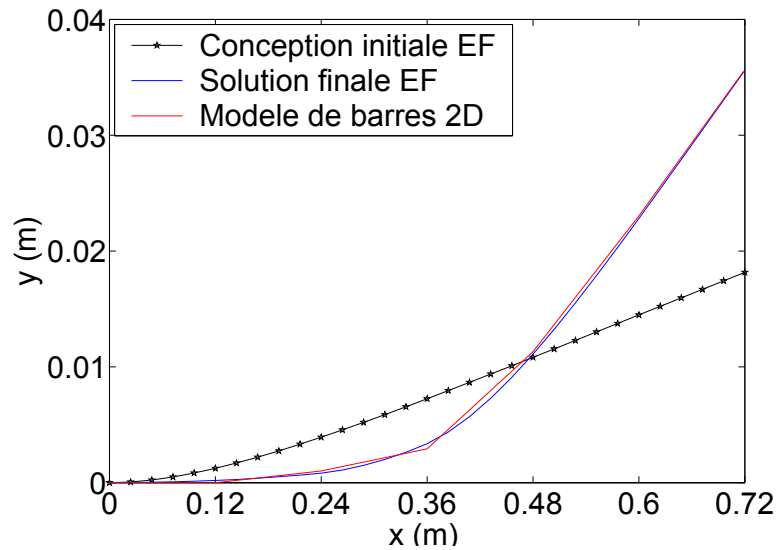


FIG. 11.4 – Comparaison des déformées pour la formulation en petits déplacements, cas de chargement 1.

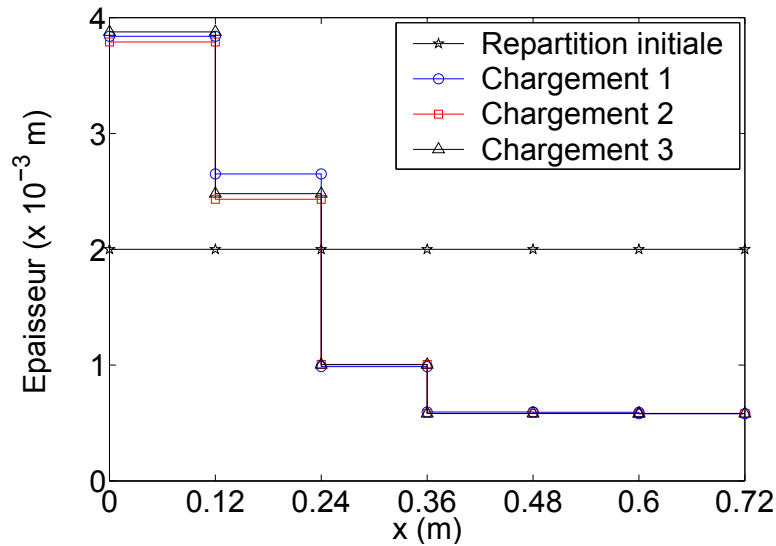


FIG. 11.5 – Profils d'épaisseur : identification en grands déplacements.

de chargement. La sensibilité aux cas de chargement augmente en grands déplacements probablement parce que les déplacements sont plus importants qu'en petits déplacements et ils dépendent des contraintes mécaniques auxquelles la structure est soumise. Les solutions en petits et grands déplacements restent néanmoins proches. La différence fondamentale est que les solutions en grands déplacements sont plus épaisses dans la région la plus proche du pied du nageur (cf. comparaison en Figure 11.8) car pour un même chargement, l'analyse en grands déplacements

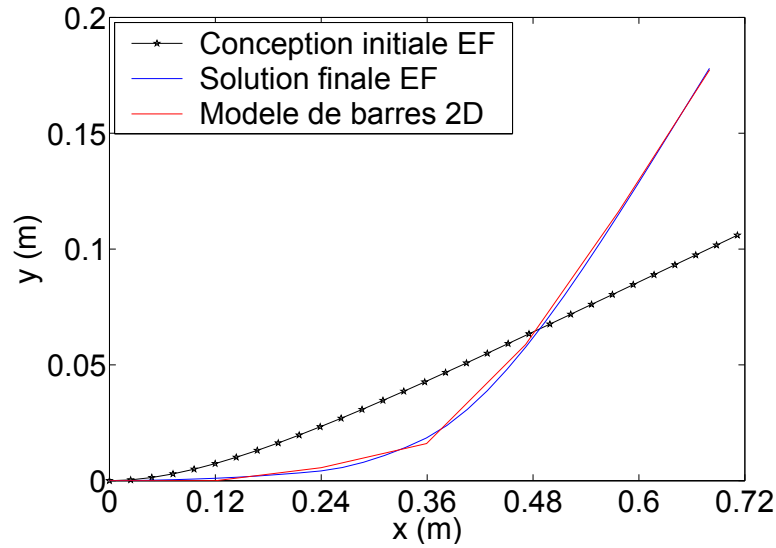


FIG. 11.6 – Comparaison des déformées pour la formulation en grands déplacements, cas de chargement 1.

Cas de chargement	Formulation statique	
	petits déplacements	grands déplacements
1	2.4842	2.5584
2	2.4777	2.5137
3	2.4720	2.5208

TAB. 11.1 – Première fréquence propre ($\tilde{f}_1 = \frac{\tilde{\omega}_1}{2\pi}$), en Hz , pour les solutions aux formulations statiques de l'identification 2D \rightarrow 3D.

produit des déplacements plus grands dans la région de l'encastrement que l'analyse en petits déplacements.

Bien que les fréquences propres ne soient pas prises en compte dans les formulations statiques de d'identification, la Table 11.1 donne la première fréquence propre \tilde{f}_1 des modèles 3D identifiés. Vis à vis de ce critère dynamique aussi, les résultats des différentes identifications statiques sont proches. La différence en \tilde{f}_1 est inférieure à 0.5% et 1.8% entre les solutions en petits et grands déplacements, respectivement. En considérant toutes les solutions statiques confondues, la différence reste inférieure à 3.5%.

11.2.3 Identification en dynamique

La formulation dynamique de l'identification ($\mu = 0$ dans l'Equation (11.1)) vise à rapprocher la première fréquence propre du système 3D de celle du modèle de barres 2D optimal, qui est $f_1 = \frac{\omega_1}{2\pi} = 1.5400 \text{ Hz}$ (cf. calcul donné en Annexe K). Il a été observé numériquement avec GBNM (qui fournit plusieurs solutions si le nombre d'analyses maximal le permet) que cette formulation possède plusieurs solutions très proches, c'est-à-dire plusieurs répartitions d'épaisseur correspondant à la fréquence cible f_1 . La Figure 11.7 montre les solutions trouvées par l'algorithme GBNM en une exécution de 1500 analyses, dont les écarts à f_1 sont plus faibles que 10^{-4} Hz , et les modes propres associés.

Il existe plusieurs solutions qui vont de palmes épaisses à la base et fines au bout, jusqu'à des palmes à épaisseur presque constante. L'amplitude maximale du mode propre est toujours en bout de palme mais la courbure maximale change de position, se déplaçant de la région de l'encastrement pour la Solution 1, vers le bord de fuite pour les Solutions 2 et 3.

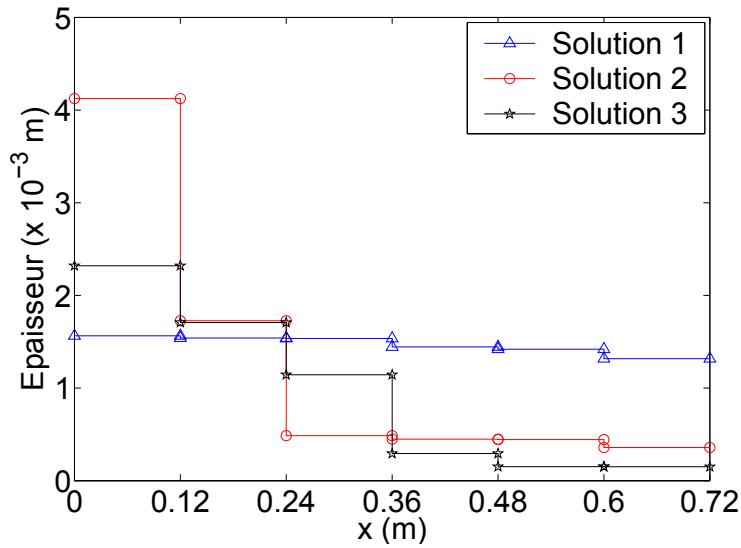
11.2.4 Formulation mixte (statique-dynamique) de l'identification

Le critère mixte ($\mu = 0.5$ dans l'Equation (11.1)) permet de prendre en compte à la fois les comportements statique et dynamique de la monopalme dans la traduction $2D \rightarrow 3D$. Dans la partie statique de la formulation mixte, seul le cas de chargement 1 en petits déplacements est considéré, puisque les sensibilités aux cas de chargement et aux non-linéarités géométriques sont négligeables (Section 11.2.2).

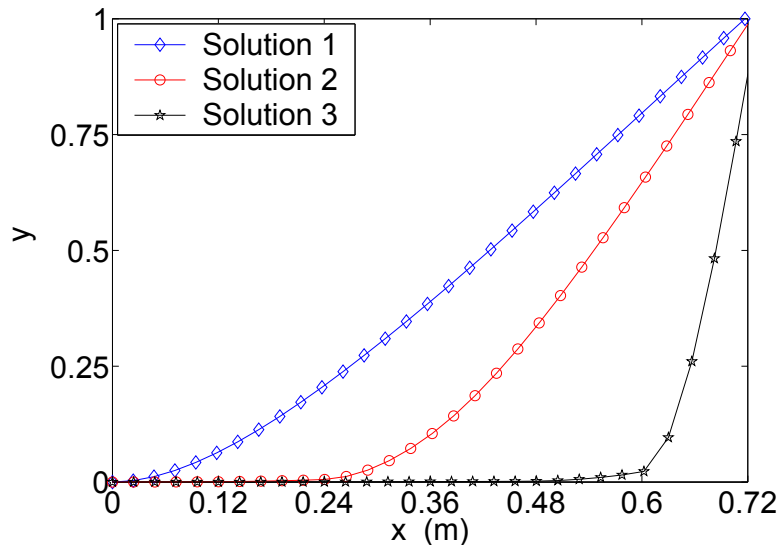
La Figure 11.8 compare les solutions aux formulations mixte ($\mu = 0.5$), statique linéaire (petits déplacements et $\mu = 1$), et statique non linéaire (grands déplacements et $\mu = 1$). La Figure 11.9 montre les déformées du modèle de barres 2D optimal et de la ligne $z = 0$ du plan inférieur du modèle 3D éléments finis (EF) identifié. La fréquence propre de la palme identifiée en formulation mixte est $\tilde{f}_1 = \frac{\tilde{\omega}_1}{2\pi} = 2.2115 \text{ Hz}$.

En comparant les solutions des formulations statiques et mixte, on constate que cette dernière est plus fine à la base de la palme et plus épaisse au bout. Le déplacement de matière vers le bout de palme permet de réduire la valeur de f_1 pour s'approcher de celle du modèle 2D. En contrepartie, la déformée statique s'éloigne légèrement de la déformée cible.

On constate donc que, pour la monopalme, les formulations statique et dynamique de la tra-



(a)



(b)

FIG. 11.7 – Profils d'épaisseur obtenus avec l'identification en dynamique (a) et les premiers modes propres correspondants, normalisés par le déplacement vertical maximal (b).

duction 2D \rightarrow 3D ne donnent pas de résultats compatibles. En effet, le modèle 2D, contrairement à la palme, a une répartition de masse constante le long de la corde, ainsi la palme ne peut pas à la fois reproduire son comportement dynamique et statique. Pour répondre à ce dilemme, il est préférable d'utiliser la formulation mixte statique/dynamique pour la traduction.

Concernant les analyses statiques, un modèle éléments finis en petits déplacements est suffisant.

Il a été constaté que les formulations statiques en petits et grands déplacements n'induisent pas de différences significatives dans la conception finale de la monopalme. De plus, lorsque la formulation mixte est considérée, des hautes précisions dans les analyses statiques ne sont pas essentielles car les équivalences statique et dynamique entre les modèles 2D et 3D sont un compromis entre les deux formulations et, ainsi, il n'y a plus de concordance exacte des déformées statiques.

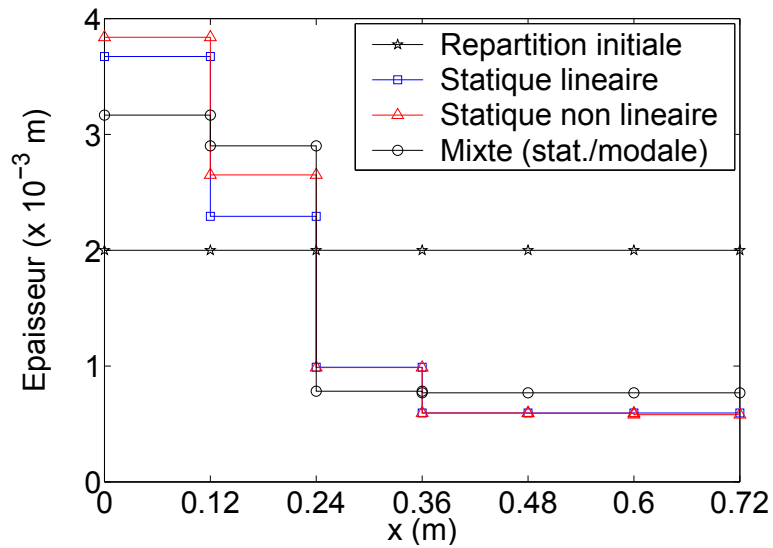


FIG. 11.8 – Comparaison des profils d'épaisseur entre les 3 procédures d'identification.

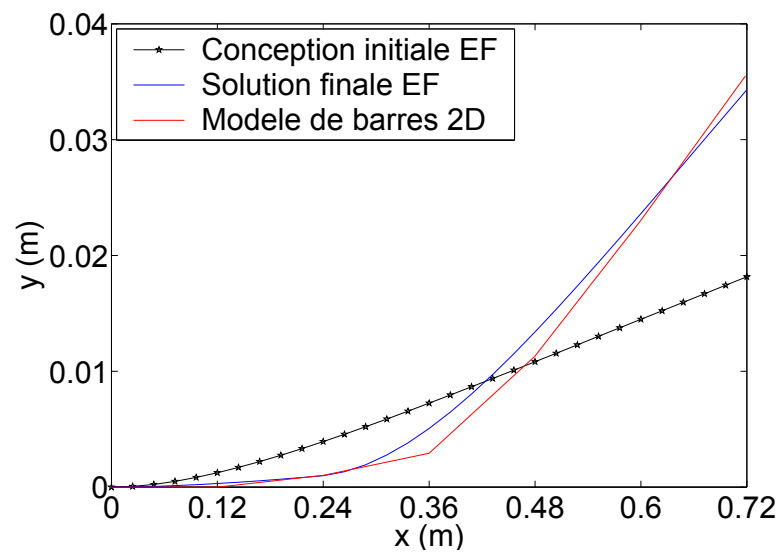


FIG. 11.9 – Comparaison des déformées statiques pour la formulation mixte, cas de chargement 1.

Chapitre 12

Identification des positions des sauts de plis lors d'un changement de tissu dans la fabrication des monopalmes

Alors qu'un objet, une monopalme en l'occurrence, est en fabrication en série, il arrive fréquemment que les matériaux constitutifs, du tissu préimprégné ici, changent. Il s'agit alors de redimensionner l'objet fait du nouveau matériau pour qu'il conserve ses fonctions initiales.

Dans l'exemple qui nous concerne, l'objectif est de conserver les propriétés mécaniques pertinentes d'une monopalme dont l'épaisseur du préimprégné qui la compose augmente. La finalité d'un tel changement est de pouvoir réduire le nombre total de plis dans la palme, donc son temps de fabrication. La différence de matériau constitutif est compensée par une relocalisation des positions de sauts de plis (cf. Figure 12.1).

Les notions couramment utilisées par les nageurs pour caractériser les palmes sont la “dureté”, c'est-à-dire, en termes mécaniques, la rigidité de flexion, le “centre de courbure” et la “nervosité” qui mélange la rigidité de flexion et l'inertie. Ces critères résument un comportement mécanique mieux décrit par la déformée sous chargement statique et les fréquences propres. Les positions des sauts de plis des palmes en tissu plus épais sont optimisées pour que la déformée et les fréquences propres soient aussi proches que possible de la palme de référence. L'optimisation est effectuée par l'algorithme GBNM.

12.1 Méthodologie

Comme pour la traduction 2D \rightarrow 3D du Chapitre 11, on cherche une équivalence statique et dynamique des deux monopalmes. Pour l'équivalence statique, les déformées des structures sous un chargement combiné de force et de moment en bout de la palme et encastées au pied du nageur (cf. schéma en Figure 12.2) sont comparées. Ces conditions limites visent à imiter le test manuel de flexion de palme des experts en monopalmes. Pour l'équivalence dynamique ne sont considérées que les premières fréquences propres qui sont les plus importantes. Les calculs de déformées et les analyses modales sont réalisées au moyen d'un modèle éléments finis 3D en employant le même code utilisé que pour la traduction 2D \rightarrow 3D (cf. Section 11.1 et [53]). L'option en petits déplacements est adoptée pour l'obtention des déformées statiques. Du fait de la symétrie de la palme et pour réduire la mémoire ordinateur occupée et le temps de calcul, ce qui est critique pour l'identification à venir, une moitié de palme est modélisée, l'autre moitié ayant un comportement symétrique. Cette modélisation ne permet pas d'avoir les fréquences associées aux modes propres non-symétriques par rapport au plan de symétrie. Cependant, on a observé empiriquement que le premier mode de la palme complète est symétrique par rapport à ce plan. Les deux premières fréquences propres du modèle avec symétrie sont considérées pour l'équivalence dynamique. Une analyse modale effectuée sur une palme complète et illustrée en Figure 12.11 indique, par comparaison avec la Figure 12.9 et les valeurs des fréquences associées, que les 2 premiers modes symétriques correspondent en fait au premier et troisième modes.

La Figure 12.3 représente le maillage éléments finis de la palme de référence, les points de contrôle où les déplacements u sont calculés, et indique la région où le chargement est appliqué.

Les positions des sauts des plis, xp_i , sont trouvées à travers la minimisation de

$$J = \mu J_{static} + (1 - \mu)(J_{freq1} + J_{freq2}) , \quad (12.1)$$

où J_{static} et J_{freq} sont des critères d'erreur quadratiques relatives,

$$J_{static} = \frac{\sum_{i=1}^{NCP} ((u_x^i - \tilde{u}_x^i)^2 + (u_y^i - \tilde{u}_y^i)^2 + (u_z^i - \tilde{u}_z^i)^2)}{\sum_{i=1}^{NCP} ((u_x^i)^2 + (u_y^i)^2 + (u_z^i)^2)} , \quad (12.2)$$

$$J_{freq1} = \frac{|\omega_1^2 - \tilde{\omega}_1^2|}{\omega_1^2} \quad \text{et} \quad J_{freq2} = \frac{|\omega_2^2 - \tilde{\omega}_2^2|}{\omega_2^2} .$$

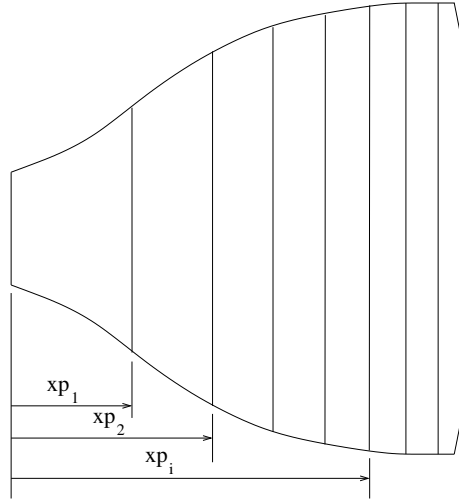


FIG. 12.1 – Paramétrisation des positions des sauts de plis.



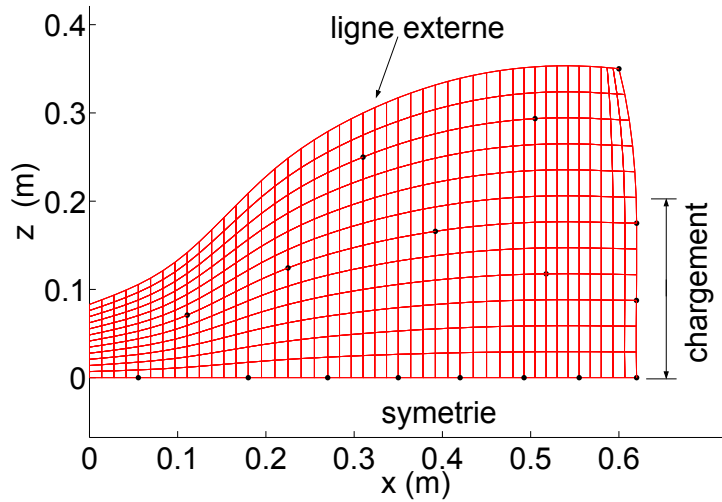
FIG. 12.2 – Chargement statique de la monopalme (vue de coté).

(u_x^i, u_y^i, u_z^i) sont les déplacements cibles de la palme originale aux NCP points de contrôle (cf. Figure 12.3), $(\tilde{u}_x^i, \tilde{u}_y^i, \tilde{u}_z^i)$ sont les déplacements de la palme à identifier aux points de contrôle, ω_1 et ω_2 les deux premières fréquences propres de la palme originale, et $\tilde{\omega}_1$ et $\tilde{\omega}_2$ les deux premières fréquences propres de la palme à identifier. μ est un facteur de pondération entre les critères statique et dynamique. Ici, une importance égale est accordée aux deux critères, ainsi $\mu = 0.5$.

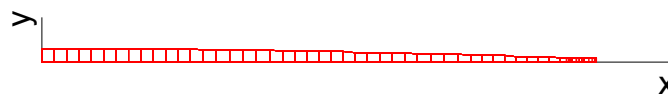
L'identification est finalement formulée comme le problème d'optimisation suivant,

$$\left\{ \begin{array}{l} \min_{xp_i} J, \\ \text{tel que,} \\ xp_i^{min} \leq xp_i \leq xp_i^{max}, \quad i = 1, NSP, \quad \text{et} \\ xp_i \leq xp_{i+1}, \quad i = 1, NSP - 1, \end{array} \right. \quad (12.3)$$

où xp_i sont les positions des sauts de plis suivant le repère $0x$ donné en Figure 12.3, $xp_i^{min} = 0.02 \text{ m}$ et $xp_i^{max} = 0.58 \text{ m}$, $i = 1, NSP$, NSP étant le nombre de sauts de plis de la palme à identifier. L'algorithme GBNM est utilisé pour résoudre ce problème d'optimisation. Les dernières contraintes $xp_i \leq xp_{i+1}$, $i = 1, NSP - 1$, sont introduites pour respecter la condition de répartition de nombre de plis décroissant de la base vers le bout de palme. Ces contraintes sont traitées de la



(a)



(b)

FIG. 12.3 – Maillage de la monopalme à 10 plis. Vue de dessus : points de contrôle et zone de chargement (a). Vue de coté avec les épaisseurs agrandies 10 fois (b). Les points de contrôle sont positionnés dans le plan inférieur ($y = 0$).

même façon que les dernières contraintes du problème (11.3), c'est-à-dire, par remise en ordre des variables xp_i avant l'analyse par éléments finis.

Tout comme pour le modèle de la monopalme cible, les points de contrôle sur le maillage de la palme à identifier sont dans le plan inférieur ($y = 0$). Un changement d'épaisseur équivaut donc à la modification des positions des nœuds du plan supérieur.

La palme à identifier est remaillée à chaque évaluation de J pour que les nœuds coïncident avec les sauts de plis. Les points de contrôle ne sont pas forcément sur les nœuds. C'est pourquoi les déplacements aux points de contrôle résultent de l'interpolation des déplacements aux nœuds de l'élément contenant chaque point de contrôle. Les fonctions d'interpolation sont celles de la formulation de l'élément.

