



HAL
open science

Modélisation numérique de rubans par éléments en courbures

Raphaël Charrondière

► **To cite this version:**

Raphaël Charrondière. Modélisation numérique de rubans par éléments en courbures. Modélisation et simulation. Université Grenoble Alpes [2020-..], 2021. Français. NNT : 2021GRALM038 . tel-03545017v2

HAL Id: tel-03545017

<https://inria.hal.science/tel-03545017v2>

Submitted on 16 Feb 2022

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.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : Mathématiques et Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Raphaël CHARRONDIÈRE

Thèse dirigée par **Florence BERTAILS-DESCOUBES**, Directrice de recherche, Université Grenoble Alpes
et codirigée par **Sébastien NEUKIRCH**, Sorbonne Université

préparée au sein du **Laboratoire Laboratoire Jean Kuntzmann**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

Modélisation numérique de rubans par éléments en courbures

Numerical modeling of ribbons by curved elements

Thèse soutenue publiquement le **14 octobre 2021**,
devant le jury composé de :

Madame FLORENCE BERTAILS DESCOUBES

DIRECTEUR DE RECHERCHE, INRIA CENTRE GRENOBLE-RHONE-ALPES, Directrice de thèse

Monsieur DAMIEN ROHMER

PROFESSEUR, ECOLE POLYTECHNIQUE, Examinateur

Monsieur BASILE AUDOLY

DIRECTEUR DE RECHERCHE, CNRS DELEGATION ILE-DE-FRANCE SUD, Président

Monsieur SEBASTIEN NEUKIRCH

DIRECTEUR DE RECHERCHE, CNRS DELEGATION PARIS CENTRE, Co-directeur de thèse

Monsieur ETIENNE VOUGA

PROFESSEUR ASSISTANT, The University of Texas at Austin, Rapporteur

Monsieur OLIVIER BRÜLS

PROFESSEUR, Université de Liège, Rapporteur



Modélisation numérique de rubans par éléments en courbure

Raphaël Charrondière

1^{er} décembre 2021

Table des matières

Table des matières	3
Notations et conventions	5
Introduction	7
I État de l'art	II
1 Bref tour d'horizon	13
1.1 Modèles de tiges	13
1.2 Modèles de coque	15
1.3 Surfaces développables	16
2 Sur une description théorique des rubans	19
2.1 Un bref historique	19
2.2 Une description géométrique	20
2.3 Une description énergétique	24
II Modèle numérique chaîné de ruban	27
3 Vers un modèle numérique de rubans	29
3.1 Fonctions de forme de haut degré	29
3.2 Intégration numérique par séries entières	30
3.3 Compensation catastrophique et contrôle de l'erreur	31
3.4 Statique de ruban sous gravité	32
3.4.1 Énergie discrète de Wunderlich	32
3.4.2 Énergie discrète de pesanteur	33
3.5 Vers l'implémentation du modèle	34
3.5.1 Implémentation des grandeurs en jeu	34
3.5.2 Choix d'un algorithme de minimisation d'énergie	35
4 Premières expériences numériques	41
4.1 Adimensionnement	41
4.2 Comparaison aux méthodes de continuation	42
4.3 Comparaison à un modèle de coque	44

4.4	Comparaison au modèle des tiges	46
4.5	Expérience de la poutre console	48
III Modèle numérique mixte		51
5	Vers un modèle mixte	53
5.1	Encastrer les deux extrémités du ruban	53
5.2	Un modèle mixte pour plus de vitesse	56
5.3	Comparaison : l'ajout de contraintes pour une meilleure vitesse	59
5.3.1	Scenario encasté libre	59
5.3.2	Scenario bi-encasté	59
5.3.3	Encastrement en début du ruban	61
6	Ajout de murs et rubans circulaires	65
6.1	Contraintes à l'encastrement	65
6.1.1	Un véritable ruban de Moebius	67
6.2	Ajout de contact entre le ruban et un plan	70
6.2.1	Ruban de Moebius sur la table	75
6.2.2	L'expérience de Roman	75
7	Discussions	83
7.1	Contrainte à l'encastrement sur η	83
7.2	L'expérience de décalage latéral des encastresments	85
7.3	L'extensibilité du ruban	91
Conclusion et perspectives		97
A	Sur les multiplicateurs de Lagrange	99
A.1	Cas bijectif	100
A.2	Cas injectif	101
Table des figures		103
Liste des Algorithmes		105
Bibliographie		107

Notations et conventions

Nous allons suivre un certain nombre de notations et conventions pour faciliter la lecture :

- \mathbf{v} un vecteur de \mathbb{R}^n de dimension quelconque.
- v_i désigne la coordonnée i du vecteur \mathbf{v} .
- \mathcal{A} une matrice à coefficients dans \mathbb{R} de taille quelconque.
- \mathbf{A}_i désigne la colonne i de la matrice \mathcal{A} .
- $A_{i,j}$ désigne le coefficient en i -ième ligne, j -ième colonne de la matrice \mathcal{A} .
- $\langle \mathbf{u}, \mathbf{v} \rangle$ le produit scalaire usuel.
- $a^{(i)}$ un exposant mis entre parenthèses désigne toujours un indice.
- $\mathbf{u} \times \mathbf{v}$ le produit vectoriel.
- $a \gg b$ la relation a est grand par rapport à b .
- $a \ll b$ la relation a est petit par rapport à b .
- $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ est la base canonique de \mathbb{R}^3 .
- ∇E est le gradient de E .
- Les angles sont tous exprimés en radians.
- Si f est une fonction et K une constante, on pourra noter $f = K$ pour dire que f est une fonction constante de valeur K .

Important : Des temps de calculs sont indiqués dans les différents chapitres. Ceux-ci dépendent entre autres de la machine utilisée, de la version des bibliothèques utilisées, des options de compilation. Cependant, je garantis que les temps de calculs restent cohérents au sein d'un même chapitre.

Les notations utilisées restent cohérentes entre les différents chapitres, pour permettre au lecteur de retrouver rapidement leur signification, grâce au récapitulatif suivant :

- a, b degrés de liberté pour la courbure normale ($\omega(s) = as + b$)
- D rigidité de flexion d'une plaque élastique
- $E_{\mathcal{W}}$ énergie potentielle élastique selon le modèle de Wunderlich
- $E_{\mathcal{S}}$ énergie potentielle élastique selon le modèle de Sadowsky
- $E_{\mathcal{P}}$ énergie potentielle de pesanteur
- \mathbf{e}_g vecteur définissant la verticale
- g accélération de la gravité
- \mathbf{g} vecteur gravité ($\mathbf{g} = -g \mathbf{e}_g$)
- h épaisseur du ruban
- $\boldsymbol{\kappa}$ vecteur courbure
- ℓ longueur du ruban
- n, m degrés de liberté pour la direction de la règle ($\eta(s) = ns + m$)
- η direction de la règle
- $\boldsymbol{\Omega}$ vecteur de Darboux
- p nombre de segments

- \mathbf{q} vecteur directeur d'une règle (non normé)
- ρ masse volumique
- \mathbf{r} position le long de la centerline
- \mathcal{R} repère matériel le long de la centerline
- s abscisse curviligne le long de la centerline
- $\hat{s}^{(i)}$ abscisse locale pour le segment i
- $\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z$ les trois axes du repère matériel
- u abscisse le long des règles
- ν coefficient de Poisson
- w largeur du ruban
- ω courbure normale
- $\check{\omega}$ courbure normale naturelle
- Y module de Young

Introduction

Bien qu'il s'agisse à première vue d'objets anecdotiques, les rubans peuvent servir à modéliser une large gamme d'objets à échelle variée comme illustré en figure 1. Par exemple, à l'échelle nanoscopique on retrouve les protéines ou molécules d'ADN qui peuvent être représentées par des rubans. À l'échelle mésoscopique on retrouve les bandes de tissu ou de papier. Et à l'échelle macroscopique les rubans deviennent attrayants en architecture, servant par exemple à réaliser des structures déployables.

Par définition un ruban est un objet fin et long, donc une structure à mi-chemin entre les tiges et les coques. Une tige est un objet mathématiquement caractérisé par une courbe dans l'espace. Les propriétés du matériau et la géométrie de la section vont alors contraindre le comportement de la courbe. Un ruban peut donc être vu comme une tige dont la section est un rectangle fin. Une coque est un objet fin, elle est mathématiquement caractérisée par une surface. En choisissant une surface adéquate on obtient alors un ruban. Dans notre travail, nous supposons que le ruban est une surface rectangulaire d'épaisseur constante. Aussi nous ne considérons que des rubans élastiques. C'est-à-dire que le ruban revient spontanément à sa configuration de base en l'absence de force extérieure. Les déformations sont alors réversibles.

En partant de ces informations, notre ruban est un pavé caractérisé par une longueur ℓ , une largeur w et une épaisseur h . Le ruban étant fin, l'épaisseur est petite comparée à la largeur. Le ruban étant long, la longueur est grande comparée à la largeur. En effet, les adjectifs «fin» et «long» ne sont pas une notion absolue, mais relative.

Finalement nous ajoutons une notion d'inextensibilité au ruban, c'est-à-dire que les forces en jeu (le poids du ruban, les forces à l'encastrement, etc.) ne sont pas suffisantes pour allonger celui-ci. Au vu de ses dimensions, nous supposons que cette inextensibilité n'est pas un effet moyenné le long du ruban, mais qu'elle est présente sur toute la surface. Ainsi toute déformation



Crédits de gauche à droite : Wikipedia Commons, R. CHARRONDIÈRE, [Van17].

FIGURE 1 – Illustration des rubans dans notre environnement. De gauche à droite : une protéine, des rubans de mercerie, une structure architecturale nommée «gridshell» en anglais.

du ruban est isométrique à la configuration où le ruban est plat. Cela signifie que toute courbe tracée sur le ruban aura la même longueur quelle que soit la déformation du ruban.

Si de nombreux modèles numériques ont été développés ces dernières décennies pour les tiges et les plaques, il en existe en revanche très peu qui soient spécifiques aux rubans. Dans cette thèse, nous proposons un modèle numérique robuste et efficace pour calculer les états d'équilibre stables d'un ruban naturellement plat ou courbé. Notre contribution se décompose en trois axes : tout d'abord l'élaboration du modèle, ensuite des comparaisons avec d'autres modèles pour valider et motiver l'apport de notre modèle, et finalement l'expérimentation.

Sur l'élaboration du modèle

Si un ruban est inextensible et rectangulaire, la surface le représentant est alors développable, car elle est isométrique à une surface plane¹. On peut alors s'appuyer sur la description analytique de van der Heijden et Audoly [SH07; DA15], eux-même s'appuyant sur les descriptions historiques de Sadowsky et Wunderlich [Sad29; Wun62]. On détaille une description théorique du ruban au chapitre 2.

Nous choisissons de nous baser sur la courbure pour discrétiser le ruban. Cela signifie que nos degrés de liberté vont décrire la courbure plutôt que la position des points du ruban^{2 3}. L'avantage de cette approche est d'obtenir une surface lisse (continue et dérivable au moins deux fois) tout en garantissant son inextensibilité. De plus, la courbure est directement accessible, donc il est facile de calculer une énergie élastique. Si l'on avait choisi de se baser sur la position, on aurait un modèle nodal, la surface n'aurait pas forcément été lisse⁴, rendant le calcul des courbures difficiles. Aussi l'inextensibilité aurait dû être imposée par des contraintes.

On définit une centerline le long du ruban, munie d'une abscisse curviligne. On verra qu'il est possible de décrire la surface en spécifiant seulement deux scalaires le long de cette courbe. Nous scindons le ruban en plusieurs éléments. Chacun d'entre eux est caractérisé par une courbure normale linéaire et une torsion géodésique quadratique en l'abscisse curviligne. Cela permet une grande diversité de configurations cinématiques, c'est-à-dire que l'espace des déformations du ruban est riche. De plus, nous donnons la possibilité au ruban d'avoir une forme naturelle ayant une courbure normale non nulle. La forme naturelle est la configuration du ruban en l'absence de toute force.

Si le choix de la discrétisation basée sur la courbure est justifié, il faut tout de même pouvoir calculer la position des points de la surface. Pour cela on doit intégrer la géométrie à partir de la courbure. Il n'existe cependant pas de solution en forme close à ce problème⁵. C'est-à-dire qu'il n'est pas possible de calculer la solution exacte à partir de fonctions et d'opérations simples. On

1. Il ne s'agit que d'une implication, le fait d'être rectangulaire permet juste de dire qu'il existe une déformation isométrique permettant d'aplatir le ruban.

2. Ce n'est pas tout à fait exact pour les raisons données au chapitre 3, mais la courbure est calculable très facilement à partir des degrés de liberté avec les opérations usuelles d'addition et de multiplication.

3. Dans le chapitre 5, nous utilisons une formulation mixte, c'est-à-dire que nous utilisons la courbure et la position comme degrés de liberté. Cela permet de garder les avantages d'un modèle en courbure en le rendant plus efficace.

4. On aurait certainement dans ce cas utilisé un maillage triangulaire. Mais certaines représentations permettent d'obtenir une surface lisse comme les B-splines rationnelles non uniformes (aussi nommés NURBS en anglais).

5. Dans le cas où la courbure est constante on peut trouver une telle solution [Ber09]. Mais nous verrons qu'il est trop restrictif de se restreindre à ce cas.

utilise alors l'idée de Casati [CB13] de développer la fonction position le long du ruban par des séries entières.

Le ruban est dit à l'équilibre lorsque l'énergie potentielle associée à sa déformation actuelle correspond à un minimum local. Ici, nous avons l'énergie de pesanteur et l'énergie élastique. On peut utiliser deux formulations pour cette dernière : celle de Sadowsky ou de Wunderlich. La formulation de Wunderlich est cependant plus correcte (en fait elle est exacte sous l'hypothèse d'inextensibilité du ruban si le ruban est naturellement plat). Dans les chapitres 5 et 6, on enrichit le problème de minimisation par des contraintes. Cela permet d'encastrer le ruban par l'un ou ses deux côtés, de ne pas l'encastrer du tout, d'attacher ses extrémités entre elles, et enfin de rajouter du contact entre le ruban et un ou plusieurs plans.

Sur les comparaisons

Notre seconde contribution (chapitre 4) se focalise sur la validité et l'intérêt de notre modèle.

Il est important d'être sûr que l'implémentation du modèle est correcte. On procède par comparaison. Ainsi, il ne s'agit pas de comparer les résultats obtenus par notre modèle à ceux de n'importe quel autre modèle. Il faut respecter les hypothèses d'inextensibilité du ruban et utiliser la même énergie. Sans cela, les différences obtenues sur l'équilibre calculé pourraient être dues au choix du modèle et non à une implémentation erronée. La vérification numérique est alors faite avec deux méthodes de continuation : par les logiciels MATHEMATICA et AUTO.

Nous avons présenté le ruban comme une structure intermédiaire entre les tiges et les coques. Il est donc naturel de comparer ces deux modèles avec le nôtre.

Nous avons choisi d'utiliser le modèle de tige des super-clothoïdes [CB13]. Comme ce modèle est inextensible, la comparaison avec notre modèle est pertinente. Cependant, par définition, pour une tige la largeur et l'épaisseur devraient être similaire en ordre de grandeur. En conséquent, la rigidité de courbure du modèle de tige avec une section rectangulaire doit être adaptée pour correspondre à la rigidité de flexion d'une plaque élastique. Même ainsi, le modèle de tiges ne parvient pas aux mêmes résultats que notre modèle. Nous en déduisons le besoin d'un modèle spécifique aux rubans.

Pour comparer notre modèle aux coques, nous avons choisi le modèle de Naghdi [Nag63]. Ce modèle est extensible. Son implémentation est faite par des éléments finis grâce au logiciel FENICS-SHELL [Hal+18]. On constate que dans la limite où l'épaisseur de la coque devient très petite par rapport à la largeur (environ deux ordres de grandeur), les résultats coïncident. On constate aussi que plus la coque est fine, et plus le solveur a du mal à converger (phénomène de verrouillage numérique). La convergence peut alors prendre plusieurs minutes. Notre modèle au contraire ne met qu'une seconde environ à converger⁶. Ainsi on en déduit que le modèle de coques est sans doute trop général pour simuler efficacement un ruban inextensible. Notre modèle montre alors tout son intérêt.

6. Pour cette comparaison le ruban a de la courbure naturelle et il pend simplement sous son propre poids, voir la section 4.3.

Quelques expériences numériques

La mise en pratique de notre modèle sur diverses expériences constitue une contribution majeure de cette thèse (chapitres 6 et 7).

- Nous commençons tout d'abord avec l'expérience de la poutre console. On accroche horizontalement une poutre sans courbure naturelle. On appelle x la distance entre les deux côtés de la poutre projetée sur l'horizontale et y la distance projetée sur la verticale, on peut tracer une courbe $\frac{y}{x}(T)$ où T est un paramètre de pesanteur dépendant des propriétés géométriques et matérielles.
- Étant un grand classique il n'est pas possible de passer à côté du ruban de Moebius, mais comme notre modèle le permet, nous posons le ruban de Moebius sur la tranche sur un plan.
- Nous reproduisons l'expérience que Benoît Roman décrit dans sa thèse, qui consiste à attacher un ruban sur une plaque en rapprochant ses extrémités de sorte qu'il forme une bosse. Puis, il est écrasé par une autre plaque parallèle à la première. Ainsi le ruban va former des vagues entre les deux plaques. Benoît Roman met en avant une loi d'échelle entre le nombre de vagues, la force du ruban sur la plaque supérieure et la distance entre les deux plaques.
- Nous continuons avec une expérience de déversement latéral. On encastre un ruban tel que la direction du ruban à l'encastrement soit suivant l'horizontale et que la normale à sa surface soit aussi horizontalement, à la manière d'un drapeau. Selon le paramètre T (dépendant de la géométrie et du matériau), le ruban va rester droit ou fléchir sur le côté.
- La dernière expérience reprise de [Hua+20; YH19] consiste à encastrent les deux bords puis à décaler latéralement les deux encastresments. Suivant le décalage le ruban aura une forme particulière : plutôt en U ou en S, avec une zone de transition entre les deux. En observant la courbe altitude du point au milieu du ruban en fonction du décalage, on repère clairement la forme du ruban.

Tandis que les trois premières expériences sont réalisées et validées avec succès, les deux dernières expériences échouent à reproduire les résultats théoriques attendus pour un modèle de coques. Ce point remet en cause l'hypothèse d'inextensibilité dans certains cas.

Via ces trois axes le travail réalisé ici forme un tout, de l'élaboration jusqu'à l'expérimentation d'un modèle du ruban élastique inextensible par éléments en courbure. Ce modèle montre une grande efficacité par sa vitesse et la richesse des scénarii qu'il peut modéliser.

Première partie

État de l'art

Chapitre 1

Bref tour d’horizon

Un modèle de ruban se trouve à mi-chemin entre un modèle de coque et un modèle de tige. Les tiges sont des structures élancées respectant la définition géométrique : $\ell \gg h, \ell \gg w$, $h \approx w$ c’est-à-dire que la longueur est grande devant la largeur et l’épaisseur qui sont comparables entre elles. Les coques respectent la relation $h \ll w, h \ll \ell$ c’est-à-dire l’épaisseur que de la structure est fine. Nous commençons donc par présenter les deux aspects ces deux types de structures minces. Puis nous introduisons brièvement la notion de développabilité, car la surface du ruban est développable (conséquence de l’hypothèse d’inextensibilité). Une présentation détaillée sur les rubans est laissée au chapitre 2.

1.1 Modèles de tiges

Une tige est un objet long. L’étude des tiges remonte à plus de 150 ans. La section d’une tige peut avoir différentes formes. Naturellement le mot tige rappelle plutôt un cylindre métallique jouant le rôle de support, ou une plante ([GNo6]), et tous deux ont une section circulaire, mais on peut donner d’autres exemples répondant à la définition : le fil d’une pelote de laine ([Cir+14]) ou d’une bobine, les cheveux, une poutre, un rail de chemin de fer. Pour ces deux derniers exemples la section n’est pas circulaire mais la définition de tige est bien respectée.

Le modèle de Kirchhoff [Kir59] est l’un des premiers modèles classiques existants, très repris jusqu’à aujourd’hui. Celui-ci se caractérise par un axe central (appelé aussi centerline) inextensible, une section transversale indéformable, pas de cisaillement et une contrainte linéaire en fonctions des trois courbures de l’axe central, y compris la torsion. Depuis, de nombreux modèles numériques ont vu le jour [BB94; CJ98; Sim85; LLA11; SCB14; Paio2; SBT07; GPL08; Ber+08; Ren+18; Ber+06], basés sur les équations de Kirchhoff. En graphique, [Ber+08] est l’un des modèles les plus classiques.

Pour représenter une tige, il faut spécifier la position de la tige et un repère matériel. Ce dernier sert à caractériser la torsion, car elle n’est pas calculable à partir de la position de la centerline (si l’on tend un fil entre ses mains et que l’on tord une extrémité, on crée de la torsion sans que celle-ci soit visible). On verra plus tard la relation entre ce repère matériel et les courbures.

Pour résoudre la cinématique, un couplage entre la courbure, le repère matériel et la position de la centerline apparaît. Bien qu’il s’agisse d’un système de Cauchy classique, on ne peut pas le résoudre analytiquement¹. Ainsi dans l’implémentation numérique, il convient d’utiliser une

1. Du moins pas facilement d’après nos connaissances. Dans [Iva01] une solution est proposée pour une courbure affine, mais la solution utilise la fonction hypergéométrique dégénérée. Elle n’est donc pas exprimée en forme

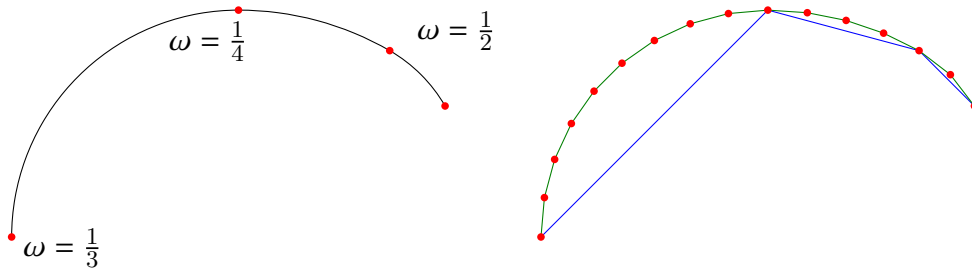


FIGURE 1.1 – Comparaison d’un modèle nodal et d’un modèle en courbure, illustration dans le cas 2D. La courbe à gauche est décrite par des éléments en courbure (ω) de longueur variable. Il y a trois éléments délimités par les points rouges. À droite, le modèle nodal décrit la même courbe. Si l’on se contente d’utiliser les mêmes points que pour les éléments en courbure, on obtient le tracé bleu, très imprécis. Il faut utiliser beaucoup plus de points pour avoir une représentation correcte de la même courbe.

technique d’intégration numérique.

Il existe principalement deux façons de résoudre le problème. La première est d’utiliser un modèle nodal, c’est-à-dire d’échantillonner des points sur la tige, puis d’interpoler la géométrie via des paramètres associés à chacun de ses points. La solution la plus simple est d’associer une position à chaque point, et de les relier entre eux par des segments, mais rien n’empêche d’ajouter par exemple un angle pour décrire la torsion ou une tangente. La deuxième est d’utiliser des coordonnées réduites, c’est-à-dire que le système est paramétré par un nombre minimal de degrés de liberté, ce qui rend une visualisation plus abstraite. On illustre en figure 1.1 ces deux approches.

Clairement, sur la figure 1.1, il semble plus judicieux de prendre la courbure comme degré de liberté pour avoir une courbe bien lisse. Mais le modèle nodal a lui aussi un gros avantage. La matrice du système est plus facile à calculer, car elle est creuse. En effet, l’énergie est exprimée localement avant d’être intégrée sur la tige. Localement, elle ne dépendra que des points voisins, donc d’un nombre limité de degré de liberté. C’est la raison pour laquelle le hessien est creux. Cela n’est pas le cas pour un modèle utilisant des coordonnées réduites. Mais l’approche nodale a aussi des inconvénients. Pour forcer l’inextensibilité, il faut rajouter des contraintes. En effet, les positions sont spécifiées à chaque nœud et non de manière relative à leur voisin. Aussi, le calcul des courbures pose problème. On a besoin des courbures pour calculer certaines énergies élastiques. La courbure se calcule à partir de la seconde dérivée de la courbe. N’ayant qu’un ensemble discret de point, il faut au moins deux voisins pour estimer correctement la courbure en un point. Le nombre de points discrets de la tige doit donc être conséquent. Sur ce point, un modèle réduit montre tout son intérêt : il n’y a pas d’approximation a priori², ni de contraintes. La cinématique est intégrée à partir des degrés de liberté uniquement. Cependant, le système devient dense, car un point dépend de tous les degrés de liberté³. Son coût de calcul est donc bien plus élevé⁴.

close, c’est-à-dire sous forme de combinaisons d’opérations mathématiques simples. Cela n’est pas satisfaisant pour nous, vu que de ce fait l’évaluation de cette solution est numériquement compliquée.

2. La courbure reste cependant contrainte par le choix de la discrétisation.
3. Cela ne concerne pas tout les points, mais le calcul du point en fin de ruban dépend bien de toutes les courbures de la tige.
4. À nombre de degrés de liberté égal. Ainsi, de manière absolue il faut nuancer ce propos.

1.2 Modèles de coque

Historiquement un modèle de plaque est proposé par Kirchhoff, puis Love le développe [Lov88]. Une plaque étant fine, elle est représentée par une surface médiane. Kirchhoff propose trois hypothèses :

- Un segment orthogonal à la surface médiane reste rectiligne lors des déformations.
- Un segment orthogonal à la surface médiane reste orthogonal à la surface lors des déformations.
- L'épaisseur de la plaque est invariante par transformation.

Cela signifie entre autre qu'il n'y a pas de cisaillement transversal à la surface.

Un modèle de coque est un modèle de plaque pouvant gérer de la courbure naturelle. En l'absence de force externe une plaque sera plate, une coque aura une forme quelconque. Classiquement, on utilise une paramétrisation de Monge pour représenter une surface. Elle est alors caractérisée par une fonction $\mathbf{u} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Deux paramètres sont utilisés puisqu'une surface est un objet 2D, le domaine de ces paramètres est Ω . Facilement cette paramétrisation correspond aux coordonnées cartésiennes de la surface aplatie dans un plan. Mais elle peut aussi être plus complexe, par exemple correspondre à des coordonnées polaires. Dans la section 2.2, on utilisera une paramétrisation qui n'est même pas matérielle, c'est-à-dire que pour deux déformations \mathbf{u}_1 et \mathbf{u}_2 et pour un point $\mathbf{t} \in \Omega$, $\mathbf{u}_1(\mathbf{t})$ et $\mathbf{u}_2(\mathbf{t})$ ne correspondent pas à la même particule de la surface.

Dans ce travail, nous utilisons le modèle de coques de Naghdi [Nag63] avec FENICS-SHELL [Hal+18]. Ce dernier est un logiciel de résolution d'équations aux dérivées partielles en éléments finis. On cherche à calculer une fonction f , solution d'une équation différentielle (f peut être la position des points par exemple). Il est généralement impossible de calculer une solution analytique. La surface est d'abord discrétisée via un maillage. On va supposer ici pour simplifier que les conditions aux bords sont nulles. L'équation est alors intégrée sur le domaine Ω . Cela va permettre de scinder le problème. On choisit des fonctions de base v_i , dont la somme vaudra 1 dans le domaine et 0 en dehors. Généralement on prendra une fonction polynomiale valant 1 sur exactement un des nœuds du maillage (P_i), 0 sur les autres. Le maillage est en quelque sorte un support visuel pour les fonctions de base. On peut alors substituer f par $\sum_i v_i f_i$, après quelques calculs on peut par linéarité de l'intégrale retrouver des problèmes indépendants faisant intervenir f et une seule des fonctions de base. L'intérêt de cette décomposition est de pouvoir approcher f par une fonction f_i dans un espace de dimension finie (par exemple un polynôme). Cette approximation n'est pas dérangeante, vu que le produit $f v_i$ est non nul seulement au voisinage du point P_i . On obtiendra alors une solution approchée du problème $\hat{f} = \sum_i (f_i v_i)$.

Dans le cas du modèle de Naghdi les fonctions à résoudre sont la position définie via ses trois coordonnées et la direction de la normale via deux angles. En théorie, il est possible de recalculer toutes les valeurs à partir de la position, mais dû à la discrétisation, comme dans le cas des tiges, il peut être difficile d'estimer la courbure qui, rappelons-le, se calcule à partir du développement de Taylor à l'ordre 2 de la surface (la fonction \mathbf{u}).