12.2 Résultats et discussion

Deux monopalmes sont identifiées. En raison de la localisation des sauts de plis, les monopalmes originales ont des "duretés" différentes désignées "souple" et "dureté moyenne" (cf. Annexe L). Ces monopalmes sont les structures de référence pour l'identification, et ses caractéristiques sont résumées ci-après :

- Forme : cf. dessins en Annexe L
- 7 sauts de plis, positionnés à :
 - monopalme "souple" : $[0.18 \ 0.27 \ 0.35 \ 0.42 \ 0.48 \ 0.53 \ 0.58] \ m$
 - monopalme "dureté moyenne" : $[0.25 \ 0.31 \ 0.37 \ 0.43 \ 0.49 \ 0.55 \ 0.59] \ m$
- Nombre de couches de la base à l'extrémité : $[10 \ 9 \ 8 \ 7 \ 6 \ 5 \ 4 \ 3]$
- Epaisseur de chaque couche : $0.15 \ mm$
- Matériau : HexPly M10/43%/7268 (structure taffetas, renfort verre "E", masse nominale de renfort (grammage) : $200 \ g/m^2$)
- Propriétés matériaux (ici, 1 et 2 représentent, respectivement, les directions de chaîne et de trame du tissu) :
 - Module $E_1 = 18 \ GPa$
 - Masse volumique $\rho = 1970 \ kg/m^3$ (calculée en supposant un taux volumique de fibre de 55%)
 - $E_2 = E_1 = 18 \ GPa$, $G_{12} = 2.6 \ GPa$, $\nu_{12} = 0.13$ ¹

¹Propriétés estimées d'après [24], page 70 où un pli de tissu équilibré, taffetas, verre E, 50% taux volumique

Les caractéristiques des nouvelles monopalmes sont :

- Forme plane : identique aux palmes originales
- Matériau : Vicotex M10/43%/664 (structure satin 8, renfort verre "E", masse nominale de renfort : 305 g/m^2)
- Epaisseur de chaque couche : 0.21 mm
- Répartition du nombre de couches : [6 5 4 3 2] ou [7 6 5 4 3]
- 4 sauts de plis dont les positions sont à identifier ($NSP = 4$)
- Propriétés matériaux :
 - Module $E_1 = 20 \text{ GPa}$
 - Masse volumique $\rho = 1970 \text{ kg/m}^3$ (calculée en supposant un taux volumique de fibre de 55%)
 - $E_2 = E_1 = 20 \text{ GPa}$, $G_{12} = 2.6 \text{ GPa}$, $\nu_{12} = 0.13^2$

12.2.1 Monopalme "souple"

Dans un premier temps, une répartition de nombres de couches [6 5 4 3 2] de tissu de verre E 305 g/m^2 Vicotex M10/43%/664 (nommée palme à 6 plis) est utilisée pour l'identification de la monopalme équivalente "souple". Le meilleur point trouvé par GBNM, après 5000 évaluations, est [0.3632 0.4061 0.4832 0.5521] m . La performance de ce point est $J = 1.83951 \times 10^{-1}$ ($J_{static} = 1.40292 \times 10^{-1}$, $J_{freq1} = 2.27604 \times 10^{-1}$, $J_{freq2} = 5.95112 \times 10^{-6}$). Avec une répartition [7 6 5 4 3] de tissu de verre E 305 g/m^2 Vicotex M10/43%/664 (nommée palme à 7 plis), après 5000 évaluations, le meilleur point trouvé est [0.1685 0.3475 0.4077 0.4687] m , avec une performance $J = 8.62974 \times 10^{-5}$ ($J_{static} = 1.72546 \times 10^{-4}$, $J_{freq1} = 1.85810 \times 10^{-8}$, $J_{freq2} = 3.02251 \times 10^{-8}$). Le point initial pour l'identification de la palme à 7 plis est [0.10 0.20 0.30 0.50] m . Les deux premières fréquences propres du modèle de la palme à 10 plis, des meilleures palmes à 6 et 7 plis, et du point initial à 7 plis, sont données en Table 12.1. Les Figures 12.4 et 12.5 comparent les déformées de la ligne de symétrie dans les plans inférieur et supérieur de la monopalme, respectivement. Les Figures 12.6 et 12.7 présentent des comparaisons des déformées de la ligne externe. La Figure 12.8 montre les déformées statiques en 3D pour le chargement statique.

de renfort a les modules $E_1 = E_2 = 20 \text{ GPa}$, $\rho = 1900 \text{ kg/m}^3$, $\nu_{12} = 0.13$ et $G_{12} = 2.85 \text{ GPa}$. Le module G_{12} est obtenu par proportionnalité des E_1 . ν_{12} est gardé constant.

²Malgré l'augmentation de E_1 et E_2 , la valeur du module G_{12} est considérée égale à celle de la palme originale, car la structure du tissu est de type satin.

	Cible (10 plis)	Identif. 6 plis	Identif. 7 plis	Point initial (7 plis)
f_1 (Hz)	1.6625	1.4611	1.6625	1.4984
f_2 (Hz)	7.2373	7.2372	7.2373	6.4921

TAB. 12.1 – Identification de la monopalme “souple” : comparaison entre les deux premières fréquences propres des différentes palmes.

On remarque que la monopalme à 6 plis présente des comportements statique et dynamique éloignés de la monopalme originale. La déformée est très écartée de la déformée cible et la valeur de la première fréquence propre possède une erreur de l'ordre de 0.2 Hz (12%). Les deuxièmes fréquences sont très proches.

En revanche, la palme à 7 plis qui a été identifiée, présente des comportements équivalents à ceux de la palme originale. Les déformées statiques sont très proches et les fréquences propres ont des erreurs inférieures à 10^{-6} Hz (0.00006%). Les Figures 12.9 et 12.10 montrent les deux modes propres, normalisés par le déplacement vertical maximum, de la palme cible et de la palme à 7 plis identifiée, respectivement. On remarque que les modes de la palme identifiée correspondent bien aux modes de la palme cible.

En conclusion, avec un nombre maximal de 6 plis de tissu de verre E 305 g/m^2 Vicotex M10/43%/664, il n'est pas possible de reproduire le comportement mécanique de la monopalme “souple” (10 plis de tissu de verre E 200 g/m^2 HexPly M10/43%/7268). 7 plis de tissu de verre E 305 g/m^2 Vicotex M10/43%/664 permettent, par contre, une équivalence au sens de J_{static} et J_{freq} .

D'autres répartitions de sauts de plis produisent des convergences locales. La Table 12.2 liste les solutions trouvées par GBNM pour l'identification de la monopalme “souple” à 7 plis.

Positions des sauts de plis, xp_i (m)	J
[0.2020 0.3105 0.4100 0.4999]	4.70573×10^{-4}
[0.1685 0.3475 0.4077 0.4687](*)	8.62974×10^{-5}
[0.2404 0.2668 0.4222 0.4998]	4.28694×10^{-4}
[0.1829 0.3225 0.4166 0.4836]	1.17843×10^{-4}

(*) meilleure solution trouvée

TAB. 12.2 – Solutions obtenues par GBNM pour la monopalme “souple” à 7 plis en 5000 analyses.

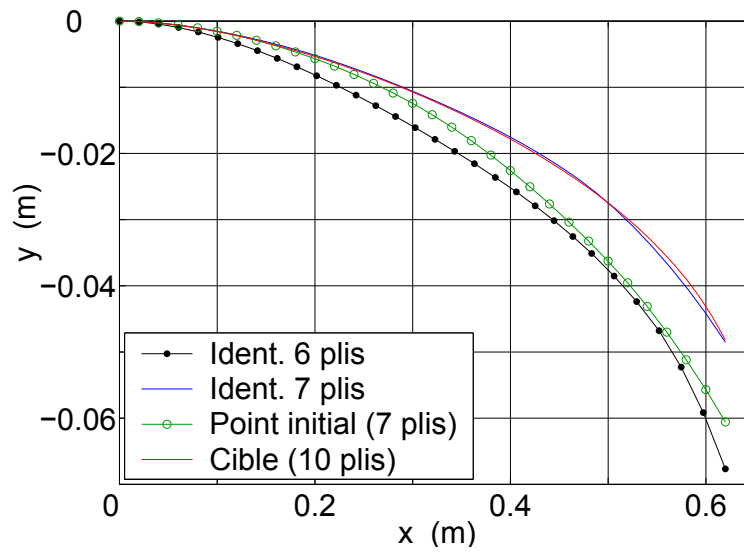


FIG. 12.4 – Comparaison des déformées statiques de la ligne de symétrie au plan inférieur de la monopalme “souple”.

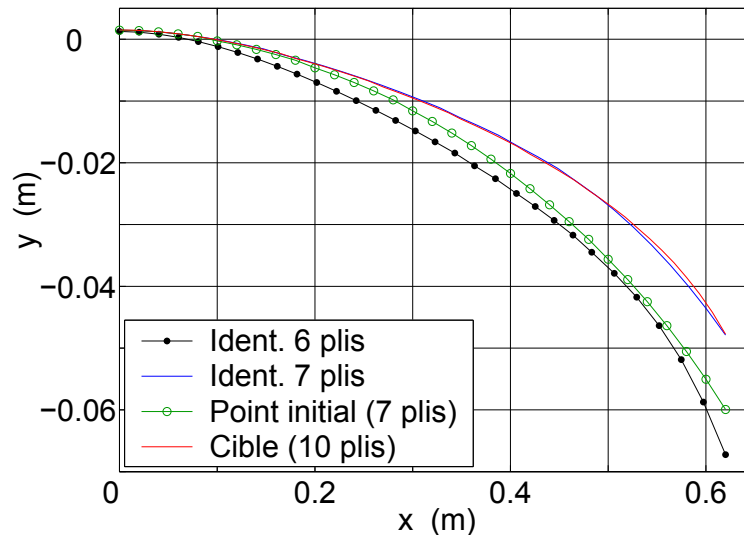


FIG. 12.5 – Comparaison des déformées statiques de la ligne de symétrie au plan supérieur de la monopalme “souple”.

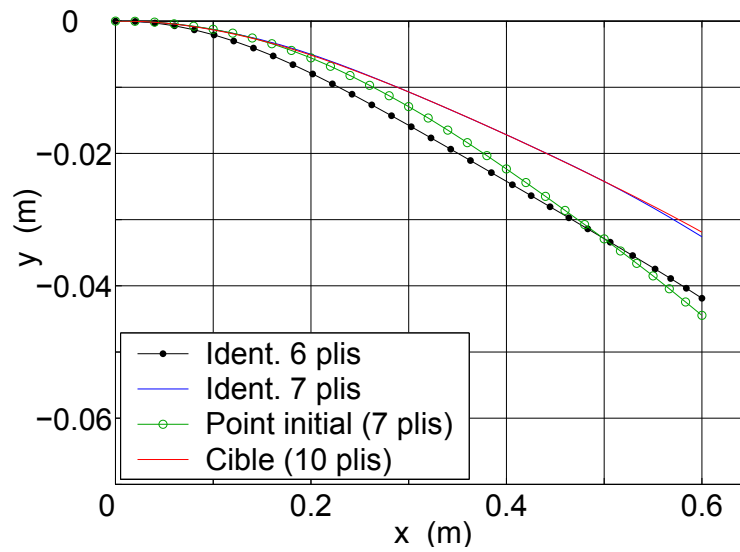


FIG. 12.6 – Comparaison des déformées statiques de la ligne externe au plan inférieur de la monopalme “souple”.

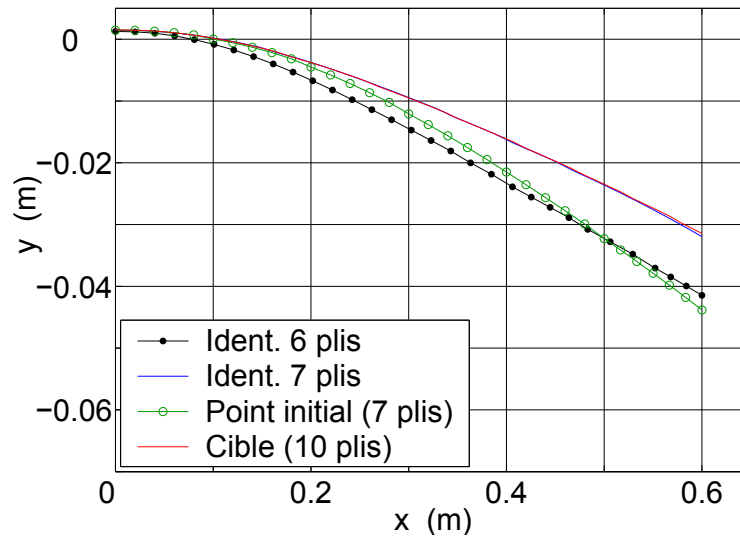
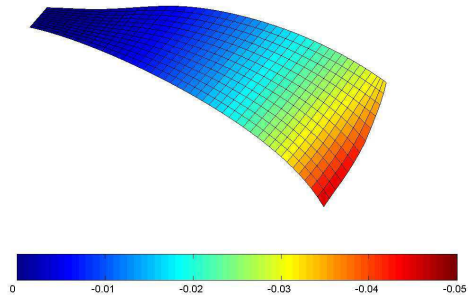
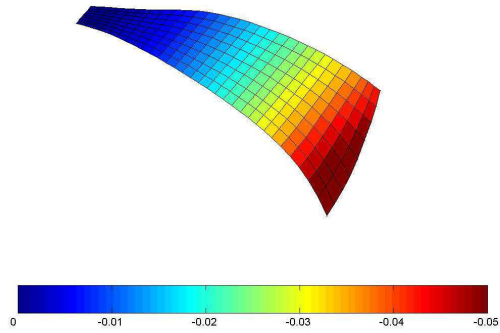


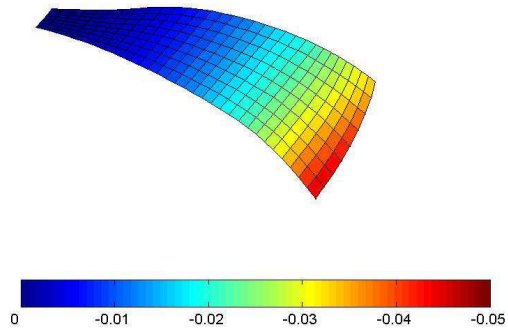
FIG. 12.7 – Comparaison des déformées statiques de la ligne externe du plan supérieur de la monopalme “souple”.



(a)

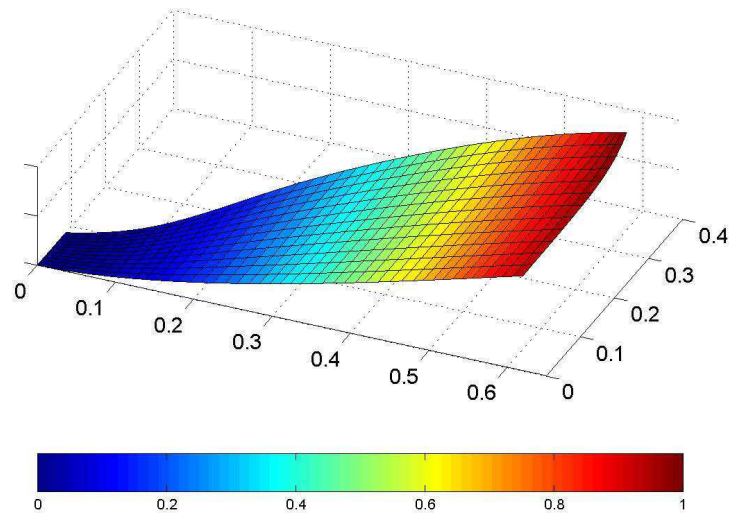


(b)

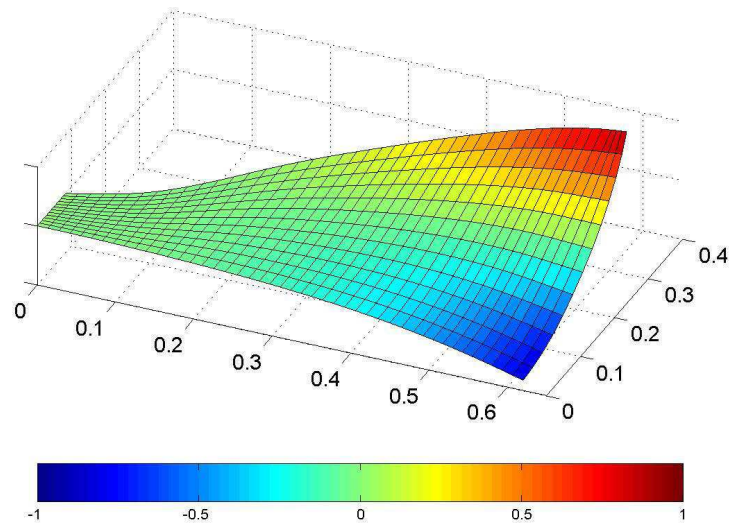


(c)

FIG. 12.8 – Déformées 3D pour le chargement statique : monopalmes “souples” cible (a), monopalmes identifiées à 6 plis (b), et monopalmes identifiées à 7 plis (c).

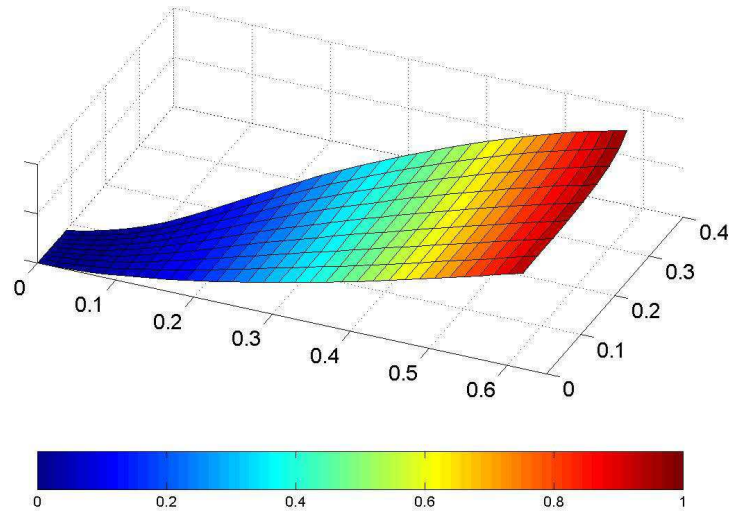


(a) $f = 1.66 \text{ Hz}$

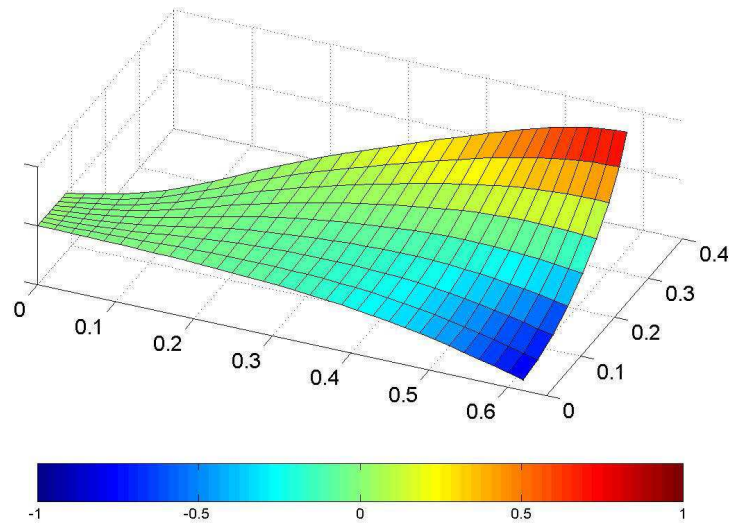


(b) $f = 7.24 \text{ Hz}$

FIG. 12.9 – Les deux premiers modes propres de la monopalme “souple” originale (modélisation avec condition de symétrie).

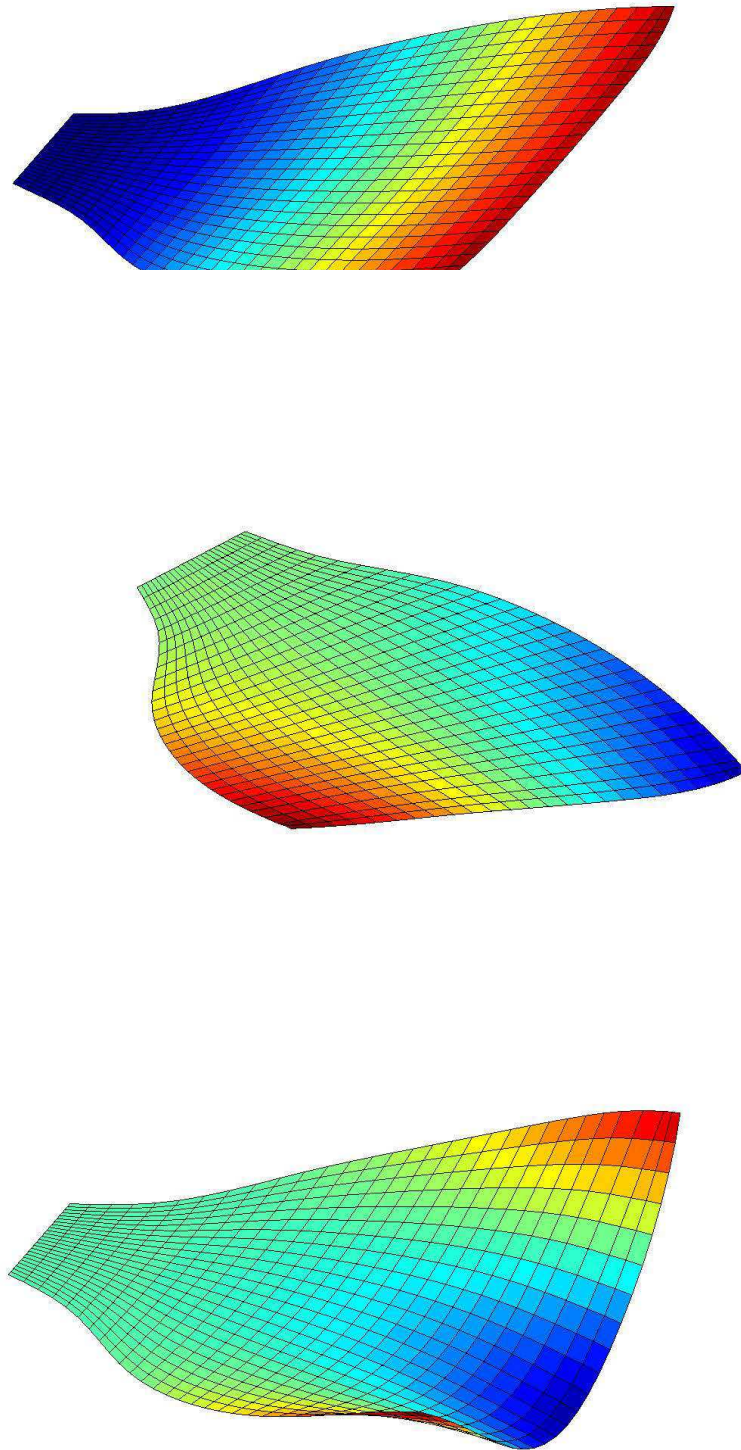


(a) $f = 1.66 \text{ Hz}$



(b) $f = 7.24 \text{ Hz}$

FIG. 12.10 – Les deux premiers modes propres de la monopalme “souple” identifiée à 7 plis (modélisation avec condition de symétrie).



(c) $f = 7.24 \text{ Hz}$

FIG. 12.11 – Les trois premiers modes propres de la monopalme “souple” identifiée à 7 plis. La palme est encastrée à sa base et le modèle n’impose pas de symétrie.

12.2.2 Monopalme “dureté moyenne”

Les configurations à faible nombre de couches, comme la répartition [6 5 4 3 2] seront trop souples pour la palme “dureté moyenne” puisqu’elles étaient déjà trop souples pour la palme “souple” étudiée dans la solution précédente. C’est pourquoi la répartition [7 6 5 4 3] de tissu de verre E 305 g/m^2 Vicotex M10/43%/664 est utilisée au départ pour l’identification de la monopalme équivalente “dureté moyenne”. Les solutions trouvées par GBNM après 5000 évaluations sont listées dans la Table 12.3. Le meilleur point est [0.2028 0.3629 0.4258 0.5057] m . La performance de ce point est $J = 4.70899 \times 10^{-5}$ ($J_{static} = 9.02186 \times 10^{-5}$, $J_{freq1} = 3.46689 \times 10^{-6}$, $J_{freq2} = 4.94341 \times 10^{-7}$). Le point initial est [0.10 0.20 0.30 0.50] m . Les deux premières fréquences propres du modèle de la palme à 10 plis, de la palme à 7 plis identifiée, et du point initial, sont données dans la Table 12.4. Les Figures 12.12 et 12.13 comparent les déformées de la ligne de symétrie dans les plans inférieur et supérieur de la monopalme, respectivement. Les Figures 12.14 et 12.15 montrent des comparaisons des déformées de la ligne externe.

On remarque ainsi que la monopalme identifiée a des comportements équivalents à ceux de la monopalme “dureté moyenne” originale.

Positions des sauts de plis, xp_i (m)	J
[0.2028 0.3629 0.4258 0.5057]	4.70899×10^{-5}
[0.2045 0.3463 0.4225 0.4973]	5.84558×10^{-5}
[0.2208 0.3626 0.4111 0.5219]	8.67783×10^{-5}
[0.1978 0.3581 0.4354 0.4960]	1.03081×10^{-4}

TAB. 12.3 – Solutions obtenues par GBNM pour la monopalme “dureté moyenne” en 5000 analyses.

	Cible (10 plis)	Identif. 7 plis	Point initial (7 plis)
f_1 (Hz)	1.6572	1.6572	1.4984
f_2 (Hz)	7.5693	7.5693	6.4921

TAB. 12.4 – Identification de la monopalme “dureté moyenne” : comparaison entre les deux premières fréquences propres des différentes palmes.