On associe trois énergies à une coque : les énergies d'extension, de cisaillement et de flexion. Chacune est associée avec une déformation. On les illustre en figure 1.2. Ces trois énergies constituent l'énergie élastique. Seule la flexion est une déformation isométrique

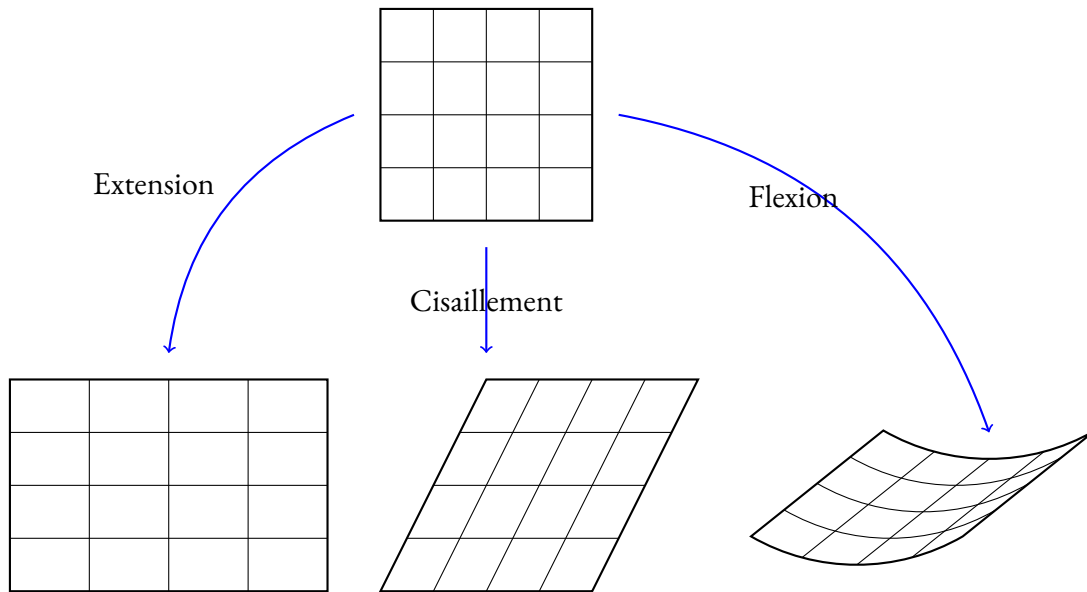


FIGURE 1.2 – Les trois types d’énergie élastique d’une coque ou plaque.

1.3 Surfaces développables

Les surfaces développables sont de plus en plus présentes en informatique graphique. Par exemple, dans [WCS21], les auteurs simulent un livre en 3D, les pages sont alors représentées par une surface développable. Dans [Dud+16], les auteurs utilisent les origamis pour contrôler la courbure, or les origamis sont fait à partir de feuilles de papier, ils sont donc développables. Aussi, la communauté à tendance à appliquer ses modèles à la fabrication. En effet par définition une surface développable est localement isométrique à une surface plane. On peut donc facilement manufacturer une telle surface en coupant et déformant une surface plane inextensible. Des exemples de surfaces développables sont les surfaces côniques, les cylindres généralisés. Par contre, une sphère n’est pas une surface développable. L’inextensibilité n’est généralement qu’une approximation, il y aura toujours une extension, mais elle peut être très minime comme dans le cas du papier. C’est d’autant plus pratique, car ce matériel est courant.

Une des propriétés les plus utilisées pour représenter une surface développable est qu’elle est réglée. C’est-à-dire qu’elle est l’union d’un ensemble de segments de longueur non nulle. Ces segments sont appelés règles. Il est important de préciser que ce n’est pas une condition suffisante pour que la surface soit développable. Les hyperboloïdes sont par exemple réglées, mais non développables. On illustre en figure 1.3 deux surfaces réglées dont un hyperboloïde. Il suffit de rajouter la condition que le plan tangent soit le même le long de la règle pour que la surface devienne développable.

Pour rester développable une surface aura tendance à créer des plis (ou coins), voir à se déchirer. Dans ce cas les plis (ou coins) sont les endroits où la surface n’est pas dérivable. On peut alors décomposer la surface en morceaux développables et lisses. On peut citer par exemple [Sch+15; SRH17] sur le froissement et le déchirement du papier.

Plusieurs méthodes ont vu le jour pour concevoir des surfaces développables. On citera en premier [SF96], [BW07], [Sol+12] qui ont l’avantage de s’appuyer strictement sur la définition de surface réglée. La surface est décomposée en plusieurs parties chacune développable. Elles sont reliées via des règles qui doivent correspondre aux bords connectés. Plus récemment,

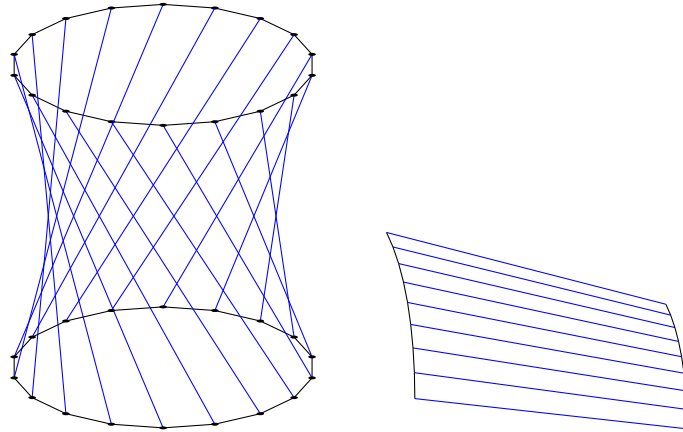


FIGURE 1.3 – Deux surfaces réglées. Les règles sont tracées en bleu. À gauche, un hyperboloïde qui n'est pas développable, à droite un morceau de cône qui est bien une surface développable.

d'autres techniques sont mises en œuvre permettant de travailler directement sur un maillage. On citera notamment [RHS18] et [SGC18]. Dans ce dernier cas les auteurs cherchent à approcher une surface lisse par une surface développable. La surface obtenue est bien continue, mais n'est plus lisse.

Chapitre 2

Sur une description théorique des rubans

2.1 Un bref historique

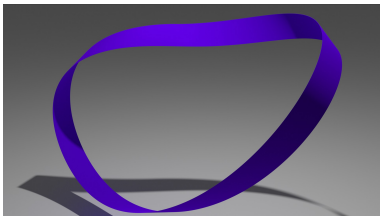


FIGURE 2.1 – Le ruban de MOEBIUS n’a qu’une seule face.

L’intérêt pour les rubans remonte à 1929 où M. Sadowsky publie un premier article sur le ruban de Moebius [Sad29] (originellement en allemand, voir [HF15] pour une traduction en anglais). Le ruban de Moebius a la particularité de n’avoir qu’une seule face, comme le montre la figure 2.1. Conceptuellement, un tel ruban est juste un ruban circulaire que l’on a coupé, puis dont on a tourné une extrémité d’un demi-tour avant de le recoller. Ce qui nous intéresse dans un premier temps ce n’est pas la forme de ce ruban en particulier, mais l’énergie potentielle qui le caractérise. En effet, la forme au repos d’un ruban correspond à un minimum local de l’énergie.

Dans son article, Sadowsky néglige la force gravitationnelle, et donc ne considère que l’énergie élastique. Cette dernière est compliquée à calculer analytiquement, mais en utilisant l’hypothèse d’un ruban étroit (la largeur est très petite devant la longueur) une expression limite de l’énergie est dérivée.

Trente-trois ans plus tard, W. Wunderlich raffine l’expression de cette énergie [Wun62]. Sous l’hypothèse d’inextensibilité, l’énergie devient exacte quelle que soit la largeur. Le ratio $\frac{w}{l}$ largeur sur longueur du ruban a une influence directe sur l’équilibre. Avec Sadowsky, au contraire l’équilibre est invariant vis-à-vis de ce rapport, puisque l’énergie totale du système lui est proportionnelle.

Formellement, un ruban est une surface développable et inextensible dont l’épaisseur reste négligeable devant la largeur, cette dernière étant elle-même négligeable devant la longueur. Nous nous limitons ici au cas des rubans rectangulaires d’épaisseur fixe. Nous motivons cette limitation plus loin dans le texte. De plus nous supposons la surface infiniment lisse, ce qui en pratique contredit toute possibilité de pli. La développabilité implique que chaque déformation du ruban est isométrique : la surface ne peut ni être étirée ni cisailée. Elle ne peut se déformer que par flexion pure. Ainsi par définition, quel que soit l’état du ruban, sa surface est réglée, c’est-à-dire qu’elle est l’union de segments de longueur non-nulle appelés règles. La présence de règles s’avère souvent utile pour paramétrer judicieusement un ruban [DA15; SH07].

E. L. Starostin et G. H. M. van der Heijden sont à notre connaissance les premiers à indiquer comment dériver les équations d’équilibre du ruban depuis l’énergie de Sadowsky et de

Wunderlich [SHo7]. L'article de M. Dias et B. Audoly [DA15] décrit assez précisément la dérivation analytique de la cinématique et de l'énergie élastique du ruban. Les auteurs y dressent aussi un parallèle entre le modèle historique de Wunderlich et le modèle classique des tiges de Kirchhoff [APro]. Nous nous en servons pour la création de notre modèle numérique car il est plus didactique que l'article de Starostin et van der Heijden.

2.2 Une description géométrique

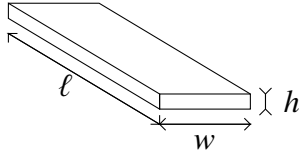


FIGURE 2.2 – Schéma d'un ruban

La figure 2.2 illustre les dimensions du ruban : on note sa longueur ℓ , sa largeur w et son épaisseur h . On rappelle que par définition $\ell \gg w \gg h$.

Une surface développable peut être caractérisée de différentes manières. Voici trois définitions possibles :

1. surface isométrique à une configuration plane ;
2. surface dont la courbure de Gauss est nulle partout ;
3. surface réglée dont le plan tangent est le même le long des génératrices.

La première définition est simple, mais elle n'est pas évidente à utiliser. La seconde semble plus exploitable : dans le domaine de l'informatique graphique, les différents types de courbures sont couramment utilisés. Mais cette définition n'est pas facile à exploiter non plus car contraindre la courbure géodésique analytiquement n'est pas facile¹. Nous nous basons alors sur la troisième approche pour donner une description du ruban.

Une surface réglée est une surface formée par un ensemble de segments appelés génératrices ou règles. Mathématiquement, les segments devraient être des droites mais notre surface est finie. Comme la définition de surface développable l'indique, cette condition est nécessaire pour la développabilité mais pas suffisante.

Nous définissons le long du ruban une abscisse curviligne que l'on note s , puis une courbe $\mathbf{r}(s)$ nommée centerline. Elle traverse le ruban en son centre comme représenté en noir sur la figure 2.3. Ensuite on définit un repère matériel $\mathcal{R}(s) = (\mathbf{t}_x(s) \ \mathbf{t}_y(s) \ \mathbf{t}_z(s))$ le long de cette courbe. Celui-ci s'aligne sur la surface du ruban. Les trois colonnes de cette matrice représentent les axes de ce repère matériel. Ainsi, on aligne $\mathbf{t}_z(s)$ avec la tangente à la centerline, $\mathbf{t}_y(s)$ avec la normale à la surface et $\mathbf{t}_x(s)$ est imposé par le fait que le repère matériel est orthonormal.

$$\mathbf{t}_z(s) = \frac{d\mathbf{r}(s)}{ds} \quad (2.1)$$

Si l'on voulait modéliser le ruban comme une tige, sa géométrie serait caractérisée en définissant une section transversale (dans le plan $(\mathbf{t}_x, \mathbf{t}_y)$) rectangulaire le long de la centerline. On suppose que toutes les génératrices passent par la centerline, on note alors l'ensemble des points d'une génératrice $L(s)$. Il suffit de spécifier un angle $\theta(s)$ entre $\mathbf{t}_x(s)$ et $L(s)$, dans le plan tangent pour reconstituer la surface. Cependant, au lieu d'utiliser l'angle $\theta(s)$, on peut spécifier un scalaire $\eta(s)$, puis un vecteur $\mathbf{q}(s) = \mathbf{t}_x(s) + \eta(s)\mathbf{t}_z(s)$ directeur de la génératrice. On note que ce vecteur n'est pas normé. La surface va alors être représentée par une fonction

¹. des travaux ont été faits dans ce sens. Dans [SGC18], les auteurs proposent une telle approche, mais travaillent directement sur un maillage donc sur l'approximation discrète d'une surface.

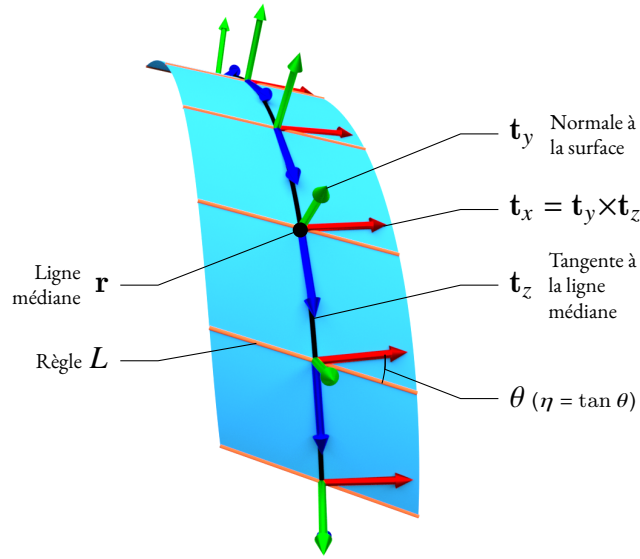


FIGURE 2.3 – Paramétrisation du ruban, avec sa ligne médiane \mathbf{r} , le repère matériel \mathcal{R} qui lui est attaché, et le champ scalaire $\eta = \tan \theta$.

$\phi(s, u) = \mathbf{r}(s) + u\mathbf{q}(s)$. Le domaine de définition est manquant pour l'instant, il faut le déterminer par un raisonnement présenté ci-dessous. Une génératrice est le sous domaine représenté par ϕ à abscisse curviligne fixe.

Il y a un cas où le ruban ne peut pas être représenté. Si la centerline est une génératrice, alors $\eta(s) = \infty$. On écarte le cas où le ruban est plat, car dans ce cas il existe un autre ensemble de génératrices possible (par exemple $\eta(s) = 0$). Il est cependant très improbable dans le cas général qu'une telle situation soit un équilibre puisque la centerline est alors rectiligne (c'est une génératrice) mais de la courbure est présente, donc l'énergie élastique est supérieure à celle au repos. On peut citer l'exemple du mètre ruban, bombé dans la largeur, mais dans ce cas l'équilibre dans cette forme est possible grâce à la courbure naturelle du ruban.

Définissons une opération d'aplatisage : depuis une configuration déformée quelconque, on déforme le ruban pour qu'il reprenne sa forme rectangulaire plate dans le plan, cette déformation est isométrique car la surface est développable. On notera par $\bar{\alpha}$ une quantité α une fois aplatie. Les vecteurs tridimensionnels vont bien sûr perdre une dimension. $\bar{\mathbf{t}}_y$ n'a pas de sens. $\bar{\mathbf{t}}_x, \bar{\mathbf{t}}_z$ sont constants (ne dépendent plus de s). Les règles restent des segments, les angles ne changent pas, donc $\eta = \bar{\eta}$. L'aplatisage est représenté en figure 2.2.

Nous avons défini l'orientation des génératrices, mais pas leur longueur, autrement dit l'espace de définition de ϕ est manquant. L'aplatisage est très efficace pour déterminer ces informations. On peut considérer le point $\bar{\mathbf{r}}_+(s)$ au bord du ruban qui correspond aux paramètres (s, u_+) . On a $\bar{\phi}(s, u_+) = \bar{\mathbf{r}}(s) + u_+\bar{\mathbf{q}}(s)$. Connaissant la largeur du ruban, on peut projeter $\bar{\phi}(s, u_+) - \bar{\phi}(s, 0) = u_+\bar{\mathbf{q}}(s)$ sur $\bar{\mathbf{t}}_x$ pour résoudre $\langle u_+\bar{\mathbf{q}}(s), \bar{\mathbf{t}}_x \rangle = \frac{w}{2}$. On obtient simplement $u_+ = \frac{w}{2}$.

Ce résultat est de prime abord plutôt surprenant, mais c'est surtout une explication du choix d'utiliser le scalaire η plutôt qu'un angle. De manière totalement symétrique on obtient une

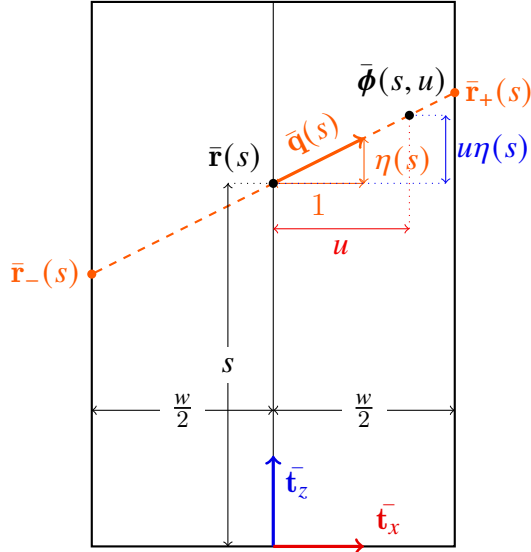


FIGURE 2.4 – Paramétrisation du ruban par s , v et η , sur la configuration plate.

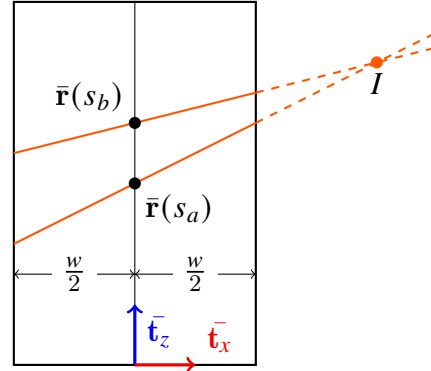


FIGURE 2.5 – Illustration d'un point singulier. Aux points sur la ligne médiane d'abscisse linéaire s_a et s_b passent deux règles, dessinées en orange sur la configuration à plat. La surface étant développable, les droites portant les règles ne peuvent pas se croiser à l'intérieur du ruban.

borne inférieure sur u . On va donc compléter la définition :

$$\begin{aligned} \phi &: [0, \ell] \times [-\frac{w}{2}, \frac{w}{2}] \rightarrow \mathbb{R}^3 \\ (s, u) &\mapsto \mathbf{r}(s) + u\mathbf{q}(s) \end{aligned} \quad (2.2)$$

Il est important de noter que la paramétrisation de la surface n'est pas matérielle. Selon η , le point de paramètre (s, u) ne sera pas le même. Pour des considérations statiques comme c'est le cas dans tout ce travail ça n'est pas très gênant, mais cette erreur se retrouve dans la littérature [She+15].

On peut maintenant expliquer pourquoi les rubans considérés dans ce travail sont rectangulaires. Si ce n'était pas le cas, l'espace de définition de ϕ serait complexe, et dépendrait à priori de η .

L'espace de définition de ϕ est quelque peu problématique. Les deux bords fins du rubans coïncident avec les génératrices $L(0)$ et $L(\ell)$. Or une fois aplatie la forme devrait être rectangulaire. Ce n'est que le cas si $\eta(0) = \eta(\ell) = 0$, c'est-à-dire si les génératrices sont perpendiculaires à la centerline. Or η dépend de la déformation du ruban, donc aux deux bords la représentation du ruban est erronée. Étant donné que la longueur est grande devant la largeur, on ignorera cette imprécision. Tant que η n'est pas trop grand (< 1 par exemple) ce qui est le cas en pratique, les bords seront juste un peu pointus. On reviendra sur ce point plus loin lors du calcul des énergies.

Pour décrire la forme de la surface, la courbure peut-être utilisée. Le vecteur de Darboux peut décrire cette courbure. Le repère matériel $\mathcal{R}(s)$ étant orthonormal, par continuité de celui-ci en fonction de l'abscisse curviligne s , il existe un vecteur $\Omega(s)$ défini par l'équation de Darboux :

$$\mathcal{R}'(s) = [\Omega(s)]_{\times} \mathcal{R}(s) \quad (2.3)$$

L'opérateur $[\cdot]_{\times}$ transforme un vecteur tridimensionnel \mathbf{v} en une matrice $[\mathbf{v}]_{\times}$, telle que

pour tout vecteur \mathbf{u} , $\mathbf{v} \times \mathbf{u} = [\mathbf{v}]_{\times} \mathbf{u}$, où \times est l'opérateur produit vectoriel. La définition plus explicite est donnée ci-dessous :

$$\left[\begin{pmatrix} a \\ b \\ c \end{pmatrix} \right]_{\times} = \begin{pmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{pmatrix}$$

On définit trois courbures (scalaires) en projetant ce vecteur sur le repère matériel. La courbure normale $\kappa_1 = \langle \boldsymbol{\Omega}, \mathbf{t}_x \rangle$ est la rotation autour de l'axe \mathbf{t}_x , la courbure géodésique $\kappa_2 = \langle \boldsymbol{\Omega}, \mathbf{t}_y \rangle$ celle autour de l'axe \mathbf{t}_y et la torsion géodésique $\kappa_3 = \langle \boldsymbol{\Omega}, \mathbf{t}_z \rangle$ celle autour de \mathbf{t}_z . Ordinairement, on interprétera facilement $\boldsymbol{\kappa}$, alors que le vecteur de Darboux reste plus abstrait, mais heureusement, comme expliqué en [CB13], un changement de base nous permet de réécrire l'équation 2.3 en :

$$\mathcal{R}'(s) = \mathcal{R}(s) [\boldsymbol{\kappa}(s)]_{\times} \quad (2.4)$$

Cette équation (2.4) est plus pratique à mettre en œuvre que l'équation (2.3). Il suffit alors de spécifier la courbure sur la centerline seulement pour reconstituer le ruban : on spécifie une condition initiale ($\mathbf{r}(0) = \mathbf{p}_0, \mathcal{R}(0) = \mathcal{F}_0$), on intègre le repère matériel grâce à l'équation 2.4, puis la position en intégrant 2.1. Le tout forme alors un système de Cauchy à résoudre.

On se retrouve ainsi avec la forme du ruban au voisinage de la centerline. Il suffit de prolonger les règles pour obtenir le ruban. Jusqu'à ce point la courbure et les règles ont été traitées indépendamment, il faut donc les relier. Par définition, nous avons $\boldsymbol{\Omega} = \kappa_1 \mathbf{t}_x + \kappa_2 \mathbf{t}_y + \kappa_3 \mathbf{t}_z$. Par commodité on n'écrit pas forcément la dépendance en s . Ici, on insiste sur la différence entre $\boldsymbol{\kappa}$ qui est le vecteur courbure exprimé dans la base du repère matériel, et $\boldsymbol{\Omega}$ qui est le vecteur courbure exprimé dans la base canonique. Dérivons $\mathbf{q}(s) = \mathbf{t}_x(s) + \eta(s)\mathbf{t}_z(s)$:

$$\begin{aligned} \mathbf{q}'(s) &= \mathbf{t}'_x(s) + \eta'(s)\mathbf{t}_z(s) + \eta(s)\mathbf{t}'_z(s) \\ &= \boldsymbol{\Omega}(s) \times \mathbf{t}_x(s) + \eta'(s)\mathbf{t}_z(s) + \eta(s)\boldsymbol{\Omega}(s) \times \mathbf{t}_z(s) \\ &= -\kappa_2 \mathbf{t}_z + \kappa_3 \mathbf{t}_y + \eta' \mathbf{t}_z + \eta \kappa_2 \mathbf{t}_x - \eta \kappa_1 \mathbf{t}_y \\ &= \eta \kappa_2 \mathbf{t}_x + (\kappa_3 - \eta \kappa_1) \mathbf{t}_y + (\eta' - \kappa_2) \mathbf{t}_z \end{aligned}$$

Par définition, le ruban étant une surface développable le plan tangent au ruban est le même le long d'une génératrice, la normale à ce plan est donc \mathbf{t}_y . Cela implique d'avoir une projection nulle de \mathbf{q}' sur \mathbf{t}_y . Il en résulte une relation primordiale :

$$\kappa_3 - \eta \kappa_1 = 0 \quad (2.5)$$

Pour finir, on peut spécifier la courbure géodésique κ_2 , rien n'empêche à la celle-ci de ne pas être arbitraire a priori, ce n'est pas incompatible avec le fait d'être développable. C'est alors la forme du ruban qui nous informe sur cette courbure : en utilisant l'aplatissement $\bar{\mathbf{q}}' = \bar{\mathbf{q}}'$ (l'ordre des opérations dériver et aplatir n'a pas d'importance) et l'égalité $\eta = \bar{\eta}$ on en déduit $\kappa_2 = \bar{\kappa}_2$, puis le ruban étant rectangulaire, sa centerline droite, $\bar{\kappa}_2 = 0$ on obtient :

$$\kappa_2 = 0 \quad (2.6)$$

Ainsi, la géométrie est unidimensionnelle et caractérisée par deux degrés de liberté (2 DdL). Une dernière question doit être posée : Quel est le domaine de chaque variable ? Les courbures n'ont a priori aucune raison d'être bornées. Cependant la direction des règles est soumise à des spécificités géométrique : les règles ne peuvent pas se croiser.



FIGURE 2.6 – Trois configurations d'un ruban. En haut les règles ne se croisent pas. Au milieu, elles se touchent au bord du ruban formant ainsi un cône dont le sommet est le point d'intersection, appelé aussi point singulier, on nomme cette configuration événement de Wunderlich. En bas les règles se croisent au milieu du ruban, on voit clairement un point singulier apparaître, pour former une telle configuration on devrait déchirer la surface.

jetant sur $\bar{\mathbf{t}}_x$, on obtient alors $u_a = u_b$. Puis en projetant sur \mathbf{t}_z , on obtient $s_a + u_a \eta(s_a) = s_a + \delta s + u_a \eta(s_a + \delta s)$ car $\bar{\mathbf{r}}(s) = s \bar{\mathbf{t}}_z$. On peut alors utiliser le développement de η à l'ordre 1 pour en déduire $\eta'(s_a) = \frac{1}{u_a}$. I étant à l'extérieur du ruban on obtient l'inégalité $u_a \geq \frac{w}{2}$ comme montré sur la figure 2.5 ou $u_a \leq -\frac{w}{2}$ si I est de l'autre côté du ruban. Nous obtenons alors la contrainte :

$$\forall s \in \mathbb{R}, |\eta'(s)| \leq \frac{2}{w} \quad (2.7)$$

2.3 Une description énergétique

Un équilibre stable est un minimum local d'énergie potentielle. Nous prenons deux énergies en compte : l'énergie élastique du ruban et l'énergie de pesanteur. Ces deux énergies doivent être intégrées sur toute la surface.

Il faut d'abord spécifier certains paramètres physiques. La constante gravitationnelle g et la verticale \mathbf{e}_g définissent la gravité $\mathbf{g} = -g \mathbf{e}_g$. La masse volumique du ruban est notée ρ , son module de Young Y , son coefficient de Poisson ν . La rigidité de flexion est alors définie par $D = \frac{Yh^3}{12(1-\nu^2)}$.

Nous donnons la possibilité au ruban d'avoir une courbure normale naturelle $\check{\omega}$, c'est-à-dire qu'en l'absence de toute force extérieure la courbure du ruban à l'équilibre sera $\boldsymbol{\kappa}(s) = (\check{\omega}(s), 0, 0)$

Commençons par calculer da l'élément d'intégration. Comme $\boldsymbol{\phi}$ n'est pas matérielle, on n'a pas $da = ds dv$ mais $da(s, u) = \left| \frac{d\boldsymbol{\phi}(s, u)}{ds} \times \frac{d\boldsymbol{\phi}(s, u)}{du} \right| ds dv$. Ainsi $da(s, u) = |(\mathbf{t}_z(s) + u \mathbf{q}'(s)) \times \mathbf{q}(s)| ds dv$.

On illustre le croisement des génératrices sur la figure 2.6. Pour avoir une meilleure intuition, je vous propose de prendre une bande de papier et de la courber pour faire apparaître un point au bord, nommé point singulier illustré sur la figure du milieu. Si l'on essaie de déplacer le sommet du cône à l'intérieur de la surface, le papier va se froisser ou se déchirer. Le cône n'est évidemment pas formé par tout le ruban, mais seulement une partie. On parle d'événement de Wunderlich pour désigner la formation de ce point singulier.

L'aplatissement est pratique pour visualiser et raisonner sur l'apparition du point critique. Nous illustrons cela sur la figure 2.5. On va considérer deux règles aux abscisses s_a et s_b , si elles ne sont pas parallèles elles se croisent en un point I . Le point I peut toucher le bord du ruban mais jamais rentrer dans le ruban. On va noter $s_b = s_a + \delta s$, avec δs aussi petit que voulu.

Exprimons I au bord du ruban comme intersection des deux règles. Ce point aura deux paramétrisations, puisqu'il est sur deux génératrices distinctes : $I = \bar{\boldsymbol{\phi}}(s_a, u_a) = \bar{\boldsymbol{\phi}}(s_b, u_b)$. En pro-

En reprenant la section précédente, avec les contraintes (2.5) et (2.6), $\mathbf{q}'(s) = \eta'(s)\mathbf{t}_z(s)$, on calcule alors $da(s, u) = |1 + u\eta'| dsdv$.

Notons que da s'annule pile quand la contrainte 2.7 est active, ce qui est logique, car les règles s'intersectent en un point.

L'énergie potentielle de pesanteur est définie de manière classique :

$$E_{\mathcal{P}} = - \int_0^\ell \int_{-\frac{w}{2}}^{\frac{w}{2}} \rho h \langle \boldsymbol{\phi}(s, u), \mathbf{g} \rangle da(s, u) \quad (2.8)$$

L'intégration de (2.8) en v donne :

$$E_{\mathcal{P}} = - \int_0^\ell \rho h w \left\langle \mathbf{r}(s) + \frac{\eta'(s)w^2}{12} \mathbf{q}(s), \mathbf{g} \right\rangle ds \quad (2.9)$$

La longueur du ruban étant bien plus grande que sa largeur, le terme $\frac{\eta'(s)w^2}{12} \mathbf{q}(s)$ peut être considéré comme négligeable. Nous omettons donc ce terme dans la suite ce qui a pour effet de simplifier les calculs et ainsi gagner en vitesse de simulation (voir section 3.2). En pratique, nous n'avons pas observé de problème lié à cette approximation. On écrit alors cette forme simplifiée :

$$E_{\mathcal{P}} = - \int_0^\ell \rho h w \langle \mathbf{r}(s), \mathbf{g} \rangle ds \quad (2.10)$$

L'énergie élastique de Wunderlich sans expliciter la dépendance en s s'écrit [SH07; DA15] :

$$E_{\mathcal{W}} = \frac{Dw}{2} \int_0^\ell \left[\kappa_1^2 (1 + \eta^2)^2 \frac{1}{w\eta'} \ln \left(\frac{1 + \frac{\eta'w}{2}}{1 - \frac{\eta'w}{2}} \right) - 2\check{\omega}\kappa_1 (1 + \nu\eta^2) \right] ds \quad (2.11)$$

$E_{\mathcal{W}}$ est calculée pour un matériel hookéen, isotropique et homogène. Pour le terme impliquant la courbure naturelle, la propriété $w \ll \ell$ est utilisée, ce qui signifie concrètement qu'en présence de courbure naturelle, cette énergie n'est pas valide pour une plaque. Cependant, si la courbure naturelle est nulle, pour notre surface développable, cette énergie est exacte quelle que soit la largeur w du ruban.

Dans la limite $\eta'w \rightarrow 0$, l'énergie de Wunderlich $E_{\mathcal{W}}$ se simplifie en l'énergie de Sadowsky,

$$E_S = \frac{Dw}{2} \int_0^\ell \kappa_1^2 (1 + \eta^2)^2 - 2\check{\omega}\kappa_1 (1 + \nu\eta^2) ds \quad (2.12)$$

Il n'est pas courant d'utiliser la contrainte (2.7) avec le modèle de Sadowsky. Cependant, nous l'utilisons systématiquement ici puisqu'elle provient de la développabilité de la surface.

Avant de discrétiser nos équations en vue de la construction du modèle numérique, nous pouvons d'ores et déjà anticiper un certain nombre de difficultés. Tout d'abord, la non-convexité de l'énergie potentielle totale laisse présager des soucis pour les techniques standard d'optimisation (comme la méthode de Newton). Une autre difficulté provient du terme logarithmique dans l'énergie de Wunderlich, qui diverge lorsque la contrainte (2.7) devient active. D'une part, nous devons minimiser l'énergie sous cette contrainte, et d'autre part, il nous faut construire un modèle limite pour l'énergie de Wunderlich lorsque la contrainte devient active.

Deuxième partie

Modèle numérique chaîné de ruban

Chapitre 3

Vers un modèle numérique de rubans

Nous avons annoncé vouloir modéliser un ruban basé sur la courbure. Au vu du chapitre 2, c'est le long de la centerline que l'on discrétise la fonction courbure $s \mapsto \kappa(s)$. Cependant, si l'on discrétise κ directement, la contrainte 2.5 impose $\eta = \frac{\kappa_3}{\kappa_1}$. Des problèmes vont alors survenir car η n'est numériquement pas défini si $\kappa_1 \approx 0$. Nous choisissons alors de discrétiser κ_1 et η . Pour ne pas alourdir les notations, nous notons $\omega = \kappa_1$ la courbure normale. Pour intégrer la cinématique du ruban grâce à ces variables, il faut tout de même spécifier un «point de départ», c'est ce que l'on appelle ici l'encastrement du ruban. Au point $s = 0$, la position est fixée $\mathbf{r}(0) = \mathbf{p}_0$ et le repère matériel aussi, $\mathcal{R}(0) = \mathcal{F}_0$.

Nous découpons le ruban en p segments de longueurs $\ell^{(i)}$, $i \in \llbracket 1, p \rrbracket$. Nos expériences numériques indiquent que $p = 20$ est une bonne valeur¹, au-delà, on ne voit pas de différence géométrique². Sur chacun des segments nous approchons les fonctions ω et η par un polynôme, ce qui nous garantit un nombre de degré de liberté fini par segment.

3.1 Fonctions de forme de haut degré

Contrairement aux modèles de tiges paramétrés en courbures [Ber+06; CB13], prendre nos variables constantes par morceaux ne permet pas d'obtenir une géométrie lisse (dérivable au moins une fois) pour le ruban : en effet, si la fonction η n'est pas une fonction continue de s , alors le ruban n'est même pas continu aux frontières des segments (voir figure 3.1). Afin de garantir une surface suffisamment lisse, nous optons donc pour des fonctions η et ω continues et *linéaires* par morceaux (linéaire par segment). En conséquence, la torsion géodésique $\kappa_3 = \eta \omega$ prend la forme d'une fonction quadratique par morceaux.

Pour un segment i donné, nous notons l'abscisse de départ $s^{(i)}$ puis $\hat{s}^{(i)} = s - s^{(i)}$, et définissons alors $\omega(\hat{s}^{(i)}) = a^{(i)}\hat{s}^{(i)} + b^{(i)}$ et $\eta(\hat{s}^{(i)}) = n^{(i)}\hat{s}^{(i)} + m^{(i)}$, avec $a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)} \in \mathbb{R}$. Cette notation assez chargée ne sert qu'à préciser que les paramètres ne sont spécifiques qu'à un segment, $\hat{s}^{(i)}$ correspondant à une abscisse locale. Par continuité des fonctions ω et η , nous

1. Cela ne concerne que ce chapitre et ses applications. Par la suite, nous enrichissons le modèle de nouvelles conditions aux bords et 20 segments peuvent s'avérer insuffisants.

2. Pour préciser cela de manière formelle, on dira qu'il n'y a pas de différence visuelle ou géométrique entre deux rubans, si la distance de Hausdorff n'excède pas 1‰ de la longueur du ruban. Nous définissons la distance entre un point P et un ensemble \mathcal{S} comme le minimum de la distance (euclidienne) entre P et un point de l'ensemble $d(P, \mathcal{S}) = \min_{Q \in \mathcal{S}} \|P - Q\|$, puis la distance d'un ensemble \mathcal{S} à un autre \mathcal{Z} par $d_a(\mathcal{S}, \mathcal{Z}) = \sup_{P \in \mathcal{S}} d(P, \mathcal{Z})$. Finalement, la distance de Hausdorff est simplement le maximum des deux : $d(\mathcal{S}, \mathcal{Z}) = \max(d_a(\mathcal{S}, \mathcal{Z}), d_a(\mathcal{Z}, \mathcal{S}))$.

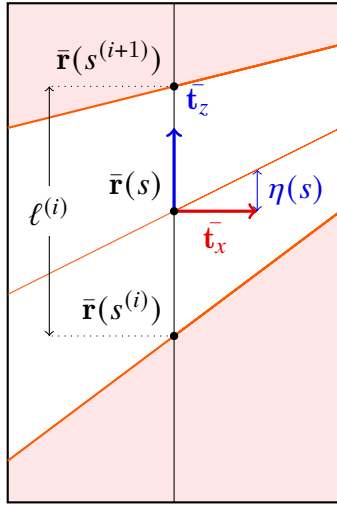


FIGURE 3.1 – Notre élément trapézoïdal (en blanc) sur la configuration aplatie.



FIGURE 3.2 – Prendre le vecteur courbure constant par élément conduit à des règles discontinues, et donc à des éléments disjoints (en haut); en revanche, une courbure normale linéaire et une continuité des règles, mène à une surface de ruban lisse (en bas).

avons les relations

$$b^{(i+1)} = b^{(i)} + \ell^{(i)} a^{(i)} \quad (3.1)$$

$$m^{(i+1)} = m^{(i)} + \ell^{(i)} n^{(i)} \quad (3.2)$$

Une présentation schématique est proposée en figure 3.3. Pour les p segments, nous avons ainsi $2 + 2p$ degrés de liberté. S'il n'y a qu'un seul segment, nous ne précisons pas les indices i pour faciliter la lecture.

La difficulté de notre nouveau problème de Darboux approché (2.4) est, de manière analogue à [CB13], qu'il ne présente pas de solution en forme close. C'est-à-dire que la solution n'est pas exprimable sous forme de combinaisons d'opérations mathématiques simples. Cela est dû à nos éléments de degré 2, en effet pour rappel la torsion géodésique est le produit de η et ω tout deux linéaires par segment. À titre d'exemple les intégrales de Fresnel $\int_0^x \sin(t^2) dt$ et $\int_0^x \cos(t^2) dt$ n'ont pas de forme close. Le recours à une méthode d'intégration numérique est donc nécessaire, avec le risque de transformer cette étape cruciale de construction de nos éléments en étape coûteuse. Heureusement, une méthode très efficace d'intégration, basée sur les séries entières, a été déployée dans [CB13]. Nous montrons que cette méthode s'adapte parfaitement à notre problème de cinématique discrète de ruban.

3.2 Intégration numérique par séries entières

Si le vecteur courbure $\boldsymbol{\kappa}$ est développable en série entière (ce qui est le cas sur chaque segment, car nous avons choisi des éléments polynomiaux), alors \mathbf{r} et \mathcal{R} le sont aussi. En ne considérant qu'un seul segment, nous pouvons ainsi écrire le vecteur courbure comme un polynôme de degré 2, $\boldsymbol{\kappa}(s) = \boldsymbol{\lambda}_0 + \boldsymbol{\lambda}_1 s + \boldsymbol{\lambda}_2 s^2$, avec $\boldsymbol{\lambda}_0 = (b, 0, mb)^\top$, $\boldsymbol{\lambda}_1 = (a, 0, nb + ma)^\top$, $\boldsymbol{\lambda}_2 = (0, 0, na)^\top$. En développant \mathbf{r} et \mathcal{R} sous forme de séries entières $\mathbf{r} = \sum_{n=0}^{+\infty} \mathbf{v}^{(n)} s^n$, $\mathcal{R} =$

$\sum_{n=0}^{+\infty} \mathcal{M}^{(n)} s^n$, on obtient :

$$\begin{cases} \mathbf{v}^{(0)} = \mathbf{p}_0, \mathcal{M}^{(0)} = \mathcal{F}_0 \\ \sum_{n=1}^{+\infty} n \mathcal{M}^{(n)} s^{n-1} = \sum_{n=0}^{+\infty} \mathcal{M}^{(n)} s^n [\boldsymbol{\kappa}(s)]_{\times} \\ \sum_{n=1}^{+\infty} n \mathbf{v}^{(n)} s^{n-1} = \sum_{n=0}^{+\infty} \mathcal{M}^{(n)} s^n \mathbf{e}_z \end{cases} \quad (3.3)$$

On rappelle que le couple $(\mathbf{p}_0, \mathcal{F}_0)$ spécifie l'encastrement du ruban, ou ici du segment. Par unicité des coefficients d'une série entière, notre problème de Darboux revient à résoudre les récurrences suivantes :

$$\begin{cases} \mathbf{v}^{(0)} = \mathbf{p}_0 \\ \mathcal{M}^{(0)} = \mathcal{F}_0 \\ \mathcal{M}^{(1)} = \mathcal{M}^{(0)} [\boldsymbol{\lambda}_0]_{\times} \\ \mathcal{M}^{(2)} = \frac{1}{2} \left(\mathcal{M}^{(1)} [\boldsymbol{\lambda}_0]_{\times} + \mathcal{M}^{(0)} [\boldsymbol{\lambda}_1]_{\times} \right) \\ \forall n \geq 3, \mathcal{M}^{(n)} = \frac{1}{n} \left(\sum_{k=0}^2 \mathcal{M}^{(n-1-k)} [\boldsymbol{\lambda}_k]_{\times} \right) \\ \forall n > 0, \mathbf{v}^{(n)} = \frac{1}{n} \mathcal{M}^{(n-1)} \mathbf{e}_z \end{cases} \quad (3.4)$$

À la différence de [CB13] dans le cas des tiges, nous obtenons ici une récurrence à trois termes pour le repère matériel (au lieu de deux). Néanmoins, la structure du problème reste similaire. En particulier, ce type de série a l'avantage de présenter une convergence très rapide, et ce pour un ordre n relativement faible [CB13; Neh99]. En pratique, il suffit donc de sommer un nombre restreint de termes (au plus une centaine) pour évaluer les séries de manière précise. Cependant, si l'on somme naïvement les différents termes de ces séries en utilisant le calcul flottant, on se rend compte rapidement d'une perte drastique de précision. Ce problème est connu sous le nom de *compensation catastrophique*.

3.3 Compensation catastrophique et contrôle de l'erreur

Les problèmes de perte de précision apparaissent lorsque l'on somme des réels d'ordres de grandeurs très différents, et de signes opposés. Dans les ordinateurs actuels, un réel est représenté par trois grandeurs. Tout d'abord une mantisse $m = (m_i)_{i \in \llbracket 1, k \rrbracket}$ qui est une suite finie de booléens, ensuite un signe $s = \pm 1$ et un exposant n pouvant être négatif. La valeur du réel est alors calculée par $s \times \overline{1}, \overline{m} \times 2^n$, où $\overline{1}, \overline{m} = 1 + \sum_{i=1}^k m_i 2^{-i}$ ³. Pour des calculs en double flottants, $n \in \llbracket -1022, 1023 \rrbracket$ et la mantisse est de taille $k = 52$ bits. Du coup lorsque l'on somme par exemple 10^{16} et 1, on obtient 10^{16} . C'est le phénomène d'absorption : la mantisse n'est pas assez grande pour conserver l'information. Si maintenant on retranche 10^{16} à la somme initiale, on obtient 0 au lieu de 1, soit une perte de précision de 100%! En règle générale ce problème peut être ignoré. Mais dans notre cas, nos séries entières ont des termes qui deviennent rapidement très grands en norme, avant de devenir très petits. De plus, les termes à sommer sont des matrices avec des coefficients de signe positif ou négatif (produits par des matrices antisymétriques). En conséquence, tous les ingrédients sont réunis pour rencontrer le problème de compensation catastrophique.

Dans [CB13], il est expliqué comment borner les facteurs de la série $\mathcal{M}^{(n)} s^n$ pour conserver la précision numérique. Si l'on travaille avec la représentation des réels en double flottant dont la

3. La norme IEEE 754 décrit plus précisément la représentation des types simple précision (32 bits), double précision (64 bits)

mantisse est de $m = 52$ bits, on voudra par exemple une précision égale à la moitié de cette taille, soit $2^{-26} \approx 1.5 \cdot 10^{-8}$, nous devons alors nous assurer que la plus grande valeur de $\|\mathcal{M}^{(n)} s^n\|$ soit bornée par $2^{26} \approx 5 \cdot 10^7$. Casati [Cas15] nous donne le résultat suivant :

$$\|\mathcal{M}^{(n)} s^n\| \leq e^{C(s)} \quad \text{avec } C(s) = 2|s| \sum_{k=0}^2 \|\lambda_k s^k\|$$

On calcule s_{max} une borne sur l'abscisse de telle façon que pour $s \leq s_{max}$ on garantit l'absence de problème de compensation catastrophique. Si on la suppose inférieure à 1, on obtient : $2^{\frac{m}{2}} = \exp(2s_{max} \sum_{k=0}^2 \|\lambda_k\|)$, d'où l'expression simple suivante :

$$s_{max} = \frac{m}{\sum_{k=0}^2 \|\lambda_k\|} \frac{\ln 2}{4}$$

Dans le cas contraire $s_{max} > 1$ on obtient la racine cubique de l'expression précédente :

$$s_{max} = \sqrt[3]{\frac{m}{\sum_{k=0}^2 \|\lambda_k\|} \frac{\ln 2}{4}}$$

Cela nous fournit une borne pour la convergence, que nous appelons rayon de convergence numérique. Ce résultat nous montre que plus la courbure et ses dérivées sont grandes, plus le rayon est petit. La somme de la fonction $C(s) = 2|s| \sum_{k=0}^2 \|\lambda_k s^k\|$ concerne en fait tous les coefficients du polynôme courbure. Ainsi, si l'on augmente les degrés des polynômes ω et η , le rayon de convergence numérique sera en moyenne réduit. C'est l'une des raisons pour lesquelles nous nous sommes limités à des éléments d'ordre 2.

Si le rayon de convergence numérique est inférieur à la taille du segment, il faut alors couper ce dernier, potentiellement en plus de deux sous-segments comme illustré figure 3.3. L'indice d'un segment i est alors complété d'un indice j pour indiquer le sous-segment, ce qui donne $a^{(i,j)} = a^{(i)}$, $b^{(i,j)} = b^{(i)} + a^{(i)} \sum_{k=1}^{j-1} \ell^{(i,k)}$, $n^{(i,j)} = n^{(i)}$ et $m^{(i,j)} = m^{(i)} + n^{(i)} \sum_{k=1}^{j-1} \ell^{(i,k)}$.

Ayant $a^{(i,j)}$ et $n^{(i,j)}$ invariants, mais $b^{(i,j)}$ et $m^{(i,j)}$ croissants en valeur absolue en fonction de j , les sous-segments successifs vont être de plus en plus courts. Cela peut sembler inquiétant, mais dans les simulations numériques effectuées, on constate qu'il n'y a souvent qu'une voire deux découpes maximum par segment.

3.4 Statique de ruban sous gravité

Il nous faut maintenant calculer l'énergie potentielle totale correspondant à notre choix de cinématique pour le ruban. Rappelons que sur chaque segment du ruban, η et ω sont des polynômes d'ordre 1 en s , que l'on écrit $\omega(s) = as + b$ et $\eta(s) = ns + m$.

3.4.1 Énergie discrète de Wunderlich

En développant les formes données dans la section 2.3, l'énergie élastique de Wunderlich (2.11) devient alors, pour un unique segment :

$$E_{\mathcal{W}} = \frac{D_{\mathcal{W}}}{2} [H(n)P(a, b, n, m) - Q(a, b, n, m)]$$

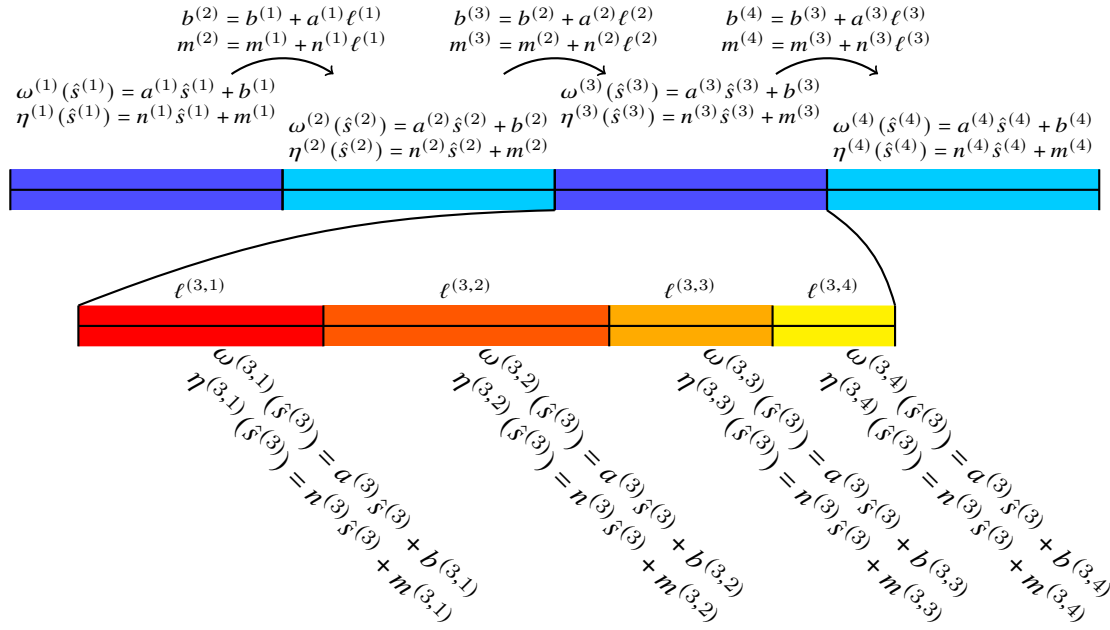


FIGURE 3.3 – Tracé d'un ruban de quatre segments en bleu/violet. Un zoom est fait sur le troisième segment : on voit en dégradé rouge vers jaune les quatre sous-segments. Ceux-ci sont dus au rayon de convergence numérique des séries.

$$\text{avec } H(n) = \frac{1}{nw} \ln \left(\frac{1 + \frac{nw}{2}}{1 - \frac{nw}{2}} \right),$$

$$P(a, b, n, m) = \int_0^\ell (as + b)^2 [1 + (ns + m)^2]^2 ds$$

$$Q(a, b, n, m) = \int_0^\ell 2\check{\omega}(as + b) (1 + \nu(ns + m)^2) ds$$

La seule difficulté ici est le traitement de H . D'une part, H n'est pas défini si $n = \pm \frac{2}{w}$, mais notre contrainte (2.7) doit en théorie nous permettre d'éviter d'atteindre ces bornes. D'autre part, vers $n = 0$, des erreurs numériques apparaissent, voir figure 3.4. Dans ce cas, nous utilisons le développement de Taylor $H(n) = 1 + \frac{1}{3} \left(\frac{nw}{2} \right)^2 + o(n^3)$. Nous basculons vers ce développement dès que $\left| \frac{nw}{2} \right| < \epsilon_T$, où ϵ_T est une valeur seuil empirique. En pratique, nous avons constaté que choisir $\epsilon_T = 10^{-5}$ mène à des résultats satisfaisants en double précision. Ce choix est fait grâce au tracé de la courbe H : visuellement, les erreurs numériques sont flagrantes à partir d'une valeur seuil. Nous avons choisi de représenter la courbure normale naturelle par une fonction linéaire, comme pour la courbure normale. Q s'intègre alors facilement en tant que polynôme. Pour utiliser l'énergie élastique de Sadowsky, il suffit de remplacer H par 1, ce qui simplifie aussi l'implémentation des deux énergies.

3.4.2 Énergie discrète de pesanteur

Comme mentionné plus haut, la longueur du ruban étant très supérieure à sa largeur, nous approchons l'énergie potentielle de pesanteur (2.8) en concentrant la masse sur la ligne médiane,

Problèmes numériques sur la fonction H

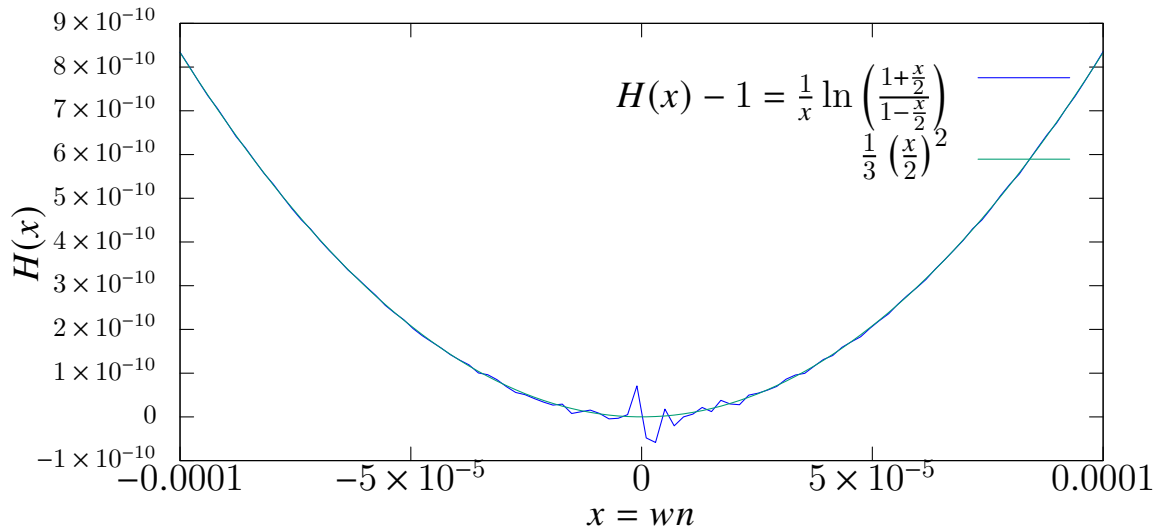


FIGURE 3.4 – Tracé de la fonction H au voisinage de 0 et de son développement limité à l'ordre 2. Pour la visibilité des valeurs de l'axe y , on décale la courbe de 1. On voit des erreurs numériques apparaître si x est trop petit, on choisit alors de substituer H par un développement limité quand les erreurs apparaissent.

$E_{\mathcal{P}} \approx \int_0^\ell wh\rho \langle \mathbf{r}(s), \mathbf{g} \rangle ds$. Cette intégrale peut être calculée à partir des séries entières $E_{\mathcal{P}} = \langle \sum_{k=0}^{+\infty} \mathbf{E}^{(k)} s^k, \mathbf{g} \rangle$. Le calcul exact de l'énergie de pesanteur est possible sans grande difficulté mathématique, cependant il n'est pas nécessaire, car les expériences numériques ne montrent pas de différence géométrique entre la forme complète et la forme simplifiée ⁴.

3.5 Vers l'implémentation du modèle

La discrétisation étant faite, il faut coder le modèle. D'abord, il faut implémenter toutes les quantités nécessaires et leur dérivées première et seconde, cela concerne principalement l'énergie et la géométrie du ruban. Ensuite, l'utilisation d'un algorithme de minimisation appliqué à l'énergie du système permet de trouver un équilibre stable. Nous verrons les principales difficultés d'implémentation, puis nous comparerons plusieurs algorithmes de minimisation d'énergie.

3.5.1 Implémentation des grandeurs en jeu

Pour pouvoir appeler une méthode de minimisation, il faut dans la plupart des algorithmes calculer l'énergie, son gradient et son hessien. J'ai implémenté cela avec le langage C++. C'est un langage compilé, donc offrant de bonnes performances, avec une bibliothèque standard bien fournie. De nombreuses bibliothèques permettent d'ajouter des fonctionnalités, comme la bibliothèque d'algèbre linéaire Eigen. Ce choix est cependant personnel, et j'invite toute personne voulant réimplémenter mon modèle à utiliser le langage qui lui conviendrait le mieux.

4. Cf. la définition au début de ce chapitre.

Pour rendre l'utilisation du code simple, je n'ai pas d'interface graphique, car celle-ci restreindrait l'usage. Au contraire l'utilisation du code en tant que bibliothèque, ou interfacé avec un langage de script (dans mon cas LUA) permet à un utilisateur un contrôle précis de ce qu'il désire faire.

Pour rendre la sortie du code plus riche, il est possible de générer un maillage, c'est-à-dire une représentation tridimensionnelle de la configuration déformée du ruban lisible avec de nombreux logiciels opensource tels que meshlab, Blender⁵ ou commerciaux comme Maya.

Le calcul de la géométrie est relativement simple. En partant du bord du ruban, l'utilisateur a défini un point d'attache $(\mathbf{p}_0, \mathcal{F}_0)$, les paramètres $(a^{(1)}, b^{(1)}, n^{(1)}, m^{(1)})$ permettent alors de réaliser les sous-découpes, et le code peut évaluer chaque série pour obtenir la position des points du segment. Pour passer au second segment, il suffit alors de calculer la position et le repère matériel en fin du premier segment. On itère ainsi sur chaque segment jusqu'à la fin du ruban.

L'intégration de l'énergie de pesanteur se fait sur le même principe. L'énergie élastique (de Wunderlich ou de Sadowsky) ne dépend que des paramètres a, b, n, m et son calcul est plus simple, car son expression mathématique ne comprend pas de série entière. Cependant, chaque segment dépend de son point d'accroche qui correspond à l'extrémité du segment précédent (ou à l'encastrement pour le premier segment). On peut parler de chaînage, car chaque élément dépend de tous ses prédécesseurs. En conséquence, le calcul du gradient est de complexité quadratique, celui du hessien cubique. On note aussi que cette dernière matrice est dense.

Un hessien dense n'est pas forcément problématique, cependant les algorithmes de minimisation utilisent l'inversion de celui-ci qui est plus efficace s'il est creux. De plus on peut arguer que le nombre de degrés de liberté est relativement faible par rapport à ce qu'il serait pour un modèle nodal. Pour quantifier la dimension du problème, il faut un ordre de grandeur du nombre de segments. Dans le simple cas de figure présenté dans les premiers chapitres, le ruban pend à partir d'un simple encastrement, il est seulement soumis à la gravité. Nous rappelons que par expérimentation numérique, nous avons trouvé que 20 segments par ruban semble être une quantité optimale. Le problème a ainsi une cinquantaine de degrés de liberté.

3.5.2 Choix d'un algorithme de minimisation d'énergie

L'énergie rend le problème de minimisation ardu, car elle n'est pas convexe. De plus nous avons une contrainte d'inégalité 2.7. Celle-ci se traduit par une borne sur les variables $n^{(i)}$:

$$\forall i \in \llbracket 1, p \rrbracket, \left| n^{(i)} \right| \leq \frac{2}{w} \quad (3.5)$$

Il existe une grande diversité d'algorithmes pour trouver un minimum local d'une fonction. J'ai essayé un panel représentatif d'algorithmes de minimisation avant de pouvoir trouver une méthode de résolution robuste. Ces algorithmes sont présentés plus loin dans le texte, on trouvera aussi une liste dans le tableau comparatif 3.2. Finalement, le choix s'est porté sur la bibliothèque opensource IPOPT [WBo6], une méthode par point intérieur utilisant entre autre une régularisation du hessien et une recherche linéaire, traitant les contraintes d'égalité, d'inégalités, et les bornes sur les variables⁶. Jusqu'ici nous n'avons que les bornes (3.5) et aucune contrainte

5. Pour illustrer ce manuscrit, les figures tridimensionnelles ont toutes été générées via le script LUA, puis le rendu a été fait sous le logiciel Blender.

6. On peut voir ces bornes comme des contraintes, mais la librairie IPOPT requiert des bornes sur les variables (potentiellement infinies).

complémentaire. Nous donnons à cet algorithme la valeur, le gradient et le hessien des énergies et des contraintes s'il y en a. Ces valeurs sont calculées de manière exacte via les équations explicites tout au long de ce chapitre. S'il n'y a aucune difficulté humaine à calculer et dériver l'énergie, due à sa non-convexité cette dernière reste compliquée à traiter pour un algorithme de minimisation.