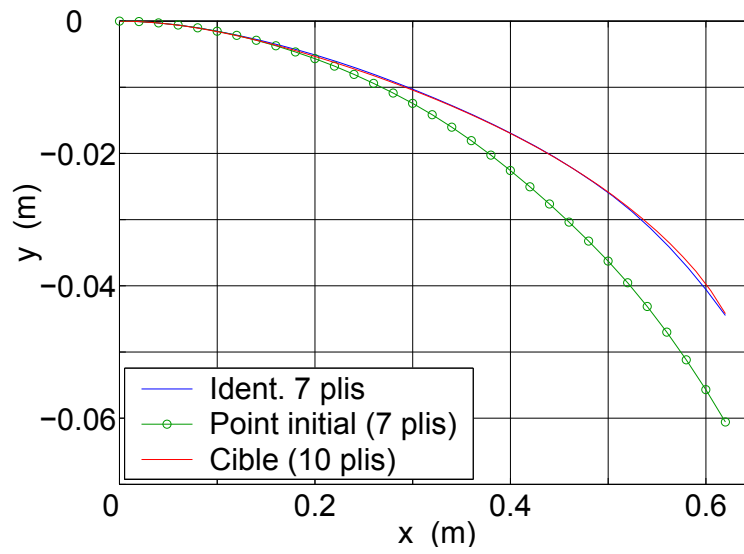


FIG. 12.12 – Comparaison des déformées statiques de la ligne de symétrie au plan inférieur de la monopalme “dureté moyenne”.

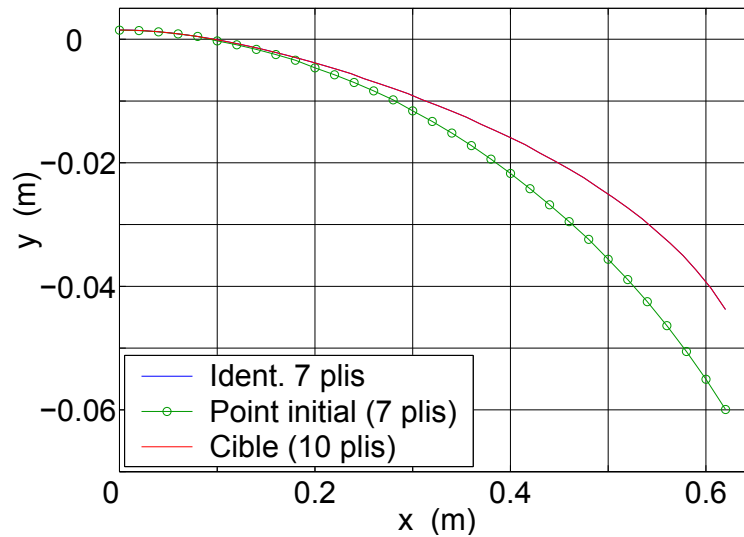


FIG. 12.13 – Comparaison des déformées statiques de la ligne de symétrie au plan supérieur de la monopalme “dureté moyenne”.

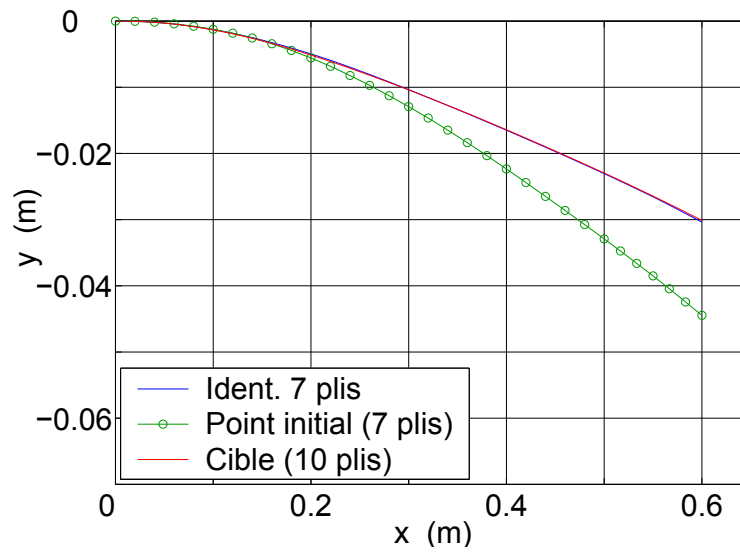


FIG. 12.14 – Comparaison des déformées statiques de la ligne externe au plan inférieur de la monopalme “dureté moyenne”.

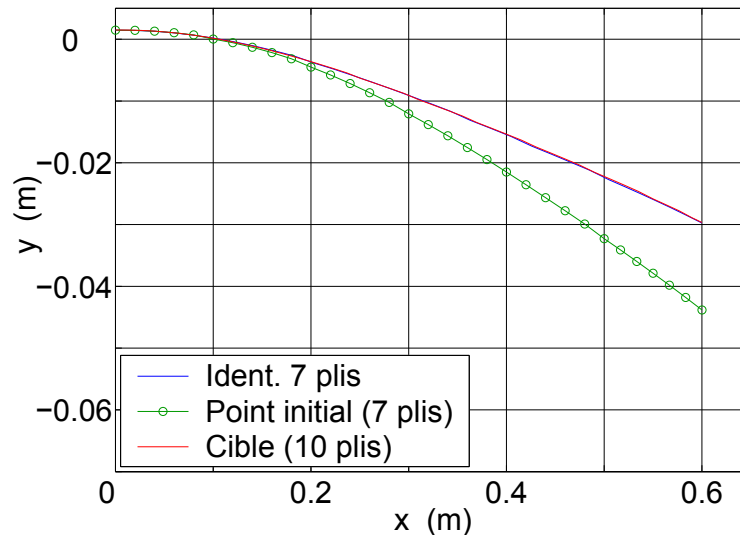


FIG. 12.15 – Comparaison des déformées statiques de la ligne externe au plan supérieur de la monopalme “dureté moyenne”.

Chapitre 13

Conclusions de la Partie III

L'algorithme d'optimisation GBNM, développé dans la première partie de ce mémoire, a été appliqué à deux problèmes d'identification de monopalmes. Le premier problème visait à traduire un modèle 2D précédemment optimisé en un modèle 3D, ce qui impliquait de déterminer la répartition d'épaisseur d'une monopalme 3D. Notons que l'approche globale de l'optimisation d'une monopalme mise en œuvre dans les Parties II et III à savoir, l'optimisation d'un système couplé fluide-structure en 2D puis la transformation de ses résultats en 3D, est numériquement moins coûteuse que l'optimisation directe du système couplé 3D. La faisabilité même de l'optimisation de la palme couplée en 3D reste aujourd'hui douteuse en raison de son coût.

Plusieurs formulations et cas de chargement ont été testés. Il a été constaté que, dans les formulations statiques avec les conditions d'identifiabilité respectées, le cas de chargement n'a pas d'influence sur les résultats finaux. En outre, les différences entre les analyses en petits et grands déplacements peuvent être négligées. La formulation réalisant le meilleur compromis entre l'identifiabilité, la prise en compte des rigidités de flexion, et de l'inertie est la formulation mixte (statique/modale).

La seconde application a permis d'identifier les positions de sauts de plis avec l'objectif de trouver des monopalmes qui ont des comportements mécaniques équivalents bien qu'elles soient fabriquées avec des tissus préimpregnés différents. Dans ce cas à nouveau, une correspondance mixte statique/modale entre les monopalmes 3D a été recherchée. A la condition que les nombres de plis des deux monopalmes autorisent des épaisseurs comparables, des comportements très proches entre les structures ont pu être identifiés.

Récapitulatif et perspectives

Ce travail est consacré au développement d'un nouvel algorithme d'optimisation, GBNM (*Globalized and Bounded Nelder-Mead*), et à son application à la conception et l'identification de structures composites. En particulier, la problématique de la conception de monopalmes de nage est détaillée.

La première Partie de ce mémoire décrit une méthode d'optimisation pour les problèmes en variables continues. Cette méthode, appelée GBNM, est essentiellement locale, mais elle devient globale lorsque le nombre d'analyses croît. Elle a été testée sur des fonctions analytiques multimodales et utilisée dans la résolution de problèmes de conception de stratifiés composites. Des comparaisons avec une autre méthode globale, un algorithme évolutionnaire, ont montré la supériorité de GBNM en précision de convergence et en robustesse pour des problèmes de dimensions 2 à 12.

A l'avenir, le développement de GBNM mérite d'être poursuivi dans deux directions. Il faut d'une part paralléliser GBNM, c'est à dire décliner la méthode pour un environnement informatique distribué. La parallélisation passera par l'initialisation de plusieurs recherches locales et l'affectation préférentielle de ressources informatiques aux recherches les plus prometteuses. D'autre part, le "pragmatisme" de GBNM ne sera complet que lorsqu'il pourra traiter des problèmes en variables mixtes (i.e., continues et discrètes) car les choix discrets sont communs en pratique (choix d'un matériau dans un catalogue par exemple).

Les seconde et troisième Parties du mémoire concernent le difficile problème de la conception d'une monopalme de nage. Cette structure, en apparence simple, implique une modélisation en fait complexe (structure 3D orthotrope en grands déplacements dont la dynamique est couplée au fluide et pilotée par un nageur) dont le coût numérique interdit toute optimisation. Une contribution de ce travail est de proposer une décomposition du problème de la conception d'une monopalme et de résoudre les sous-problèmes d'optimisation créés par GBNM.

Dans la seconde Partie du mémoire, un système simplifié de nageur-fluide-monopalme est proposé puis optimisé. Un modèle d'écoulement particulière bidimensionnel avec non-séparation entre le fluide et la structure est adopté. Le nageur et la palme sont représentés par des segments linéaires, les segments de la palme étant liés par des ressorts de flexion. La palme est en équilibre dynamique avec le fluide. La cinématique du nageur est imposée. L'optimisation maximise la puissance propulsive pour une puissance totale limitée. Les variables sont les rigidités de flexion de la monopalmes 2D.

Notons qu'une perspective intéressante de cette approche serait d'optimiser le mouvement du nageur en même temps que la palme. Pour préciser les contraintes auxquelles la cinématique serait soumise, des études approfondies sur les techniques de nage et la physiologie du mouvement humain seraient nécessaires.

La troisième Partie est dédiée à la traduction des résultats de l'optimisation 2D en un profil d'épaisseur d'une monopalmes 3D. La forme plane et le matériau de la palme étant fixés par des considérations industrielles, l'objectif est de trouver les épaisseurs qui correspondent mécaniquement à la répartition optimale de raideurs. Cela est accompli par la minimisation des écarts entre les comportements statique et/ou modaux du modèle 2D optimal et d'un modèle 3D en éléments finis. Le critère d'identification purement dynamique induit une formulation non-identifiable mais tenant compte de l'inertie des palmes. L'identifiabilité est retrouvée avec la formulation statique et un chargement faisant jouer tous les degrés de liberté de la structure. Dans cette partie, une seconde application sœur est d'identifier les positions de sauts de plis lors d'un changement de tissu composite dans la fabrication de monopalmes. La même démarche de correspondance statique et modale entre deux modèles a été employée.

Sous l'hypothèse réaliste que les ressources informatiques disponibles vont continuer à croître, il sera important d'arriver à prendre en compte les effets d'écoulement 3D et de séparation entre l'écoulement et l'obstacle dans l'optimisation des monopalmes.

L'autre perspective de continuation de développement de la monopalmes est de remplacer les simulations numériques par des expériences réelles car la fabrication d'un prototype de monopalmes est relativement peu coûteuse. L'évaluation de la performance serait réalisée par des nageurs. Cette procédure peut s'appuyer sur les résultats rapportés dans cette thèse et sur la technique de plans d'expériences (DOE pour *Design of Experiments* ([68])) pour guider la confection des prototypes.

Bibliographie

- [1] ABBOTT, I.H., VON DOENHOFF, A.E. *Theory of Wing Sections*. Dover Publications, Inc. (1959).
- [2] BALY, L., DUREY, A., FAVIER, D. Etude des fréquences et amplitudes ondulatoires chez le nageur avec palmes de haut niveau. *Journées de l'Ecole Doctorale des Sciences de la Vie et de la Santé, Univ. de Méditerranée, France* (2001).
- [3] BARHEN, J., PROTOPOPESCU, V., REISTER, D. TRUST : A Deterministic Algorithm for Global Constrained Optimization. *Science*, **276**, 1094–1097 (1997).
- [4] BATCHELOR, G.K. *An Introduction to Fluid Dynamics*. Cambridge University Press, Boston (1985).
- [5] BÄCK, T. *Evolutionary Algorithms in Theory and Practice*. Oxford Univ. Press (1996).
- [6] BÄCK, T., HAMMEL, U., SCHWEFEL, H.-P. Evolutionary Computation : Comments on the History and Current State. *IEEE Transactions on Evolutionary Computations*, **1**(1), 3–17 (1997).
- [7] BERTHELOT, J.-M. *Matériaux Composites*. Tec & Doc, third edn. (1999).
- [8] BESSON, J., FOERCH, R. Large scale object-oriented finite element code design. *Computer Methods in Applied Mechanics and Engineering*, **142**, 165–187 (1997).
- [9] BESSON, J., FOERCH, R. Object-oriented programming applied to the finite element method - part 1 : General concepts. *Revue Européenne des Eléments Finis*, **7**(5), 535–566 (1998).
- [10] BREITKOPF, P., NACEUR, H., RASSINEUX, A., VILLON, P. Moving Least Square Response Surface Approximation : Formulation and Metal Forming Applications. (*soumis à*) *Computer and Structures* (2003).
- [11] BRENT, R. *Algorithms for Minimization without Derivatives*. Prentice-Hall, Inc. (1973).

- [12] COLGATE, J.E., LYNCH, K.M. Mechanics and Control of Swimming : A Review. *IEEE Journal of Oceanographic Engineering*, **(to appear)** (2004).
- [13] COMBES, S.A., DANIEL, T.L. Flexural stiffness in insect wings - II.Spatial distribution and dynamic wing bending. *The Journal of Experimental Biology*, **206**, 2989–2997 (2003).
- [14] CULIOLI, J.-C. *Introduction à l'Optimisation*. Ellipses (1994).
- [15] DANIEL, T., JORDAN, C., GRUNBAUM, D. Hydromechanics of Swimming. *Advances in Comparative and Environmental Physiology - Chapter 2*, **11**, 17–49 (1992).
- [16] DANIEL, T.L., COMBES, S.A. Flexible Wings in Fins : Bending by Inertial or Fluid-Dynamic Forces? *Integr. Comp. Biol.*, **42**, 1044–1049 (2002).
- [17] DENNIS, J.E., TORCZON, V. Direct search methods on parallel machines. *SIAM Journal on Optimization*, **1**(4), 448–474 (1991).
- [18] DI PRAMPERO, P.E., PENDERGAST, D.R., WILSON, D.W., RENNIE, D.W. Energetics of swimming in man. *Journal of Applied Physiology*, **37**(1), 1–5 (1974).
- [19] DOLAN, E.D., LEWIS, R.M., TORCZON, V. On the local convergence of pattern search. *SIAM Journal on Optimization*, **14**(2), 567–583 (2003).
- [20] DUDA, O.R., HART, P.E., STORK, D.G. *Pattern Classification*. John Wiley & Sons, New York, second edn. (2001).
- [21] DURAND, N., ALLIOT, J.-M. A Combined Nelder-Mead Simplex and Genetic Algorithm. *Disponible sur <http://www.recherche.enac.fr/opti/papers/>* (1999).
- [22] FLETCHER, R. *Practical Methods of Optimization*. John Wiley & Sons, second edn. (2000).
- [23] FOLDAGER, J., HANSEN, J.S., OLHOFF, N. A General Approach Forcing Convexity of Ply Angle Optimization in Composite Laminates. *Structural Optimization*, **16**, 201–211 (1998).
- [24] GAY, D. *Matériaux Composites*. Hermes, fourth edn. (1997).
- [25] GILL, P.E., MURRAY, W., WRIGHT, M.H. *Practical Optimization*. Academic Press, London (1981).
- [26] GOLDBERG, D.E., VOESSNER, S. Optimizing Global-Local Search Hybrids. In *GECCO 99 - Genetic and Evolutionary Computation Conference*, pp. 220–228 (1999).
- [27] GÉRADIN, M., RIXEN, D. *Théorie des Vibrations - Application à la Dynamique des Structures*. Masson, Paris, second edn. (1996).

- [28] GRAMMATICOS, B. La Technique de Nage en Monopalme. *Disponible sur <http://web.ccr.jussieu.fr/gmpib/basile/FStech.html>* (1999).
- [29] GÜRDAL, A., HAFTKA, R.T., HAJELA, P. *Design and Optimization of Laminated Composite Materials*. John Wiley & Sons, Inc. (1999).
- [30] GRÉDIAC, M. A Procedure for Designing Laminated Plates with Required Stiffness Properties. Application to Thin Quasi-Isotropic Quasi-Homogeneous Uncoupled Laminates. *Journal of Composite Materials*, **33**(20), 1939–1956 (1999).
- [31] GROSSET, L., LE RICHE, R., HAFTKA, R.T. A double-distribution statistical algorithm for composite laminate optimization. *Submitted to the Structural and Multidisciplinary Optimization* (2004).
- [32] HAFTKA, R.T. Combining Global and Local Approximations. *AIAA Journal*, **29**(9), 1523–1525 (1991).
- [33] HAFTKA, R.T., GÜRDAL, Z. *Elements of Structural Optimization*. Kluwer Academic Publishers, third edn. (1993).
- [34] HICKERNELL, F.J., YUAN, Y.-X. A Simple Multistart Algorithm for Global Optimization. *OR Transactions*, **1**(2) (1997).
- [35] HU, X., SHONKWILLER, R., SPRUILL, M.C. Random Restarts in Global Optimization. Tech. rep., School of Mathematics, Georgia Institute of Technology, Atlanta (jan 1994).
- [36] JONES, R.M. *Mechanics of Composite Materials*. Hemisphere (1975).
- [37] KATZ, J., PLOTKINS, A. *Low Speed Aerodynamics, from wing theory to panel methods*. Mc Graw-Hill Books Co, New York (1991).
- [38] KATZ, J., WEIHS, D. Hydrodynamic propulsion by large amplitude oscillation of an airfoil with chordwise flexibility. *Journal of Fluid Mechanics*, **88**, 485–497 (1978).
- [39] KELLEY, C.T. Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. *SIAM Journal on Optimization*, **10**(1), 43–45 (1999).
- [40] KOLDA, T.G., LEWIS, R.M., TORCZON, V. Optimization by Direct Search : New Perspectives on Some Classical and Modern Methods. *SIAM Journal on Optimization*, **45**(3), 385–482 (2003).
- [41] KVAMSDAL, T., JENSSEN, C., OKSTAD, K., AMUNDSEN, J. Fluid-structure interaction for structural design. In *T. Kvamsdal et al., editors, Proc. Int. Symp. on Comput. Methods*

- for *Fluid-Structure Interaction (FSI'99)*, Tapir Publishers, Trondheim, Norway, pp. 211–238 (1999).
- [42] L.A., SCHMIT, B., FARSHI. Optimum design of laminated fibre composite plates. *International Journal for Numerical Methods in Engineering*, **11**, 623–640 (1977).
- [43] LAGARIAS, J.C., REDDS, J.A., WRIGHT, M.H., WRIGHT, P.E. Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, **9**(1), 112–147 (1998).
- [44] LANCASTER, P., SALKAUSKAS, K. *Curve and Surface Fitting : An Introduction*. Academic Press Inc., London (1986).
- [45] LE MAÎTRE, O., HUBERSON, S., SOUZA DE CURSI, E. Application of a non convex model of fabric deformation to sail cut analysis. *Journal of Wind Analysis and Industrial Aerodynamics* (1997).
- [46] LE MAÎTRE, O., HUBERSON, S., SOUZA DE CURSI, E. Unsteady model of sail and flow interaction. *Journal of Fluids and Structures*, **13**, 37–59 (1999).
- [47] LE RICHE, R. Cours : Introduction à l'Optimisation Globale. *Disponible sur [http ://www.emse.fr/~leriche](http://www.emse.fr/~leriche)* (2004).
- [48] LE RICHE, R., GAUDIN, J. Design of Dimensionally Stable Composites by Evolutionary Optimization. *Composite Structures*, **41**, 97–111 (1998).
- [49] LE RICHE, R., GAUDIN, J. Interfaçage dynamique dans LAMKIT. Rapport final d'étude, LMR, INSA de Rouen / EADS CCR, Rouen, France (2001).
- [50] LE RICHE, R., GUYON, F. Dual Evolutionary Optimization. *Artificial Evolution, Lecture Notes in Computer Science*, **2310**, 281–294 (2001). Selected paper of the 5th International Conference on Artificial Evolution, EA 2001, Collet, P., Lutton, E., Schoenauer, M., Fonlupt, C. and Hao, J.-K., eds., Le Creusot, France.
- [51] LE RICHE, R., HAFTKA, R.T. Optimization of Laminate Stacking Sequence for Buckling Load Maximization by Genetic Algorithm. *AIAA Journal*, **31**(5), 951–970 (1993).
- [52] LE RICHE, R., SAOUAB, A., BRÉARD, J. Coupled compression RTM and composite layup optimization. *Composites Science and Technology, Elsevier publ.*, **63**(15), 2277–2287 (2003).

- [53] LEMOSSE, D. *Eléments Finis Isoparamétriques Tridimensionnels pour l'Etude des Structures Minces*. Thèse de Doctorat, Laboratoire de Mécanique, Institut National des Sciences Appliquées de Rouen, 76801, Saint Etienne du Rouvray, France (2000).
- [54] LEMOSSE, D., DHATT, G. Study of Shell-like Structures by Brick Finite Element Model. In *Proceedings of the Fifth International Conference on Computational Structures Technology, Leuven, Belgium*, vol. H, pp. 183–190 (2000).
- [55] LEVY, A.V., MONTALVO, A. The tunneling algorithm for the global minimization of functions. *SIAM J. of Sci. and Stat. Comp.*, **1**(15), 15–29 (1985).
- [56] LEWIS, R.M., TORCZON, V. Pattern search methods for bound constrained minimization. *SIAM Journal on Optimization*, **9**(4), 1082–1099 (1999).
- [57] LEWIS, R.M., TORCZON, V. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. *SIAM Journal on Optimization*, **12**(4), 1075–1089 (2002).
- [58] LEWIS, R.M., TORCZON, V., TROSSET, M.W. Direct search methods : then and now. *Journal of Computational and Applied Mathematics*, **124**(1-2), 191–207 (2000).
- [59] LINDSEY, C.C. Form, function and locomotory habits in fish. *Fish Physiology - Vol. III - Locomotion*, pp. 1–100 (1978).
- [60] LUERSEN, M.A., LE RICHE, R., GUYON, F. A constrained, globalized, and bounded Nelder-Mead method for engineering optimization. *Structural and Multidisciplinary Optimization*, **27**, 43–54 (2004).
- [61] LUERSEN, M.A., LE RICHE, R., LEMOSSE, D. Identification of a 3D Shape from a 2D Design : Application to a Swimming Monofin. In B. H. V. Topping, C. A. Mota Soares (eds.), *Proceedings of the Fourth International Conference on Engineering Computational Technology*, Stirling, United Kingdom (2004). Civil-Comp Press. paper 30.
- [62] MACIVER, M.A., FONTAINE, E., BURDICK, J.W. Designing future underwater vehicles : principles and mechanisms of weakly electric fish. *IEEE Journal of Oceanographic Engineering*, **(to appear)** (2004).
- [63] MCKINNON, K.I.M. Convergence of the Nelder-Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization*, **9**(1), 148–158 (1998).
- [64] MICHALEWICZ, Z., SCHOENAUER, M. Evolutionary Algorithms for Constrained Parameter Optimization. *Evolutionary Computation*, **4**(1), 1–32 (1997).

- [65] MIKI, M. Material Design of Composite Laminates with Required In-Plane Elastic Properties. In T. Kayashi, K. Kawata, S. Umekawa (eds.), *Proceedings of the Forth International Conference on Composite Materials (ICCM-IV), Tokio*, vol. 2, pp. 1725–1731 (1982).
- [66] MINOUX, M. *Mathematical Programming : Theory and Algorithms*. John Wiley & Sons (1986).
- [67] MOLLENDORF, J.C., FELSKE, J.D., SAMIMY, S., PENDERGAST, D.R. A Fluid/Solid Model for Predicting Slender Body Deflection in a Moving Fluid. *Journal of Applied Mechanics*, **70**, 346–350 (2003).
- [68] MONTGOMERY, D.C. *Design and Analysis of Experiments*. John Wiley & Sons, fifth edn. (2002).
- [69] MOSCATO, P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts : Towards Memetic Algorithms. C3p report 826, Caltech Concurrent Computation Program (jan 1989).
- [70] NELDER, J.A., MEAD, R. A simplex for function minimization. *Computer Journal*, **7**, 308–313 (1965).
- [71] OKAMOTO, M., NONAKA, T., OCHIAI, S., TOMINAGA, D. Nonlinear numerical optimization with use of a hybrid Genetic Algorithm incorporating the Modified Powell method. *Applied Mathematics and Computation*, **91**, 63–72 (1998).
- [72] PEDERSEN, P. On sensitivity analysis and optimal design of specially orthotropic laminates. *Engineering Optimization*, **11**, 305–316 (1987).
- [73] PERTTUNEN, C.D., JONES, D.R., STRUCKMAN, B.E. Lipschitzian optimization without a lipschitz constant. *Journal of Optimization Theory and Applications*, **79**(1), 157–181 (1993).
- [74] POWELL, M.J.D. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, **7**, 155–162 (1964).
- [75] PRESS, W.H., TEUKOLSKY, S.A., VETTERLING, W.T., FLANNERY, B.P. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press (1992).
- [76] RENDERS, J.-M., BERSINI, H. Hybridizing Genetic Algorithms with Hill-Climbing Methods for Global Optimization : Two Possible Ways. *IEEE* (1994).
- [77] ROCKAFELLAR, R.T. Lagrange multipliers in optimization. *Nonlinear Programming, Proc. SIAM-AMS*, **9**, 145–168 (1976).

- [78] ROUX, W.J., STANDER, N., HAFTKA, R.T. Response surface approximations for structural optimization. *International Journal for Numerical Methods in Engineering*, **42**, 517–534 (1998).
- [79] SFAKIOTAKIS, M., LANE, D.M., DAVIES, B.C. Review of Fish Swimming Modes for Aquatic Locomotion. *IEEE Journal of Oceanic Engineering*, **24**(2), 237–252 (1999).
- [80] SHANG, Y., WAN, Y., FROMHERZ, M.P.J., CRAWFORD, L. Toward Adaptive Cooperation between Global and Local Solvers for Continuous Constraint Problems. In *CP'01 Workshop on cooperative Solvers in Constraints Programming*, Cyprus (Decembre 2001).
- [81] SPENDLEY, W., G.R. HIMSWORTH, F.R. Sequential Application of Simplex Designs in Optimisation and Evolutionary Operation. *Technometrics*, **4**, 441–461 (1962).
- [82] SYSWERDA, G. A Study of Reproduction in Generational and Steady State Genetic Algorithms. In G.J.E. Rawlins (ed.), *Foundations of Genetic Algorithms*, San Mateo, CA (1991). Morgan Kaufman.
- [83] TORCZON, V. On the convergence of the multidirectional search algorithms. *SIAM Journal on Optimization*, **1**(1), 123–145 (1991).
- [84] TORCZON, V. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, **7**(1), 1–25 (1997).
- [85] TÖRN, A.A. A Search-Clustering Approach to Global Optimization. *Towards Global Optimization 2*, pp. 49–62 (1978).
- [86] TÖRN, A.A., ZILINSKAS, A. *Global Optimization*. Springer-Verlag, Berlin (1989).
- [87] TRIANTAFYLLOU, G.S., TRIANTAFYLLOU, M.S., GROSENBAUCH, M.A. Optimal Thrust Development in Oscillating Foils with Application to Fish Propulsion. *Journal of Fluids and Structures*, **7**, 205–224 (1993).
- [88] TRIANTAFYLLOU, M., TRIANTAFYLLOU, G. Un poisson robot. *Pour la Science*, (211), 52–58 (1995).
- [89] TU, W., MAYNE, W. Studies of multi-start clustering for global optimization. *International Journal for Numerical Methods in Engineering*, **53**, 2239–2252 (2002).
- [90] VITALI, R., HAFTKA, R.T., SANKAR, B.V. Multi-fidelity design of stiffened composite panel with a crack. *Structural and Multidisciplinary Optimization*, **23**(5), 347–356 (2002).
- [91] WOLFRAM, S. *The Mathematica Book*. Wolfram Media & Cambridge University Press, third edn. (1996).

- [92] WOLPERT, D.H., MACREADY, W.G. The No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, **1**(1), 67–82 (1997).
- [93] WRIGHT, M.H. Direct Search Methods : One Scorned, Now Respectable. *Dundee Biennial Conference in Numerical Analysis, Harlow, UK*, pp. 191–208 (1996).
- [94] ZAMPARO, P., PENDERGAST, D.R., TERMIN, B., MINETTI, A.E. How fins affect the economy and efficiency of human swimming. *The Journal of Experimental Biology*, **205**, 2665–2676 (2002).

Annexe A

Adaptive linear penalty as generalized Lagrangian

¹ The adaptive linear penalty scheme of (2.18) and (2.19) can be analyzed using generalized Lagrangian theory (cf. [77, 66]). In the following, operators $<$, \leq , $>$ and \geq used with vectors apply to all components independently.

Proposition 1 (\bar{L} generalized Lagrangian). *Let \bar{L} be defined as*

$$\begin{cases} \bar{L}(x, \lambda) = f(x) + \lambda \cdot g^+(x) & \text{if } \lambda \geq 0, \\ -\infty & \text{if } \exists k, 1 \leq k \leq m / \lambda_k < 0, \end{cases} \quad (\text{A.1})$$

where $g^+(x) = \max(0, g(x))$. \bar{L} is a generalized Lagrangian.

This ensues directly from the definition of generalized Lagrangians : \bar{L} is a closed concave function of λ , and, introducing the essential objective function \bar{f} ,

$$\begin{cases} \bar{f}(x) = f(x) & \text{if } x \text{ feasible}, \\ +\infty & \text{otherwise}, \end{cases} \quad (\text{A.2})$$

one readily shows that $\bar{f}(x) = \max_{\lambda \in \mathbb{R}^m} \bar{L}(x, \lambda)$, which completes the definition.

Denoting by x^* a solution of the primal problem (P), a given problem, defined by f and g ,

¹Annexe extraite de l'article [60].

has a saddle point with respect to \bar{L} if,

$$\begin{aligned} f(x^*) &= \min_x \bar{f}(x) = \min_x \max_{\lambda} \bar{L}(x, \lambda) = \\ & \max_{\lambda} \min_x \bar{L}(x, \lambda) = \max_{\lambda} \bar{\varphi}(\lambda), \end{aligned} \tag{A.3}$$

where

$$\bar{\varphi}(\lambda) = \min_x \bar{L}(x, \lambda) \tag{A.4}$$

is the dual function associated to \bar{L} . As (A.3) shows, if a problem has a saddle point with respect to the generalized Lagrangian \bar{L} , there is a founded way to build a penalty function that leads to x^* : the primal constrained formulation (P) can be replaced equally by the dual problem,

$$(\bar{D}) \quad \max_{\lambda \in \mathfrak{R}^m} \bar{\varphi}(\lambda). \tag{A.5}$$

If a problem has a saddle point and $\bar{\lambda}$ is a solution of (\bar{D}) , x^* is obtained by minimizing the penalty function $\bar{L}(x, \bar{\lambda})$. In terms of λ , (\bar{D}) is easy to solve because $\bar{\varphi}(\lambda)$ is a concave function (see proof in [66]). Moreover, there is a convenient expression for its sub-gradient.

Proposition 2 (Sub-gradient of the dual function). *A sub-gradient of the (generalized) dual function $\bar{\varphi}$ at λ is $g^+(\bar{x}(\lambda))$, where $\bar{x}(\lambda)$ is such that $\bar{L}(\bar{x}(\lambda), \lambda) = \min_x \bar{L}(x, \lambda)$.*

The proof is the direct application of Theorem 4', Chap. 6, in [66]. Proposition 2 shows that calculating a sub-gradient of $\bar{\varphi}$ is a by-product of calculating φ itself since both involve minimizing $\bar{L}(\cdot, \lambda)$. Such minimization remains the most costly operation when solving (\bar{D}) . Therefore, in this work, the dual function and its sub-gradient are not calculated by the complete resolution of (A.4) but are approximated by the heuristic of (2.19). The condition $\bar{L}(x^{\text{new}}, \lambda^k) \leq \bar{L}(x^{\text{best}}, \lambda^k)$ and the update of x^{best} guarantee that the approximation of $\bar{x}(\lambda)$ improves at each iteration. In light of the definition of $\bar{\varphi}$'s sub-gradient, the λ 's updating formula in (2.19) is interpreted as a fixed step gradient algorithm that solves (\bar{D}) , where $g(\bar{x}(\lambda))$ is approximated by $g(x^{\text{best}})$.

All the above considerations assume that the problem has a saddle point. Perturbation analysis ([77]) helps characterizing the existence of saddle points. For a short while, let \bar{L} denote any generalized Lagrangian. The ordinary Lagrangian, $L(x, \lambda) = f(x) + \lambda.g(x)$ is an instance of \bar{L} . Perturbation analysis uses the perturbational representation, \bar{F} , and the optimal value function, $\bar{\phi}$. Let Y be a real linear space and y be a vector in Y . The perturbational representation is

defined as

$$\bar{F}(x, y) = \max_{\lambda \in \mathbb{R}^m} (\bar{L}(x, \lambda) - \lambda \cdot y) \quad (\text{A.6})$$

and the optimal value function is

$$\bar{\phi}(y) = \min_x \bar{F}(x, y) . \quad (\text{A.7})$$

The following Theorem, from [66], states that if there is a supporting hyperplane of $\bar{\phi}$ at $y = 0$, then solving (\bar{D}) also solves (P) .

Theorem 1. *A necessary and sufficient condition for $\bar{\lambda}$ to be a saddle point multiplier is that,*

$$\forall y \in Y \quad , \quad \bar{\phi}(y) \geq \bar{\phi}(0) - \bar{\lambda} \cdot y . \quad (\text{A.8})$$

Using (A.7), the optimal value function of the ordinary Lagrangian is

$$\phi(y) = \min_{x / g(x) \leq y} f(x) . \quad (\text{A.9})$$

From now on, \bar{L} denotes back the only generalized Lagrangian discussed here, the linear adaptive penalty function of (A.1). It can be shown that the associated optimal value function is,

$$\bar{\phi}(y) = \begin{cases} +\infty & \text{if } \exists k, 1 \leq k \leq m / y_k < 0 , \\ \phi(y) & \text{if } y \geq 0 . \end{cases} \quad (\text{A.10})$$

Figure A.1 shows the optimal value functions ϕ and $\bar{\phi}$ on two example problems. On the left, the problem does not have a saddle point in terms of the ordinary Lagrangian, but has saddle points in terms of the linear adaptive penalty function. On the right, L and \bar{L} both have saddle points. Because $\bar{\phi}$ is equal to ϕ in the subspace of Y such that $y \geq 0$ and is infinite elsewhere, it has supporting hyperplanes at $y = 0$ more frequently than ϕ does.

Proposition 3 (Saddle points of \bar{L}). *The class of problems for which \bar{L} has a saddle point contains the class of problems for which ordinary Lagrangians have a saddle point.*

Two other features of the penalty function \bar{L} are visible on Figure A.1 and are presented without proof.

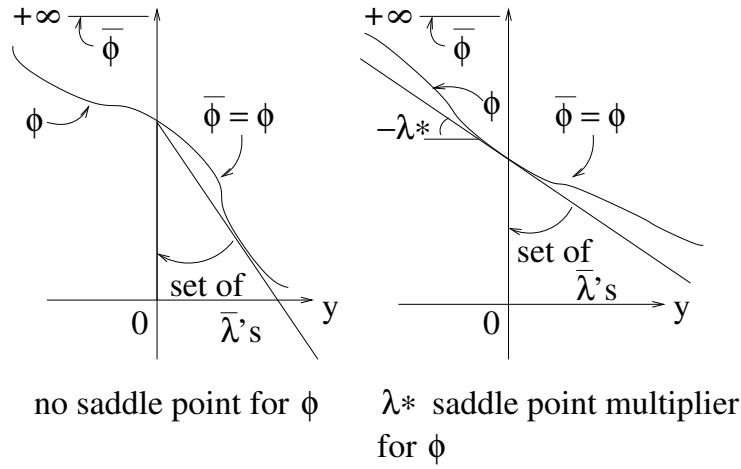


FIG. A.1 – Two examples of optimal value functions ϕ and $\bar{\phi}$ for a single constraint. λ^* and $\bar{\lambda}$ are saddle point multipliers for L and \bar{L} , respectively. On the left, unlike $\bar{\phi}$, ϕ has no supporting hyperplane at $y = 0$ (i.e., no saddle point). A problem with saddle point for the ordinary Lagrangian is sketched on the right : ϕ and $\bar{\phi}$ both have supporting hyperplanes.

Proposition 4 (Set of $\bar{\lambda}$'s). *The set of saddle point multipliers of \bar{L} , $\{\bar{\lambda}\}$ is bounded below. Furthermore, if λ^* is a saddle point multiplier of the ordinary Lagrangian L , it is the lower bound of $\{\bar{\lambda}\}$, i.e., $\bar{\lambda} \geq \lambda^*$.*

In particular, it should be stressed that the lower bound of $\{\bar{\lambda}\}$ is often finite.

Proposition 5 (finite penalty parameters). *x^* can be solution of $\min_x \bar{L}(x, \bar{\lambda})$ for finite values of $\bar{\lambda}$.*

Quadratic and higher order exterior penalty functions achieve differentiability at $g(x) = 0$ (providing f and g are differentiable) but need, in all cases, infinitely large penalty parameters in order to approach the feasible domain. On the contrary, \bar{L} is not differentiable but convergence to the optimum x^* can occur for finite penalties.

Annexe B

Mise en œuvre de GBNM dans LAMKIT

La méthode GBNM a été mise en œuvre dans Matlab et dans LAMKIT©. LAMKIT est un logiciel d'analyse et d'optimisation de structures composites développé par EADS-CCR et distribué par EADS-CIMPA (<http://www.eads.net/lamkit>). En mode calcul simple il peut réaliser des calculs statiques, de calculs de rupture, de calculs de flambement de plaques simplement supportées, des calculs du temps d'injection par les procédés type RTM (*Resin Transfer Molding*) et CRTM (*Compressed Resin Transfer Molding*) avec injection latérale ([52]), et des calculs de dispersion par simulation de Monte Carlo de plusieurs caractéristiques. En mode optimisation, les caractéristiques physiques du stratifié peuvent être manipulées par un des optimiseurs monocritère sous contraintes ou multicritères. 44 critères d'optimisation sont disponibles dans LAMKIT, et les optimiseurs, à l'exception de GBNM, sont basés sur des algorithmes évolutionnaires. LAMKIT est écrit en C++ et systématise le chargement dynamique des données, ainsi que l'instanciation dynamique des classes ([8, 9]). La modularité du programme est favorisée par des interfaces propres et logiques. L'utilisateur de LAMKIT peut ajouter des fonctionnalités au logiciel, sans en posséder les sources. Ceci facilite la maintenance du logiciel et donne la possibilité à l'utilisateur d'ajouter facilement des modules métier ([49]).

La mise en donnée de l'optimiseur GBNM est montrée ci-dessous (fichier `.opt`) pour un problème de maximisation du module d'élasticité E_x pour un composite à 4 couches. Les variables sont les orientations des fibres. Le composite est symétrique, ainsi il n'y a que deux variables.

exemple_gbnm.opt

```
****optimize gbnm
***variables
**cont_ply_angle
*continuous
  name t1
  ref 1.
  init 65.
  min -90.
  max 90.
*specific
  seq 1
  lam_type sym
**cont_ply_angle
*continuous
  name t2
  ref 1.
  init 65.
  min -90.
  max 90.
*specific
  seq 2
  lam_type sym
***criteria
  Ex      35.e+09
***convergence
  length           5.0
  reflection       1.0
  contraction      0.5
  expansion        2.0
  coef_size_gauss 0.01
  nb_random_points 10
  size_stop        0.000001
  size_degeneration 0.0000001
  max_nb_iteration 1000
****return
```

Le mot clé qui active l'optimiseur GBNM est `gbnm`, au début du fichier. Une brève description des paramètres utilisés par GBNM est donné ci-dessous :

`length` : taille initiale du simplexe ; le même que a de l'Equation 2.8.

`reflection`, `contraction` et `expansion` : coefficients des opérations auxquelles le simplexe est soumis dans l'algorithme de Nelder-Mead (voir Figure 2.4). Les valeurs les plus courants sont 1, 0.5 et 2, respectivement.

`size_stop` : tolérance pour le critère de convergence *small* (voir Section 2.3). Le ε_{small} dans l'Equation 2.12.

`size_degeneration` : tolérance pour vérifier si un simplexe est dégénéré (voir Section 2.2.2).

`nb_random_points` : nombre de points aléatoires générés pour la ré-initialisation probabilisée ; le même que N_r (voir Section 2.4.3).

`coef_size_gauss` : paramètre qui contrôle l'étalement des Gaussiennes, utilisé pour estimer la densité de probabilité p (voir Section 2.4.1) ; le même que α dans l'Equation 2.4.

`max_nb_iteration` : nombre maximum d'évaluations de la fonction coût ; le même que C_{max} , cf. Figure 2.8.

Annexe C

Comparaison entre GBNM et la fonction *fmincon* de Matlab

Matlab est un des logiciels de calcul numérique les plus utilisés dans l'industrie et dans les institutions de recherche scientifique. C'est pourquoi, dans cette annexe, des comparaisons sont réalisées entre GBNM (dont une version est programmée sous Matlab) et *fmincon*, la fonction de Matlab (Optimization Toolbox version 2.2) pour les problèmes d'optimisation sous contraintes.

Pour les problèmes généraux d'optimisation non linéaire, *fmincon* utilise la méthode SQP (programmation quadratique séquentielle) ([25]). L'estimation initiale de l'Hessien du Lagrangien est la matrice identité, et sa correction est faite par la formule de Broyden, Fletcher, Goldfarb et Shanno (BFGS ([22])). Si le gradient de la fonction coût n'est pas fourni, Matlab l'estime par différences finies.

Le sens des comparaisons effectuées peut, en toute rigueur, être contesté car GBNM peut devenir un optimiseur global alors que *fmincon* est purement local. Cependant, ces comparaisons donnent des indications pragmatiques sur la contribution de GBNM.

Les paramètres de *fmincon*, excepté les points initiaux, sont les valeurs par défaut. Pour GBNM la taille a du simplexe initial (cf. Equations (2.7) et (2.8)) est 5% de la plus petite dimension du domaine. Le paramètre qui contrôle l'étalement des Gaussiennes est $\alpha = 0.01$, et le nombre de points aléatoires qui contrôle l'erreur d'échantillonnage est $N_r = 10$.

Cinq problèmes sont résolus. Trois minimisations sans contraintes, la fonction de Rosenbrock en dimension 2, la fonction de Griewank en dimension 12, et la fonction de Fletcher et Powell en dimension 4, et deux minimisations sous contraintes (Section 3.3).

C.1 Fonction de Rosenbrock

La fonction de Rosenbrock bidimensionnelle a été présentée dans la Section 3.1. Son minimum est $f^* = 0$ en $x_1^* = x_2^* = 1$. Comme cette fonction est unimodale, la stratégie de comparaison adoptée consiste à confronter le nombre d'analyses requises pour chaque méthode pour atteindre le même ordre de grandeur dans la valeur de la fonction coût. *fmincon* est exécutée jusqu'à la convergence, et le nombre d'évaluations et la valeur de la fonction sont enregistrés. Puis GBNM est exécuté jusqu'à atteindre au moins le même ordre de grandeur dans la valeur de la fonction coût obtenue par *fmincon*. Une combinaison de trois points initiaux et de deux domaines différents sont testés. La Table C.1 présente les résultats pour $(x_1; x_2) \in [0; 20]$ et la Table C.2 pour $(x_1; x_2) \in [-20; 20]$. Parmi les six cas testés, *fmincon* trouve le minimum avec moins d'analyses que GBNM dans quatre cas, notamment ceux avec les bornes $[0; 20]^2$ où le simplexe tend à dégénérer sur les bornes (cf. Section 3.1). Pour les bornes $[-20; 20]^2$ GBNM atteint le minimum avec moins d'analyses que *fmincon* dans deux cas sur les trois testés.

Point initial	<i>fmincon</i>			GBNM		
	Nombre d'analyses	Point de convergence	Valeur de la fonction	Nombre d'analyses	Point de convergence	Valeur de la fonction
(0;0)	109	(1.0001;1.0001)	4.0365×10^{-9}	208	(0.9999;0.9999)	4.9944×10^{-9}
(5;5)	94	(1.0000;1.0000)	1.7658×10^{-10}	115	(1.0000;1.0000)	6.2415×10^{-10}
(10;10)	126	(1.0000;0.9999)	1.2032×10^{-8}	177	(1.0001;1.0001)	7.2092×10^{-9}

TAB. C.1 – Comparaison entre GBNM et *fmincon* : fonction de Rosenbrock, $(x_1; x_2) \in [0; 20]$.

Point initial	<i>fmincon</i>			GBNM		
	Nombre d'analyses	Point de convergence	Valeur de la fonction	Nombre d'analyses	Point de convergence	Valeur de la fonction
(0;0)	109	(1.0001;1.0001)	4.0365×10^{-9}	123	(1.0000;1.0001)	1.6889×10^{-9}
(5;5)	205	(1.0000;1.0000)	3.6216×10^{-9}	151	(1.0000;0.9999)	2.4406×10^{-9}
(10;10)	304	(1.0000;1.0000)	4.8639×10^{-10}	276	(1.0000;1.0000)	2.3444×10^{-10}

TAB. C.2 – Comparaison entre GBNM et *fmincon* : fonction de Rosenbrock, $(x_1; x_2) \in [-20; 20]$.

C.2 Fonction de Griewank en dimension n=12

La fonction de Griewank a été présentée dans la Section 3.2. Elle est fortement multimodale. Six essais sont réalisés, avec différents points initiaux. Comme cette fonction est multimodale.

dale, pour chaque essai on examine la valeur de la fonction objectif obtenue pour un certain nombre d'évaluations : d'abord le nombre d'analyses, identique pour les deux méthodes, est celui nécessaire à la convergence avec *fmincon*, puis, pour GBNM, le coût d'une recherche est fixé à 1000 évaluations de la fonction objectif. La Table C.3 présente les résultats. Pour les points initiaux testés, en considérant le nombre d'analyses nécessaires à la convergence de *fmincon*, *fmincon* trouve toujours un meilleur point que GBNM. Lorsque le coût de la recherche est augmenté jusqu'à 1000 évaluations, GBNM trouve des meilleures solutions que *fmincon* dans quatre cas sur les six testés.

Point initial	Nombre d'analyses (*)	Valeur de la fonction objectif		
		<i>fmincon</i>	GBNM	GBNM (1000 analyses)
(10; ...; 10)	630	-0.98768	-0.85531	-0.95081
(50; ...; 50)	371	-0.88719	-0.61844	-0.93005
(100; ...; 100)	328	-0.89522	-0.24641	-0.97739
(1000; ...; 1000)	446	-0.40979	4.4047	-0.91071
(-100; ...; -100)	492	-0.99383	-0.95056	-0.96721
(-1000; ...; -1000)	531	-0.70688	0.095532	-0.98632

(*) nombre d'analyses requises pour la convergence avec *fmincon*

TAB. C.3 – Comparaison entre GBNM et *fmincon* : fonction de Griewank.

C.3 Fonction de Fletcher et Powell de dimension n=4

La fonction de Fletcher et Powell est écrite comme ([5]),

$$f = \sum_{i=1}^n (A_i - B_i)^2 ,$$

$$\text{où } A_i = \sum_{j=1}^n (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j) \quad \text{et} \quad B_i = \sum_{j=1}^n (a_{ij} \sin x_j + b_{ij} \cos x_j) , \quad (\text{C.1})$$

$$-\pi \leq x_i \leq \pi , \quad i = 1, \dots, n.$$

Les paramètres suivants ont été choisis pour l'Equation (C.1) :

$$\begin{aligned}
 n &= 4, \\
 \alpha &= [0.435934 \quad 0.550595 \quad -1.283410 \quad -0.0734284], \\
 a &= \begin{bmatrix} -78 & 28 & 53 & -9 \\ 38 & 13 & -30 & 77 \\ -13 & -50 & -98 & 20 \\ -75 & 0 & -22 & -60 \end{bmatrix}, \quad b = \begin{bmatrix} 97 & -25 & -78 & -27 \\ -11 & -72 & 10 & 33 \\ 30 & 25 & -32 & -1 \\ 76 & 75 & 46 & 58 \end{bmatrix}.
 \end{aligned} \tag{C.2}$$

Cette fonction possède 2^n minima dans l'intervalle $-\pi \leq x_i \leq \pi$, $i = 1, n$, et parmi eux, plusieurs minima globaux $f^* = 0$. Cinq essais sont réalisés, en variant le point initial. D'abord le nombre d'analyses est identique pour les deux méthodes : *fmincon* est exécutée jusqu'à convergence, et le nombre d'analyses correspondant est utilisé comme coût de la recherche avec GBNM. Ensuite, pour GBNM, on vérifie quel est le nombre d'évaluations requises pour trouver au moins un des minima globaux. Les résultats sont présentés dans la Table C.4. On remarque que parmi les cinq points de départ différents, deux exécutions de *fmincon* trouvent un minimum global ($|f^*| < 1.0 \times 10^{-7}$). Quant à GBNM, avec 2000 évaluations et un point de départ quelconque parmi ceux essayés, on a une probabilité de 100% de trouver un minimum global. Les résultats de GBNM sont basés sur 100 exécutions indépendantes. Dans deux cas un minimum global est le premier point de convergence de GBNM si bien que moins de 2000 analyses sont nécessaires pour avoir f proche de zéro.