Le travail d'étude pour la sélection d'un algorithme de minimisation se base sur notre modèle chaîné comme présenté dans ce chapitre⁷. En particulier, le choix de l'utilisation d'IPOPT a historiquement été fait pour cette version du modèle, même si cette option avait pour avantage d'intégrer facilement des contraintes pour de futures évolutions.

Cet algorithme a démontré une capacité à résoudre un large éventail de scénarii et à surpasser les autres méthodes de minimisation selon les deux critères : absence de divergence et vitesse (voir la table 3.2).

Pour comprendre le rôle joué par l'algorithme, une comparaison de différentes méthodes de minimisation permet d'extraire des éléments jouant un rôle clef :

- Utiliser une méthode du *second ordre*. On observe systématiquement une convergence lente pour toute méthode n'exploitant pas l'information contenue dans le hessien.
- *Régulariser* le hessien pour gérer robustement les cas où il est singulier ou négatif (possède des valeurs propres nulles ou négatives).
- Utiliser une *recherche linéaire* pour améliorer la convergence.
- Finalement, *adimensionner* le problème avant de minimiser. Pour les méthodes du second ordre, nous observons un gain de temps d'un facteur 3 à 10, signifiant que le conditionnement du système est amélioré par l'adimensionnement. Cette étape dont les détails sont donnés au chapitre 4 doit cependant être réalisée au préalable par l'expérimentateur.

J'ai comparé IPOPT, nommé ici IPOPT-HESS avec cinq méthodes standard décrites ci-dessous. Chacune correspond à une instance d'un algorithme générique de minimisation (Algorithme 1). Pour une compréhension plus complète des différentes méthodes d'optimisation, le lecteur peut se référer au livre [NW06].

Algorithme 1 : Algorithme générique de minimisation

Entrées : la fonction à minimiser E , et le point initial (la «graine») \mathbf{x}

tant que le critère de convergence n'est pas atteint **faire**

```

 $\mathcal{A} \leftarrow \text{matriceSystème}(E, \mathbf{x});$ 
 $\mathbf{d} \leftarrow \text{résoudre}_{\mathcal{A}}\{\mathcal{A} \cdot \mathbf{d} = -\nabla E(x)\}$  // Direction de descente ;
 $\alpha \leftarrow \text{tailleÉtape}(E, x, \mathbf{d})$  // Étape ;
 $\mathbf{x} \leftarrow \mathbf{x} + \alpha \mathbf{d};$ 
```

Nous supposons ici que l'énergie est différentiable au moins deux fois, ce qui est le cas pour les rubans. L'opérateur va choisir un point initial \mathbf{x} aussi appelé graine. À chaque étape, une direction de descente \mathbf{d} est calculée et le point \mathbf{x} est déplacé dans cette direction. Le terme «descente» signifie que dans un voisinage de \mathbf{x} , l'énergie diminue dans cette direction (autrement dit cette diminution est juste locale.). La fonction `tailleÉtape` peut être une recherche linéaire par exemple. Elle permet d'une part de réguler l'amplitude du déplacement de \mathbf{x} afin de

7. On parlera dans le chapitre 5 de modèle mixte. La plupart des méthodes de convergence proposées dans ce chapitre ne gèrent pas les contraintes. Le nouveau modèle utilisant des contraintes, il n'est pas possible de refaire les études proposées.

respecter les bornes des variables si besoin. Et d'autre part elle peut aussi accélérer l'algorithme en cherchant un point dans la direction \mathbf{d} qui diminue au mieux possible l'énergie. Il faut quand même faire attention à ce que la recherche ne soit pas trop coûteuse pour garder de l'intérêt. Le critère de convergence est souvent basé sur l'amplitude $|\nabla E(x)|$ du gradient de l'énergie; quand celle-ci est négligeable (par exemple inférieure à une valeur seuil), un point critique (par exemple un point de selle ou un minimum) est probablement atteint.

Méthodes du premier ordre Ces méthodes n'utilisent que la valeur de l'énergie et du gradient pour effectuer la minimisation. L'algorithme standard «*descente de gradient*» nommé ici GRADDESC utilise l'approximation au premier ordre de E autour du point \mathbf{x} et suit l'opposé du gradient de l'énergie : `matriceSystème` renvoie l'identité et `tailleEtape` vaut 1. Après les premières itérations, les méthodes du premier ordre sont malheureusement lentes car le gradient devient petit autour d'un minimum.

Méthodes du second ordre Pour augmenter la vitesse de convergence, une approximation de l'énergie à l'ordre deux peut être utilisée. Dans ce cas, la fonction `matriceSystème` renvoie le hessien $\mathcal{H}E(x)$ de l'énergie. Si nous prenons `tailleEtape=1`, nous obtenons l'algorithme de *Newton*, nommé NEWTON ici. Près de la solution, l'erreur $|\nabla E(x)|$ est petite et la convergence est quadratique. Cependant, cette méthode atteint un extremum, pas nécessairement un minimum. Elle peut aussi diverger pour diverses raisons : (a) la convergence n'est en général garantie que pour les fonctions convexes, ce qui n'est pas le cas pour les rubans, (b) l'approximation au second ordre est locale, si \mathbf{d} est trop grand, le nouveau point \mathbf{x} peut se retrouver n'importe où, et les courbures du ruban seront bien trop élevées, (c) le terme en logarithme $H(n)$ dans l'énergie de Wunderlich peut diverger si la borne $|n| \leq \frac{2}{w}$ n'est pas prise en compte. Notons tout de même que ce dernier point n'est pas d'actualité si l'énergie de Sadowsky est utilisée.

Comme l'algorithme original de Newton ne calcule pas nécessairement une direction de descente dans le cas non convexe, nous introduisons une variante nommé ici NEWTONC. Quand \mathbf{d} n'est pas une direction de descente, c'est-à-dire $\mathbf{d}^T \mathcal{H}E(x) \mathbf{d} < 0$, NEWTONC effectue une étape de descente de gradient à la place d'utiliser \mathbf{d} . De plus une recherche linéaire ainsi qu'une vérification $|n| \leq \frac{2}{w}$ sont utilisées, ce qui concrètement revient à diminuer la valeur de `tailleEtape`. La convergence est améliorée dans certains cas, mais l'algorithme reste souvent bloqué dans des zones non convexes où il effectue des étapes de descente de gradient, retombant alors dans la lenteur d'une méthode du premier ordre.

NEWTONR est défini pour gérer ces régions non convexes. Comme son nom l'indique, c'est un variant de l'algorithme de Newton. La fonction `matriceSystème` introduit une régularisation : en plus de calculer le hessien, elle calcule la valeur propre minimale μ . Si $\mu < 10^{-8}$, le hessien est régularisé en lui ajoutant $10|\mu|I$ (I est la matrice identité). Une recherche linéaire complète la méthode. De cette manière, l'algorithme est quasiment aussi robuste et efficace qu'IPOT-HESS. Il est d'ailleurs probable qu'avec un réglage de paramètres plus poussé les deux méthodes se vaillent.

Méthodes quasi-Newton Comme expliqué en 3.5.1, le coût de calcul du gradient croît quadratiquement et celui du hessien cubiquement avec le nombre de segments p . Les méthodes quasi-Newton évitent de calculer le hessien de l'énergie, mais cherchent tout de même à l'estimer. Un exemple bien connu est l'algorithme BFGS [Bro70] qui construit l'approximation du hessien à partir des gradients successifs du système. Le hessien peut ainsi gagner en précision au cours de l'exécution. Sous l'hypothèse de convexité, il est prouvé que l'approximation converge

	Itérations	# {E} ^a	# {∇E} ^b	# {HE} ^c	temps (sec)
IPOPT-HESS	43	87	44	43	9.8
IPOPT-L-BFGS	> 1000 ^d	15379	1001	0	367.5

a. Nombre d'évaluations de l'énergie

b. Nombre d'évaluations de gradient

c. Nombre d'évaluations de hessien

d. L'algorithme a été arrêté à 1000 itérations, même si la convergence n'était pas atteinte.

TABLE 3.1 – Pour tester l'utilité du calcul exact du hessien de l'énergie, nous calculons l'équilibre d'un ruban avec IPOPT soit en utilisant le hessien réel, soit en demandant à IPOPT de faire lui-même l'approximation du hessien grâce à l'algorithme L-BFGS. Nous utilisons un ruban de longueur $\ell = 50\text{cm}$, largeur $w = 0.5\text{cm}$, d'épaisseur $h = 0.1\text{mm}$, de masse volumique $\rho = 2000\text{kg/m}^3$, de module de Young $Y = 1.8\text{GPa}$, de coefficient de Poisson $\nu = 0.5$ et de courbure normale naturelle constante $\check{\omega} = 80\text{m}^{-1}$. Le ruban est encastré avec un angle $\alpha = \frac{\pi}{4}$ radians par rapport à l'axe vertical.

vers le véritable hessien. Cependant cette méthode requiert un grand nombre d'itérations dans notre cas. IPOPT nous permet d'utiliser L-BFGS, une variante de BFGS où le hessien est réduit à un produit $\mathcal{A}\mathcal{A}^T$ permettant une approximation moins gourmande en mémoire. Là aussi le résultat est insatisfaisant, comme le montre la table 3.1 ⁸.

Évaluation des méthodes La plupart des méthodes citées ne supportent pas les bornes sur les variables, ou tout du moins il faudrait adapter ces méthodes pour qu'elles les supportent. Pour simplifier la comparaison des différents algorithmes, nous supprimons les bornes (3.5). Aussi nous nous restreignons à l'énergie de Sadowsky pour éviter les problèmes numériques. Même si les résultats obtenus ne répondent pas totalement aux besoins, cela permet d'avoir un bon aperçu des éléments clés pour un bon algorithme.

Nous adimensionnons les quantités physiques par la longueur et la rigidité de flexion. Concrètement, cela revient à fixer $D = 1$, $\ell = 1$. Nous choisissons un coefficient de Poisson de $\nu = 0.4$. Il reste alors seulement deux paramètres adimensionnés : $\Gamma = \frac{Mg\ell^2}{Dw}$ et $K_N = \check{\omega}\ell$, où $M = \rho\ell wh$ est la masse totale et $g = 9.81\text{m/s}^2$ est l'accélération de la gravité. La courbure normale naturelle est constante, donc la forme naturelle est un morceau d'anneau, le ruban pouvant faire plusieurs tours sur lui-même.

Trois scénarii sont envisagés, comme illustré sur la figure 3.5. Les paramètres sont choisis pour que la complexité de résolution soit croissante, ainsi le premier a une faible courbure naturelle et un faible poids ($K_N = -10$, $\Gamma = 100$) quand le dernier est bien bouclé et pesant ($K_N = -100$, $\Gamma = 7500$). La verticale est choisie telle que $\mathbf{e}_g = \mathbf{e}_x$, puis le ruban est encastré avec les orientations suivantes : $\mathbf{t}_x(s=0) = \frac{\sqrt{2}}{2}\mathbf{e}_x + \frac{\sqrt{2}}{2}\mathbf{e}_y$ et $\mathbf{t}_z(s=0) = \mathbf{e}_z$, formant ainsi un angle de $\frac{\pi}{4}$ radians par rapport à la verticale. Il faut finalement préciser la graine (un point initial) pour l'algorithme de minimisation. Nous choisissons la forme naturelle. Pour le premier scénario, nous divisons le ruban en 10 segments, pour les deux autres en 20 segments.

Les résultats obtenus sont listés en table 3.2. Les méthodes de descente de gradient, Newton pur, ou le mélange NEWTONC ne convergent pas vers l'équilibre depuis la forme naturelle. Au

8. Un esprit affûté pourrait arguer que BFGS pourrait fonctionner car il utilise une matrice pleine, là où L-BFGS utilise une représentation "compressée". Cependant dans nos tests, il est possible de régler la taille de la matrice \mathcal{A} . Nous l'avons prise de taille carrée pour se ramener à BFGS.

contraire, les méthodes utilisant le hessien et le rendant positif par régularisation convergent dans tout les cas. L'ajout d'une recherche linéaire réduit parfois la taille de l'étape et permet à l'algorithme de ne pas envoyer le point trop loin, comme expliqué plus tôt.

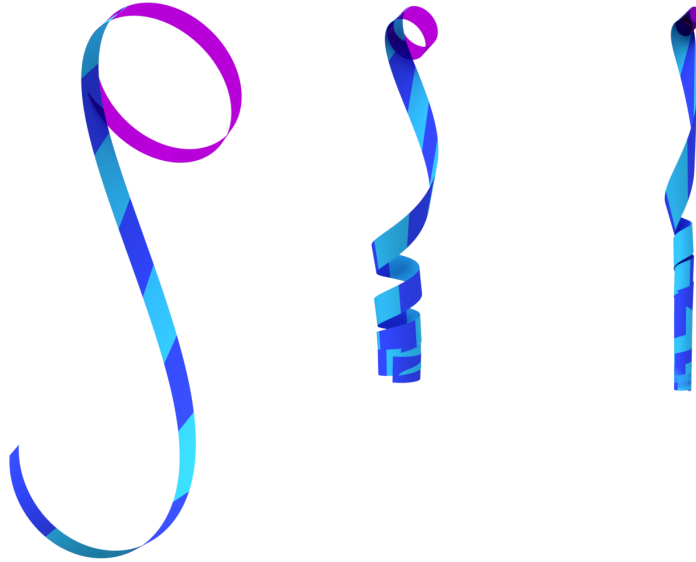


FIGURE 3.5 – Trois cas tests de complexité croissante : (gauche) $K_N = -10$ et $\Gamma = 100$, simulé avec $p = 10$ éléments, (milieu) $K_N = -40$ et $\Gamma = 1200$, simulé avec $p = 20$ éléments, (droite) $K_N = -100$ et $\Gamma = 7500$, simulé avec $p = 20$ éléments. La forme naturelle est représentée en violet. Cette configuration sert de graine (d'état initial) pour l'algorithme de convergence. Les deux tons de bleu servent à distinguer les segments du ruban à l'équilibre. Pour des besoins de comparaison, l'énergie de Sadowsky est utilisée. Les comparaisons entre algorithmes sont présentées dans la table 3.2.

Méthode	$K_N = 10, \Gamma = 100$	$K_N = 40, \Gamma = 1200$	$K_N = 100, \Gamma = 7500$
GRADDES	- (> 500 pas)	- (> 600 pas)	- (> 600 pas)
BFGS	272 pas, 0.5 sec.	506 pas, 6.2 sec.	554 pas, 30 sec.
NEWTON	- (diverge)	- (diverge)	- (diverge)
NEWTONC	- (> 500 pas)	- (> 500 pas)	- (> 500 pas)
NEWTONR	14 pas, 0.1 sec.	24 pas, 1.2 sec.	46 pas, 8.6 sec.
IPOPT-HESS	11 pas, 0.09 sec.	23 pas, 0.9 sec.	35 pas, 5.8 sec.
AUTO	30 pas, 0.11 sec.	45 pas, 0.16 sec.	167 pas, 0.9 sec.
SHOOTING	42 pas, 8 sec.	78 pas, 16 sec.	784 pas, 360 sec.

TABLE 3.2 – Mesure de la durée du calcul de l'équilibre en itérations (pas) et en secondes, sur les trois scenarii décrits dans le texte et illustrés en figure 3.5 pour divers algorithmes. Parmi les méthodes appliquées sur notre modèle discret (haut du tableau), seul NEWTONR et IPOPT-HESS sont robustes : tout deux combinent l'utilisation de l'approximation exacte du second ordre et de la régularisation. Parmi les méthodes de continuation utilisant une formulation forte des rubans dont nous parlerons dans le chapitre 4 (bas du tableau), AUTO se montre très rapide et devance même nos meilleures méthodes de minimisation (en temps).

Chapitre 4

Premières expériences numériques : où l'on montre la nécessité du modèle de ruban

Dans ce chapitre, nous réalisons les premiers tests avec notre modèle. Il est assez naturel de se comparer aux modèles existants afin de se valider. En mécanique, il est classique d'utiliser une méthode de tir pour résoudre un système. C'est ce que l'on utilise dans ce chapitre avec le modèle de Sadowsky pour valider notre implémentation. Par ailleurs, le modèle de ruban étant à mi-chemin entre un modèle de tiges et de coques, il est tout naturel de comparer le modèle de ruban avec ces derniers. On choisira l'utilisation de FENICS-SHELL, une bibliothèque spécialisée dans les modèles à éléments finis, appliquée au modèle de plaque de Naghdi [Nag63]. Le modèle de coques étant plus général que le modèle de rubans, on s'attend à ce que les résultats coïncident, avec un coût de calcul moindre pour notre modèle réduit. Pour la comparaison au modèle de tiges, on s'appuie sur [CB13], un modèle de tiges très proche de celui des rubans, puisqu'il a inspiré notre modèle avec la représentation par la courbure et la technique des séries entières pour calculer la géométrie. Si les modèles coïncident cela montrera alors un intérêt limité au modèle du ruban puisqu'un simple modèle de tige pourrait être utilisé à la place de notre modèle. Si au contraire, les équilibres calculés diffèrent, l'utilisation d'un modèle spécifique au ruban prend tout son sens.

4.1 Adimensionnement

Avant de comparer différents modèles, il convient d'ajouter une étape d'adimensionnement. Comme nous l'avons évoqué en 3.5.2, l'adimensionnement est une étape importante pour l'efficacité du modèle numérique, la raison en est assez simple : les valeurs numériques en jeu tournent autour de l'unité. Prenons par exemple les coefficients des deux énergies élastique et de pesant. Nous avons $C_e = \frac{Dw}{2}$ et $C_m = \rho whg$. On a $\nu \in [0, 0.5]$, on estime alors $12(1 - \nu) \approx 10$. On peut prendre la masse volumique égale à 1000 kg.m^{-3} . Le module de Young peut être assez variable selon la rigidité du matériel. Prenons une bande de papier usuelle d'un centimètre de large, son épaisseur est d'environ $h = 10^{-4} \text{ m}$, le module de Young $Y \approx 4 \cdot 10^9 \text{ Pa}$ on obtient $C_e \approx 10^{-4}$ et $C_m = 10^{-2}$. On se retrouve alors avec deux coefficients petits, ce qui entraîne des difficultés de résolution pour IPOPT : expérimentalement on gagne un ordre de grandeur en temps après adimensionnement.

Il y a un autre avantage à l'adimensionnement : certains paramètres sont liés. Quelques soit le scénario, on obtiendra les mêmes résultats via certaines transformations. Il convient alors de supprimer cette invariance (ou tout du moins de la calculer). Commençons par ramener la longueur à 1, la variable a devient \underline{a} dans cette nouvelle configuration. Ainsi $\underline{\ell} = 1$, $\underline{w} = \frac{w}{\ell}$, $\underline{s} = \frac{s}{\ell}$, $\underline{\omega}(s) = \omega(s)\ell$, $\underline{\check{\omega}}(s) = \check{\omega}(s)\ell$, $\underline{\eta}(s) = \eta(s)$, $\underline{\mathbf{r}}(s) = \frac{\mathbf{r}(s)}{\ell}$ et $d\underline{s} = \frac{ds}{\ell}$.

On en déduit $\eta'(s) = \frac{d\eta(s)}{ds} = \frac{d\eta(\frac{s}{\ell})}{ds} = \frac{\eta'(\frac{s}{\ell})}{\ell}$, soit $\eta'w = \underline{\eta}'\underline{w}$. Cela nous permet d'adapter l'équation de l'énergie de Wunderlich (2.11) en :

$$E_{\mathcal{W}} = \frac{Dw}{2\ell} \int_0^1 \left[\underline{\omega}^2(1 + \underline{\eta}^2)^2 \frac{1}{\underline{w}\underline{\eta}'} \ln \left(\frac{1 + \frac{\eta'w}{2}}{1 - \frac{\eta'w}{2}} \right) - 2\underline{\check{\omega}}\underline{\omega}(1 + \nu\underline{\eta}^2) \right] d\underline{s} \quad (4.1)$$

ou pour Sadowsky :

$$E_S = \frac{Dw}{2\ell} \int_0^1 \left[\underline{\omega}^2(1 + \underline{\eta}^2)^2 - 2\underline{\check{\omega}}\underline{\omega}(1 + \nu\underline{\eta}^2) \right] d\underline{s} \quad (4.2)$$

Pour l'énergie de pensateur, on obtient :

$$E_{\mathcal{P}} = \rho h w g \ell^2 \int_0^1 \langle \underline{\mathbf{r}}(s), \mathbf{e}_g \rangle d\underline{s} \quad (4.3)$$

Il ne faut pas oublier la contrainte (2.7) qui reste identique, $\forall \bar{s} \left| \underline{\eta}'(\bar{s}) \right| \leq \frac{2}{\underline{w}}$.

Nous avons donc trouvé une loi d'échelle. Pour trouver l'équilibre, il faut minimiser la somme, on peut donc diviser l'énergie par un facteur positif $\frac{Dw}{\ell}$ sans modifier les équilibres statiques, pour se retrouver avec deux paramètres, $\Gamma = \frac{\rho h g \ell^3}{D}$ et ν . En pratique dans un code non adimensionné, il faut poser $\ell = 1$, adapter la largeur du ruban en conséquence et poser $D = 1$ puis $\rho = \frac{\Gamma}{hg}$. Le coefficient de Poisson ne rentre en jeu que s'il y a une courbure normale naturelle non nulle.

Remarquons qu'en utilisant l'énergie de Sadowsky, le système ne dépend de \underline{w} que pour la contrainte (2.7), tandis que la somme des énergies, une fois mise à l'échelle ne dépend que de Γ . Ce n'est pas le cas avec l'énergie de Wunderlich où la somme des énergies dépend de \underline{w} . On illustre donc par ce point la limitation de l'approximation de Sadowsky.

Ainsi on obtient trois paramètres adimensionnés : Γ , $\frac{w}{\ell}$ et ν . Si une simulation n'est pas adimensionnée, il est toujours possible de faire la conversion a posteriori. C'est en particulier utile pour faire des comparaisons avec l'expérimental où certains paramètres sont limités par des contraintes physiques. Aussi certains simulateurs ont parfois des spécificités qui imposent un adimensionnement a posteriori.

4.2 Comparaison aux méthodes de continuation

Pour valider notre modèle, nous l'avons comparé à une méthode numérique alternative. Nous avons ainsi implémenté deux méthodes de continuation, inspirées de [MGN18], qui permettent de résoudre le problème au bord posé sur les équations différentielles ordinaires régissant la statique des rubans avec courbure naturelle.¹ Ces méthodes dérivent les équations diffé-

1. Une différence majeure entre la minimisation d'énergie et les méthodes de tir reste cependant la nature des équilibres : la minimisation comme son nom l'indique cherche les équilibres stables tandis que le tir s'intéresse à

rentielles du gradient de l'énergie du système, dont les détails sont donnés dans l'appendice de [Cha+20]. À notre connaissance, ce type de méthode n'a jamais été appliquée avec l'énergie de Wunderlich combinée avec la contrainte d'inégalité (2.7) et la courbure naturelle. En ce sens, cette méthode est moins générale que la nôtre, et nous nous contentons donc de l'énergie de Sadowsky sans contrainte, afin d'opérer une comparaison avec notre méthode.

La première méthode utilisée est une méthode classique de tir. Les équations différentielles sont intégrées avec un solveur du logiciel MATHEMATICA, et les inconnues sont trouvées grâce au solveur numérique de recherche de racines. Cette façon d'opérer est facile à mettre en place, mais n'est pas robuste.

La seconde méthode de continuation se base sur la bibliothèque Fortran AUTO [DKK91a; DKK91b; DO12]. La discrétisation spatiale est basée sur la collocation orthogonale sur des polynômes de Lagrange de degré $NCOL$ utilisant $NTST$ segments. Une fois discrétisée, le problème aux limites prend la forme d'un système algébrique non linéaire de taille $NDIM \times NCOL \times NTST$, où $NDIM$ est la taille du système différentiel (dans le cas présent $NDIM = 20$). Le tout est résolu avec une méthode de Newton-Chord utilisant la différentiation numérique pour le calcul du jacobien (l'équivalent du hessien dans formalisme de la minimisation).

Nous comparons la configuration d'équilibre obtenue par notre méthode à celle obtenue par la méthode de tir. Les paramètres utilisés sont : épaisseur $h = 0.1$ mm, largeur $w = 1.5$ cm, longueur $\ell = 50$ cm, courbure naturelle $\tilde{\omega} = 30$ m⁻¹, masse volumique $\rho = 1000$ kg/m³, coefficient de Poisson $\nu = 0.5$, et module de Young $Y = 1.8$ GPa. Les formes calculées à l'équilibre via les deux méthodes, tracées sur la figure 4.1, sont identiques, ce qui valide notre approche sur le modèle de Sadowsky.

À noter que contrairement à la méthode de tir qui peine à converger dès que la courbure naturelle du ruban est élevée, notre modèle converge en quelques secondes seulement, même pour une courbure naturelle élevée de $K_N = \tilde{\omega}\ell = 100$.

Reprenons le tableau de comparaison 3.2 des trois scénarii illustrés sur la figure 3.5. Cette fois nous cherchons à obtenir les équilibres stables par continuation.

Dans les routines de continuation, K_N est directement fixé à sa valeur cible et Γ est progressivement augmenté de 0 à la valeur désirée. Pour chaque valeur intermédiaire de Γ l'équilibre est calculé. Dans AUTO on utilise un polynôme de degré $NCOL = 4$ et $NTST = 10$ éléments pour le premier scénario, $NTST = 15$ éléments dans le second et $NTST = 35$ éléments dans le troisième. Comme affiché dans la table 3.2, la méthode de continuation AUTO reste la plus rapide même si le nombre d'itérations requis (c'est-à-dire les valeurs intermédiaires de Γ utilisées) est supérieur au nombre d'itérations requises par la minimisation.

Ainsi AUTO se montre plus performant que notre meilleur algorithme de minimisation

toute configuration critique (c'est-à-dire équilibres stables et instables). Nous nous restreignons donc aux équilibres stables dans la comparaison.

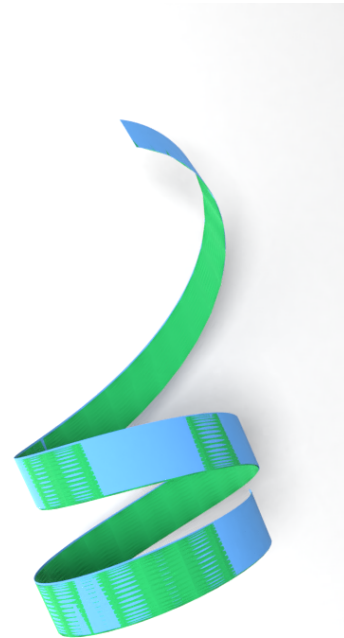


FIGURE 4.1 – La même configuration trouvée par notre méthode (en bleu) et la méthode de tir (en vert).

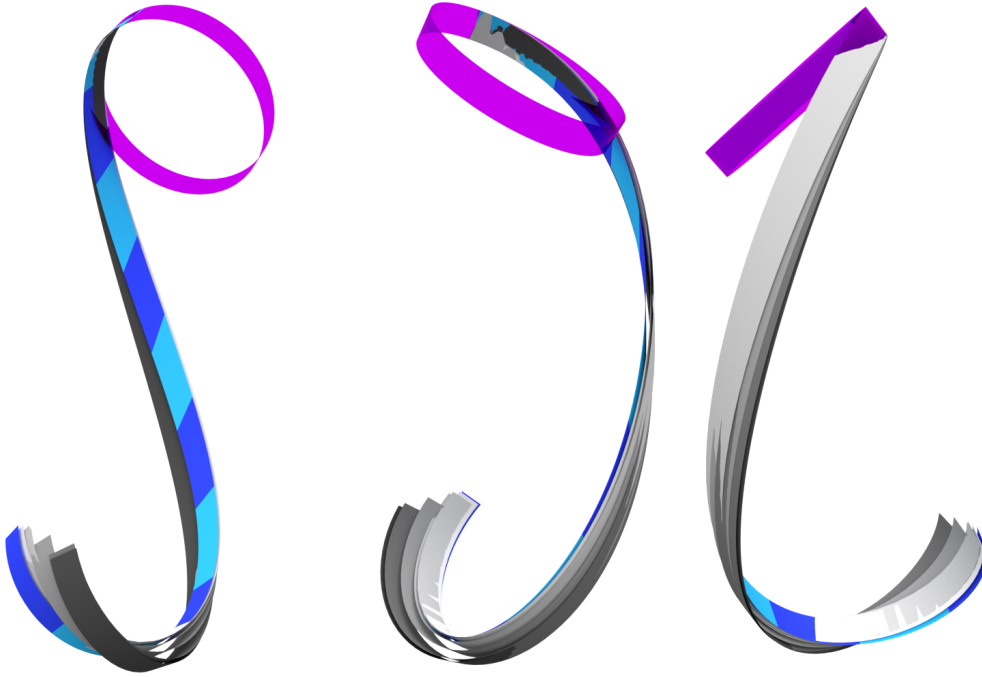


FIGURE 4.2 – Comparaison de notre méthode (en bleu) avec le code éléments finis FENICS-SHELL (5 rapports largeur/épaisseur augmentant avec une échelle de couleur grise), sous trois points de vue. Le scénario est repris de la partie 3.5.2. On voit le code éléments finis converger vers notre méthode quand l'épaisseur devient négligeable devant la largeur.

sur le problème de Sadowsky, spécialement avec un grand nombre d'éléments. Cependant, l'approche par continuation ne peut pas facilement gérer les contraintes d'inégalité. En particulier, la contrainte (2.7) du modèle de Wunderlich est difficile à mettre en place. À notre connaissance, seul [MH19] a proposé un moyen de traiter cette contrainte par une méthode de continuation en ajoutant un terme de régularisation à l'énergie de Wunderlich. Les auteurs reconnaissent cependant que cette solution ad-hoc perturbe l'équilibre du ruban quand la largeur w augmente, ce qui la rend utilisable pour une largeur assez petite. Pour les rubans plus larges, la contrainte de Wunderlich devient cruciale, et nous montrons plus tard que son absence peut mener à une divergence du code. Notre méthode au contraire est plus versatile et peut traiter cette contrainte correctement.