Point initial	Nombre d'analyses ⁽¹⁾	Valeur de la fonction objectif		GBNM
		<i>fmincon</i>	GBNM	Nombre d'analyses pour trouver un min. global
(0; 0; 0; 0)	129	19.4456	21.0265	2000 ⁽²⁾
(1; 1; 1; 1)	167	4.2080×10^{-9}	19.4581	2000 ⁽²⁾
(2; 2; 2; 2)	98	$2.1351 \times 10^{+4}$	$2.1566 \times 10^{+4}$	2000 ⁽²⁾
(-1; -1; -1; -1)	140	3.2167×10^{-8}	2.2666×10^{-2}	238 ⁽³⁾
(-2; -2; -2; -2)	144	54.2261	5.4367×10^{-5}	192 ⁽³⁾

⁽¹⁾ nombre d'analyses requises pour la convergence avec *fmincon*

⁽²⁾ basé sur 100 exécutions indépendantes, où plusieurs minima sont trouvés

⁽³⁾ le premier point de convergence est un minimum global ($|f^*| < 1.0 \times 10^{-7}$)

TAB. C.4 – Comparaison entre *fmincon* et GBNM : fonction de Fletcher et Powell.

C.4 Fonction Test 1 sous contraintes

Dans ce test la minimisation sous contraintes à effectuer est celle de l'Equation (3.5) où le minimum global du problème est $f^* = -0.0958248$ en $x_1^* = 1.2280$ et $x_2^* = 4.2454$ (cf. Section 3.3).

Cinq points de départ différents sont testés. La Table C.5 résume les résultats obtenus avec *fmincon*. Tous les points de convergence sont des points admissibles. Cependant le minimum global n'est trouvé que si le point initial en est très proche (point initial $(x_1; x_2) = (1; 4.5)$). La Table C.6 donne le nombre d'évaluations pour trouver le minimum global avec GBNM. Deux jeux de coefficients de pénalisation constants sont employés : $\lambda_A = (20; 20)$ et $\lambda_B = (5.5; 98.4)$. Ce dernier a été déterminé par adaptation des λ , i.e., par résolution numérique du problème dual (cf. Section 3.3). En utilisant les coefficients λ_A , dans quatre cas sur les six testés le premier point de convergence est le minimum global. Pour les deux autres cas, le premier point de convergence n'est pas le minimum global, mais avec un coût de 1500 évaluations plusieurs minima sont localisés dont au moins un est le minimum global. Avec le jeu de coefficients λ_B le premier point de convergence est le minimum global dans cinq cas sur six. Pour le sixième cas (point de départ $(10; 10)$), avec un coût de recherche de 500 évaluations, plusieurs minima sont obtenus, dont le minimum global. Les comparaisons à λ constants (et bien choisis) sont par définition favorables à GBNM qui n'a rien besoin d'apprendre pour prendre en compte les contraintes. L'intérêt de ces tests est de montrer que le choix de λ ne paraît pas trop critique, ce qui est une propriété de la pénalisation linéaire \bar{L} (cf. Annexe 12).

Point initial	Nombre d'analyses	Point de convergence	Valeur de la fonction coût
(1;1)	23	(1.5051;3.2893)	1.9635×10^{-6}
(2;2)	62	(1.3244;3.4304)	-0.027263
(5;5)	23	(2.0000;5.0000)	-4.8754×10^{-37}
(10;10)	53	(1.7342;4.7460)	-0.029144
(2;4)	34	(1.6740;3.8023)	-0.025812
(1;4.5)	36	(1.2280;4.2454)(*)	-0.095825

(*) optimum global

TAB. C.5 – Résultats obtenus avec *fmincon*. Minimisation sous contraintes de la fonction Test 1.

Point initial	Nombre d'analyses	
	λ_A	λ_B
(1;1)	1500 ⁽¹⁾	93 ⁽²⁾
(2;2)	74 ⁽²⁾	71 ⁽²⁾
(5;5)	72 ⁽²⁾	74 ⁽²⁾
(10;10)	1500 ⁽¹⁾	500 ⁽¹⁾
(2;4)	61 ⁽²⁾	59 ⁽²⁾
(1;4.5)	65 ⁽²⁾	61 ⁽²⁾

⁽¹⁾ plusieurs minima sont trouvés,
avec une probabilité de 100% de
trouver le minimum global
(basé sur 100 exécutions indépendantes)
⁽²⁾ le premier point de convergence est
le minimum global

TAB. C.6 – Nombre d'évaluations nécessaires pour trouver le minimum global avec GBNM. Minimisation sous contraintes de la fonction Test 1.

C.5 Fonction Test 2 sous contraintes

Le problème de la minimisation de la fonction Test 2 sous contraintes a été présenté dans la Section 3.3 (cf. Equation (3.6)). Le minimum global est $f^* = 680.6300573$, atteint en $x^* = (2.330499 ; 1.951372 ; -0.4775414 ; 4.365726 ; -0.624487 ; 1.038131 ; 1.594227)$.

Cinq points de départ différents sont testés. D'abord *fmincon* est exécuté jusqu'à convergence et la valeur de la fonction objectif et le nombre d'évaluations sont enregistrés. Puis GBNM, avec des coefficients de pénalisation constants, est exécuté en fixant comme nombre d'analyses celui de la convergence avec *fmincon*. Enfin, GBNM est étudié avec un coût de recherche fixé à 1000 évaluations. Deux jeux de coefficients de pénalisation sont employés : $\lambda_A = (20; 20; 20; 20)$ et $\lambda_B = (68.5; 26.0; 5.2; 3.8)$. Ce dernier a été obtenu par la résolution numérique du problème dual (cf. Section 3.3).

La Table C.7 récapitule les valeurs de fonctions coûts et le nombre d'analyses associé. Dans cet exemple, *fmincon* trouve toujours le minimum global. En utilisant le même nombre d'analyses nécessaires pour la convergence avec *fmincon*, GBNM s'approche du minimum global mais n'atteint pas la convergence. Quand le coût de la recherche est augmenté jusqu'à 1000 analyses, GBNM améliore les solutions trouvées, mais n'atteint pas la même précision que *fmincon*.

Il est vraisemblable que la nature polynomiale de la fonction objectif et des contraintes convienne bien aux approximations quadratiques de SQP. De plus, ce problème ne semble pas

avoir d'optima locaux. Enfin, 1000 analyses peuvent ne pas être suffisantes pour que GBNM converge pleinement dans un espace à 7 variables.

Point initial	Nombre d'analyses	Fonction coût			1000 analyses	
		$fmincon$	GBNM (λ_A)	GBNM (λ_B)	GBNM (λ_A)	GBNM (λ_B)
(0; ... ; 0)	197	680.6301	683.1407	683.9268	681.3490	682.1689
(1; ... ; 1)	144	680.6301	700.5384	691.3390	682.1079	680.8656
(5; ... ; 5)	485	680.6301	688.7653	696.8413	686.4749	683.3205
(10; ... ; 10)	511	680.6301	751.7660	720.6969	686.4749	708.2116
(-10; ... ; -10)	540	680.6301	705.1305	709.7181	682.3187	685.9200

TAB. C.7 – Comparaison entre $fmincon$ et GBNM. Minimisation de la fonction Test 2 sous contraintes.

Annexe D

Présence de minima locaux en conception de composites - Exemples particuliers

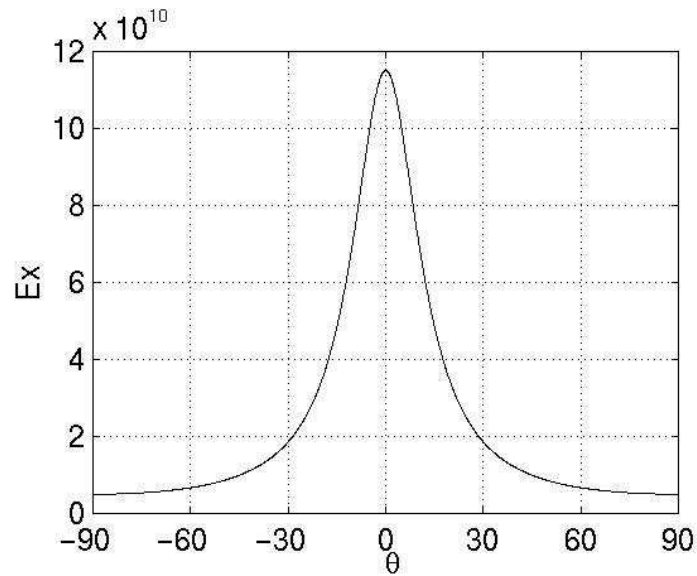
Dans cette annexe il est montré comment des critères aussi simples que les modules d'élasticité E_x et E_y des composites stratifiés à une et deux couches peuvent présenter des optima locaux.

Les Figures D.1 et D.2 montrent, pour une couche, la variation du module d'élasticité E_x , en fonction de l'angle des fibres, θ , pour les matériaux carbone-époxyde et verre-époxyde, dont les propriétés élastiques sont présentées dans la Table D.1. Les équations pour l'obtention des modules d'élasticité effectifs d'un stratifié à partir des modules des plis et en fonction des orientations des fibres peuvent être trouvées dans les références [33] et [29]. On remarque que pour les deux matériaux le maximum global se trouve à 0 degré. On remarque aussi que pour le carbone-époxyde le minimum de E_x se trouve à -90 et 90 degrés, et pour le verre-époxyde vers 78 et -78 degrés. En outre, pour le verre-époxyde il y a un maximum local de E_x à $\theta = 90^\circ$, et comme la courbe est symétrique, un autre maximum local se trouve à $\theta = -90^\circ$. Un optimiseur local, comme Nelder-Mead, peut s'arrêter sur ces optima locaux. La Figure D.3 montre le comportement de E_y pour une couche verre-époxyde. On remarque que les maxima globaux de E_y sont à 90 et -90° , mais il y a un maximum local à 0° . Pour le carbone-époxyde il n'y a pas de tels maxima locaux.

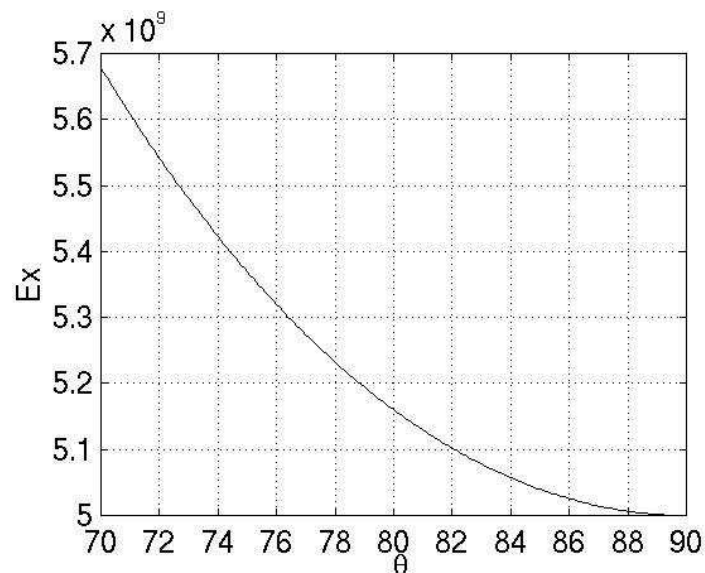
La Figure D.4 montre la variation de E_x pour un composite à deux couches de carbone-époxyde, en fonction des angles d'orientation des fibres $[\theta_1/\theta_2]$. Il y a un unique maximum (global),

Propriété	carbone-époxyde	verre-époxyde
Module d'élasticité E_1	$115 \times 10^9 Pa$	$45 \times 10^9 Pa$
Module d'élasticité E_2	$5 \times 10^9 Pa$	$10 \times 10^9 Pa$
Module de cisaillement G_{12}	$5 \times 10^9 Pa$	$4.5 \times 10^9 Pa$
Coefficient de Poisson ν_{12}	0.35	0.31

TAB. D.1 – Propriétés élastiques de deux composites unidirectionnels, selon [7].

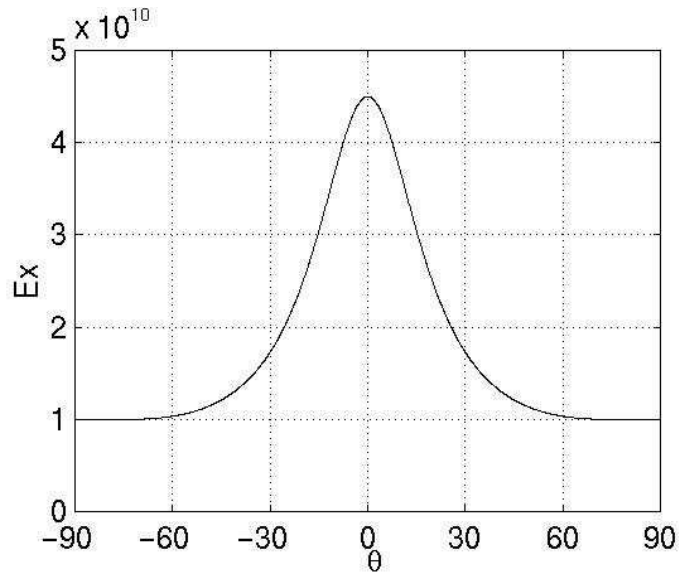


(a) Variation de l'angle θ : -90 à 90 degrés.

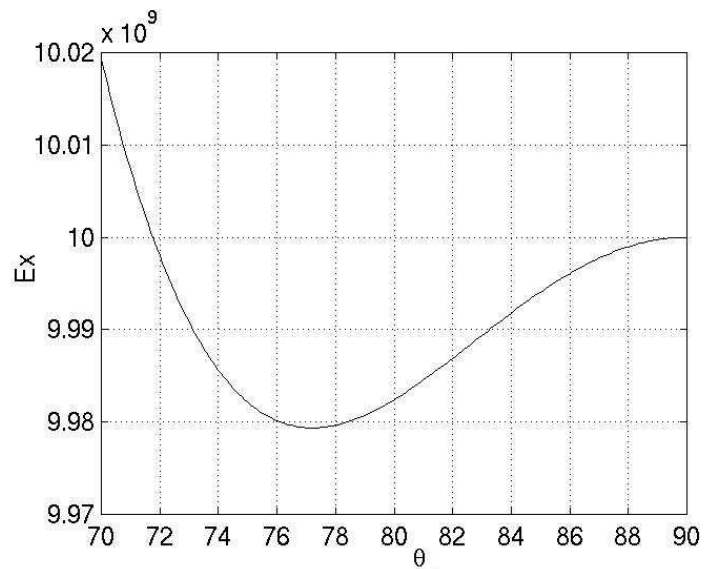


(b) Variation de l'angle θ : 70 à 90 degrés.

FIG. D.1 – Module E_x en fonction de θ pour le composite unidirectionnel à fibres de carbone. E_x présente un unique maximum (global), positionné à 0° .

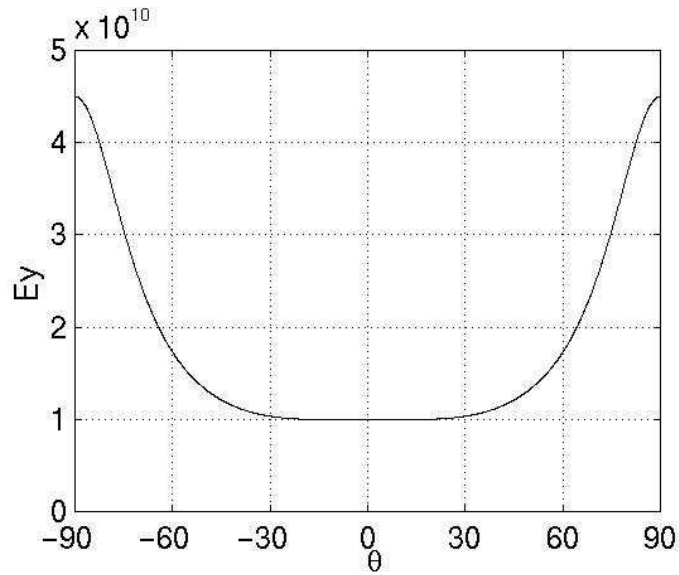


(a) Variation de l'angle θ : -90 à 90 degrés.

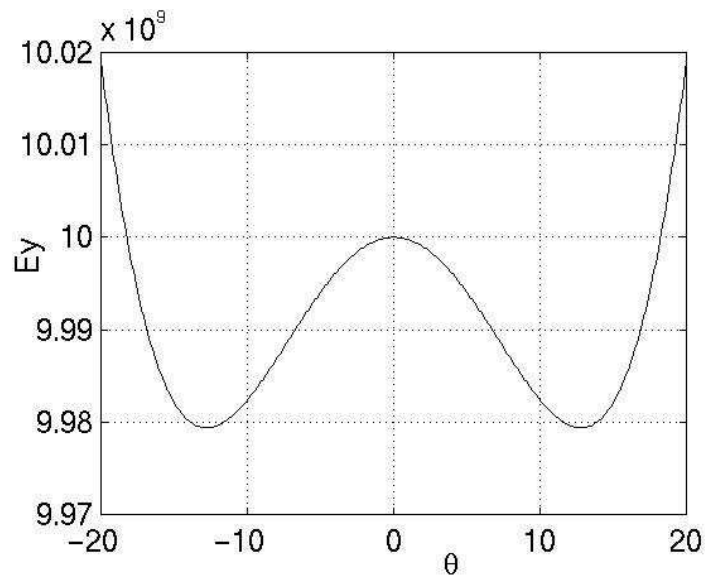


(b) Variation de l'angle θ : 70 à 90 degrés.

FIG. D.2 – Module E_x en fonction de θ pour le composite unidirectionnel à fibres de verre. E_x présente un maximum global à 0° et deux maxima locaux en -90° et 90° .



(a) Variation de l'angle θ : -90 à 90 degrés.

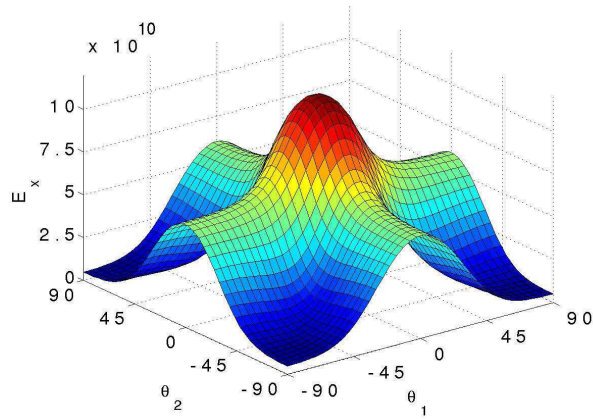


(b) Variation de l'angle θ : 20 à 20 degrés.

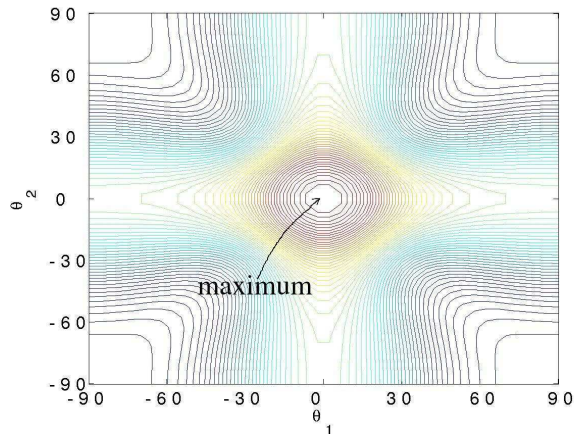
FIG. D.3 – Module E_y en fonction de θ pour le composite unidirectionnel à fibres de verre. E_y présente deux maxima globaux, en -90 et en 90 degrés, et un maximum local en 0 degré.

positionné en $[0/0]$ degrés.

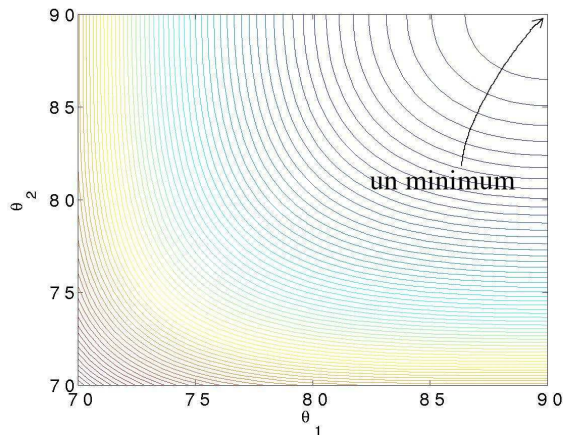
La Figure D.5 montre la variation de E_x en fonction de $[\theta_1/\theta_2]$ pour un composite à deux couches de verre-époxyde. Pour ce stratifié le maximum global se trouve aussi en $[0/0]$ et il y a huit maxima locaux : $[0/90]$, $[0/-90]$, $[90/0]$, $[-90/0]$, $[90/90]$, $[90/-90]$, $[-90/90]$ et $[-90/-90]$. Notons de plus que quatre minima globaux se trouvent vers $[78/78]$, $[78/-78]$, $[-78/78]$ et $[-78/-78]$. La Figure D.5(c) montre le minimum en $[78/78]$. Les autres minima sont positionnés de manière symétrique.



(a) Variation des angles de -90 à 90 degrés.

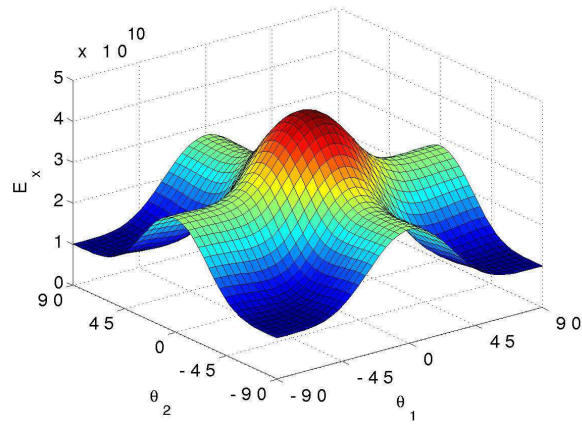


(b) Courbes de niveau : variation des angles de -90 à 90 degrés.

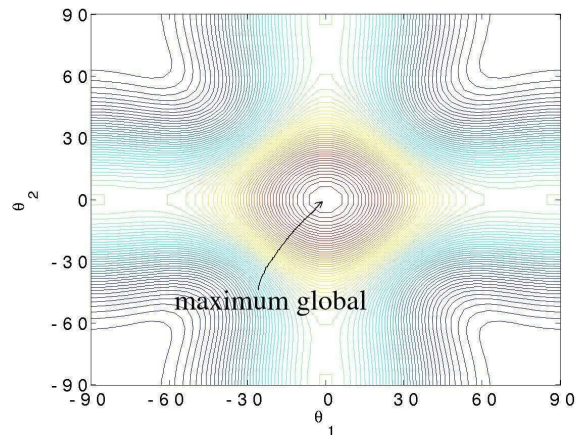


(c) Agrandissement des courbes de niveau.

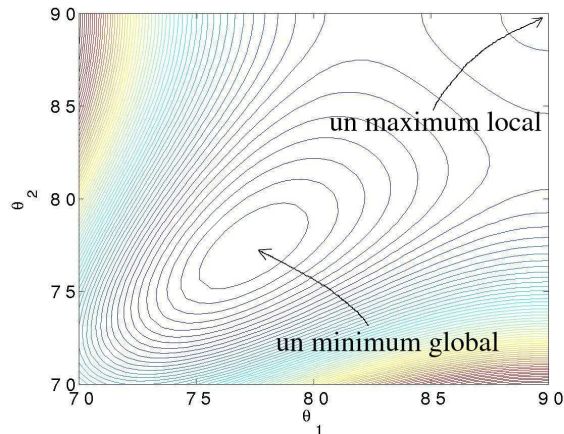
FIG. D.4 – Module E_x en fonction de $[\theta_1/\theta_2]$ pour un stratifié à deux couches en carbone-époxyde. E_x présente un unique maximum (global), en $[0^\circ/0^\circ]$, et plusieurs minima.



(a) Variation des angles de -90 à 90 degrés.



(b) Courbes de niveau : variation des angles de -90 à 90 degrés.



(c) Agrandissement des courbes de niveau, en montrant un maximum local et un minimum de E_x .

FIG. D.5 – Module E_x en fonction de $[\theta_1/\theta_2]$ pour un stratifié à deux couches en verre-époxyde. E_x présente un maximum global, en $[0^\circ/0^\circ]$, plusieurs maxima locaux, et plusieurs minima (tous globaux).

Annexe E

Un modèle d'écoulement parfait et instationnaire autour d'un corps mince

Dans cet Annexe un modèle pour l'écoulement plan autour d'un obstacle de faible épaisseur et à faible incidence est présenté. Dans l'approximation d'un fluide **incompressible non visqueux**, si le champ incident est irrotationnel, la vorticit  est nulle, dans le domaine fluide, sauf au voisinage de l'obstacle et dans le sillage. L'application de la d composition de Helmholtz au champ de vitesse permet de distinguer une contribution potentielle d'une contribution rotationnelle. Ce probl me a  t  intensivement  tudi  et sa r solution associe classiquement une m thode de surface portante et un traitement lagrangien du sillage (m thode particulaire) ([37]). L'int r t pratique de ces m thodes est de formuler le probl me sur les fronti res de l' coulement et le sillage, trait  comme une simple ligne en 2D, et donc de ne pas avoir   discr tiser l'ensemble du domaine fluide   l'aide d'un maillage.

E.1 Ecoulement potentiel

La d composition de Helmholtz conduit    crire le champ de vitesse sous la forme :

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}_\infty + \nabla\Phi + \nabla \wedge \Psi \quad (\text{E.1})$$

avec \mathbf{U}_∞ le champ incident (la vitesse d'avance du nageur), Φ le potentiel des vitesses de perturbation induit par les obstacles, et Ψ le potentiel vecteur contenant la perturbation tourbillonnaire du sillage et des couches limites. Dans le cas plan, le potentiel vecteur se r duit   sa

composante normale : $\Psi = \Psi \mathbf{z}$. On montre que, dans la limite d'un obstacle d'épaisseur nulle, l'obstacle est une surface présentant une discontinuité du potentiel des vitesses que l'on notera $\mu(s, t) \equiv \Phi^+(s, t) - \Phi^-(s, t)$. Les conditions aux limites cinématiques pour un fluide parfait sont des conditions de glissement :

$$\mathbf{U}(\mathbf{X}(s, t)) \cdot \mathbf{n}(s, t) = (\dot{\mathbf{X}}(s, t)) \cdot \mathbf{n}(s, t) , \quad (\text{E.2})$$

avec ici \mathbf{X} un point de l'obstacle, $\dot{\mathbf{X}}$ la vitesse locale de déplacement de l'obstacle (dans le repère de l'étude) et \mathbf{n} la normale unitaire à la surface de l'obstacle. Le champ de vitesse induit par une distribution de sauts de potentiel sur une surface d'équation $\mathbf{X}(s, t)$, s étant l'abscisse curviligne naturelle sur la surface, i.e., $\|\partial \mathbf{X} / \partial s\| = 1$, et où $s = L$ dénote le bord de fuite de l'obstacle, à l'instant t est donné par la relation intégrale suivante (voir [4]) :

$$\mathbf{U}_\mu(\mathbf{x}) = \nabla \Phi = \frac{1}{4\pi} \int_0^L \mu(s) \nabla \left(\frac{(\mathbf{x} - \mathbf{X}(s)) \cdot \mathbf{n}(\mathbf{X}(s))}{\|\mathbf{x} - \mathbf{X}(s)\|^3} \right) ds. \quad (\text{E.3})$$

La condition de glissement (E.2) s'appliquant en chaque point de la surface d'abscisse l , elle permet, avec l'expression de la vitesse \mathbf{U}_μ d'établir une équation intégrale (dite de Fredholm de seconde espèce) pour la distribution de sauts de potentiel $\mu(s)$:

$$\begin{aligned} \int_0^L \mu(s) \mathbf{n}(l) \nabla \left(\frac{(\mathbf{X}(l) - \mathbf{X}(s)) \cdot \mathbf{n}(\mathbf{X}(s))}{\|\mathbf{X}(l) - \mathbf{X}(s)\|^3} \right) ds \\ = 4\pi \left(-U_\infty - \nabla \wedge \Psi + \dot{\mathbf{X}}(l) \right) \cdot \mathbf{n}(l) \end{aligned} \quad (\text{E.4})$$

Cette équation est classiquement calculée en discrétisant l'obstacle en N segments (correspondants dans la pratique aux éléments finis du calcul de structure) supports d'un saut de potentiel constant ("piecewise approximation", [37]). L'écriture de Eq. (E.4) discrétisée sur les N centres géométriques des segments permet de construire un système linéaire pour les $\mu_{i=1,N}$. La matrice de ce système (matrice d'influence) est linéaire en $\dot{\mathbf{X}}$ et non linéaire en \mathbf{X} . Le membre de droite contient le terme $\nabla \wedge \Psi$ qui est la contribution tourbillonnaire du sillage à la vitesse dont l'expression est donnée en Equation (E.11).

E.2 Sillage

Le théorème de Kelvin indique que la circulation totale dans le fluide est un invariant du problème. Aussi, toute variation de la circulation autour de l'obstacle est nécessairement compensée par l'émission de vorticit  dans le sillage. Si on suppose de plus que l' coulement reste attach  tout le long de la surface de l'obstacle pour s'en d tacher au bord de fuite, ce qui exclut les situations de d crochage (statiques pour les grandes incidences et / ou dynamiques du fait de fortes acc l rations de l'obstacle), la variation de la circulation Γ autour de l'obstacle est donn e par :

$$\frac{d\Gamma}{dt} = \frac{d\mu(s=L)}{dt} . \quad (\text{E.5})$$

E.2.1 Emission

L' mission de la vorticit  dans le sillage a donc lieu au bord de fuite de l'obstacle et la circulation totale Ω_e  mise dans l'intervalle de temps $[t; t + \Delta t]$ est donc

$$\Omega_e(t + \Delta t) = - \int_t^{t+\Delta t} \frac{d\Gamma}{dt} dt = \mu(L, t) - \mu(L, t + \Delta t) . \quad (\text{E.6})$$

Le support de cette circulation est la portion de ligne d' mission issue du bord de fuite entre t et $t+\Delta t$. Cette repr sentation permet de satisfaire d'une part le th or me de Kelvin et d'autre part la condition de Kutta-Joukowski. La m thode particulaire consiste en une repr sentation du sillage par des particules discr tes supportant une circulation. Cette m thode est bien adapt e pour traiter le cas de sillages enroul s. La circulation est introduite dans le fluide par une proc dure au cours de laquelle une particule de position \mathbf{X}_w supportant la circulation $\Omega_w(n\Delta t) = \Omega_e(n\Delta t)$ est plac e dans le fluide au point \mathbf{X}_e au temps $n\Delta t$:

$$\mathbf{X}_e = \mathbf{X}(L, n\Delta t) + \mathbf{U}(\mathbf{X}(L, n\Delta t)) \frac{\Delta t}{2} . \quad (\text{E.7})$$

E.2.2 Evolution du sillage

En  crivant les  quations d'Euler en formulation vitesse-tourbillon, on montre que l' volution du champ tourbillonnaire conduit   r soudre le syst me d' quations suivant pour l'ensemble des

particules :

$$\frac{d\Omega_w}{dt} = 0 \quad , \quad \frac{d\mathbf{X}_w}{dt} = \mathbf{U}(\mathbf{X}_w) \quad . \quad (\text{E.8})$$

Les conditions initiales sont données par :

$$\Omega_w(n\Delta t, t = n\Delta t) = \Omega_e(n\Delta t) \quad , \quad \mathbf{X}_w(n\Delta t, t = n\Delta t) = \mathbf{X}_e(n\Delta t) \quad . \quad (\text{E.9})$$

Ainsi, la vorticit  de chaque particule  mise reste constante,

$$\Omega_w(n\Delta t, t) = \Omega_e(n\Delta t) \quad . \quad (\text{E.10})$$

La vitesse induite au temps $n\Delta t$ par l'ensemble des particules du sillage est obtenue au moyen de la loi de Biot et Savart :

$$\mathbf{U}_w(\mathbf{x}) = \nabla \wedge \Psi = \sum_{n=1}^m \Omega_w(n\Delta t) \wedge \nabla \left(\frac{1}{\|\mathbf{x} - \mathbf{X}_w(n\Delta t)\|} \right) \quad . \quad (\text{E.11})$$

E.3 R solution du probl me fluide

Le probl me fluide est non lin aire puisque la circulation autour de l'obstacle et la position d' mission ne sont pas reli es par une relation lin aire. Un processus it ratif est mis en  uvre :   partir du sillage et de la configuration de l'obstacle connus   $t + \Delta t$, l' quation int grale (E.4) est r solue fournissant la circulation et la position d'une nouvelle particule selon les Equations (E.6) et (E.7). En consid rant cette nouvelle particule dans le sillage, on r sout de nouveau (E.4) calculant ainsi une autre estimation pour \mathbf{X}_e et Ω_e . Le processus est renouvel  tant qu'un crit re de convergence n'est pas atteint. Ceci fait, l'ensemble des particules est convect  selon (E.8) et une nouvelle it ration temporelle peut commencer.

E.4 Efforts exerc s par le fluide

En l'absence de viscosit , le fluide n'exerce qu'un effort de pression normal sur l'obstacle. Cet effort de pression est donn  par la relation de Bernoulli instationnaire,  crite ici sous sa forme normalis e :

$$\mathbf{F}(s, t) = \left(\frac{\partial \mu(s, t)}{\partial t} + \mathbf{U}(\mathbf{X}(s)) \cdot \nabla \mu(s) \right) \mathbf{n}(s) ds \quad . \quad (\text{E.12})$$

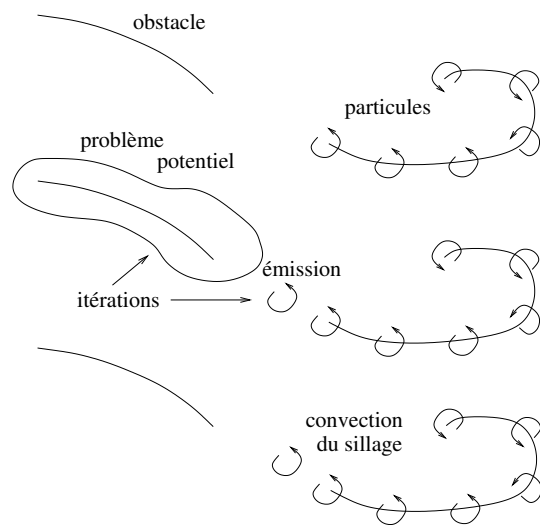


FIG. E.1 – Représentation schématique du problème fluide.

Annexe F

Equations des n barres liées élastiquement

Les équations du mouvement du système de barres de la Figure 7.3 sont calculées par application des équations de Lagrange. L'effet de la pesanteur sur la palme est négligé. L'énergie potentielle de la palme est

$$E_p = \frac{1}{2} \sum_{i=2}^n C_i \theta_i^2 . \quad (\text{F.1})$$

L'énergie cinétique de translation de l'ensemble de barres est

$$E_{cT} = \frac{1}{2} \rho_L \left(\sum_{i=1}^n l_i \right) (x_1^2 + y_1^2) . \quad (\text{F.2})$$

Une barre i est en rotation autour de P à la vitesse $\sum_{j=1}^i \dot{\theta}_j$. D'après le théorème de Koenig, son énergie cinétique est la somme de l'énergie cinétique de rotation autour de son centre de gravité G_i et l'énergie cinétique de translation de sa masse appliquée en G_i ,

$$E_{cRi} = \frac{1}{2} I_{G_i} \left(\sum_{j=1}^i \dot{\theta}_j \right)^2 + \frac{1}{2} (\rho_L l_i) v_{G_i}^2 , \quad (\text{F.3})$$

où

$$I_{G_i} = \frac{\rho_L l_i^3}{12} . \quad (\text{F.4})$$

La vitesse en G_i est calculée à partir de sa position,

$$\begin{cases} x_{G_i} = \sum_{j=1}^{i-1} [l_j \cos(\sum_{k=1}^j \theta_k)] + \frac{l_i}{2} \cos(\sum_{k=1}^i \theta_k) \\ y_{G_i} = \sum_{j=1}^{i-1} [l_j \sin(\sum_{k=1}^j \theta_k)] + \frac{l_i}{2} \sin(\sum_{k=1}^i \theta_k) \end{cases} \quad (\text{F.5})$$

que l'on dérive par rapport au temps,

$$\begin{cases} \dot{x}_{G_i} = \sum_{j=1}^{i-1} [-l_j (\sum_{k=1}^j \dot{\theta}_k) \sin(\sum_{k=1}^j \theta_k)] - \frac{l_i}{2} (\sum_{k=1}^i \dot{\theta}_k) \sin(\sum_{k=1}^i \theta_k) \\ \dot{y}_{G_i} = \sum_{j=1}^{i-1} [l_j (\sum_{k=1}^j \dot{\theta}_k) \cos(\sum_{k=1}^j \theta_k)] + \frac{l_i}{2} (\sum_{k=1}^i \dot{\theta}_k) \cos(\sum_{k=1}^i \theta_k) . \end{cases} \quad (\text{F.6})$$

La vitesse en G_i est,

$$v_{G_i}^2 = \dot{x}_{G_i}^2 + \dot{y}_{G_i}^2 . \quad (\text{F.7})$$

Globalement, l'énergie cinétique de la palme est

$$E_c = E_{cT} + \sum_{i=1}^n E_{cRi} . \quad (\text{F.8})$$

Le système de barres est soumis aux efforts extérieurs suivants : les forces et le couple d'encastrement en 1 qui sont produits par le nageur, F_{x1} , F_{y1} et M_1 , respectivement ; les forces en 1 et les couples en chaque articulation produits par le fluide, F_{x1f} , F_{y1f} , $M_{if}, i = 1, n$, respectivement.

Soient f_i^x et f_i^y les forces fluides s'exerçant horizontalement et verticalement sur le i -ième segment de palme (de coordonnées initiales (x_i, y_i)), on a,

$$F_{x1f} = \sum_{i=1}^n f_i^x , \quad (\text{F.9})$$

$$F_{y1f} = \sum_{i=1}^n f_i^y , \quad (\text{F.10})$$

$$M_{if} = \sum_{j=i}^n [f_j^y (\bar{x}_j - x_i) - f_j^x (\bar{y}_j - y_i)] , \quad (\text{F.11})$$

où (\bar{x}_j, \bar{y}_j) sont les coordonnées du point d'application des résultantes des forces sur la barre j .

Les équations de Lagrange sont écrites ci-après, pour les degrés de liberté du système,

$$\frac{d}{dt} \frac{\partial E_c}{\partial \dot{x}_1} - \frac{\partial E_c}{\partial x_1} + \frac{\partial E_p}{\partial x_1} = F_{x_{1f}} + F_{x_1} \quad (\text{F.12})$$

$$\frac{d}{dt} \frac{\partial E_c}{\partial \dot{y}_1} - \frac{\partial E_c}{\partial y_1} + \frac{\partial E_p}{\partial y_1} = F_{y_{1f}} + F_{y_1} \quad (\text{F.13})$$

$$\frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}_1} - \frac{\partial E_c}{\partial \theta_1} + \frac{\partial E_p}{\partial \theta_1} = M_{1f} + M_1 \quad (\text{F.14})$$

$$\frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}_r} - \frac{\partial E_c}{\partial \theta_r} + \frac{\partial E_p}{\partial \theta_r} = M_{rf} \quad r = 2, n. \quad (\text{F.15})$$

En remplaçant les énergies par leurs expressions dans les Equations (F.12) à (F.15), on obtient les équations du mouvement (7.6) à (7.9).

Dans les équations du mouvement, $x_1(t)$, $\dot{x}_1(t)$, $\ddot{x}_1(t)$, $y_1(t)$, $\dot{y}_1(t)$, $\ddot{y}_1(t)$, $\theta_1(t)$, $\dot{\theta}_1(t)$ et $\ddot{\theta}_1(t)$ sont connus (mouvement du point qui représente le pied du nageur). Les efforts fluides $F_{x_{1f}}$, $F_{y_{1f}}$ et M_{if} , $i = 1, n$ sont des fonctions de $x_1(t)$, $\dot{x}_1(t)$, $y_1(t)$, $\dot{y}_1(t)$, $\theta_i(t)$, $\dot{\theta}_i(t)$, $i = 1, n$ qui définissent les positions et vitesses des points de la palme. Les inconnues des équations du mouvement sont $\theta_i(t)$, $\dot{\theta}_i(t)$, $\ddot{\theta}_i(t)$, $i = 2, n$, $F_{x_1}(t)$, $F_{y_1}(t)$ et $M_1(t)$. Le schéma utilisé pour résoudre les équations est présenté en Annexe G.

Annexe G

Intégration des équations dynamiques de la palme

Pour résoudre les équations non-linéaires (7.9) dont les inconnues sont $\theta_i(t + \Delta t)$, $\dot{\theta}_i(t + \Delta t)$ et $\ddot{\theta}_i(t + \Delta t)$, $i = 2, n$, on utilise le schéma temporel implicite de Newmark couplé à une méthode itérative de Newton-Raphson ([27]) et GBNM. L'état à l'instant t , $\theta_i(t)$, $\dot{\theta}_i(t)$ et $\ddot{\theta}_i(t)$, est supposé connu.

L'équation non-linéaire d'équilibre (7.9) est ré-écrite sous la forme d'un résidu,

$$R_i = \frac{d}{dt} \frac{\partial E_c}{\partial \dot{\theta}_i} - \frac{\partial E_c}{\partial \theta_i} + \frac{\partial E_p}{\partial \theta_i} - M_{if} \quad i = 2, n. \quad (\text{G.1})$$

Le schéma temporel de Newmark lie les degrés de libertés temporels par les formules

$$\dot{\theta}_i(t + \Delta t) = \dot{\theta}_i(t) + [(1 - \gamma)\ddot{\theta}_i(t) + \gamma\ddot{\theta}_i(t + \Delta t)]\Delta t, \quad (\text{G.2})$$

$$\theta_i(t + \Delta t) = \theta_i(t) + \dot{\theta}_i(t)\Delta t + [(\frac{1}{2} - \beta)\ddot{\theta}_i(t) + \beta\ddot{\theta}_i(t + \Delta t)]\Delta t^2, \quad (\text{G.3})$$

où γ et β sont les paramètres de la quadrature. Les valeurs les plus utilisées sont $\gamma = 0.5$ et $\beta = 0.25$, ce qui correspond à adopter, dans l'intervalle de temps, une valeur moyenne de l'accélération, et ce qui donne une stabilité inconditionnelle à l'intégration.

Une valeur de $\ddot{\theta}_i(t + \Delta t)$ est choisie lors des itérations de résolution de (G.1). A la première itération d'équilibre, la prédiction est $\ddot{\theta}_i(t + \Delta t) = 0$. $\dot{\theta}_i(t + \Delta t)$ et $\theta_i(t + \Delta t)$ résultent des équations (G.2) et (G.3). Si ces valeurs ne satisfont pas l'équation (G.1), une correction $\Delta\theta_i$ est

appliquée à $\theta_i(t + \Delta t)$ qui provient de la formule de Newton-Raphson,

$$\mathbf{S}(t + \Delta t)\Delta\Theta = -\mathbf{R}(t + \Delta t) , \quad (\text{G.4})$$

où $\Delta\Theta$ est le vecteur des corrections $\Delta\theta_i$, $i = 2, n$, \mathbf{R} le vecteur résidu, et \mathbf{S} la matrice jacobienne :

$$\mathbf{S} = \frac{d\mathbf{R}}{d\Theta} . \quad (\text{G.5})$$

Les corrections de la vitesse et de l'accélération découlent du schéma de Newmark (équations (G.2) et (G.3)) en remplaçant θ_i par $\theta_i + \Delta\theta_i$, $\dot{\theta}_i$ par $\dot{\theta}_i + \Delta\dot{\theta}_i$ et $\ddot{\theta}_i$ par $\ddot{\theta}_i + \Delta\ddot{\theta}_i$:

$$\begin{aligned} \Delta\dot{\theta}_i &= \frac{\gamma}{\beta\Delta t}\Delta\theta_i , \\ \Delta\ddot{\theta}_i &= \frac{1}{\beta\Delta t^2}\Delta\theta_i . \end{aligned} \quad (\text{G.6})$$

Après correction de l'état, les itérations sont poursuivies à un même instant tant que le résidu n'est pas inférieur à un seuil de précision :

$$\|\mathbf{R}(t + \Delta t)\| < \epsilon\|\mathbf{M}_f(t + \Delta t)\| . \quad (\text{G.7})$$

M_{if} est calculé à chaque itération d'équilibre par un appel du code fluide. Après convergence dans le pas de temps, on utilise les équations (7.6), (7.7) et (7.8) pour calculer les efforts F_{x1} , F_{y1} et M_1 .

La méthode de Newton-Raphson n'est pas forcément convergente. C'est pourquoi une régularisation du type Levenberg-Marquardt a été faite. En conséquence l'Equation (G.4) est remplacée par,

$$(\mathbf{S}^T\mathbf{S} + \lambda\mathbf{I}) = -\mathbf{S}^T\mathbf{R} , \quad (\text{G.8})$$

où λ est le paramètre de régularisation. En outre, pour que l'algorithme soit plus robuste, quand les corrections de Newton-Raphson sont infructueuses, la méthode d'optimisation GBNM est appelée pour minimiser le résidu.

La Figure G.1 présente l'organigramme de la résolution des équations. En pratique, la matrice

jacobienne \mathbf{S} est calculée numériquement par différences finies en variant les θ_j :

$$S_{ij} = \frac{dR_i}{d\theta_j} \approx \frac{R_i(\dots, \theta_j + \delta\theta_j, \dot{\theta}_j + \delta\dot{\theta}_j, \ddot{\theta}_j + \delta\ddot{\theta}_j, \dots) - R_i(\dots, \theta_j, \dot{\theta}_j, \ddot{\theta}_j, \dots)}{\delta\theta_j} \quad (\text{G.9})$$

où les variations sur $\dot{\theta}_j$ et $\ddot{\theta}_j$ sont calculées par le schéma de Newmark à partir des variations en θ_j ,

$$\begin{aligned} \delta\dot{\theta}_j &= \frac{\gamma}{\beta\Delta t} \delta\theta_j \\ \delta\ddot{\theta}_j &= \frac{1}{\beta\Delta t^2} \delta\theta_j . \end{aligned} \quad (\text{G.10})$$

On remarque que comme le résidu R_i est fonction du moment fluide M_{if} (cf. Equation G.1), le calcul fluide est, lui aussi, perturbé dans l'évaluation de (G.9).

Equation du mouvement : $E(\theta, \dot{\theta}, \ddot{\theta}) = M(\theta, \dot{\theta})$

M = moment fluide

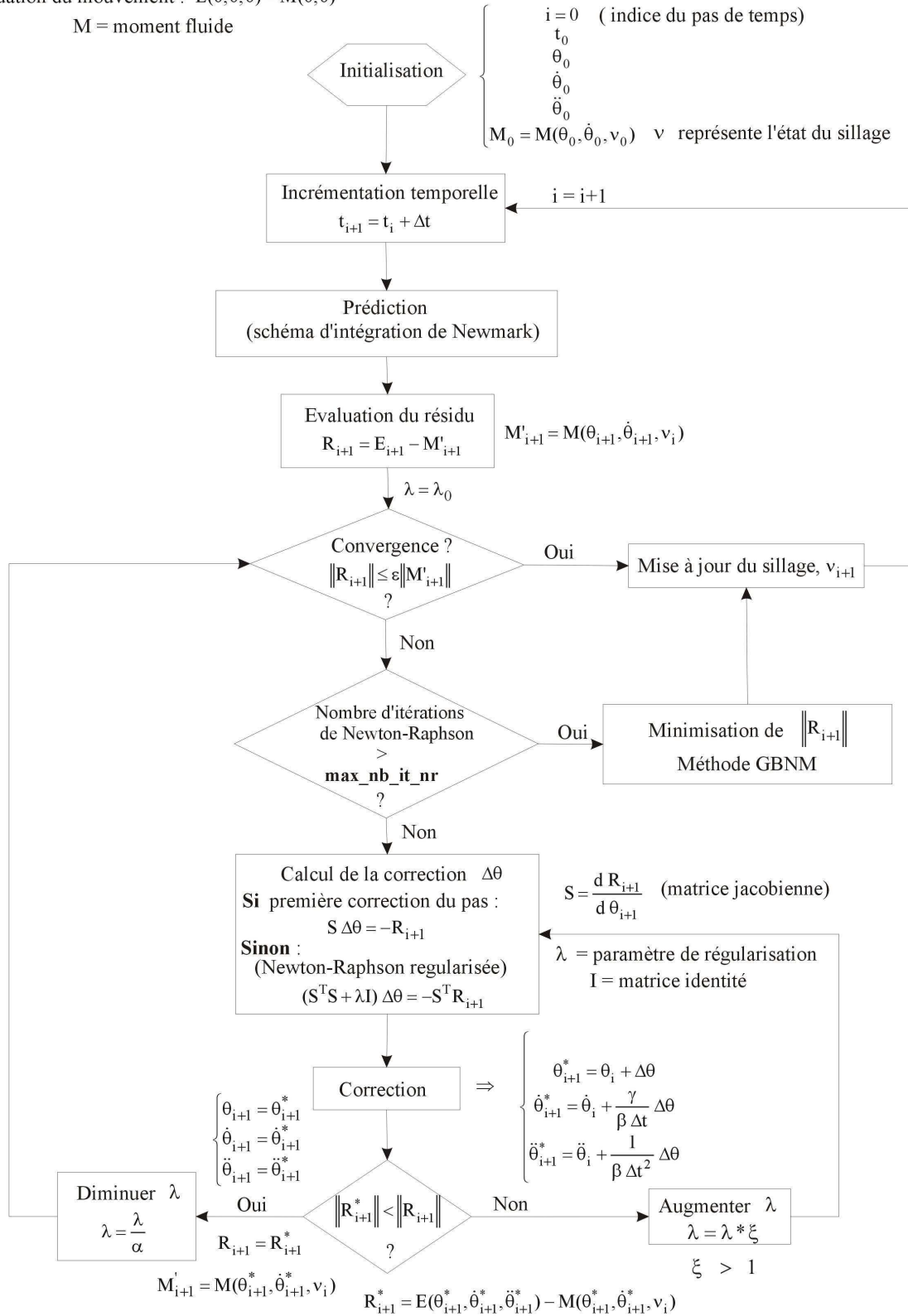


FIG. G.1 – Organigramme de la résolution des équations du système : schéma d'intégration temporelle de Newmark, couplé aux méthodes de Newton-Raphson régularisée et GBNM.