4.3 Comparaison à un modèle de coque

Nous avons également comparé notre modèle à un modèle de coque mince discrétisé par éléments finis. Cette comparaison réutilise le plus simple des scénarii présenté de la partie 3.5.2, illustré en figure 3.5 à gauche, c'est-à-dire que la courbure normale naturelle adimensionnée du ruban est fixée à $K_N = 10$, le paramètre physique adimensionné est $\Gamma = 100$, $\nu = 0.4$. Il faut définir une dimension au ruban puisque les méthodes éléments fini s'appuient sur un maillage. En théorie ce choix est inclus dans le coefficient Γ . Pour l'énergie de Sadowsky, le ratio $\frac{w}{\ell}$ n'a pas d'influence tant que la contrainte (2.7) n'est pas active. Le choix est fait de prendre $\frac{w}{\ell} = 0.03$. Le maillage triangulaire est une grille régulière de 100×4 .

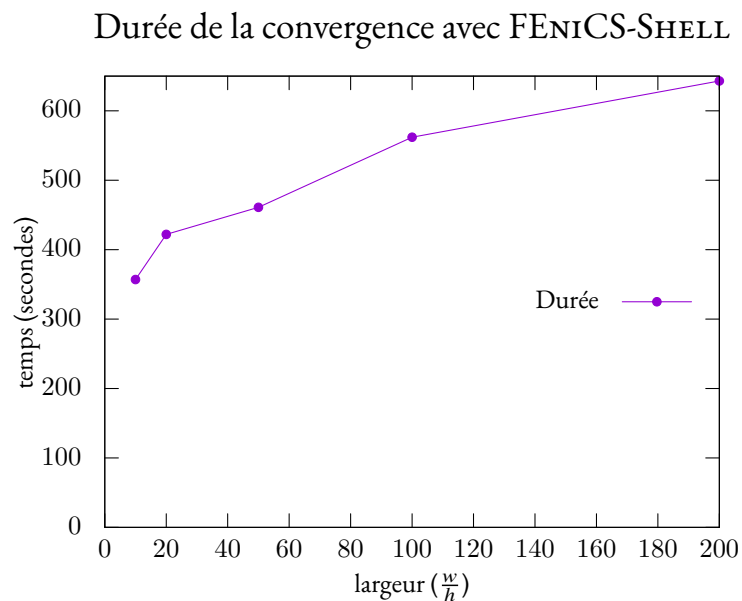


FIGURE 4.3 – Temps de convergence pour FENICS-SHELL selon l'épaisseur. La description de l'expérience est en section 4.3.

Nous utilisons le modèle de Naghdi [Nag63], codé en Python avec la bibliothèque FENICS-SHELL² pour calculer un équilibre. C'est un modèle général de coques fines basé sur une surface combinée avec directions transverses rectilignes mais non nécessairement normales à la surface (nommé couramment hypothèses cinématiques de Reissner-Mindlin). Les modèles de ruban élastiques dérivés pour des structures respectant la relation $\ell \gg w \gg h$, viennent de la théorie des coques où l'énergie d'extension et de cisaillement sont supprimées, puis l'énergie de courbure est intégrée sur la largeur w . Dans [MGN18] il est montré que la comparaison avec l'expérimental fonctionne bien pour un ratio $\frac{w}{h}$ grand, typiquement $\frac{w}{h} > 70$, mais échoue à reproduire le comportement de la structure pour de petit ratios $\frac{w}{h} < 30$.

Expérimentalement, les codes éléments finis sont souvent soumis à un problème de verrouillage et peuvent difficilement calculer l'équilibre d'une structure si celle-ci est trop fine. Pour calculer l'équilibre, nous augmentons progressivement et linéairement la courbure naturelle K_N , puis la gravité Γ . Nous utilisons 5 largeurs : $\frac{w}{h} \in \{10, 20, 50, 100\}$. La convergence prend alors en quatre et dix minutes (illustré en figure 4.3). En comparaison, notre modèle met 0.2 secondes, soit trois ordres de grandeur en moins.

Les résultats sont montrés en figure 4.2. On voit les différentes étapes de convergence. On constate effectivement un rôle important de l'épaisseur et finalement la concordance lorsque l'épaisseur tend vers zéro. Donc nous validons ici aussi notre approche. On reproche cependant au code de FENICS-SHELL d'être lent par rapport à notre approche (plusieurs minutes contre environ une seconde)³. Ainsi, pour simuler un ruban inextensible élastique, notre modèle réduit montre une belle efficacité.

2. <https://fenics-shells.readthedocs.io/en/latest/>

3. Il faut ici relativiser les propos. Notre comparaison ne montre pas que FENICS-SHELL est mauvais a priori. En effet, c'est un code de coques donc il traite des scénarii plus généraux qu'un ruban d'épaisseur négligeable.

4.4 Simulation d'une anglaise et comparaison au modèle des tiges

Il est habituel d'utiliser un modèle de tige de Kirchhoff à section rectangulaire pour simuler un ruban. Néanmoins, il a été montré que le modèle de ruban génère des configurations statiques qui peuvent être très différentes de celles engendrées par un modèle de tige [DA15; MGN18]. En effet, dans le cas des tiges la surface est bien réglée, mais les règles sont orthogonales à la centerline et la courbure géodésique est libre au lieu d'être nulle. Cela constitue une différence majeure entre les deux modèles : la surface par un modèle de tiges n'est pas développable.

Pour simuler une tige de Kirchhoff, nous utilisons le modèle discret en super-clothoïde [CB13]. L'énergie de gravité est la même pour la tige et le ruban (dû à notre approximation faite de l'équation (2.10) à (2.9)), donc seule l'énergie élastique diffère.

La matrice de rigidité pour une tige à section rectangulaire est :

$$\mathcal{K} = \begin{pmatrix} YI_1 & 0 & 0 \\ 0 & YI_2 & 0 \\ 0 & 0 & \frac{Y}{2(1+\nu)}J \end{pmatrix}$$

où $I_1 = \frac{h^3 w}{12}$, $I_2 = \frac{h w^3}{12}$, et $J = J_0 \left[1 - \frac{192}{\pi^5} \frac{h}{w} \sum_{k=0}^{\infty} \frac{\tanh\left(\frac{(2k+1)\pi w}{2h}\right)}{(2k+1)^5} \right]$ avec $J_0 = \frac{1}{3} w h^3$ [Luro5]. Les termes de cette série étant en $O\left(\frac{1}{k^5}\right)$, sa somme partielle converge rapidement : avec moins de 10 termes, la précision numérique est celle de la machine. L'énergie élastique s'écrit alors :

$$E_{\mathcal{T}} = \frac{1}{2} \int_0^{\ell} \mathcal{K}_{1,1} (\kappa_1 - \ddot{\omega})^2 + \mathcal{K}_{2,2} \kappa_2^2 + \mathcal{K}_{3,3} \kappa_3^2 ds$$

De manière à recaler la rigidité de courbure YI_1 du modèle de tige sur son équivalent pour les rubans Dw , nous multiplions YI_1 par $\frac{1}{1-\nu^2}$. Les deux modèles ont alors la même rigidité par rapport à la flexion normale.

Pour tester cette comparaison, nous simulons une anglaise, c'est-à-dire un ruban recourbé plusieurs fois sur lui-même et qui pend sous son poids propre. Pour une meilleure comparaison, nous allons faire varier la longueur du ruban. Notons que cela ne veut pas dire que la longueur change dynamiquement, mais que nous réalisons un calcul d'équilibre pour une longueur fixée. Nous choisissons les dimensions suivantes : longueur $\ell \in [1, 50]$ cm, largeur $w = 1$ cm, épaisseur $h = 0.1$ mm et les paramètres physiques suivants : masse volumique $\rho = 1000$ kg m⁻³, module de Young $Y = 1.8$ GPa, coefficient de Poisson $\nu = 0.5$, l'accélération de la gravité $g = 9.81$ m s⁻². La courbure normale naturelle est fixée à $\ddot{\omega} = 60$ m⁻¹, le ruban dans sa longueur maximale fait donc environ cinq tours sur lui-même dans sa configuration naturelle.

Numériquement, la structure est divisée en 20 segments, et nous choisissons une configuration initiale où le ruban est parfaitement plat et horizontal. Pour calculer l'équilibre de la tige, nous utilisons le modèle de super-clothoïdes [CB13]. Celui-ci est en effet proche de notre modèle du ruban car il sous divise la tige en segment à laquelle il associe une courbure, et la géométrie est aussi résolue par des développements en séries entières. De plus le code est disponible en ligne ⁴. Il faut cependant adapter le code qui est dynamique pour faire du statique : nous attendons

4. <http://bipop.inrialpes.fr/~casati/publications/codes/ssc.html>

simplement assez d'itérations pour que la tige ne bouge plus. Il a tout de même fallu adapter le code pour changer la section de la tige qui était elliptique pour une forme rectangulaire, mais cela revient juste à adapter le calcul de la matrice \mathcal{K} .

Quelques résultats sont présentés sur la figure 4.4, où quelques étapes caractéristiques montrent la superposition du modèle de ruban en bleu avec le modèle de tige en orange. À petite longueur, l'énergie élastique domine et le ruban adopte une forme proche de sa courbure naturelle. Ensuite, à partir de 5 cm, la gravité n'est plus négligeable, mais la torsion géodésique reste élevée et le ruban se déplie mais reste dans le plan. Finalement à partir de 29 cm le ruban bascule hors du plan et la torsion apparaît. La tige colle au ruban tant que celui-ci reste dans le plan, ce qui n'a rien d'extraordinaire puisque l'énergie concernant la courbure normale est identique pour les deux modèles. Quand le ruban est grand des différences notables apparaissent. Le modèle de Kirchhoff échoue à capturer la développabilité de la surface.

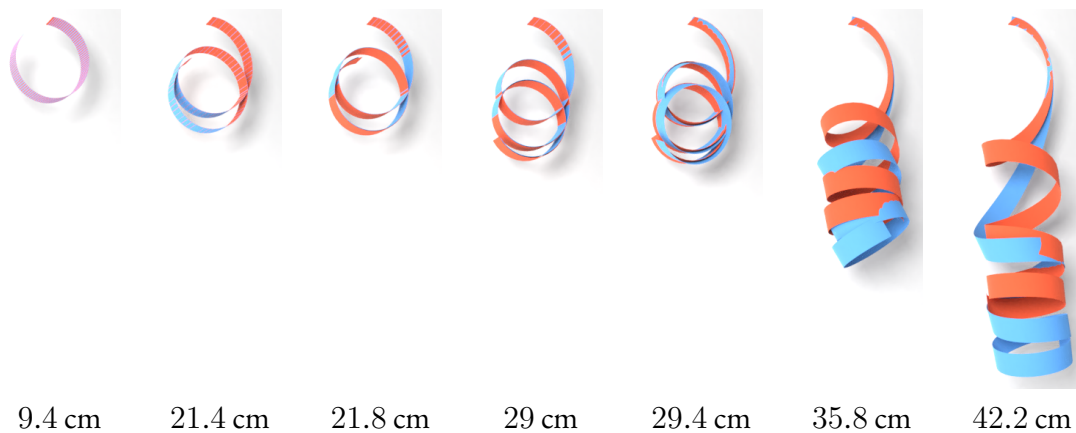


FIGURE 4.4 – Comparaison des équilibres engendrés par le modèle des rubans (en bleu) et celui des tiges à section rectangulaire (en orange) pour des longueurs de structure croissantes. Les deux modèles coïncident quand l'équilibre est planaire ($\ell \leq 29$ cm), mais diffèrent substantiellement quand la torsion entre en jeu.

L'échec du modèle de tige à reproduire le comportement du ruban est assez prévisible. Tout d'abord par définition une tige possède une section dont les dimensions caractéristiques doivent être proches, or pour un ruban, par définition $h \ll w$, donc ce n'est pas compatible. Ensuite si la section d'une tige est rectangulaire, ce n'est pas le cas pour un ruban. En effet, la coupe d'un ruban est rectangulaire le long d'une règle, mais la section orthogonale à la centerline, n'est en général pas rectangulaire. Finalement, la courbure de la centerline d'une tige n'est pas contrainte. Donc il est probable que le minimum d'énergie sans les contraintes du modèle ruban sur la courbure soit plus bas qu'avec les contraintes.

On pourrait alors tout de même se demander si un modèle de tige plus général pourrait mieux approcher le comportement d'un ruban. De manière caricaturale, le modèle de ruban est un modèle de tige puisque sa caractérisation est faite le long d'une courbe (sa centerline). Cependant si l'on considère que les trois courbures doivent être libres on peut penser qu'il faut ajouter la possibilité de cisaillement de la section. Le modèle d'Antman [Ant74] introduit cela.

Les détails du modèle d'Antman ne sont pas présentés ici, le lecteur peut se référer à l'annexe B de [Cha+20] pour retrouver les équations constitutives du modèle. Cependant, nous présentons les résultats d'une expérience comparative avec le modèle de tiges de Kirchhoff et

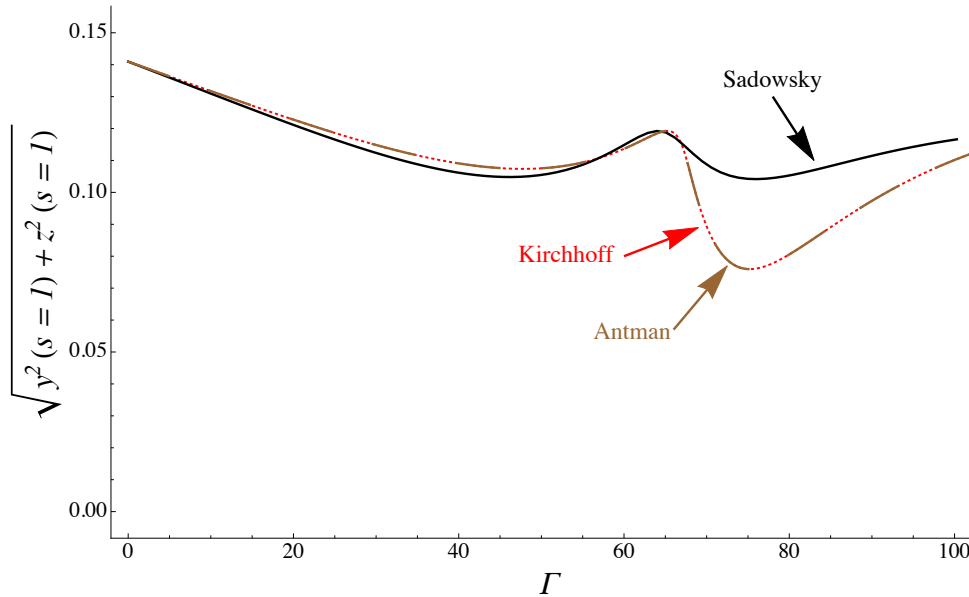


FIGURE 4.5 – Comparaison des modèles de tiges d’Antman, Kirchhoff et du modèle de ruban avec l’énergie de Sadowsky. Les paramètres expérimentaux sont $\frac{w}{\ell} = \frac{1}{30}$, $\frac{h}{w} = \frac{1}{10}$, $K_N = \omega\ell = 10$, $\nu = 0.4$. La quantité $\sqrt{y^2(s = \ell) + z^2(s = \ell)}$ correspond à la distance entre les deux extrémités du ruban dans le plan orthogonal à la gravité. Voir aussi 4.6 pour une illustration des équilibres avec $\Gamma = 80$.

le modèle de ruban avec l’énergie de Sadowsky. Le ratio $\frac{w}{\ell}$ est fixé à $\frac{1}{30}$ et $\frac{h}{w}$ à $\frac{1}{10}$, la courbure normale adimensionnée à la constante $K_N = \omega\ell = 10$, le coefficient de Poisson $\nu = 0.4$. Pour la comparaison nous traçons les graphes $(\Gamma, \sqrt{y^2(s = \ell) + z^2(s = \ell)})$, soit la distance entre les deux extrémités du ruban dans le plan orthogonal à la gravité en fonction de Γ que nous faisons varier de 0 à 100. Un exemple d’équilibre pour les trois modèles avec $\Gamma = 80$ est présenté en figure 4.6. Sur la figure 4.5 on remarque la superposition des courbes des deux modèles de tiges, quand la courbe du ruban avec Sadowsky est nettement différente.

Cet échec à retrouver la forme du ruban n’est pas étonnant. En effet, l’espace des déformées par le modèle de tige enrichi est bien supérieur à celui du ruban. En effet, le modèle de tige classique est déjà plus riche puisque qu’il y a trois courbures le long de la centerline au lieu de deux pour le ruban. Si l’on ajoute le cisaillement en plus, cela ne peut qu’enrichir l’espace des déformées.

4.5 Expérience de la poutre console

Une expérience bien connue en mécanique des objets élastiques est utilisée pour estimer les paramètres matériaux. Une poutre élastique est encastree horizontalement avec un bord libre. Sous l’action de la gravité, le rapport des dimensions suit une courbe théorique spécifique en fonction d’un paramètre adimensionné gravito-courbure ($\Gamma = \frac{\rho g h w \ell^3}{D}$), comme illustré sur la figure 4.7.

Pour comparer les prédictions de notre code à l’expérience, nous utilisons des simulations de rubans naturellement plats, de largeur $w = 5$ cm, de densité $\rho = 1200$ kg/m³, d’épaisseur

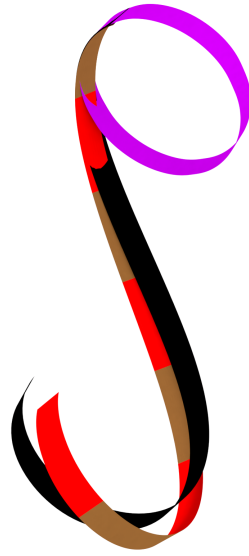


FIGURE 4.6 – Comparaison des modèles de tiges d’Antman, Kirchhoff et du modèle de ruban avec l’énergie de Sadowsky. On illustre ici l’équilibre avec les paramètres $\frac{w}{\ell} = \frac{1}{30}$, $\frac{h}{w} = \frac{1}{10}$, $\Gamma = 80$, $K_N = \omega\ell = 10$, $\nu = 0.4$. Les couleurs des différents modèles sont identiques au tracé figure 4.5, noir pour le ruban, rouge pour Kirchhoff et marron pour Antman. Puis nous rajoutons la forme naturelle en violet. Pour bien visualiser la superposition, nous alternons le tracé de la tige entre les modèles de Kirchhoff et d’Antman.

$h = 1$ mm et de coefficient de Poisson $\nu = 0.5$ fixes. Le module de Young Y est choisi dans l’intervalle $[0.5-10]$ MPa et la longueur ℓ du ruban dans l’intervalle $[5-90]$ cm. Au total, nous effectuons 2500 simulations numériques. De plus, nous résolvons les équations pour l’elastica plan avec la rigidité de flexion $D = \frac{Yh^3w}{12(1-\nu^2)}$. Enfin, Victor Romero nous a fournis les données expérimentales qu’il a produites en laboratoire, telle que celle illustrée figure 4.8. La figure 4.7 compare simultanément toutes ces approches (numérique, théorique et expérimentale) lorsque l’on calcule le rapport des dimensions du ruban en fonction de Γ .

On observe une très bonne correspondance entre notre modèle numérique, le modèle de l’elastica, et l’expérimental. Il est assez remarquable d’obtenir les bonnes précisions même avec une valeur de Γ élevé et seulement $p = 20$ segments. On peut se référer à [Rom+21] pour retrouver les résultats de cette expérience pour d’autres codes et modèles. On constate que ce scénario bien qu’en apparence simple n’est pas facile à traiter en général. Ainsi certains modèles montrent leur faiblesse vers $\Gamma = 10^4$. La facilité à capturer le bon comportement est sans doute due au choix d’utiliser la courbure normale ω (et la direction de la règle) comme degré de liberté. Il est ainsi aisé de représenter une forte courbure et un fort ratio $\frac{y}{x}$, alors qu’une représentation nodale nécessite beaucoup de points vers l’encastrement pour capturer la courbure.

Expérience de la poutre console

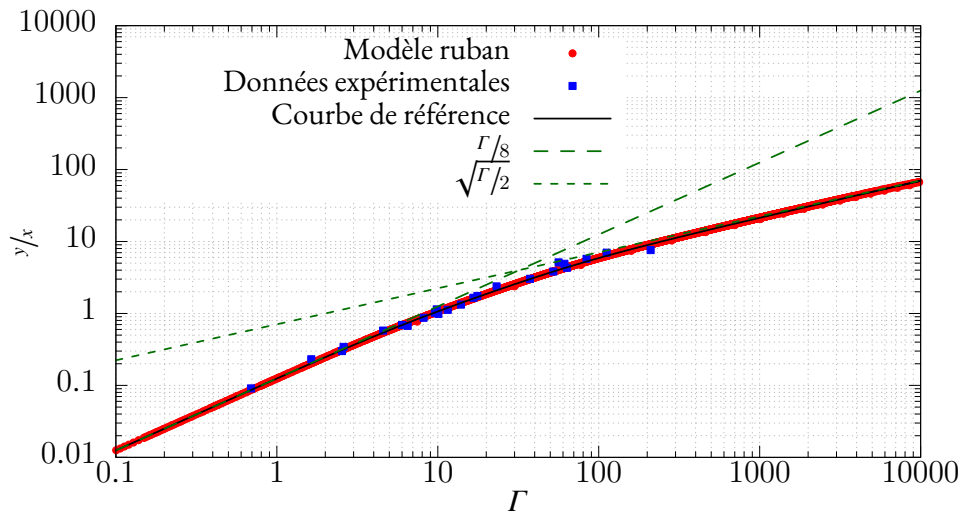


FIGURE 4.7 – Rapport y/x de l'étendue de la déformée de la poutre, en fonction du paramètre de gravito-courbure Γ , évalué de trois manières : par le calcul de l'elastica plan (en noir), par notre modèle numérique de ruban (en rouge), et par une série d'expériences (en bleu). Aussi on ajoute la limite pour des petits Γ , $\frac{y}{x} = \frac{\Gamma}{8}$ et pour les grands Γ , $\frac{y}{x} = \sqrt{\frac{\Gamma}{2}}$



FIGURE 4.8 – Expérience de la poutre console : le ruban expérimental (en gris foncé) est fixé horizontalement d'un côté et tombe sous son propre poids. La prédiction de notre modèle (en bleu) est projetée sur l'image réelle pour comparaison. Le ruban réel est découpé dans une feuille de PET (Polyethylene terephthalate), avec comme dimensions $29\text{cm} \times 3\text{cm} \times 0.143\text{mm}$. Expérience réalisée par Victor Romero.

Troisième partie

Modèle numérique mixte

Chapitre 5

Vers un modèle mixte

Dans ce chapitre, nous commençons par ajouter la possibilité d'encastrement des deux extrémités du ruban. Nous illustrons cela par un exemple en figure 5.1. Pour cela, il faut rajouter des contraintes au problème. Le choix de celles-ci n'est pas anodin et peut même être une raison d'échec de convergence pour l'algorithme de minimisation.

Avec un double encastrement, le nombre de segments requis augmente, car certaines formes peuvent avoir des courbures assez spécifiques en certains endroits, comme le ruban de Moebius que nous traiterons en section 6.1.1. Malheureusement, dans les chapitres précédents, nous avons vu que la complexité du calcul du hessien est cubique en le nombre de segments, et celle du gradient est quadratique, ce qui est assez coûteux. En comparaison, quand la géométrie est nodale, le calcul du système est de complexité linéaire. Et même si le hessien a une taille quadratique, sa forme creuse permet de conserver une représentation linéaire. Dans nos expériences au chapitre 3, le nombre de segments utilisé est cependant assez faible pour que cela ne soit pas gênant (de l'ordre de 20 segments). Ainsi chaque calcul d'équilibre prend un temps négligeable, c'est-à-dire, de l'ordre de la seconde. Mais avec l'enrichissement du système, ce n'est plus le cas : on monte parfois à 100 segments comme on verra dans le chapitre 6. De plus un système creux est souvent plus facile à traiter pour le solveur. Ainsi comme illustré en figures 5.2 et 5.4, il faut trouver une méthode pour pallier ce problème.

Pour tirer parti du meilleur entre modèle nodal ou élément de haut degré, nous développons dans ce chapitre une formulation mixte : on considère chaque segment séparément et on relie les différents segments par des contraintes. En ce sens mixte signifie modèle basé sur la courbure et nodal : chaque segment a une position et une orientation spécifiées au point de début et des degrés de liberté basés sur la courbure pour spécifier la forme du segment. La complexité du calcul du système qui en résulte devient linéaire. En contrepartie, l'algorithme de minimisation doit gérer les contraintes de raccordement entre les segments. Ainsi, il n'est plus question de continuer à développer une méthode maison telle que NEWTONR (voir la sous-section 3.5.2), alors qu'IPOPT est capable de gérer efficacement les contraintes. Pour lever une ambiguïté de vocabulaire, on nommera l'ancien modèle « modèle chaîné » et le nouveau « modèle mixte ».

5.1 Encastrement des deux extrémités du ruban

Nous avons choisi depuis le début de représenter le ruban par sa courbure. Ainsi, pour représenter un ruban, il faut spécifier l'un des encastrements pour un bord du ruban. Puis l'intégration grâce à nos fonctions de bases ω et η permet de calculer la position et l'orientation de

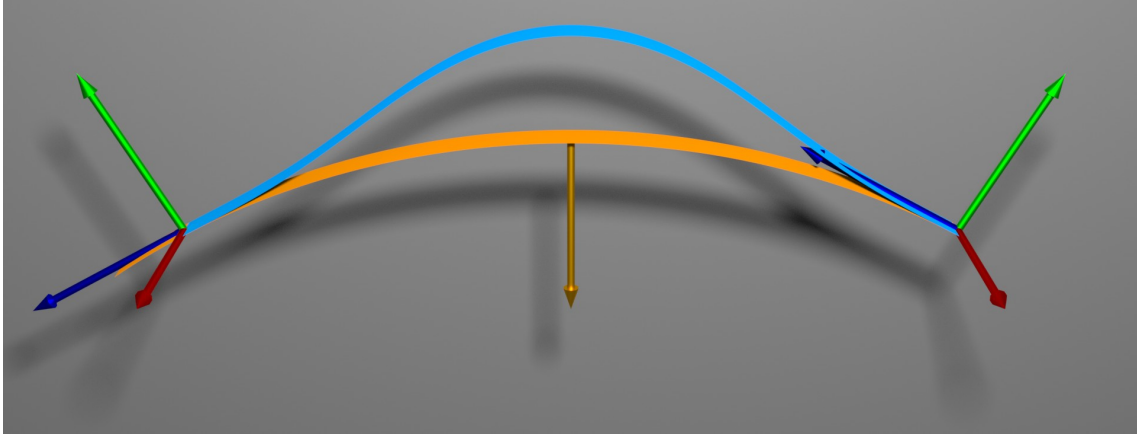


FIGURE 5.1 – Une expérience simple de double encastrement. Le ruban orangé décrit un arc de cercle parfait et sert de graine à l’algorithme de minimisation. Il n’est pas encastéré à sa fin, c’est-à-dire à gauche. Les deux repères servent d’encastrement au ruban et la flèche orange indique la direction de la gravité. Le ruban bleu est la forme à l’équilibre. Pour cette expérience, on a utilisé $\frac{w}{\ell} = \frac{1}{11}$, $\Gamma = 100$, $\nu = 0.35$.

l’autre bord. Par commodité, nous appelons cet autre bord « fin du ruban », ce qui prend sens vis-à-vis de l’abscisse curviligne s . Cette fin du ruban est calculée par la composition de fonctions calculées par des séries entières soumises aux problèmes de compensation catastrophique. Il semble alors complexe de vouloir trouver un jeu de degrés de liberté qui permettrait de respecter automatiquement un encastrement aux deux bords du ruban. À la place, introduire des contraintes mathématiques dans le problème d’optimisation est un choix naturel, d’autant que ce type de problème est connu et le solveur IPOPT sait le traiter.

L’ensemble des degrés de liberté du système forme un vecteur $\mathbf{x} \in \mathbb{R}^n$ aussi appelé point du système. Nous représentons le calcul de l’extrémité du ruban par les fonctions $\mathbf{v} : \mathbb{R}^n \mapsto \mathbb{R}^3$ et $\mathcal{G} : \mathbb{R}^n \mapsto \mathcal{SO}_3$ qui dans l’ordre représentent la position et le repère matériel en fin du ruban. Ces deux fonctions sont, rappelons-le, non-convexes. Ni leur gradient, ni leur hessien n’est nul. En définissant l’encastrement désiré $(\mathbf{u}, \mathcal{H})$ en fin de ruban, il faut résoudre la minimisation de l’énergie sous les contraintes $\mathbf{v}(\mathbf{x}) = \mathbf{u}$, $\mathcal{G}(\mathbf{x}) = \mathcal{H}$. En pratique, les contraintes sont sous la forme $f(\mathbf{x}) = 0$ où f est une fonction à valeur dans \mathbb{R} . Ces fonctions pourront alors être rassemblées dans un vecteur.

Pour gérer la position en fin de ruban, nous choisissons d’utiliser trois contraintes :

$$v_x(\mathbf{x}) - u_x = 0 \quad (5.1a)$$

$$v_y(\mathbf{x}) - u_y = 0 \quad (5.1b)$$

$$v_z(\mathbf{x}) - u_z = 0 \quad (5.1c)$$

Pour gérer le repère matériel en fin de ruban, on propose deux jeux de contraintes possibles, soit :

$$\langle \mathbf{G}_x(\mathbf{x}), \mathbf{H}_x \rangle - 1 = 0 \quad (5.2a)$$

$$\langle \mathbf{G}_y(\mathbf{x}), \mathbf{H}_y \rangle - 1 = 0 \quad (5.2b)$$

$$\langle \mathbf{G}_z(\mathbf{x}), \mathbf{H}_z \rangle - 1 = 0 \quad (5.2c)$$

soit :

$$\langle \mathbf{G}_x(\mathbf{x}), \mathbf{H}_y \rangle = 0 \quad (5.3a)$$

$$\langle \mathbf{G}_y(\mathbf{x}), \mathbf{H}_z \rangle = 0 \quad (5.3b)$$

$$\langle \mathbf{G}_z(\mathbf{x}), \mathbf{H}_x \rangle = 0 \quad (5.3c)$$

On peut interpréter les contraintes de la première option (5.2), par l'alignement des même axes des deux repères (car les vecteurs correspondant sont unitaires). Pour la seconde option (5.3), on demande aux axes différents des repères d'être orthogonaux. La première option semble plus « naturelle ». Pour la seconde solution, on se retrouve avec un problème de non-unicité de la solution, par exemple la rotation de la matrice \mathcal{H} , $\mathcal{G}(\mathbf{x}) = (-\mathbf{H}_x \quad -\mathbf{H}_z \quad \mathbf{H}_y)$ est une solution à (5.3). On s'étonnera alors que ce soit cette option que l'on retienne. Montrons pourquoi.

Concentrons-nous sur la première composante de chaque option : $\mathbb{k}_1(\mathbf{x}) = \langle \mathbf{G}_x(\mathbf{x}), \mathbf{H}_x \rangle - 1$ et $\mathbb{k}_2(\mathbf{x}) = \langle \mathbf{G}_x(\mathbf{x}), \mathbf{H}_y \rangle$. Le problème d'optimisation s'écrit alors « Minimiser $E(\mathbf{x})$ sous la contrainte \mathbb{k}_i » où i correspond au choix de la contrainte. Classiquement, on résout le système $\nabla_{\mathbf{x}, \lambda} (E(\mathbf{x}) - \langle \lambda, \mathbb{k}_i(\mathbf{x}) \rangle) = 0$ en \mathbf{x} et en λ . Il forme ce qu'on appelle les conditions d'optimalité. En observant de plus près, la composante du gradient selon λ impose juste que la contrainte $\mathbb{k}_i(\mathbf{x}) = 0$ soit respectée. Dans ce qui suit, on ignore cette composante, car ce n'est pas le point problématique dans le choix des contraintes, le système considéré est $\nabla_{\mathbf{x}} (E(\mathbf{x}) - \langle \lambda, \mathbb{k}_i(\mathbf{x}) \rangle) = 0$.

On appelle communément λ le multiplicateur de Lagrange. Ici λ est un réel, mais s'il y a plusieurs contraintes, il devient un vecteur de taille égale au nombre de contraintes. Nous en reparlerons plus tard car il peut être interprété comme une force lorsque la contrainte est une longueur. Maintenant, il faut pouvoir calculer λ de manière unique. Dans notre cas où il n'y a qu'une seule contrainte, la solution au problème est unique si le gradient de la contrainte $\nabla_{\mathbf{x}} \mathbb{k}_i$ n'est pas nul. Numériquement λ diverge quand le gradient est négligeable. On se doute bien que le gradient de la contrainte n'est pas toujours nul dans le cas du ruban, mais il peut l'être quand la contrainte est respectée. De manière plus générale, s'il y a plusieurs contraintes, pour que le système n'ait qu'une solution, le jacobien du vecteur des contraintes doit être injectif¹. On dit aussi (par abus de langage) que les contraintes sont linéairement indépendantes².

Pour vérifier si le gradient est nul ou non, on utilise une propriété géométrique sur un vecteur normé.

Propriété : Soit $\mathbf{u}(\alpha) \in \mathbb{R}^n$ un vecteur normé dépendant d'une variable α quelconque. Alors, la dérivée de \mathbf{u} est orthogonale à \mathbf{u} .

On note $\mathbf{u}(\alpha) \perp \frac{d\mathbf{u}(\alpha)}{d\alpha}$, le symbole \perp indique que les deux quantités sont orthogonales, ou de manière équivalente que leur produit scalaire est nul.

Démonstration : On regarde l'ordre 1 : $\mathbf{u}(\alpha + d\alpha) = \mathbf{u}(\alpha) + d\alpha \frac{d\mathbf{u}(\alpha)}{d\alpha}$. Le calcul de la norme au carré à l'ordre 1 donne alors $\|\mathbf{u}(\alpha + d\alpha)\|^2 = \langle \mathbf{u}(\alpha), \mathbf{u}(\alpha) \rangle + 2 \left\langle \mathbf{u}(\alpha), \frac{d\mathbf{u}(\alpha)}{d\alpha} \right\rangle = \|\mathbf{u}(\alpha)\|^2 + 2 \left\langle \mathbf{u}(\alpha), \frac{d\mathbf{u}(\alpha)}{d\alpha} \right\rangle$. Comme \mathbf{u} est normé, $\left\langle \mathbf{u}(\alpha), \frac{d\mathbf{u}(\alpha)}{d\alpha} \right\rangle = 0$, c'est-à-dire que la dérivée de \mathbf{u} n'a pas de composante selon \mathbf{u} . \square

Nous appliquons cette propriété au vecteur \mathbf{G}_x qui est bien normé puisque $\mathcal{G}(\mathbf{x}) \in \mathcal{SO}_3$

1. Par exemple, on tomberait dans le cas d'un jacobien non injectif si l'on contraignait les neuf coefficients du repère matériel, car une matrice de \mathcal{SO}_3 n'a que trois degrés de liberté.

2. Ce sont les gradients des contraintes qui forment une famille de vecteurs linéairement indépendants

avec comme variable, n'importe quelle composante de \mathbf{x} :

$$\forall j \mathbf{G}_x(\mathbf{x}) \perp \frac{d\mathbf{G}_x(\mathbf{x})}{d\mathbf{x}_j} \quad (5.4)$$

Considérons un point \mathbf{x}^* respectant la contrainte $\mathbb{k}_i(\mathbf{x}^*) = 0$. Cela n'implique pas que l'énergie soit minimale, mais suffit à souligner une difficulté de résolution. Nous allons développer les fonctions autour de ce point. Utilisons alors $\bar{\mathbf{x}} = \mathbf{x} - \mathbf{x}^*$ pour effectuer le développement à l'ordre 1 suivant : $\mathbf{G}_x(\bar{\mathbf{x}}) = \mathbf{G}_x(\mathbf{x}^*) + \sum_k \frac{d\mathbf{G}_x(\bar{\mathbf{x}})}{d\mathbf{x}_k} \bar{x}_k + o(\bar{\mathbf{x}})$. La contrainte étant respectée, $\mathbf{G}_x(\mathbf{x}^*) = \mathbf{H}_x$.

Cas \mathbb{k}_1 Le calcul du gradient au point donné est $\nabla_{\mathbf{x}} \mathbb{k}_1(\mathbf{x}^*) = \left\langle \frac{d\mathbf{G}_x(\bar{\mathbf{x}})}{d\mathbf{x}_k}, \mathbf{H}_x \right\rangle_k$. Ainsi d'après la propriété (5.4), $\nabla_{\mathbf{x}} \mathbb{k}_1(\mathbf{x}^*) = \mathbf{0}$. C'est le cas que l'on cherche à éviter, donc l'utilisation de cette contrainte est à proscrire.

Cas \mathbb{k}_2 De même, le calcul du gradient donne $\nabla_{\mathbf{x}} \mathbb{k}_2(\mathbf{x}^*) = \left\langle \frac{d\mathbf{G}_x(\bar{\mathbf{x}})}{d\mathbf{x}_k}, \mathbf{H}_y \right\rangle_k$. En pratique, ce résultat peut être nul si $\frac{d\mathbf{G}_x(\bar{\mathbf{x}})}{d\mathbf{x}_k}$ est colinéaire à \mathbf{H}_z . Du fait de la construction de la fonction \mathcal{G} par composition de fonctions et du calcul par séries entières, le résultat est assez compliqué à montrer.

Nous pouvons "intuer" une raison en utilisant un ruban très court : avec un unique segment de longueur ds . On a $\mathcal{G} = \mathcal{R}(ds)$ et quand la condition est respectée $\mathcal{H} = \mathcal{R}(ds)$. On se concentre uniquement sur les paramètres b et m . On calcule $\alpha_k = \left\langle \frac{d\mathbf{R}_x(ds)}{dk}, \mathbf{R}_y(ds) \right\rangle$ où k représente b ou m . En se plaçant à l'ordre 1, on ignore les termes en ds^2 :

$$\alpha_k = \left\langle \frac{d}{dk} \left[\mathcal{R}(0) \left(I + \begin{bmatrix} b \\ 0 \\ mb \end{bmatrix}_x \begin{pmatrix} ds \\ 0 \\ 0 \end{pmatrix} \right) \right], \mathbf{R}_y(0) \right\rangle$$

Ainsi $\alpha_k = kds$, pour peu qu'une des deux valeurs b et m ne soit pas nulle alors $\nabla_{\mathbf{x}} \mathbb{k}_2(\mathbf{x}^*)$ n'est pas nul. On suppose qu'en général ce ne sera quasiment jamais le cas. En pratique, l'utilisation de cette contrainte ne pose pas de soucis. Au contraire en utilisant les contraintes (5.2) (cas \mathbb{k}_1), la convergence avec IPOPT devient environ trois ordres de grandeur plus lente, voire impossible dans certains cas.

Le problème des contraintes de type \mathbb{k}_2 est la non-unicité des solutions. Comment tomber sur la bonne solution ? Heureusement les solutions sont des points éloignés les uns des autres. Il faut que la fin du ruban pivote de $\frac{\pi}{2}$ pour changer de solution. A priori, la solution ne devrait donc pas sauter entre les solutions pendant la minimisation. On ne peut plus donner n'importe quel point de départ (graine) et demander à IPOPT de trouver une solution. Il faut guider la convergence en utilisant un point initial judicieux, voire progresser par étapes en posant plusieurs problèmes de minimisation successifs : la solution d'un problème va donner un point initial à la seconde et ainsi de suite. C'est ce qu'on appelle le warmstart.

5.2 Un modèle mixte pour plus de vitesse

La figure 5.4 indique qu'un bon choix de contraintes pour l'encastrement en fin de ruban n'est pas suffisant pour garantir une bonne convergence. Intuitivement, les fonctions \mathcal{H}

et \mathbf{u} sont trop complexes pour résoudre la contrainte. Mais parallèlement à l'élaboration de contraintes d'encastrement de fin du ruban, une idée similaire naît : « Pourquoi ne pas traiter les segments indépendamment et les relier avec des contraintes ? ». L'encastrement en début de segment joue le rôle de l'encastrement de la fin du segment précédent. Le système devient creux et sa complexité de calcul linéaire. Cela permet d'augmenter le nombre de segments le long du ruban. Aussi l'algorithme de minimisation fonctionne mieux, car le système devient plus facile à résoudre, comme le montre la figure 5.4.

Pour ce faire on se base sur la section précédente qui nous indique un jeu de contrainte pour aligner les repères matériels entre les segments, mais cela n'est pas suffisant. Il faut bien choisir les degrés de liberté de chaque segment.

Chaque segment est caractérisé par :

- une position $\mathbf{r}(0)$ et un repère matériel $\mathcal{R}(0)$ pour le placer et l'orienter dans l'espace. Concrètement, c'est l'encastrement du ruban défini soit par l'opérateur comme précédemment soit par le segment précédent.
- sa courbure normale et la direction des génératrices le long de la centerline pour décrire sa forme. On ne modifie pas les choix précédents, et on reprend les quatre degrés de liberté : a , b , n et m .
- sa courbure normale naturelle, idem on reprend une représentation linéaire de celle-ci sur chaque segment.

Jusqu'à présent la position et le repère matériel étaient calculés depuis l'encastrement en début de ruban. Tout les calculs ont été faits sous forme matricielle. La question de la représentation de ces deux quantités n'avait donc pas vraiment d'intérêt : nous manipulons des matrices donc nous utilisons des matrices. Mais maintenant, nous voulons les expliciter en tant que degrés de liberté dans notre système. Il n'y a aucune raison de ne pas représenter la position par un vecteur de \mathbb{R}^3 . Cependant, le repère matériel est un élément de \mathcal{SO}_3 . Il ne possède donc que trois degrés de liberté, le choix de sa représentation n'est donc pas évident. Citons les trois représentations les plus classiques :

- Sous forme de matrice de rotation. Cependant, une telle matrice possède neuf coefficients, il faudra donc ajouter six contraintes pour forcer la matrice à appartenir à \mathcal{SO}_3 .
- Sous forme de quaternions. Les quaternions sont bien adaptés pour représenter un repère, mais ils ont quatre degrés de liberté, il faut les normaliser ce qui rajoute une contrainte, et dans la plupart des calculs du chapitre 3 il faut calculer la forme matricielle du repère. Ce calcul est simple à faire, mais rajoute une couche de calculs supplémentaire et va donc complexifier le travail d'ИОРТ. Pour éviter cela, il faudrait adapter les équations (3.4) à l'utilisation des quaternions, c'est-à-dire dériver une série simple en prenant en compte les problèmes de compensations catastrophiques.
- Sous forme d'angles d'Euler. La représentation matricielle correspond alors à la multiplication de trois matrices de rotations ayant chacune un axe fixe (par exemple \mathbf{e}_x , \mathbf{e}_y , \mathbf{e}_z) et un angle libre. Cette solution semble être idéale, car il ne faut pas la coupler à des contraintes, et elle pourrait très bien s'adapter aux séries entières. Cependant, cette représentation pose un problème : le blocage de cardan. C'est une configuration où l'un des degrés de liberté est perdu, c'est-à-dire un des angles de rotation ne modifie pas la matrice. Ce blocage peut arriver pour certaines directions d'encastrement par exemple.

Ainsi le choix de représentation s'est finalement porté sur les matrices. Il faut donc choisir six contraintes pour forcer l'appartenance de la matrice à \mathcal{SO}_3 . En utilisant la définition de repère orthonormé on discerne rapidement un choix de contraintes : trois pour normer les colonnes

et trois pour les rendre orthogonales. Sur la matrice \mathcal{F} on obtient le système

$$\langle \mathbf{F}_x, \mathbf{F}_x \rangle - 1 = 0 \quad (5.5a)$$

$$\langle \mathbf{F}_y, \mathbf{F}_y \rangle - 1 = 0 \quad (5.5b)$$

$$\langle \mathbf{F}_z, \mathbf{F}_z \rangle - 1 = 0 \quad (5.5c)$$

$$\langle \mathbf{F}_x, \mathbf{F}_y \rangle = 0 \quad (5.5d)$$

$$\langle \mathbf{F}_y, \mathbf{F}_z \rangle = 0 \quad (5.5e)$$

$$\langle \mathbf{F}_z, \mathbf{F}_x \rangle = 0 \quad (5.5f)$$

On peut en particulier remarquer que ces contraintes sont convexes, donc très faciles à gérer. Il ne faut pas confondre ces contraintes avec (5.2) et (5.3)³. Le seul détail reste que le repère peut ainsi être indirect. Comme pour l'encastrement en fin de ruban on va supposer que si le système est bien initialisé (warmstart) la solution ne va pas sauter vers une solution qui ne convient pas (il faudrait que toute une colonne soit multipliée par un facteur -1 par exemple).

Chaque segment possède donc 16 degrés de liberté et 6 contraintes. Il faut relier ces segments. Les besoins pour ce faire ne sont pas nouveaux. On rappelle que pour une surface lisse, la direction des règles η doit être continue (figure 3.2), ω aussi. Ces deux besoins se traduisent par une contrainte linéaire. Les deux nouveautés sont en revanche le raccord des positions et des repères matériels, qui sont très similaires à ce que l'on a vu dans la section 5.1. On construit ainsi entre les segments i et $i + 1$ le système de contraintes :

$$b^{(i+1)} - b^{(i)} - a^{(i)} \ell^{(i)} = 0 \quad (5.6a)$$

$$m^{(i+1)} - m^{(i)} - n^{(i)} \ell^{(i)} = 0 \quad (5.6b)$$

$$r_x^{(i+1)}(0) - v_x(\mathbf{r}^{(i)}(0), \mathcal{R}^{(i)}(0), a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)}, \ell^{(i)}) = 0 \quad (5.6c)$$

$$r_y^{(i+1)}(0) - v_y(\mathbf{r}^{(i)}(0), \mathcal{R}^{(i)}(0), a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)}, \ell^{(i)}) = 0 \quad (5.6d)$$

$$r_z^{(i+1)}(0) - v_z(\mathbf{r}^{(i)}(0), \mathcal{R}^{(i)}(0), a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)}, \ell^{(i)}) = 0 \quad (5.6e)$$

$$\left\langle \mathbf{R}_x^{(i+1)}(0), \mathbf{G}_y \left(\mathbf{r}^{(i)}(0), \mathcal{R}^{(i)}(0), a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)}, \ell^{(i)} \right) \right\rangle = 0 \quad (5.6f)$$

$$\left\langle \mathbf{R}_y^{(i+1)}(0), \mathbf{G}_z \left(\mathbf{r}^{(i)}(0), \mathcal{R}^{(i)}(0), a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)}, \ell^{(i)} \right) \right\rangle = 0 \quad (5.6g)$$

$$\left\langle \mathbf{R}_z^{(i+1)}(0), \mathbf{G}_x \left(\mathbf{r}^{(i)}(0), \mathcal{R}^{(i)}(0), a^{(i)}, b^{(i)}, n^{(i)}, m^{(i)}, \ell^{(i)} \right) \right\rangle = 0 \quad (5.6h)$$

En bilan, pour un segment, il y a 16 degrés de libertés et 14 contraintes, soit deux paramètres non contraints. C'est équivalent aux deux degrés de libertés par segment du premier modèle.

La dernière étape consiste à attacher le ruban en $s = 0$. Il y a deux possibilités : soit on utilise des contraintes pour attacher le ruban, soit on supprime les degrés de liberté et on considère les paramètres positions et repère du premier segment comme fixes. La première option est très facile à implémenter. La deuxième peut facilement être adaptée à partir de la première : il suffit de filtrer les contraintes et les degrés de liberté pour adapter le système. Le choix des contraintes est très élémentaire, pour un encastrement $(\mathbf{p}, \mathcal{F})$, cela se résume au système linéaire :

3. Le problème de gradient nul en la solution pour (5.2) n'a pas lieu d'être ici, car les composantes de la matrice \mathcal{F} sont toutes des degrés de liberté.

$$\forall j \in \{x, y, z\}, r_j^{(0)}(0) - \mathbf{p}_j = 0 \quad (5.7a)$$

$$\forall i, j \in \{x, y, z\}, R_{i,j}^{(0)} - F_{i,j} = 0 \quad (5.7b)$$

Comme montré plus tard en 5.3.3 les deux méthodes se valent. Nous utilisons par défaut des contraintes pour l'encastrement. Ce choix est un peu plus facile à mettre en place dans le code.

5.3 Comparaison : l'ajout de contraintes pour une meilleure vitesse

Nous rappelons que nous avons nommé l'ancien modèle par l'adjectif «chaîné», cela fait référence aux compositions de fonctions entre les segments. On rappelle aussi que le nouveau modèle est nommé «mixte». Commençons donc à comparer les deux modèles.

On construit deux scénarii pour la comparaison. Le premier reste basique, on va tout simplement faire pendre un ruban sous gravité. On s'attend à voir un temps d'exécution de l'algorithme de minimisation inférieur pour le modèle mixte avec une différence croissante en fonction du nombre de segments. Pour le deuxième scénario, on ajoute un encastrement à la fin du ruban. Ce scénario doit nous permettre de voir le rôle des contraintes sur un cas assez complexe. Nous utilisons des valeurs adimensionnées dans tous les scénarii présentés.

5.3.1 Scénario encasté libre

Prenons un ruban de longueur $\ell = 1$, largeur $w = 0.1$. Puis fixons $D = 1$, $\nu = 0.35$ et $\Gamma = 3$. Nous fixons la verticale à $\mathbf{e}_g = \mathbf{e}_z$ et encastrons le ruban avec un angle de 0.2 radians dans le plan $(\mathbf{e}_y, \mathbf{e}_z)$. Rappelons que la gravité est définie par le vecteur $\mathbf{g} = -g\mathbf{e}_g$, et $\Gamma = \frac{\rho h g \ell^3}{D}$ (voir section 4.1). Sa tangente à l'origine est alors selon \mathbf{e}_x . La graine est un ruban presque plat : $\omega(s) = 10^{-5}s + 10^{-4}$, $\eta(s) = 10^{-5}s + 10^{-4}$. Nous mesurons les performances du calcul d'équilibre en temps et en itérations en utilisant $p = 10, 20, 40$ puis 100 segments pour les énergies de Sadowsky et de Wunderlich. Nous illustrons l'équilibre en figure 5.3.

Les résultats présentés en figure 5.2 montrent un temps de calcul cubique en le nombre de segments pour le modèle chaîné et linéaire pour le modèle mixte⁴. On peut noter que l'énergie utilisée n'a pas une grande influence. La partie intéressante, et quelque peu surprenante se trouve pour un nombre de segments inférieur ou égal à 20. Le calcul pour le modèle chaîné y est plus rapide. Remarquons tout de même que pour ces points nous restons dans le cadre des chapitres 3 et 4, c'est-à-dire que la fin du ruban n'est pas encastree. On peut expliquer qu'avec un nombre faible de segments le calcul d'équilibre soit plus long pour le modèle mixte, car IPOPT doit dans ce cas traiter des contraintes absentes pour le modèle chaîné.

5.3.2 Scénario bi-encasté

Prenons un ruban de longueur $\ell = 1$, largeur $w = 0.01$. Puis fixons $D = 1$, $\nu = 0.5$ et $\Gamma = 1$. Nous fixons la verticale à $\mathbf{e}_g = \mathbf{e}_z$ et encastrons le ruban verticalement à son début,

4. Ces mesures nous rappellent les complexités pour résoudre un système plein (le cas du modèle chaîné) et un système creux (le cas du modèle mixte). Le lien n'est pas forcément direct puisque le nombre de pas de l'algorithme doit aussi être pris en compte.

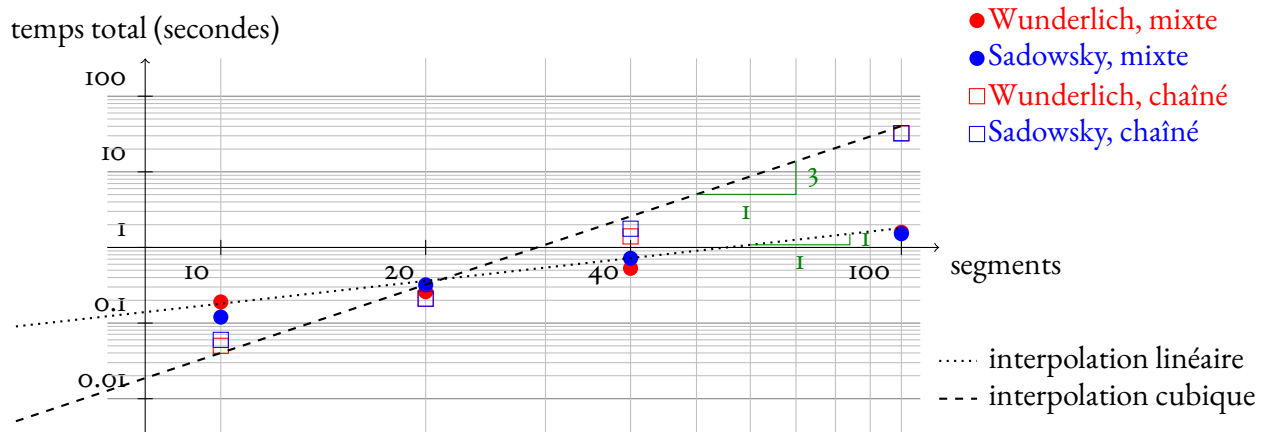


FIGURE 5.2 – Mesure de l'efficacité du modèle mixte par rapport au modèle chaîné (attention à l'échelle logarithmique), et interpolation manuelle de la complexité. Le ruban n'est encastré que d'un seul côté (détails en section 5.3.1). On voit que le modèle mixte est efficace pour un grand nombre de segments. On trouve un temps de calcul cubique en le nombre de segments pour le modèle chaîné et linéaire pour le modèle mixte. Jusqu'à 20 segments, le modèle chaîné reste cependant plus efficace.

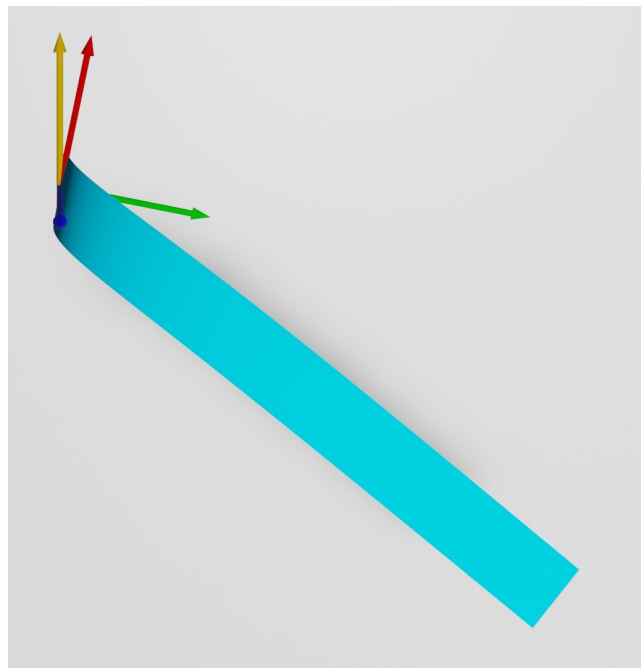


FIGURE 5.3 – Illustration de l'équilibre du ruban décrit dans le « Scenario encastré libre » (section 5.3.1). L'encastrement est indiqué avec les trois flèches rouge, verte et bleue, et la verticale par une flèche jaune. La différence entre l'utilisation de l'énergie de Sadowsky et de Wunderlich n'est pas visible à l'œil nu.

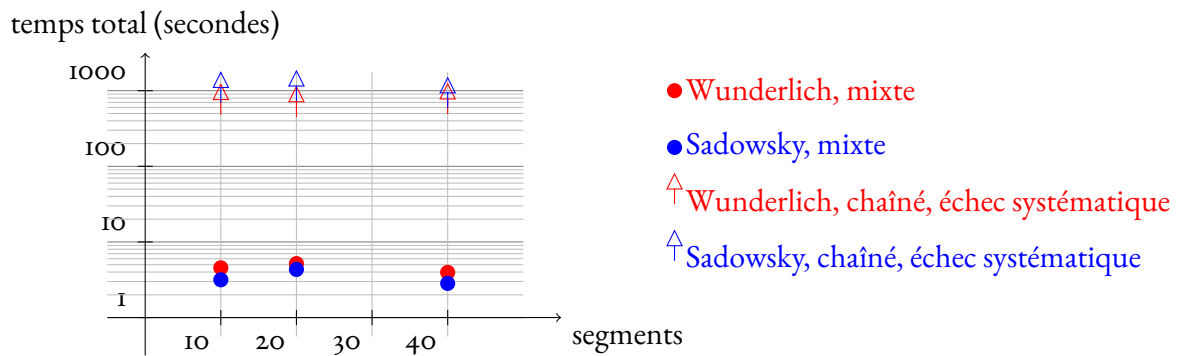


FIGURE 5.4 – Efficacité du modèle mixte et chaîné dans un scénario bi-encasté (voir 5.3.2 pour les détails). Le modèle chaîné échoue systématiquement avec le statut «Restoration failed» pour Wunderlich, et pour Sadowsky le nombre maximum d'itération (1000) est atteint. Cela démontre clairement l'utilité du modèle mixte où la minimisation réussit systématiquement en quelques secondes.

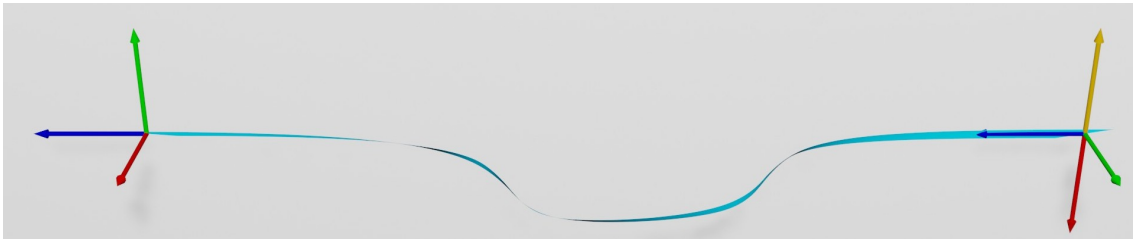


FIGURE 5.5 – Illustration de l'équilibre du ruban décrit dans le « Scénario bi-encasté » (section 5.3.2). Les encastresments sont indiqués par des flèches rouges, vertes et bleu, et la verticale par une flèche jaune. La différence entre l'utilisation de l'énergie de Sadowsky et de Wunderlich n'est pas visible à l'œil nu.

horizontalement à sa fin, les deux encastresments distants de 0.95. La graine est un ruban presque plat : $\omega(s) = 10^{-4}s + 10^{-3}$, $\eta(s) = 10^{-4}s + 10^{-3}$. Comme précédemment nous utilisons les énergies de Wunderlich et de Sadowsky. Nous répétons la minimisation pour 10, 20 et 40 segments. Nous illustrons l'équilibre en figure 5.5.