Annexe H

Influence de la discrétisation des barres, de l'incrément de temps et du temps total d'une simulation de la palme 2D

Dans cette Section, à travers une étude paramétrique, on observe l'influence des paramètres du modèle de palme 2D à 5 ressorts, en regardant les puissances moyennes et le temps d'exécution d'une simulation sur un PC standard. Les paramètres étudiés sont la discrétisation des barres de la palme pour la résolution du problème fluide, l'incrément de temps Δt , et le temps total d'une simulation T_f ¹. La discrétisation du nageur est invariable, à savoir, 60 segments de même longueur. La cinématique de nage est celle présentée en Section 7.3.1.

Dans les études paramétriques, les raideurs C_i varient ensemble mais restent constantes le long de la palme. La variation du temps machine d'une simulation en fonction du changement des C_i est négligeable. Les Tables ci-après résument les résultats des tests réalisés.

Parmi les paramètres essayés, la Table H.1 donne les résultats les plus précis, puisque ils sont obtenus avec la discrétisation des barres la plus fine (10 segments/barre), l'incrément de temps le plus petit ($\Delta t = 0.004$ s), et le temps total de simulation le plus grand $T_f = 12$ s. En revanche, la simulation est coûteuse : le temps d'exécution atteint 50 *min*, ce qui interdit l'optimisation. En observant les résultats avec les différentes réglages, on remarque que les paramètres : discrétisation = 5 segments/barre, incrément de temps $\Delta t = 0.004$ s et temps total de simulation $T_f = 8$ s (Table H.4) sont un bon compromis entre la précision et le temps d'exécution. Ce sont

¹Comme T_f est utilisé dans l'évaluation des moyennes temporelles, sa valeur est choisie comme étant un multiple de la période du mouvement.

ces paramètres qui ont été choisis pour l'optimisation des C_i dans le Chapitre 8.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1342.07	-763.28	-823.41	-1065.82	-855.94
\bar{P}_f	-1470.94	-1020.26	-1379.94	-3921.65	-5273.39
ν_P	0.912	0.748	0.597	0.272	0.162

TAB. H.1 – 10 segments par barre, $\Delta t = 0.004$ s, $T_f = 12$ s, temps d'exécution = 50 min.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1348.95	-748.38	-808.69	-1056.38	-850.28
\bar{P}_f	-1481.19	-1006.61	-1367.90	-3923.13	-5277.93
ν_P	0.911	0.743	0.591	0.269	0.161

TAB. H.2 – 5 segments par barre, $\Delta t = 0.004$ s, $T_f = 12$ s, temps d'exécution = 33 min.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1349.53	-748.82	-809.19	-1054.03	-846.95
\bar{P}_f	-1481.72	-1007.45	-1369.07	-3920.99	-5273.79
ν_P	0.911	0.743	0.591	0.269	0.161

TAB. H.3 – 5 segments par barre, $\Delta t = 0.004$ s, $T_f = 10$ s, temps d'exécution = 21 min.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1350.54	-749.39	-809.67	-1052.15	-846.08
\bar{P}_f	-1482.55	-1008.32	-1369.88	-3917.57	-5270.33
ν_P	0.911	0.743	0.591	0.269	0.161

TAB. H.4 – 5 segments par barre, $\Delta t = 0.004$ s, $T_f = 8$ s, temps d'exécution = 17 min.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1366.03	-766.19	-843.55	-1222.74	-1072.51
\bar{P}_f	-1515.35	-1037.20	-1415.01	-4210.27	-5885.64
ν_P	0.901	0.739	0.596	0.290	0.182

TAB. H.5 – 5 segments par barre, $\Delta t = 0.005$ s, $T_f = 12$ s, temps d'exécution = 20 min.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1367.25	-766.64	-844.30	-1222.04	-1070.46
\bar{P}_f	-1516.34	-1037.80	-1416.02	-4209.48	-5883.51
ν_P	0.902	0.739	0.596	0.290	0.182

TAB. H.6 – 5 segments par barre, $\Delta t = 0.005$ s, $T_f = 10$ s, temps d'exécution = 17 min.

$C_i, i = 2, 6$	125	500	1000	5000	15000
\bar{P}_{fx}	-1369.80	-767.38	-845.57	-1221.02	-1066.89
\bar{P}_f	-1518.42	-1038.74	-1417.67	-4208.47	-5879.5
ν_P	0.902	0.739	0.596	0.290	0.181

TAB. H.7 – 5 segments par barre, $\Delta t = 0.005$ s, $T_f = 8$ s, temps d'exécution = 13 min.

Annexe I

Flexural stiffness identifiability

¹ When choosing the load cases under which J_{static} is calculated (Equation (12.2)), care should be taken to guarantee that the optima are locally unique (local identifiability property). In this regard, it is convenient and sufficiently descriptive to study under which conditions on the loads F_i the 2D bars system of Figure I.1 has a unique set of stiffnesses C_i associated to an equilibrium position θ_i^* , $i = 1, n$.

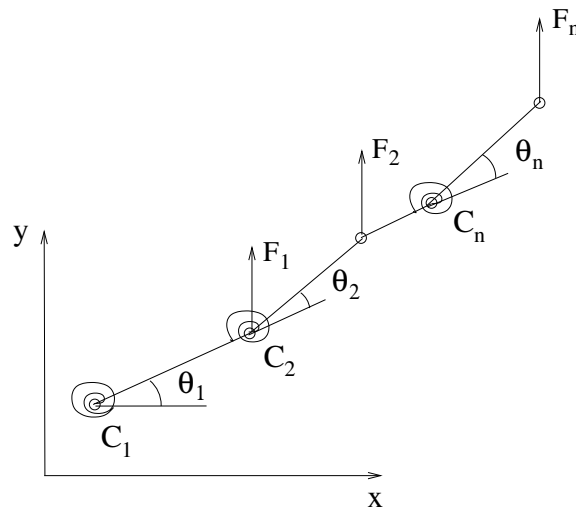


FIG. I.1 – 2D bars system. All bars have length l , C_i is the stiffness of the i -th joint, the loads F_i are vertical.

¹Annexe extraite de l'article [61].

The equilibrium equations of the system are,

$$\left\{ \begin{array}{l} C_1\theta_1 = lF_1 \cos \theta_1 + lF_2(\cos \theta_1 + \cos(\theta_1 + \theta_2)) + \cdots + \\ \quad + lF_n(\cos \theta_1 + \cdots + \cos(\theta_1 + \cdots + \theta_n)) \\ \quad \dots \\ C_n\theta_n = lF_n \cos(\theta_1 + \cdots + \theta_n) , \end{array} \right. \quad (\text{I.1})$$

or,

$$h_i(\theta_1, \dots, \theta_n) = 0 , \quad i = 1, n . \quad (\text{I.2})$$

Let θ_i^* , $i = 1, n$, satisfy the equilibrium equations (I.2). To see how a change in C_i 's affects the equilibrium of the system, a first order approximation to the h_i 's is written at θ^* ,

$$\left[\frac{\partial h}{\partial C}(\theta^*) \right] \Delta C = 0 , \quad (\text{I.3})$$

where components of the Jacobian matrix are, in general,

$$\left[\frac{\partial h}{\partial C}(\theta^*) \right]_{ij} = \frac{\partial h_i}{\partial C_j}(\theta^*) = \frac{dh_i}{dC_j}(\theta^*) + \sum_{k=1}^n \frac{\partial h_i}{\partial \theta_k}(\theta^*) \frac{\partial \theta_k}{\partial C_j}(\theta^*) . \quad (\text{I.4})$$

Cases of interest are the problematic ones where, around θ_i^* , a change in C_i 's induces no change in the equilibrium, i.e., cases where there are an infinite number of C_i 's associated to the same deflected shape θ^* (same J_{static}). At such non-identifiable points, by definition,

$$\frac{\partial \theta_k}{\partial C_j}(\theta^*) = 0 \quad (\text{I.5})$$

and the Jacobian has null eigenvalues whose associated eigenvectors (stiffness change ΔC) induce no variation of the equilibrium. When (I.5) holds,

$$\left[\frac{\partial h}{\partial C}(\theta^*) \right]_{ij} = \frac{\partial h_i}{\partial C_j}(\theta^*) = \theta_i^* \delta_{ij} , \quad (\text{I.6})$$

where $\delta_{ij} = 1$ if $i = j$; $\delta_{ij} = 0$ otherwise.

This establishes that the Jacobian eigenvalues are the equilibrium angles. Local non-identifiability occurs when some of the equilibrium angles are null, which is intuitive since the associated springs have no action. Two typical scenarii where some θ_i^* are null are depicted in Figure I.2, first when

the tip of the system is not loaded, then when the moments cancel at a joint.

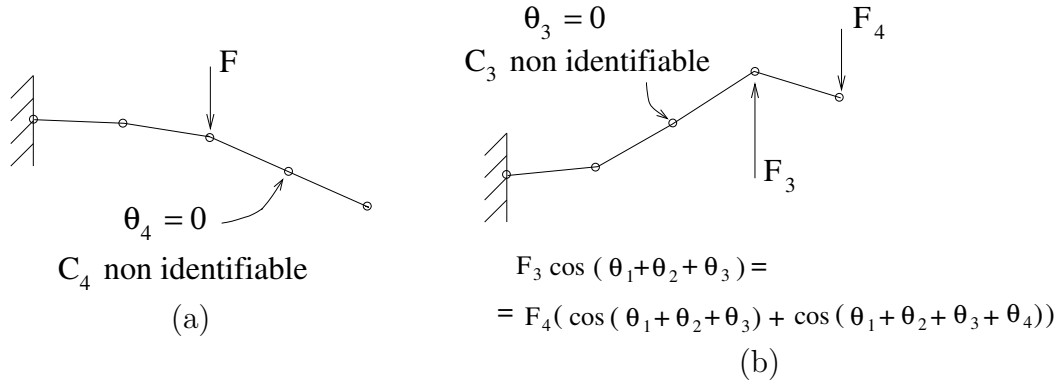


FIG. I.2 – Examples of loads such that the flexural stiffnesses are not identifiable, (a) the tip is not loaded, (b) moments cancel at joint 3.

Besides the load case, we note, without formal proof, that the control points used to calculate J_{static} should be numerous and well spread on the system in order to guarantee the uniqueness of $\arg \min_{C \text{ or } h} J_{static}$. As a counter-example (see Figure I.3), if there is only a control point at the tip with a target displacement (u^t, v^t) , it should be clear that there is an infinite number of choices of $(C_1^*, \dots, C_n^*, \theta_1^*, \dots, \theta_n^*)$ such that

$$\begin{cases} \text{(I.2) is satisfied and} \\ l(\cos \theta_1^* + \dots + \cos(\theta_1^* + \dots + \theta_n^*)) = u^t \\ l(\sin \theta_1^* + \dots + \sin(\theta_1^* + \dots + \theta_n^*)) = v^t \end{cases} \quad (\text{I.7})$$

because (I.7) is a system of $(n + 2)$ equations in $2n$ unknowns.

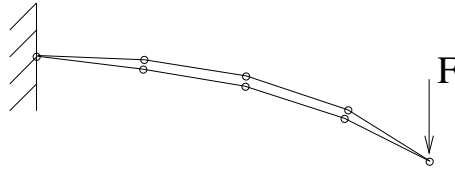


FIG. I.3 – Two deflected shapes that have the same displacements at the tip. This illustrates why C_i 's would not be identifiable if there was only a control point at the tip.

Annexe J

Correction des raideurs 2D optimales

Les résultats préliminaires de l'identification d'épaisseurs obtenus suivant la méthodologie exposée en Section 11.1 ont été un ordre de grandeur supérieur à des monopalmes réalistes. Les Figures J.1 et J.2 montrent les répartitions obtenues pour la condition $\bar{P}^{min} = -2000 W$ avec les formulations statiques en petits et grands déplacements, respectivement. Les monopalmes de sprint fabriquées avec un matériau équivalent à celui utilisé dans l'identification ont des épaisseurs plus faibles, la région la plus épaisse étant de l'ordre de $1.7 \times 10^{-3} m$ (cf. Figure J.3).

Malgré que le profil utilisé dans les vraies monopalmes ne soit pas issu d'une optimisation, on considère que l'ordre de grandeur des épaisseurs est convenable. Les écarts entre les épaisseurs optimisées et celles utilisées proviennent de la surestimation des raideurs 2D qui sont elles mêmes la conséquence de la surestimation des efforts. La surestimation des efforts, quant à elle, est due à la non-séparation structure-fluide et au fait que, dans le modèle 2D, le fluide ne peut pas passer par les cotés.

Ces observations nous ont conduit à corriger les efforts fluides puis les raideurs optimales 2D. Dans la littérature, on peut trouver des relations entre la distribution d'efforts le long de l'envergure et les efforts 2D, mais pour des cas d'écoulements stationnaires et des grands élancements ([1]). Ici, la correction exploite la connaissance des profils typiques de palmes réelles pour en déduire un ordre de grandeur des efforts. Ce type de correction a posteriori à partir de données expérimentales est comparable à la technique du *Beta Correction* ([32, 90]) utilisée en conception aéronautique et qui emploie des modèles de niveaux de précision différents. Les phases des efforts en fonction du déplacement du nageur et de la palme sont conservées. Par contre, l'intensité, qui est trop grande par rapport à la réalité, est corrigé par l'intermédiaire d'un scalaire multiplicatif,

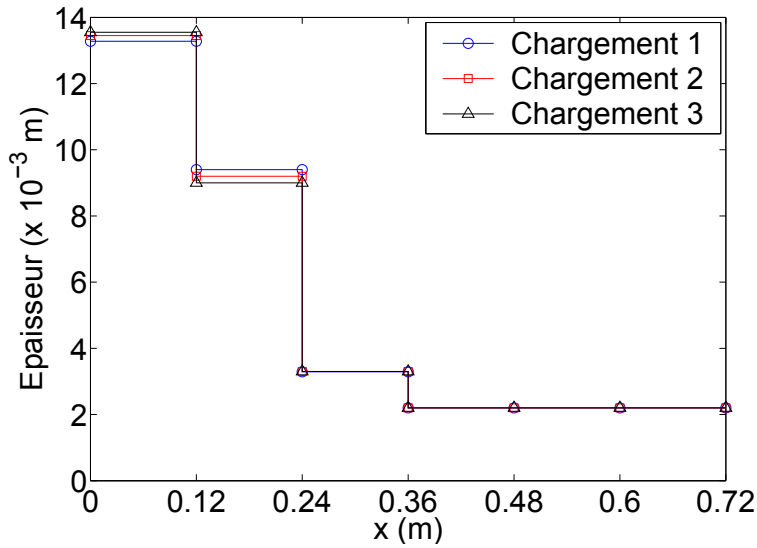


FIG. J.1 – Profil d'épaisseur : identification en petits déplacements, $\bar{P}^{min} = -2000 W$, sans correction des raideurs.

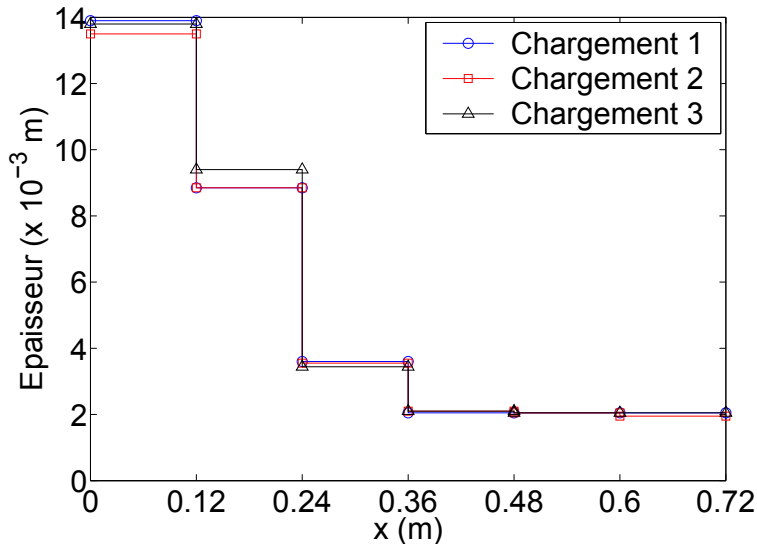


FIG. J.2 – Profil d'épaisseur : identification en grands déplacements, $\bar{P}^{min} = -2000 W$, sans correction des raideurs.

le coefficient de correction de charges 2D/3D, β . Après avoir trouvé β , la répartition de raideurs 2D optimale est corrigée, et la répartition d'épaisseur est identifiée à nouveau. La procédure de correction est réalisée en trois temps :

1. Obtention du coefficient de charge β : β est trouvé en minimisant l'écart entre la déformée du modèle 2D avec les raideurs optimales et la déformée du plan de symétrie d'un modèle

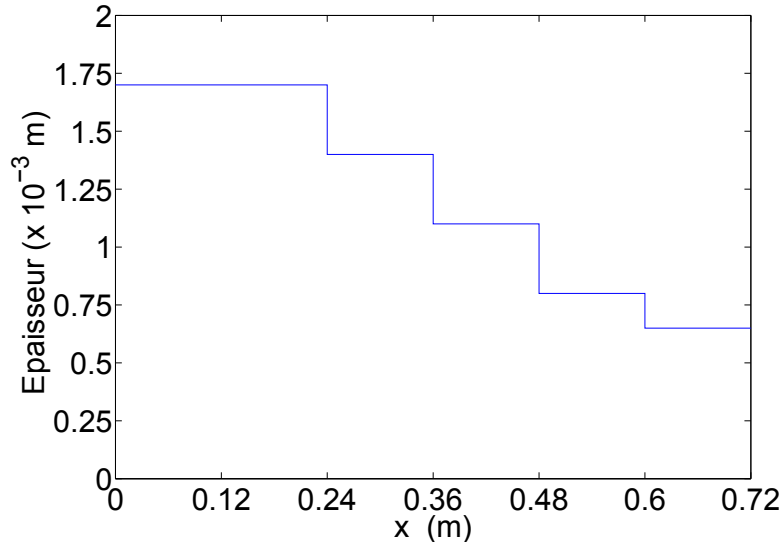


FIG. J.3 – Profil d'épaisseur typique d'une monopalme de compétition du type sprint.

éléments finis 3D de monopalme dont les épaisseurs sont réalistes sous le chargement 2D multiplié par β (et, bien entendu, réparti sur la largeur de la palme). L'écart est moyenné sur plusieurs instants. L'inertie de la palme est négligée.

2. Correction de la répartition de raideurs 2D : en utilisant un cas de chargement connu, les nouvelles raideurs sont trouvées en minimisant l'écart entre la déformée du modèle 2D avec des raideurs optimales (la cible), et la déformée 2D sous un chargement multiplié par β .
3. Nouvelle identification des épaisseurs : les raideurs 2D corrigées deviennent la nouvelle cible des identifications d'épaisseurs de palmes 3D (cf. Chapitre 11).

Les deux premières étapes sont détaillées ci-après, la troisième étape étant l'objet du Chapitre 11.

J.1 Obtention du coefficient de charge β

Le coefficient de charge est trouvé en minimisant la fonction erreur

$$J_\beta = \frac{1}{NP \ NC \ NPC} \sum_{k=1}^{NP} \sum_{j=1}^{NC} \sum_{i=1}^{NPC} ((u_i - u_i^t)^2 + (v_i - v_i^t)^2), \quad (\text{J.1})$$

où (u_i^t, v_i^t) sont les déplacements cibles du modèle 2D avec des raideur optimales, (u_i, v_i) sont les déplacements du modèle d'éléments finis 3D en grands déplacements, NP est le nombre de profils

d'épaisseurs, NC le nombre de cas de chargements, et NPC le nombre de points de contrôle (les mêmes que ceux de J , cf. Section 11.1). Le modèle éléments finis 3D de la monopalme est celui de la Figure 11.1.

Un profil d'épaisseur ($NP = 1$) est utilisé en (J.1) : [1.7 1.7 1.4 1.1 0.8 0.65] *mm*. Il correspond à une monopalme de sprint de chez Breier S.A.S.

Les chargements sont issus du problème 2D couplé fluide-structure avec les raideurs optimales, et correspondent à cinq instants différents d'un même cycle ($NC = 5$) : 4.252, 4.500, 4.644, 4.748 et 4.996 *s*. Dans le modèle 3D, les efforts bidimensionnels suivant x et y sont repartis uniformément le long de la largeur, en prenant en compte la symétrie. L'inertie étant négligée, J_β est évalué par des analyses statiques.

La minimisation de J_β est réalisée avec l'algorithme GBNM et la valeur trouvée pour le coefficient de correction de charge est $\beta = 1.542 \times 10^{-2}$.

J.2 Correction de la répartition de raideurs 2D

Pour trouver la nouvelle répartition de raideurs, l'écart entre la déformée du modèle 2D avec les raideurs optimales, et la déformée 2D sous le chargement corrigé par le facteur β , est minimisé :

$$\min_{C_i} J_C , \quad (\text{J.2})$$

où

$$J_C = \frac{1}{NPC} \sum_{i=1}^{NPC} ((u_i - u_i^t)^2 + (v_i - v_i^t)^2) , \quad (\text{J.3})$$

(u_i^t, v_i^t) étant les déplacements cibles du modèle 2D avec des raideur optimales et le chargement original, (u_i, v_i) les déplacements du modèle 2D avec le chargement corrigé, et NPC le nombre de points de contrôle, positionnés sur les articulations du modèle 2D ($NPC = 5$). Le chargement et les conditions aux limites utilisées sont ceux du cas 1, décrit en Section 11.2 et en Figure 11.2(a).

Après la résolution de (J.2), en utilisant GBNM, les valeurs corrigées obtenues pour les raideurs sont : $C_2 = 87.944$, $C_3 = 64.032$, $C_4 = 5.067$, $C_5 = 4.541$, $C_6 = 4.610$ *Nm/rad*. La Figure J.4 représente la répartition des raideurs le long de la palme. Ce profil est considéré comme solution de la conception 2D dans l'identification d'épaisseurs réalisée au Chapitre 11.

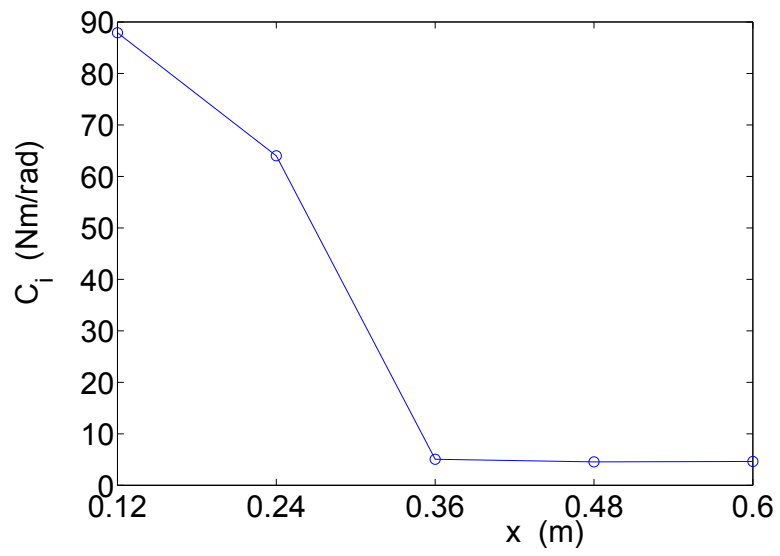


FIG. J.4 – Répartition corrigée de raideurs 2D, $\bar{P}^{min} = -2000 W$.

Annexe K

Analyse modale du modèle de barres 2D

Les équations pour effectuer l'analyse modale du modèle de barres 2D sont obtenues à partir de la linéarisation des équations du mouvement présentées dans l'Annexe F. Cette linéarisation donne le système suivant,

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F\} , \quad (\text{K.1})$$

où $[M]$, $[C]$ et $[K]$ sont les matrices de masse, d'amortissement, et de raideur, respectivement, $\{x\}$, $\{\dot{x}\}$ et $\{\ddot{x}\}$ sont les vecteurs des déplacements généralisés, des vitesses, et des accélérations, respectivement, et F est le vecteur des forces extérieures. Notons que dans un système linéaire, $[M]$, $[C]$ et $[K]$ ne dépendent ni de $\{x\}$, ni de $\{\dot{x}\}$, ni de $\{\ddot{x}\}$. Après l'obtention des Equations (K.1), ou plus précisément des matrices $[K]$ et $[M]$, les fréquences propres et les modes associés sont trouvés par la résolution du problème aux valeurs propres suivant ([27]),

$$([K] - \omega_j^2[M])\{\xi_j\} = \{0\} , \quad (\text{K.2})$$

ω_j étant les fréquences propres, et $\{\xi_j\}$ les modes associés.

Les équations du système de barres 2D sont maintenant reprises afin d'obtenir $[K]$ et $[M]$ pour le système linéaire. La stratégie utilisée est d'écrire les expressions des énergies potentielle et cinétique linéarisées, puis d'appliquer les équations de Lagrange. On entend par linéarisation d'une énergie une expression quadratique dans les déplacements généralisés ou leurs dérivées de telle sorte que les forces généralisées associées soient linéaires ([27]). Les conditions aux limites sont celles utilisées dans la procédure d'identification d'épaisseurs, c'est-à-dire, l'encastrement de

la structure au point qui représente le pied du nageur. Par conséquent, le système ne présente que des degrés de liberté en rotation (θ). L'expression de l'énergie potentielle E_p reste identique à celle donnée par l'Equation (F.1) car les C_i sont constants. Ainsi cette expression est déjà quadratique. Comme la palme est encastree, l'énergie cinétique de translation du système, donnée par l'Equation (F.2), est nulle, et l'énergie cinétique totale E_c est donnée par l'énergie cinétique de rotation (cf. Equations (F.8), (F.3), (F.4) et (F.7)), et re-écrite ci-après,

$$E_c = E_{cR} = \sum_{i=1}^n E_{cRi} = \sum_{i=1}^n \left[\frac{1}{24} \rho_L l_i^3 \left(\sum_{j=1}^i \dot{\theta}_j \right)^2 + \frac{1}{2} (\rho_L l_i) (\dot{x}_{G_i}^2 + \dot{y}_{G_i}^2) \right], \quad (\text{K.3})$$

où \dot{x}_{G_i} et \dot{y}_{G_i} sont les dérivées par rapport au temps de la position du centre de gravité d'une barre i . Les positions x_{G_i} et y_{G_i} sont données par l'Equation (F.5). En faisant maintenant l'approximation pour des petites rotations, $\cos(\theta) \approx 1$ et $\sin(\theta) \approx \theta_i$, elles se réduisent à,

$$\begin{cases} x_{G_i} \approx \sum_{j=1}^{i-1} l_j + \frac{l_i}{2}, \\ y_{G_i} \approx \sum_{j=1}^{i-1} (l_j \sum_{k=1}^j \theta_k) + \frac{l_i}{2} \sum_{k=1}^i \theta_k, \end{cases} \quad (\text{K.4})$$

et si l'on dérive par rapport au temps,

$$\begin{cases} \dot{x}_{G_i} \approx 0, \\ \dot{y}_{G_i} \approx \sum_{j=1}^{i-1} (l_j \sum_{k=1}^j \dot{\theta}_k) + \frac{l_i}{2} \sum_{k=1}^i \dot{\theta}_k. \end{cases} \quad (\text{K.5})$$

En substituant (K.5) dans l'Equation (K.3), l'énergie cinétique E_c est alors écrite comme,

$$E_c \approx \sum_{i=1}^n \left[\frac{1}{24} \rho_L l_i^3 \left(\sum_{j=1}^i \dot{\theta}_j \right)^2 + \frac{1}{2} \rho_L l_i \left(\sum_{j=1}^{i-1} \left(l_j \sum_{k=1}^j \dot{\theta}_k \right) + \frac{l_i}{2} \sum_{k=1}^i \dot{\theta}_k \right)^2 \right]. \quad (\text{K.6})$$

En possession des expressions de l'énergie potentielle E_p et de l'énergie cinétique E_c linéarisées, les équations du mouvement linéarisées, et en conséquence les matrices de raideur $[K]$ et de masse $[M]$ sont obtenues par application des équations de Lagrange. Le logiciel de calcul symbolique Mathematica ([91]) est utilisé pour cela. Les commandes de Mathematica et les matrices obtenues sont montrées ci-dessous.

```

ClearAll["Global'*"]

(* ---- Nomenclature : *)
(* ---- n --> nombre de barres *)
(* ---- l[i] --> longueur des barres *)
(* ---- theta[i][t] --> angles relatifs entre les barres (fonction du temps t) *)
(* ---- rho1 --> masse lineique des barres *)
(* ---- Ct[i] --> raideurs *)
(* ----- *)

n = 6; (* nombre de barres *)
Do[l[i]=1,{i,1,n}]; (* toutes les barres ont longueur l identique *)

(* ----- Energie cinetique : *)
Ec = Sum[(rho1 l[i]^3/24 (Sum[theta'[k][t],{k,1,i}]^2 +
      rho1 l[i]/2 ((Sum[l[j] Sum[theta'[k][t],{k,1,j}],{j,1,i-1}] +
      l[i]/2 Sum[theta'[k][t],{k,1,i}]^2)),{i,1,n}];

(* ----- Energie potentielle : *)

Ep = 1/2 (Sum[(Ct[i](theta[i][t])^2),{i,2,n}]);

(* ----- Equations du mouvement : application des equations de Lagrange : *)

SetAttributes[rho1,Constant]; (* rho1 et l ne dependent pas du temps t *)
SetAttributes[l,Constant];
Moment = Array[moment,n]; (* vecteur des efforts exterieurs *)

Do[(moment[i] = Dt[D[Ec,theta'[i][t]],t] - D[Ec,theta[i][t]] + D[Ep,theta[i][t]]),{i,1,n}];

(* ---- Obtention des matrices de raideur [K] et de masse [M] ----*)

K = Array[k,{n,n}];
M = Array[m,{n,n}];

Do[(Do[(k[i,j] = Coefficient[moment[i],theta[j][t]]),{j,1,n}]),{i,1,n}];
Do[(Do[(m[i,j] = Coefficient[moment[i],theta''[j][t]]),{j,1,n}]),{i,1,n}];

(* ----- Conditions aux limites : theta[1][t] = 0 et theta''[1][t] = 0
--> eliminer la premiere ligne et la premiere colonne de chaque matrice *)

Kc1 = K[[Range[2,n],Range[2,n]]]; (* matrice de raideur reduite *)
Mc1 = M[[Range[2,n],Range[2,n]]]; (* matrice de masse reduite *)

(* ----- Impression des matrices : *)

MatrixForm[K]

```

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & Ct[2] & 0 & 0 & 0 & 0 \\ 0 & 0 & Ct[3] & 0 & 0 & 0 \\ 0 & 0 & 0 & Ct[4] & 0 & 0 \\ 0 & 0 & 0 & 0 & Ct[5] & 0 \\ 0 & 0 & 0 & 0 & 0 & Ct[6] \end{pmatrix}$$

MatrixForm[M]

$$\begin{pmatrix} \frac{72 l^3 rhol}{325 l^3 rhol} & \frac{325 l^3 rhol}{125 l^6 rhol} & \frac{112 l^3 rhol}{88 l^3 rhol} & \frac{45 l^3 rhol}{18 l^3 rhol} & \frac{32 l^3 rhol}{26 l^3 rhol} & \frac{17 l^3 rhol}{7 l^3 rhol} \\ \frac{112 l^6 rhol}{88 l^3 rhol} & \frac{88 l^3 rhol}{88 l^3 rhol} & \frac{64 l^3 rhol}{64 l^3 rhol} & \frac{27 l^3 rhol}{27 l^3 rhol} & \frac{20 l^3 rhol}{20 l^3 rhol} & \frac{11 l^3 rhol}{11 l^3 rhol} \\ \frac{45 l^3 rhol}{32 l^3 rhol} & \frac{18 l^3 rhol}{26 l^3 rhol} & \frac{27 l^3 rhol}{20 l^3 rhol} & \frac{9 l^3 rhol}{14 l^3 rhol} & \frac{14 l^3 rhol}{8 l^3 rhol} & \frac{4 l^3 rhol}{5 l^3 rhol} \\ \frac{32 l^3 rhol}{17 l^3 rhol} & \frac{7 l^3 rhol}{7 l^3 rhol} & \frac{11 l^3 rhol}{11 l^3 rhol} & \frac{4 l^3 rhol}{4 l^3 rhol} & \frac{5 l^3 rhol}{5 l^3 rhol} & \frac{1 l^3 rhol}{1 l^3 rhol} \end{pmatrix}$$

MatrixForm[Kcl]

$$\begin{pmatrix} Ct[2] & 0 & 0 & 0 & 0 \\ 0 & Ct[3] & 0 & 0 & 0 \\ 0 & 0 & Ct[4] & 0 & 0 \\ 0 & 0 & 0 & Ct[5] & 0 \\ 0 & 0 & 0 & 0 & Ct[6] \end{pmatrix}$$

MatrixForm[Mcl]

$$\begin{pmatrix} \frac{125 l^3 rhol}{88 l^3 rhol} & \frac{88 l^3 rhol}{64 l^3 rhol} & \frac{18 l^3 rhol}{27 l^3 rhol} & \frac{26 l^3 rhol}{20 l^3 rhol} & \frac{7 l^3 rhol}{11 l^3 rhol} \\ \frac{88 l^3 rhol}{18 l^3 rhol} & \frac{64 l^3 rhol}{27 l^3 rhol} & \frac{27 l^3 rhol}{9 l^3 rhol} & \frac{14 l^3 rhol}{8 l^3 rhol} & \frac{4 l^3 rhol}{5 l^3 rhol} \\ \frac{26 l^3 rhol}{7 l^3 rhol} & \frac{20 l^3 rhol}{11 l^3 rhol} & \frac{14 l^3 rhol}{4 l^3 rhol} & \frac{8 l^3 rhol}{5 l^3 rhol} & \frac{5 l^3 rhol}{1 l^3 rhol} \end{pmatrix}$$

Il suffit maintenant de substituer les valeurs numériques de la masse linéique $rhol$, de la longueur l , et des raideurs $Ct[i]$, $i = 2, 6$, dans les matrices de raideur $[Kcl]$ et de masse $[Mcl]$, puis résoudre le problème aux valeurs propres (K.2). Pour $rhol = 2 \text{ kg/m}$, $l = 0.12 \text{ m}$, $Ct[2] = 87.944 \text{ Nm/rad}$, $Ct[3] = 64.032 \text{ Nm/rad}$, $Ct[4] = 5.067 \text{ Nm/rad}$, $Ct[5] = 4.541 \text{ Nm/rad}$ et $Ct[6] = 4.610 \text{ Nm/rad}$, les fréquences propres $f_j = \frac{\omega_j}{2\pi}$ du système de barres 2D, encasté dans la région du pied du nageur, sont : $f_1 = 1.5400 \text{ Hz}$, $f_2 = 7.8097 \text{ Hz}$, $f_3 = 17.6597 \text{ Hz}$, $f_4 = 31.8568 \text{ Hz}$ et $f_5 = 76.9945 \text{ Hz}$.

Annexe L

Dessins et propriétés matériaux de monopalmes

Les dessins ci-dessous ont été fournis par l'entreprise Breier S.A.S. Les notes concernant l'échelle ne sont pas valables.

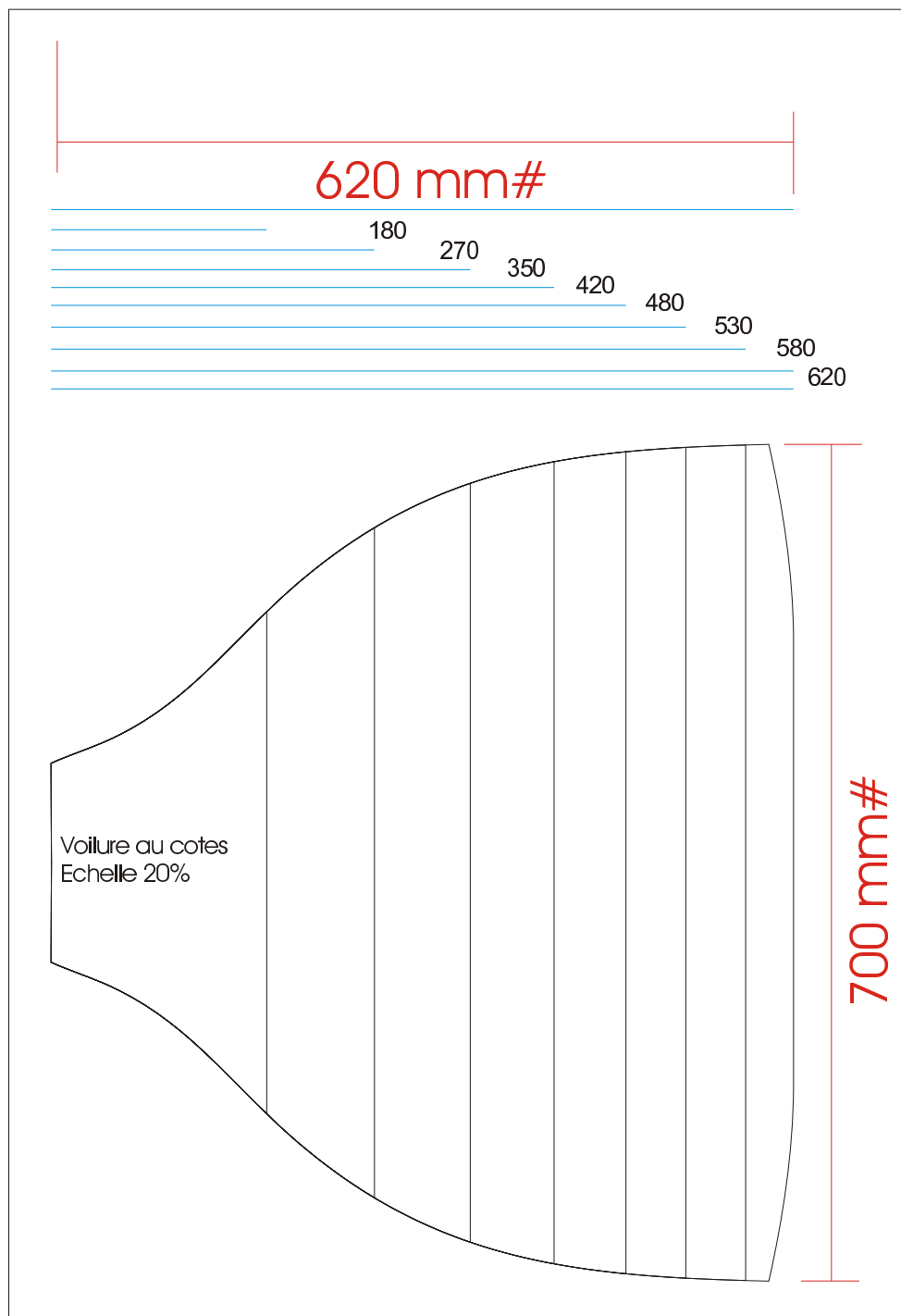
MONOPALME 1 :

loisir dureté souple :

10 plis de tissu de verre taffetas 200g/m²: épaisseur 0,15 mm
voir fiche sur HR200P-40 ou HexPly M10/43%/7628

Épaisseur au plus épais : 1.5 mm

Épaisseur au plus fin : 0.5 mm



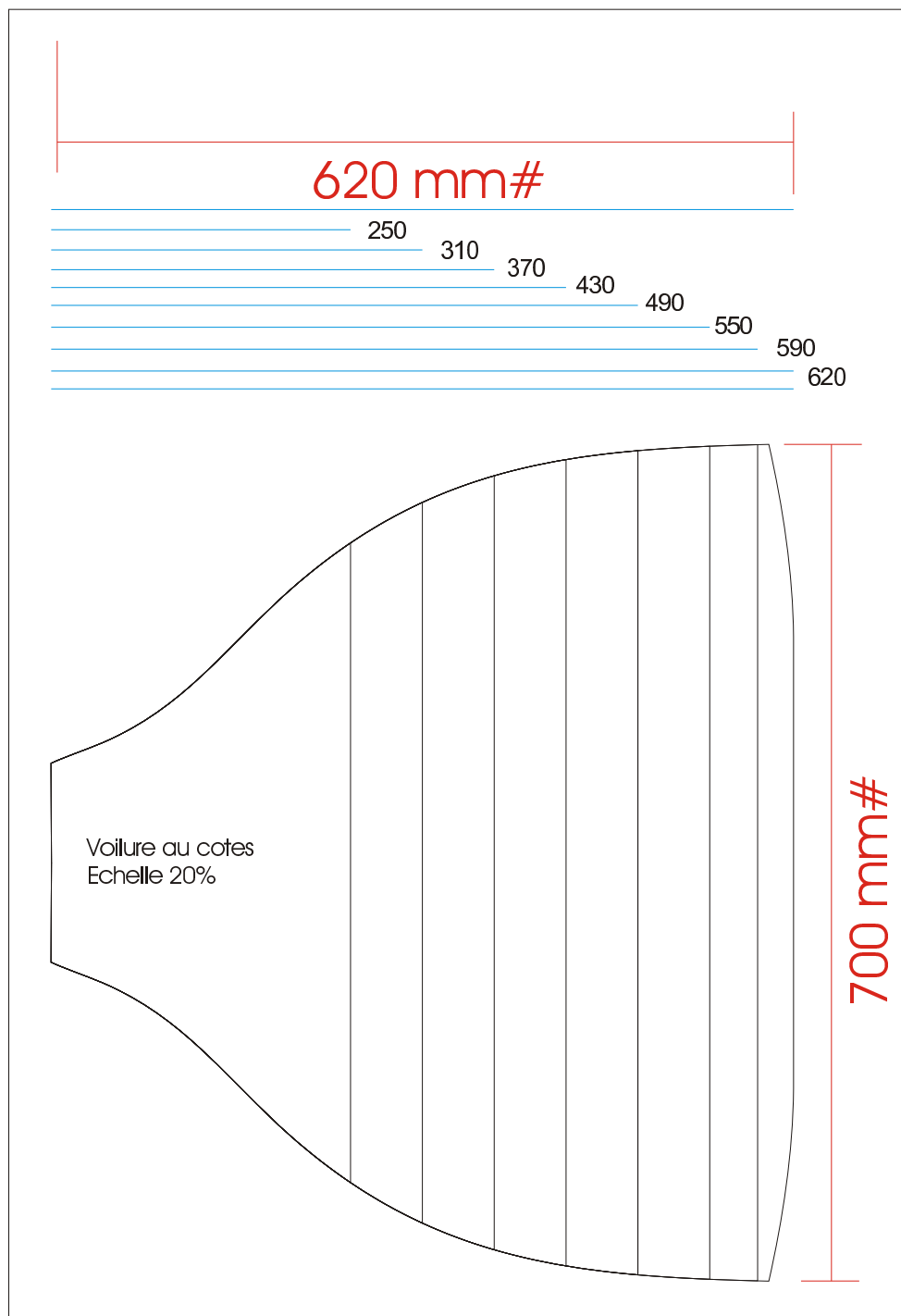
MONOPALME 2 :

loisir dureté moyenne :

10 plis de tissu de verre taffetas 200g/m²: épaisseur 0,15 mm
voir fiche sur HR200P-40 ou HexPly M10/43%/7628

Épaisseur au plus épais : 1.5 mm

Épaisseur au plus fin : 0.5 mm



Préimprégnés Hexply® M10
Système Epoxyde 120 °C

Fiche Produit

Désignation

Epoxyde / Tissu verre E 200 g/m²

HexPly® M10/43%/7628 - 110 cm

(Résine / Taux massique de résine / Référence renfort - Largeur)

Préimprégné :

Masse nominale : 350 g/m²
Volume nominal fibre : 37,96 %
Densité résine : 1,20 g/cm³
Epaisseur par pli polymérisé : 0,203 mm
Durée de vie maximale à 23 ± 2 °C : 60 jours

Renfort :

Masse nominale : 200 g/m²
Structure : Taffeats
Densité fibre : 2,60 g/cm³
Fibre : Verre E

Mise en œuvre

Le produit présente une grande souplesse de mise en œuvre.

Cycle de cuisson : de 15 h/85 °C à 10 min/150 °C

Pression : 0,3 à 3 bar

Stockage / destockage

Durée de vie maximale : 18 mois à - 18 °C / 6 mois à + 5 °C

Conserver le produit dans son emballage d'origine ou après utilisation partielle, dans un emballage étanche. Ne pas dérouler avant retour à température ambiante, du fait des risques de condensation d'eau.

HexPly® M10/43%/7628 - 110 cm

Propriétés mécaniques sur stratifié

Mise en œuvre (60 min à 120 °C, vide 0,85 bar, pression 3 bar)

	Traction Chaîne	Flexion Chaîne	C.I.L. Chaîne	Compression Chaîne
Contrainte (MPa)	550	650	50	500
Module (GPa)	18	19	-	-
Normes	NF T57.101	NF T57.105	NF T57.104	DIN 65380

Nota : Les valeurs figurant ci-dessus sont des valeurs moyennes obtenues sur un stratifié présentant un taux volumique de fibre de 55 % (soit une épaisseur par pli polymérisé de 140 µm environ).

Important

Les renseignements contenus dans la présente fiche produit sont fondés sur nos connaissances actuelles et sur les résultats d'essais effectués avec un constant souci d'objectivité. Ils doivent être adaptés à chaque cas particulier. Les performances du produit après utilisation étant liées aux conditions particulières de mise en œuvre, elles ne sauraient engager notre responsabilité.

Fibredux, Vicotex et Strafil sont des marques déposées Hexcel Composites.

JANVIER 98

Préimprégnés Vicotex® M10
Système Epoxyde 120 °C

Fiche Produit

Désignation

Epoxyde / Tissu verre E 305 g/m²

Vicotex® M10/43%/664 - 120 cm

(Résine / Taux massique de résine / Référence renfort - Largeur)

Préimprégné :

Masse nominale : 535 g/m²
Volume nominal fibre : 37,96 %
Densité résine : 1,20 g/cm³
Epaisseur par pli polymérisé : 0,304 mm
Durée de vie maximale à 23 ± 2 °C : 60 jours

Renfort :

Masse nominale : 305 g/m²
Structure : Satin 8
Densité fibre : 2,60 g/cm³
Fibre : Verre E

Mise en œuvre

Le produit présente une grande souplesse de mise en œuvre.

Cycle de cuisson : de 15 h/85 °C à 10 min/150 °C

Pression : 0,3 à 3 bar

Stockage / destockage

Durée de vie maximale : 18 mois à - 18 °C / 6 mois à + 5 °C

Conserver le produit dans son emballage d'origine ou après utilisation partielle, dans un emballage étanche. Ne pas dérouler avant retour à température ambiante, du fait des risques de condensation d'eau.

Vicotex[®] M10/43%/664 - 120 cm

Propriétés mécaniques sur stratifié

Mise en œuvre (60 min à 120 °C, vide 0,85 bar, pression 3 bar)

	Traction Chaîne	Flexion Chaîne	C.I.L. Chaîne	Compression Chaîne
Contrainte (MPa)	600	700	50	500
Module (GPa)	20	20	-	-
Normes	NF T57.101	NF T57.105	NF T57.104	DIN 65380

Nota : Les valeurs figurant ci-dessus sont des valeurs moyennes obtenues sur un stratifié présentant un taux volumique de fibre de 55 % (soit une épaisseur par pli polymérisé de 210 µm environ).

Important

Les renseignements contenus dans la présente fiche produit sont fondés sur nos connaissances actuelles et sur les résultats d'essais effectués avec un constant souci d'objectivité. Ils doivent être adaptés à chaque cas particulier. Les performances du produit après utilisation étant liées aux conditions particulières de mise en œuvre, elles ne sauraient engager notre responsabilité.

Fibredux, Vicotex et Strafil sont des marques déposées Hexcel Composites.

JANVIER 98