Les résultats sont présentés sur la figure 5.4. Le modèle chaîné montre systématiquement son incapacité à converger, c'est d'ailleurs la raison de l'absence du point "100 segments". Le modèle mixte réussit à converger. On peut cependant noter que le temps de convergence est plus conséquent que pour le scénario précédent, il passe de moins d'une seconde à quelques secondes, soit un ordre de grandeur supérieur, même si ce temps reste très raisonnable pour quelques expériences isolées. Le double encastrement n'est donc absolument pas anodin.

5.3.3 Encastrement en début du ruban

Nous avons évoqué deux méthodes d'attache du ruban. Soit nous ajoutons des contraintes, soit nous supprimons les degrés de liberté correspondant du premier segment du ruban. Toujours en adimensionné, nous définissons un ruban avec $\ell = 1$, $D = 1$, $\nu = 0.5$, $\Gamma = 10$, $p = 20$

segments. Nous définissons cinq scénarii illustré en figure 5.3.3 :

1. L'encastrement de début est horizontal, l'encastrement de fin est au même endroit, mais pivoté de $\frac{\pi}{4}$ radians par rapport à la tangente au ruban. Graine : $\omega(s) = -2\pi + 10^{-4}s$, $\eta(s) = 0.01 + 10^{-4}s$.
2. Le ruban est encasté horizontalement aux deux extrémités et les encastresments sont espacés par $0.1\mathbf{R}_x(0)$. Graine : $\omega(s) = -2\pi + 10^{-4}s$, $\eta(s) = 0.01 + 10^{-4}s$.
3. L'encastrement au début est horizontal, et l'encastrement en fin est tel que le ruban puisse former un arc de cercle parfait d'angle o.i. Graine : $\omega(s) = 0.02 + 10^{-4}s$, $\eta(s) = 10^{-4}s$.
4. L'encastrement au début est horizontal, en fin l'encastrement est la rotation du repère matériel de $\frac{\pi}{4}$ radians autour de son propre axe $\mathbf{R}_x(\ell)$. Graine : $\omega(s) = 5s$ si $s \leq 0.5$, $\omega(s) = 5 - 5s$ sinon, $\eta(s) = 0$.
5. Les encastresments sont ceux du protocole 2 une fois ceux-ci pivotés de $\frac{\pi}{4}$ radians autour de la tangente au ruban. Graine : $\omega(s) = -2\pi + 10^{-4}s$, $\eta(s) = 0.01 + 10^{-4}s$.

Sur chaque scenario, nous comparons les deux méthodes d'attache. Les résultats sont listés dans la table 5.1. La différence est insignifiante : pour attacher le ruban en $s = 0$ l'utilisation de contraintes ne pénalise pas le temps de calcul. Pour une implémentation plus facile⁵, on peut donc se contenter de l'utilisation de ces contraintes.

5. Il s'agit ici d'une impression. Selon les auteurs il est plus facile de traiter les segments de la même manière puis d'ajouter des contraintes que de traiter le premier segment autrement.

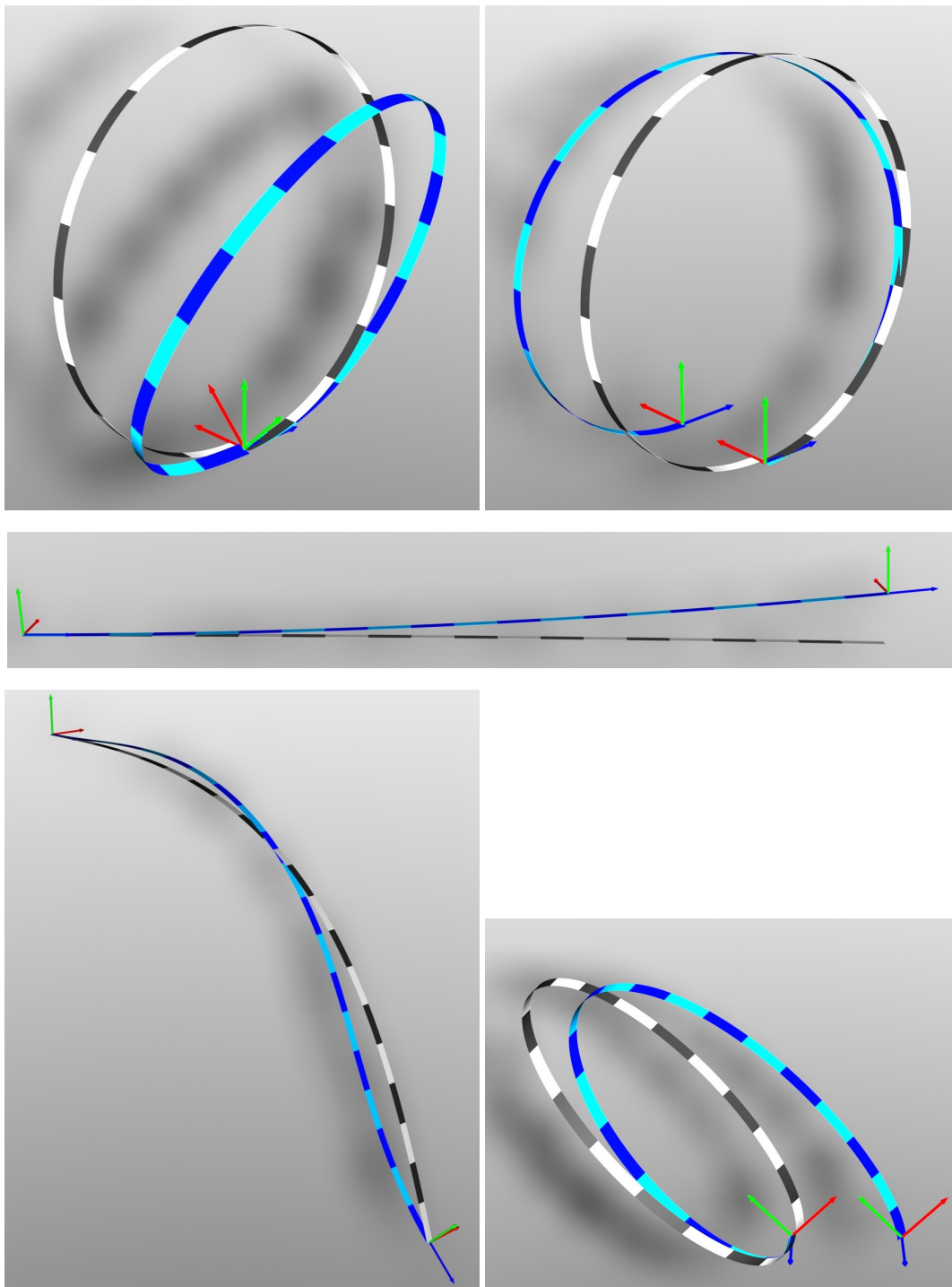


FIGURE 5.6 – Les cinq scenarii pour tester l'encastrement, dans le même ordre que dans le texte 5.3.3. La graine correspond au ruban en teintes grises (pour rappel, c'est la configuration initiale de l'algorithme de minimisation). L'alternance des couleurs correspond aux segments. Les encastresments sont aussi indiqués par les flèches tricolores qui collent aux axes des repères matériels. Le choix d'utiliser un encastrement par contraintes ou par suppression de degrés de liberté ne modifie pas l'équilibre calculé. On trace ce dernier en bleu.

Encastrement	itérations	erreur KKT	temps (sec)
Scenario 1			
Contraintes	19	9.09e-11	2.9
DdL fixés	20	9.09e-11	2.6
Scenario 2			
Contraintes	6	9.09e-11	0.9
DdL fixés	6	9.09e-11	0.7
Scenario 3			
Contraintes	22	9.09e-11	3.2
DdL fixés	25	1.76e-07	3.0
Scenario 4			
Contraintes	6	9.09e-11	0.9
DdL fixés	6	9.09e-11	0.7
Scenario 5			
Contraintes	6	9.09e-11	0.9
DdL fixés	11	9.65e-10	1.3

TABLE 5.1 – Comparaison de l'encastrement en début de ruban via des contraintes ou en fixant des degrés de liberté (DdL fixés) en utilisant les scénarii décrits en section 5.3.3 et illustrés en figure 5.3.3. Dans les quatre premiers cas, il vaut mieux fixer les degrés de liberté car le temps d'exécution est plus court. Dans le cinquième cas, c'est l'inverse avec une différence plus conséquente. Cependant, la différence n'est pas si nette.

Chapitre 6

Ajout de murs et rubans circulaires

Nous avons vu au chapitre 5 comment gérer les contraintes pour encastrier un ruban aux deux extrémités, et ce faisant nous sommes passés à un modèle mixte. Nous restons avec ce modèle mixte dans ce chapitre. Pour élaborer ce modèle, nous avons choisi une série de contraintes pour raccorder les segments du ruban les uns aux autres puis encastrier la fin du ruban. À présent nous allons ajouter d'autres contraintes.

Nous allons représenter un ruban de Moebius comme montré en figure 2.1. Comme dans le problème originel, l'énergie de gravité est ignorée, car supposée négligeable devant l'énergie élastique. L'ajout par rapport aux chapitres précédents est la fermeture du ruban sur lui-même. C'est-à-dire, visuellement le ruban n'a plus d'extrémité. On peut tout d'abord proposer d'encastrier les extrémités identiquement (même position, même repère matériel aux deux extrémités). Ce n'est cependant pas suffisant, car la continuité de la surface est assurée par la continuité de la fonction η . Il faut alors contraindre la direction de la génératrice au raccord.

Une fois que l'on peut simuler le ruban de Moebius, on aimerait le poser sur la tranche. Cela implique d'ajouter de la pesanteur au problème, qui jusque-là avait été négligée. La partie délicate est dans le traitement des encastresments. En effet, poser le ruban implique de ne fixer la position d'aucun point. Aussi, pour poser le ruban il faut ajouter du contact. Il ne s'agit pas d'auto-contact (c'est-à-dire de contact entre plusieurs point du ruban), mais de contact entre le ruban et un plan. Une invariance par translation et rotation apparaît. En effet, le ruban peut être déplacé parallèlement au plan ou tourné sur le plan sans changer son énergie potentielle.

Avec la possibilité de contact du ruban avec un plan, nous allons reproduire l'expérience de Benoît Roman [RP99]. Il s'agit de contraindre un ruban entre deux plaques. Celui-ci va alors former des vagues, comme montré sur la figure 6.11.

6.1 Contraintes à l'encastrement

Un ruban circulaire, (indépendamment de sa courbure naturelle qui peut être nulle) a ces deux bords collés ensemble de façon à être déformable pour former un cylindre de hauteur w et de rayon $\frac{\ell}{2\pi}$. Un ruban de Moebius est très similaire sauf que la somme du vrillage et de la torsion sur la longueur est égale à π radians. Ces deux types de rubans sont illustrés en figure 6.1. Dans notre cas les rubans de Moebius auront une courbure naturelle nulle. Pour créer un ruban de Moebius, on peut suivre la construction déjà évoquée en section 2.1 : on coupe un ruban cylindrique dans sa largeur, on tord d'un demi-tour l'un des bords et on recolle.

Le ruban de Moebius requiert un peu d'attention. Il faut faire attention au mot «côté».

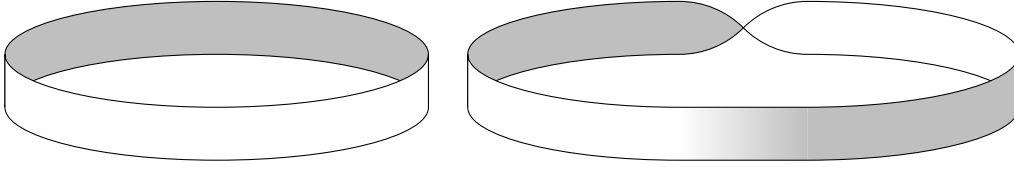


FIGURE 6.1 – Vue schématique d'un ruban circulaire (à gauche) et d'un ruban de Moebius (à droite). L'un des côtés du ruban est plus foncé. Pour le ruban de Moebius il y a de la torsion de π radians schématisée à «l'arrière» du ruban donc les côtés «s'inversent» comme montré avec le dégradé de couleur. On dit alors aussi qu'il n'y a qu'un seul côté.

Usuellement on dira qu'un ruban a deux côtés. Dans la description géométrique de notre modèle les deux côtés sont définis par la normale à la surface \mathbf{t}_y . Cependant, au point de recollement, c'est-à-dire à la jonction entre $s = 0$ et $s = \ell$ la normale va s'inverser. En effet pour garder une surface lisse avec la somme du vrillage et de la torsion égale à π radians, le repère matériel est tourné d'un demi-tour autour de la tangente à la centerline. On a ainsi $\mathbf{t}_x(0) = -\mathbf{t}_x(\ell)$, $\mathbf{t}_y(0) = -\mathbf{t}_y(\ell)$ et $\mathbf{t}_z(0) = \mathbf{t}_z(\ell)$. Pour cette raison, on dira couramment qu'il n'y a qu'un seul côté, c'est comme si l'un côté devenait l'autre (ce que l'on représente avec le fondu blanc gris en figure 6.1). Ici, on garde la version deux côtés.

Ainsi on peut calculer une contrainte sur η au point de recollement. En début de ruban, par définition, nous avons $\mathbf{q}(0) = \mathbf{t}_x(0) + \eta(0)\mathbf{t}_z(0)$ et en fin du ruban $\mathbf{q}(\ell) = -\mathbf{t}_x(0) + \eta(\ell)\mathbf{t}_z(0)$. Pour que les deux règles soient identiques, les vecteurs directeurs des règles \mathbf{q} doivent être colinéaires. On trouve $\mathbf{q}(0) = -\mathbf{q}(\ell)$ en imposant $\eta(\ell) = -\eta(0)$.

On peut s'inspirer des contraintes (5.6) pour relier le segment final du ruban au segment initial. Il suffit de substituer l'indice i par p et l'indice $i + 1$ par 1 avec p le nombre de segments du ruban. Bien évidemment, cette contrainte remplace la contrainte d'encastrement en fin de ruban. Rappelons comme vu en section 5.1 que la contrainte de raccord entre les segments est satisfaite lorsque le repère matériel de fin de segment est identique au repère matériel de début du segment suivant, mais aussi lorsque l'un est tourné de π radians autour de son axe de la tangente par rapport à l'autre repère¹. Il n'y a donc pas besoin d'adapter les contraintes pour le cas du ruban de Moebius. Aussi comme c'était le cas pour l'encastrement en fin de ruban, il faut faire attention lors du calcul de l'état d'équilibre pour ne pas tomber sur une solution aux contraintes mathématiquement correcte mais non désirée. On verra en section 6.1.1, sur l'exemple du ruban de Moebius comment calculer l'équilibre.

On introduit la valeur $\delta = 1$ pour le cas des rubans circulaires et $\delta = -1$ pour le cas des rubans de Moebius. Pour garder une surface continue, les règles au point de recollement doivent correspondre. On définit alors la contrainte analytiquement $\eta(\ell) = \delta\eta(0)$, puis adaptée à notre modèle numérique avec un ruban de p segments.

$$m^{(p)} + \ell^{(p)}n^{(p)} = \delta m^{(0)} \quad (6.1)$$

On considère une alternative possible. On conserve les deux encastrements en les considérant séparément comme dans le chapitre 5. Puis on impose $\eta(0) = \eta(\ell) = 0$. Cette alternative est restrictive, car on impose aux règles des extrémités du ruban d'être perpendiculaire à la centerline. Elle peut cependant se justifier dans certains cas, par exemple pour supprimer une inva-

1. Ce ne sont pas les deux seules solutions.

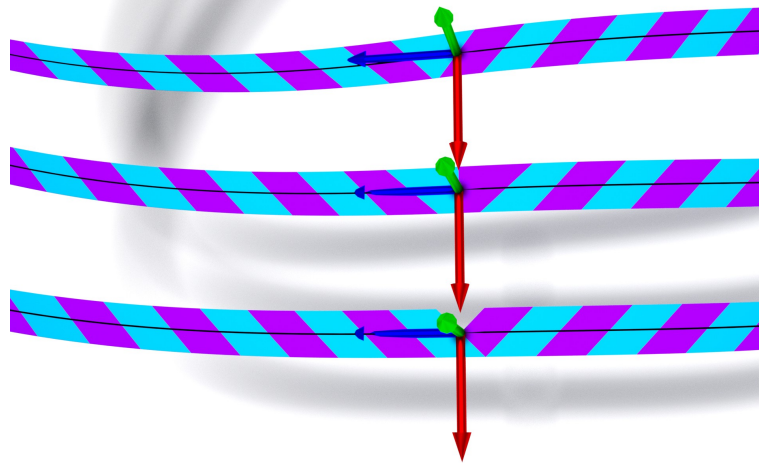


FIGURE 6.2 – Raccord sur Moebius – On illustre ici la nécessité de contraindre η sur un ruban de Moebius. Le ruban est en teinte bleu/violet pour distinguer les segments, et le repère matériel est tracé au point $s = 0$. En haut, on force $\eta(0) = -\eta(\ell)$, au milieu $\eta(0) = \eta(\ell) = 0$ et en bas il n'y a pas de contrainte sur η . Il apparaît clairement que sans contrainte le ruban n'est pas continu. Un événement de Wunderlich apparaît sur le ruban (on rappelle que cela correspond à la formation d'un point singulier, quand les génératrices s'intersectent sur le bord du ruban en $u = \pm \frac{w}{2}$). Suivant que la valeur de η soit fixée ou non, l'événement sera centré en $s = 0$ ou non. La contrainte $\eta(0) = -\eta(\ell)$ permet au ruban de « glisser ». C'est-à-dire que la forme est la même à transformation solide près, mais l'encastrement a changé d'endroit. Voir les courbes correspondantes sur la figure 6.3

riance qui peut entraver la convergence de l'algorithme de minimisation. Cependant, on verra en 7.1 un exemple où fixer η en un point change radicalement l'équilibre. Il faut alors justifier la possibilité d'usage de cette alternative.

Nous illustrons maintenant le propos abordé dans cette section. Nous calculons le ruban de Moebius en utilisant différentes contraintes sur η .

6.1.1 Un véritable ruban de Moebius

Nous allons calculer la forme d'un ruban de Moebius. Nous utilisons l'énergie de Wunderlich. De même que la gravité était négligée dans les articles de Sadowsky et Wunderlich [Sad29; Wun62], nous choisissons de ne pas mettre d'énergie de pesanteur pour la forme finale ($\Gamma = 0$). Nous choisissons un ratio de $\frac{w}{\ell} = 0.01$ pour la dimension et un coefficient de Poisson $\nu = 0.5$. En pratique, nous prenons $D = 1$ et $\ell = 1$. Finalement nous divisons le ruban en $p = 100$ segments de même longueur. Pour choisir ce nombre de segments, nous regardons la convergence en fonction du nombre de segments. Nous avons comme précédemment répété le calcul d'équilibre en utilisant de plus en plus de segments. Quand nous voyons une superposition des équilibres, nous déduisons que le nombre de segments utilisé est suffisant.

Il faut une procédure de calcul d'équilibre. On peut commencer par utiliser un ruban circulaire naturellement plat, encasté aux deux extrémités. Il est très facile de partir d'un tel ruban. On lui donne simplement une forme circulaire, sa courbure sera ainsi $\omega = \frac{2\pi}{\ell}$ et $\eta = 0$ puisqu'il n'y a pas de torsion. Puis nous tournons l'un des encastresments d'un demi-tour. En pratique,

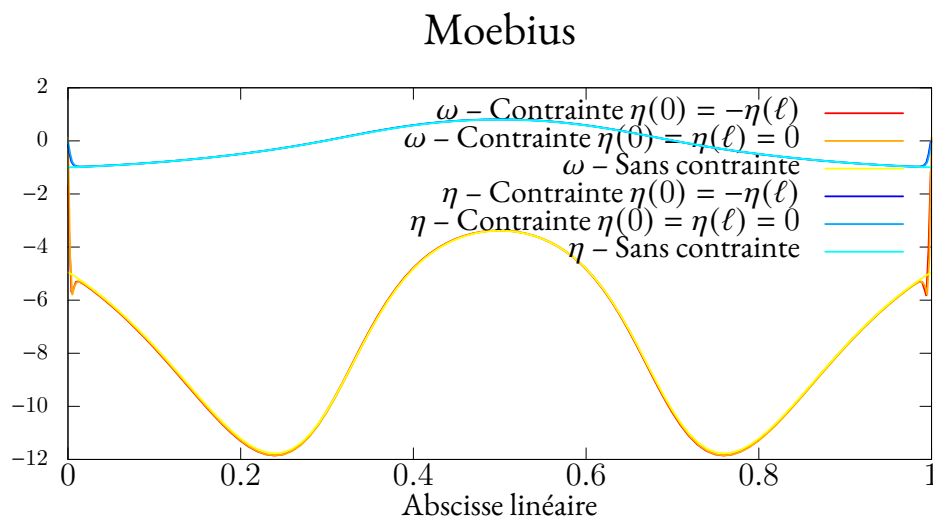


FIGURE 6.3 – Raccord sur Moebius – Tracé de la courbure normale ω et de la direction de la génératrice η d'un ruban de Moebius pour trois variations de contraintes. À l'équilibre, nous avons choisi les paramètres $\ell = 1$, $w = 0.01$, $\nu = 0.5$, $\Gamma = 0$. Le ruban est divisé en 100 segments de longueur identique.

Pour le tracé avec la contrainte $\eta(0) = -\eta(\ell)$ nous avons translaté les courbures selon s pour avoir superposition avec les autres courbes. En l'absence de gravité, il y a invariance de l'énergie par translation et rotation du ruban. Cela justifie la translation que l'on a nommée « glissement » en figure 6.2

nous utilisons l'encastrement en fin de ruban. Il ne faut pas effectuer ce demi-tour en une fois, mais en plusieurs étapes, on se retrouve avec des rubans dans des états intermédiaires qui permettent au ruban de ne pas tomber dans une mauvaise solution à l'encastrement. La dernière étape consiste à reformer le ruban, c'est-à-dire ajouter la contrainte nécessaire sur η et de connecter le premier et dernier segment du ruban à la place d'utiliser un encastrement en fin de ruban². Pour ne pas rester bloqué dans une configuration symétrique, il est généralement utile de perturber un peu l'état initial. On ajoutera aux valeurs $a^{(i)}$, $b^{(i)}$, $n^{(i)}$, $m^{(i)}$ ($i \in \llbracket 1, p \rrbracket$) un epsilon, généralement 10^{-4} . De plus on constate qu'un peu de gravité aide IPOPT à converger plus rapidement. En effet, le rang du hessien diminue sans l'énergie de gravité, il faut donc le régulariser. Ainsi, nous ne mettrons Γ à 0 qu'à l'étape finale.

On reprend et on détaille ces étapes dans l'algorithme 2. Pour perturber l'état initial, on a choisi $\omega(s) = 2\pi + 10^{-4}s$ et $\eta(s) = 10^{-2} + 10^{-4}s$. Les contraintes de Moebius font référence aux contraintes entre le premier et dernier segment, y compris (6.1).

Algorithme 2 : Algorithme calcul du Moebius

```

InitialiserRuban( $\ell = 1, w = 0.01, D = 1, \nu = 0.5, \Gamma = 0.1, p = 100$ );
FixerVerticale( $\begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$ );
pour  $i$  de 1 à  $p$  faire
    InitialiserSegment( $i, \ell^{(i)} = \frac{1}{100}, a^{(i)} = 10^{-4}, b^{(i)} =$ 
         $-2\pi + (i - 1)\frac{10^{-4}}{100}, n^{(i)} = 10^{-4}, b^{(i)} = 10^{-2} + (i - 1)\frac{10^{-4}}{100}$ );
 $\mathcal{F}_0 \leftarrow \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ ;
 $\mathbf{p}_0 \leftarrow \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$ ;
pour  $\theta$  de 0 à  $\pi$  par incrément de  $\frac{\pi}{5}$  faire
     $\mathcal{M} \leftarrow \begin{pmatrix} 0 & 0 & 1 \\ \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \end{pmatrix}$ ;
    FixerDoubleEncastrement( $\mathbf{p}_0, \mathcal{F}_0, \mathbf{p}_0, \mathcal{M}$ );
    AppelSolveur();
ChangerParametre( $\Gamma = 0$ );
FixerEncastrement( $\mathbf{p}_0, \mathcal{F}_0$ );
AjoutContraintesMoebius();
AppelSolveur();

```

Nous proposons deux variantes. La première se contente de ne pas appliquer la contrainte sur η (6.1). Le résultat attendu est une surface discontinue. C'est bien ce que l'on observe en figure 6.2. La deuxième variante est d'utiliser une contrainte $\eta(0) = \eta(\ell) = 0$ à la place de $\eta(0) = -\eta(\ell)$, et un double encastrement au lieu de relier le dernier au premier segment du ruban. Cela correspond à l'alternative donc on a parlé en section précédente.

Nous avons insisté sur le besoin de justifier la validité de la contrainte alternative. Tout d'abord, il existe un point où η est nul. Comme le ruban est circulaire, on peut prolonger les fonctions ω et η . Il faut cependant faire attention à la spécificité des rubans de Moebius. Nous

2. On peut se contenter d'un double encastrement si besoin.

avons déjà vu que la direction de la règle s'inversait en changeant de côté, or pour changer de côté, dans la cadre de la prolongation des fonctions, c'est ajouter ℓ à l'abscisse curviligne s . Ainsi $\eta(s + \ell) = -\eta(s)$. La prolongation de η reste continue par définition. D'après le théorème des valeurs intermédiaires, il existe un point dans $[0, 2\ell]$ tel que η s'annule³, donc dans $[0, \ell]$ aussi. Ensuite, sans gravité (ni autre force extérieure) n'importe quelle transformation solide laisse un état d'équilibre à l'équilibre. On déduit alors qu'imposer $\eta(0) = \eta(\ell) = 0$ est légitime, car le placement de l'encastrement correspond ainsi à un point où η s'annule.

On compare les équilibres obtenus par la méthode et ses deux variantes. Pour les nommer, on utilise la contrainte sur η . L'encastrement est visualisé en figure 6.2 et les fonctions ω et η sur le graphe 6.3. On observe que les courbures ne sont quasiment pas impactées par la gestion de l'encastrement. Notons qu'entre la version originale et l'alternative, il y a une translation des courbes selon s , puisque cette translation n'a pas de sens physique comme nous avons vu, les courbes « Contrainte $\eta(0) = -\eta(\ell)$ » sont décalées selon s pour se superposer aux autres. Sur la figure 6.2 on voit cette translation.

On remarque que vers le point $s = 0$ ou $s = \ell$ les courbes évoluent très vite. On va donc diviser les segments aux bords du ruban pour pouvoir mieux observer ce point de recollement. On rappelle en effet qu'il n'est pas nécessaire d'avoir des segments de longueur identique. Visuellement, tout semble indiquer un événement de Wunderlich (qui correspond à une intersection des génératrices au bord du ruban, on rappelle que le point d'intersection est nommé point singulier) à cet endroit. On choisit par exemple de diviser les trois premiers et trois derniers segments en six. Nous utilisons l'alternative avec la contrainte $\eta(0) = \eta(\ell) = 0$, sinon l'événement de Wunderlich peut « glisser » vers une autre abscisse. Il faudrait alors sous-diviser tout le ruban. L'équilibre calculé fait bien apparaître le point singulier comme prédit, voir figure 6.4. Il est intéressant de constater que même si dans cette zone, la courbure varie vite et nécessite de nombreux segments pour la capturer, on ne visualise pas de changement géométrique par rapport à avant. Les deux formes se superposent.

La présence de l'événement de Wunderlich n'est pas nouveau, Starostin et van der Heijden en parlent dans [SH07]. La largeur du ruban a un impact sur cette zone. Récemment, van der Heijden et Audoly ont dérivé la forme paramétrique de la courbure ($\kappa_1(s), \kappa_3(s)$) à l'événement de Wunderlich pour la limite des petites largeurs [HA22]. On trace donc ce graphe en figure 6.5 pour les largeurs de ruban de plus en plus petites. On remarquera que la direction des règles reste dans l'intervalle $[-1, 1]$. La question suivante concerne la taille de l'événement de Wunderlich. On la fait correspondre à l'abscisse du point de rebroussement (ce qui correspond aussi à la limite de la courbe théorique). Cette valeur n'est pas visible sur le graphe, car celui-ci est paramétrique. Nous la traçons sur le graphe en figure 6.6. L'abscisse qui définit la taille de l'événement de Wunderlich est notée s_w . On remarque que cette taille est quasiment linéaire en fonction de la largeur du ruban, le logiciel gnuplot nous interpole la fonction $s_w = 0.79 \frac{w}{\ell}$.

6.2 Ajout de contact entre le ruban et un plan

Lors d'une simulation dynamique, il est possible de calculer des points de contact précis où deux objets collisionnent. On peut alors ajouter une force ou une contrainte à cet endroit

3. Le théorème des valeurs intermédiaires spécifie que pour une fonction f continue sur un intervalle réel $[a, b]$, alors pour tout réel k entre $f(a)$ et $f(b)$, il existe un réel $c \in [a, b]$ tel que $f(c) = k$. Ici on prendra $a = 0.5\ell$ et $b = 1.5\ell$, η est continue entre a et b . $f(0.5\ell) = -f(1.5\ell)$ donc 0 est bien entre ces deux valeurs. Le théorème s'applique et donne le résultat.

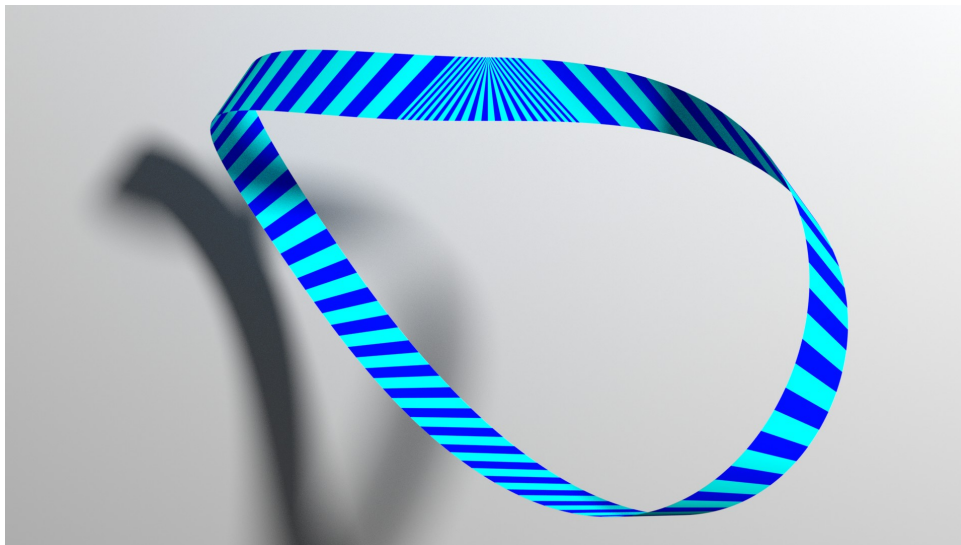


FIGURE 6.4 – Division des segments autour de l'événement de Wunderlich. On voit bien le point singulier apparaître.

pour que le système reste valide. En complément une étape de correction peut intervenir pour rester dans une configuration valide. Si l'on n'est intéressé que par l'équilibre, le fait qu'un pas de temps soit très amorti ou un peu imprécis n'a pas forcément beaucoup d'importance.

Une première solution serait alors de passer à un modèle « pseudo-dynamique », où sans se poser de question on confondrait la force et le gradient du système. On utilise le qualificatif « pseudo » car le gradient du système est calculé sur des coordonnées généralisées. Il y a deux inconvénients à cette approche.

D'abord, il y a un sacrifice de vitesse. Comme vu en section 3.5.2, n'utiliser que le gradient et non le hessien du système revient à une méthode de minimisation du premier ordre, inefficace en temps de calcul comparé à une méthode de second ordre. Cependant, il n'y a parfois pas le choix pour capturer certains phénomènes. Par exemple pour simuler un plectonème on tord un ruban pour lui faire emmagasiner de la torsion, puis on rapproche les deux bords. Le ruban forme alors une boucle et les deux côtés s'enroulent l'un autour de l'autre comme illustré en figure 6.7. Si l'on essaye de simuler ce comportement, un minimiseur d'énergie aura tendance à sauter vers un équilibre sans torsion, car moins coûteux en énergie. Un modèle « pseudo dynamique » au contraire ne pourra pas sauter vers un équilibre sans torsion à cause du contact.

Ensuite, il faut gérer les contraintes que l'on a introduites avec le modèle mixte. La dynamique rend cela assez complexe. On ne peut pas se contenter d'une approximation linéaire des contraintes par pas de temps, car les segments se détacheraient les uns des autres⁴. Le contact n'aurait alors plus trop de sens puisque le ruban ne serait plus une surface continue. De plus, si les repères matériels à la jonction entre les segments s'éloignent trop, par non-unicité des solutions aux contraintes d'alignement des repères, les segments pourraient mal se reconnecter entre eux. Ainsi, dû aux contraintes de raccord entre les segments, le coût d'un pas de temps serait alors conséquent. Ces raisons nous semblent suffisantes pour ne pas creuser cette direction de recherche.

4. S'il n'y a pas d'encastrement en fin de ruban, on pourrait imaginer une méthode pour conserver la continuité du ruban, en recollant manuellement les segments entre eux après chaque pas de temps.

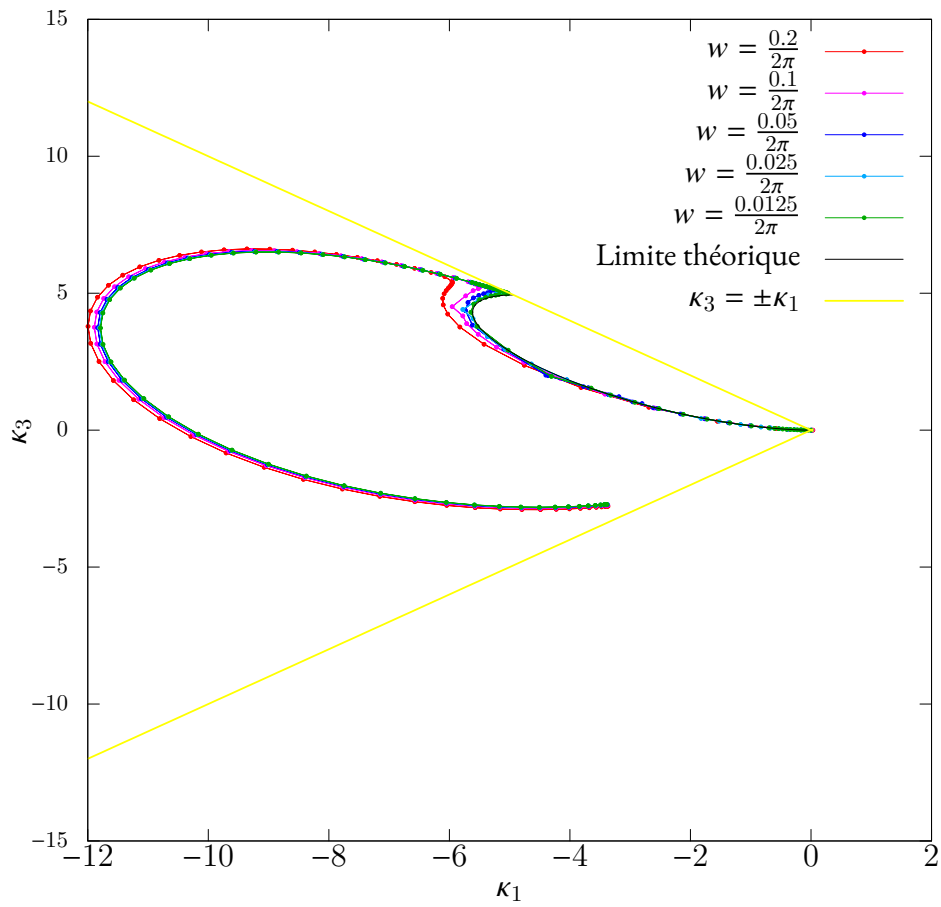
Moebius, Wunderlich ($\ell = 1$)

FIGURE 6.5 – Tracé paramétrique de la courbure du ruban de Moebius pour différentes largeurs. La courbe tend vers une limite [HA22] lorsque la largeur w tend vers 0. On remarque aussi que la direction des règles $\eta = \frac{\kappa_3}{\kappa_1}$ est inférieure à 1 en valeur absolue.

Moebius, Wunderlich, taille de l'évènement de Wunderlich adimensionné

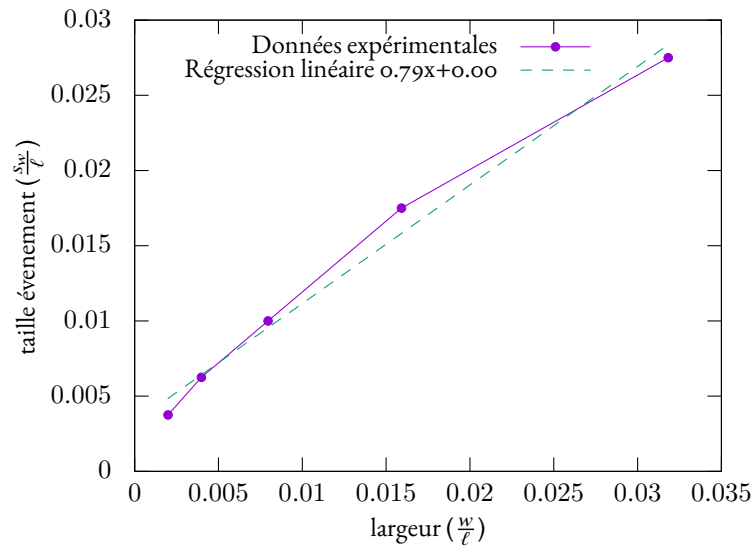


FIGURE 6.6 – Tracé de la taille de l'évènement de Wunderlich en fonction de la largeur du ruban.

La solution retenue est donc d'enrichir le problème statique. On ajoute des contraintes au modèle pour traduire la non-pénétration. L'utilisation d'IPOPT nous impose un certain format pour poser le problème de minimisation. En particulier, on ne peut pas ajouter ou supprimer des contraintes au cours de la simulation. Il faut indiquer toutes les contraintes avant de commencer la minimisation. Cela aurait permis d'ajouter une contrainte lorsque l'on détecte un contact et ainsi de contrôler et limiter leur nombre. Cependant, le problème ne serait plus continu. Pour obtenir ce contrôle, il faudrait certainement utiliser ou concevoir un algorithme alternatif à IPOPT.

Gérer l'auto-contact via des contraintes, semble être complexe à cause de la géométrie basée sur la courbure du ruban. De plus si l'on compte une contrainte par segment, on obtient un nombre quadratique de contraintes par rapport au nombre de segment. On risque de surcontraindre le problème si trop de contraintes deviennent actives à la fois. Le choix a donc été fait de n'ajouter que du contact qu'avec un objet externe, en l'occurrence un plan. Pour simplifier encore plus, nous ne considérons les contacts qu'en début du ruban, en fin de ruban et à chaque jonction entre les segments. Si ces derniers sont assez petits, cela ne devrait pas perturber les équilibres. Notons alors que le nombre de segments requis par un ruban va grossir. Sur ce point, la complexité linéaire du calcul du système pour le modèle mixte en le nombre de segments est bien pratique quand celle du modèle chaîné rendrait le contact en pratique ingérable.

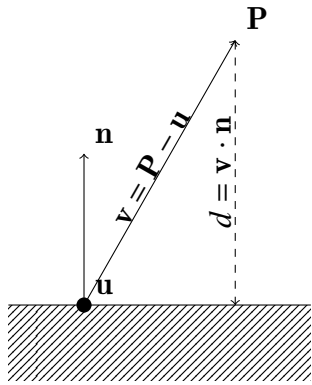
On peut placer un unique point de contact potentiel par segment. On choisit $\mathbf{r}^{(i)}(0)$ qui est directement donné par les degrés de liberté. Ainsi la contrainte et ses dérivées sont numériquement très facile à calculer. Comme les segments sont collés les uns aux autres il n'y a pas besoin de rajouter une contrainte pour une potentielle collision en fin du segment, excepté pour le dernier segment.

Détecter le contact uniquement sur la centerline est un début, mais le ruban ne va en général pas rester en équilibre sur une seule courbe. De façon analogue, si l'on pose un objet très étroit sur la tranche, il va tomber. On peut alors utiliser les deux bords du ruban ($u = \pm \frac{w}{2}$) pour



FIGURE 6.7 – Un plectonème de ruban (photographie). Le ruban emmagasine de la torsion qui donne cette forme caractéristique.

pouvoir poser un ruban sur la tranche. Similairement au dernier paragraphe on place un point de contact aux deux extrémités des règles en début de segment.



Si on a un point \mathbf{P} et un plan (\mathbf{u}, \mathbf{n}) où \mathbf{u} est un point du plan et \mathbf{n} sa normale, on peut projeter le vecteur $\mathbf{P} - \mathbf{u}$ sur la normale pour calculer une distance signée d de \mathbf{P} au plan. On illustre cela sur la figure 6.8. Il suffit alors que cette distance soit positive pour être sûr que le point reste du même côté. Notons que le sens de la normale est donc important. Nous définissons ainsi les contraintes de non-collision avec le plan (\mathbf{u}, \mathbf{n}) pour un segment i :

$$\left\langle \mathbf{r}^{(i)}(0) + \frac{w}{2} \left(\mathbf{t}_x^{(i)}(0) + m^{(i)} \mathbf{t}_z^{(i)}(0) \right) - \mathbf{u}, \mathbf{n} \right\rangle \geq 0 \quad (6.2a)$$

$$\left\langle \mathbf{r}^{(i)}(0) - \frac{w}{2} \left(\mathbf{t}_x^{(i)}(0) + m^{(i)} \mathbf{t}_z^{(i)}(0) \right) - \mathbf{u}, \mathbf{n} \right\rangle \geq 0 \quad (6.2b)$$

FIGURE 6.8 – Distance signée d'un point à un plan

On peut généraliser la contrainte (6.2) en n'utilisant pas le début du segment, mais une abscisse locale quelconque.

On obtient : $\left\langle \mathbf{r}^{(i)}(\hat{s}^{(i)}) \pm \frac{w}{2} \left(\mathbf{t}_x^{(i)}(\hat{s}^{(i)}) + \left(m^{(i)} + \hat{s}^{(i)} n^{(i)} \right) \mathbf{t}_z^{(i)}(\hat{s}^{(i)}) \right) - \mathbf{u}, \mathbf{n} \right\rangle \geq 0$.

Bien évidemment on ne peut pas appliquer cette contrainte de manière continue, il faut choisir des abscisses discrètes. Jusqu'ici on avait pris les abscisses $\hat{s}^{(i)} = 0$ (deux points en début de chaque segment), et $\hat{s}^{(p)} = \ell^{(p)}$ (deux points à la fin du dernier segment).

On peut se demander si on pourrait utiliser de points de contact additionnels. On pourrait alors utiliser la contrainte en milieu de segment ou à intervalle régulier sur l'abscisse curviligne. A priori, plus il y a de points où l'on détecte le contact, mieux ce sera. Si cela est vrai, on peut aussi utiliser plus de segments pour le même résultat. Certes en doublant le nombre de segments,

le coût de calcul est doublé. Mais d'un autre côté, pour évaluer la contrainte et ses dérivées en milieu de segment, il faut aussi recalculer la cinématique en ce point. Le surcoût en ajoutant une contrainte en milieu de segment est quasiment le même que le surcoût pour doubler le nombre de segments. De plus, l'ajout de contraintes rajoute un risque de sur-contraindre le problème. Il peut donc être préférable d'utiliser plus de segments pour raffiner la gestion du contact.

À présent, nous allons illustrer l'utilisation du contact sur deux exemples.

6.2.1 Ruban de Moebius sur la table

Notre modèle permet de simuler efficacement sans grande difficulté un ruban de Moebius sur la table. En pratique une table est un plan infini. On peut partir du ruban de Moebius classique, section 6.1.1. Si les encastremets correspondent bien à l'événement de Wunderlich, alors on a les symétries $\eta(s) = \eta(\ell - s)$ et $\omega(s) = \omega(\ell - s)$. Ainsi, pour poser le ruban sur la tranche, on place le plan au bon endroit, c'est-à-dire, en $\mathbf{r}\left(\frac{\ell}{2}\right)$ et tangent au ruban. Finalement on fixe la verticale $\mathbf{e}_g = \mathbf{t}_y\left(\frac{\ell}{2}\right)$ orthogonale au plan. Contrairement à ce qui est fait en 6.1.1, on n'annule pas la gravité. De cette façon, la gravité ne va pas faire basculer le ruban car le moment de la force de gravité sur tout le ruban est nulle. En lâchant le ruban, c'est-à-dire en supprimant les contraintes d'encastrement, il devrait alors tenir sur la tranche. Si on choisit un plan quelconque, c'est-à-dire indépendamment de la déformation du ruban, en lâchant ce dernier, il va généralement basculer et tomber. On illustre cela en figure 6.9.

Pour lâcher le ruban, on supprime les contraintes d'encastrement, on utilise la contrainte adéquate, pour relier le dernier segment au premier, et on force $\eta(0) = \eta(\ell) = 0$. Cette dernière contrainte sur η a l'avantage de conserver l'événement de Wunderlich localisé en $s = 0$ comme on le voulait pour retrouver la symétrie annoncée au paragraphe précédent. Aussi la position et le repère matériel au milieu du ruban correspondent à un début de segment.

Notons que ce résultat d'expérience n'est pas validé par une comparaison avec un autre modèle, mais l'expérience suivante est bien validé par la théorie.

6.2.2 L'expérience de Roman

L'expérience de Roman est décrite dans [RP99] et dans la thèse de Benoît Roman [Romoo]. On la schématise en figure 6.10. Un ruban est encasté aux deux extrémités sur une plaque. Les deux repères d'encastrement sont identiques et sont tels que la tangente au ruban en ses points soit colinéaire à la direction entre les encastremets. Ensuite, le ruban est écrasé par une autre plaque. Deux grandeurs sont introduites : H la distance entre les plaques et Δ la fraction de longueur du ruban qu'il manque entre les encastremets pour que le ruban devienne plat. Dans un cadre adimensionné ℓ vaut 1.

L'expérience est 2D, c'est-à-dire que toute coupe du ruban dans sa longueur (à u fixé) sera planaire et aura la même forme. Il existe cependant un cas un peu particulier illustré sur la figure 6.11 : si le ruban glisse latéralement, il peut sortir de sa configuration plane. Ici on se limite au cas 2D. Selon l'écrasement H , le ruban va pouvoir former des vagues. Si la centerline ne touche la plaque qu'en un point, on parle de contact linéique. Si la vague est en contact sur tout un domaine (surface non nulle), c'est un contact plan. Sinon, il n'y a pas de contact entre la vague et la plaque, on parle de pli en suspens. Chaque type de pli est illustré en figure 6.11. L'altitude d'un point \mathbf{P} est la valeur $P_y - r_y(0)$, l'orientation du repère canonique est précisé sur le schéma 6.10.

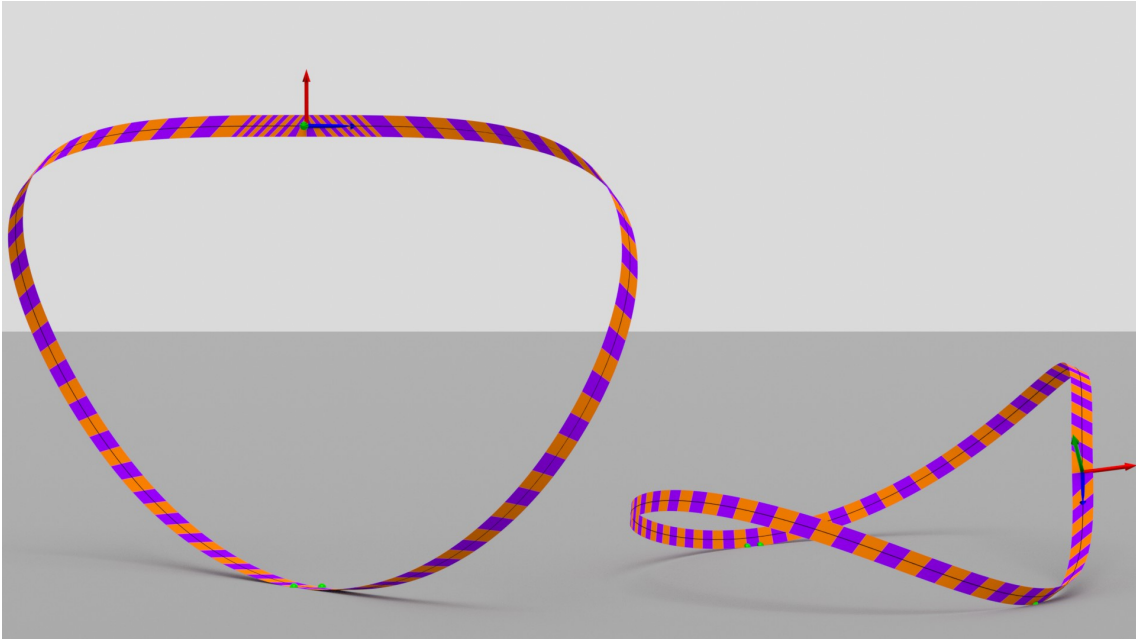


FIGURE 6.9 – Rubans de Moebius posés sur un plan – Les rubans ont été calculés avec les paramètres physiques $Dw = 1$, $\frac{w}{\ell} = 0.01$, $\nu = 0.5$, $\Gamma = 10$. À gauche, le ruban est posé sur la tranche suivant la méthode décrite en 6.2.1. Grâce à l'utilisation de la contrainte $\eta(0) = \eta(\ell) = 0$, l'emplacement de l'événement de Wunderlich n'a pas bougé, comme le montrent les petits segments à cet emplacement. À droite, le ruban est tombé. Ces rubans sont composés initialement de 100 segments identiques, mais à droite les six segments autour de l'événement ont été scindés en 4 segments chacun. Les points de contact au sol sont représentés avec des sphères vertes.

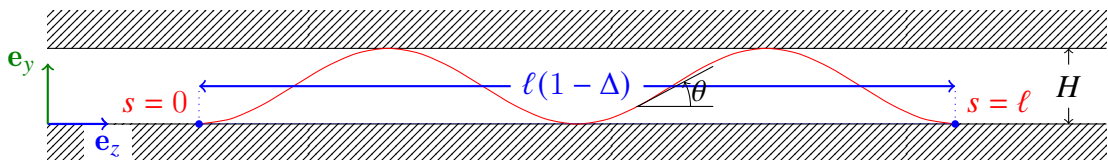


FIGURE 6.10 – Schéma de l'expérience de Roman – L'expérience de Roman est 2D. Un ruban (en rouge) est encastré sur une plaque (aux points bleus), puis aplati par une deuxième plaque parallèle à la première. Des vagues vont apparaître en fonction de l'écrasement H et de l'écartement Δ entre les encastresments.

Pour décrire la centerline du ruban, Benoît Roman décompose le problème. En partant de l'hypothèse qu'il n'y a que des contacts linéiques, il prend un morceau du ruban constituant une demie vague, c'est-à-dire d'un point d'altitude minimale (o) au point d'altitude maximale (H) le plus proche, ou l'inverse. On retrouve alors l'équation de l'elastica :

$$Dw \frac{d^2\theta(s)}{ds^2} = -p \sin(\theta(s)) + q \cos(\theta(s)) \quad (6.3)$$

où $\theta(s)$ est l'angle entre la tangente et la plaque du dessous (cf. figure 6.10). Pour rappel D est la rigidité de flexion. Le scalaire p est le multiplicateur de Lagrange fixant l'altitude du point maximal, il peut donc être interprété comme la force du ruban sur la plaque du dessus. La valeur de q correspond à la longueur de la demi vague, c'est-à-dire la distance entre les points extrémaux projetée sur l'axe e_z . On note $F(H)$ la force du ruban sur la plaque supérieure en fonction de l'écrasement. Dans sa thèse Benoît Roman explique comment obtenir une loi d'échelle qui relie cette force F avec l'écrasement et le nombre de vagues. Si on multiplie les distances d'un facteur λ , l'équation (6.3) devient $YI\lambda^2\ddot{\theta} = -p \sin(\theta) + q \cos(\theta)$, c'est comme si p et q étaient multipliés par le facteur $\frac{1}{\lambda^2}$. Maintenant imaginons que l'on ait une vague avec un contact linéique (un seul point de contact de la centerline sur la plaque du haut), on a la force p . On choisit $\lambda = \frac{1}{N}$, c'est-à-dire qu'on divise les dimensions par N . Ensuite on duplique le résultat N fois et on "colle" les vagues les une à la suite des autres. On a ainsi obtenu un ruban formant N vagues identiques. La largeur totale $1 - \Delta$ est égale à la largeur d'origine pour une vague et l'écrasement H a été divisé par N . La figure 6.13 représente cette transformation. Chaque vague exerce une force pN^2 sur la plaque supérieure, soit au total pN^3 . On obtient ainsi la loi d'échelle :

$$F\left(\frac{H}{N}\right) = N^3 F(H) \quad (6.4)$$

Notons que cette loi d'échelle fonctionne aussi en passant de k à nk vagues. Il est important de remarquer que cette loi ne fonctionne que pour des contacts linéiques.

Dans l'algorithme 3, nous décrivons un moyen de tester cette loi d'échelle. La largeur du ruban a arbitrairement été mise à 0.2. Pour tester une loi d'échelle cette valeur n'a pas d'importance. Cependant, utiliser une grande largeur permet au ruban de rester dans une configuration 2D. Le choix a été fait d'utiliser la matrice identité \mathcal{I}_3 pour repère matériel aux encastresments (en cohérence avec les axes sur le schéma 6.10). Pour ne pas avoir de problèmes de résolution le facteur ϵ fixé à 10^{-6} précise bien de quel côté de la plaque est le ruban à l'état initial. Il y a deux faits importants dans l'algorithme. D'abord il faut utiliser plus de segments si l'écrasement est faible que s'il est grand. En effet, s'il y a plus de vagues il vaut mieux avoir plus de points pour détecter la collision. Pour cela le coefficient `toleranceSubdivision` va permettre de doubler le nombre de segments (c'est la fonction `SousDiviserRuban`) chaque fois que H est divisé par deux. Ensuite H n'est pas baissé linéairement mais en suivant une loi exponentielle. La raison en est simple : le nombre de vagues n'est pas linéaire, mais plutôt en $\frac{1}{H}$ ⁵.

Nous avons précédemment mentionné dans le texte une façon de calculer une force. Nous la rappelons ici. Si une contrainte correspond à une distance, ou plus précisément au vecteur entre un point fixe et un point du ruban, nous pouvons déduire que la force associée à cette

5. Benoît Roman précise dans sa thèse que le nombre de vagues n'est pas une fonction bien définie, car il y a plusieurs équilibres possibles à écrasement H fixé. Si l'on veut une approximation, on peut remplacer le ruban par des triangles, ce qui donne $N = \frac{\ell\Lambda}{2Y}$. On peut aussi utiliser une forme sinusoïdale, mais l'ordre de grandeur reste le même.

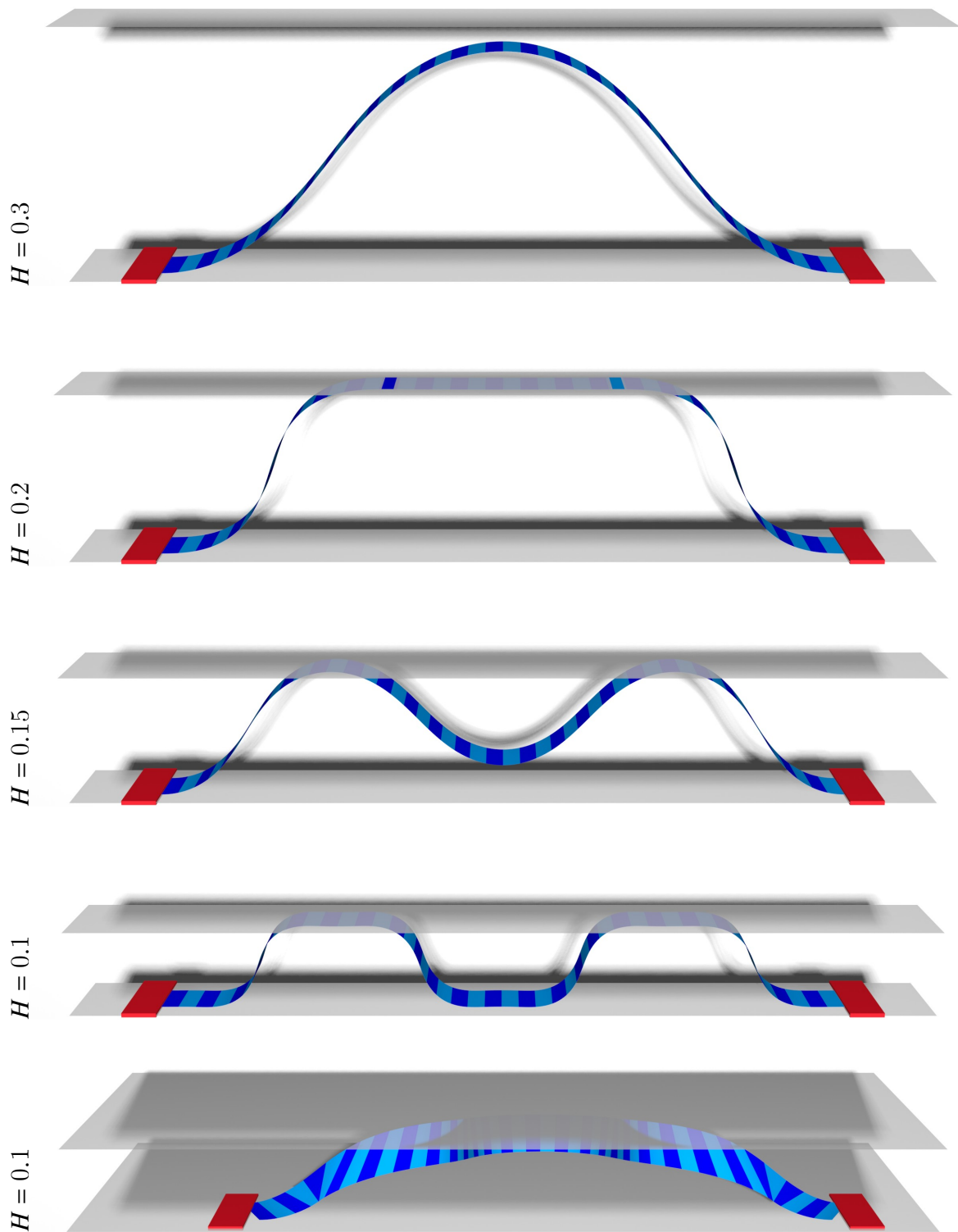


FIGURE 6.11 – Expérience de Roman - Un ruban est attaché entre deux plaques que l'on rapproche progressivement. Sur les quatre premières vues, nous voyons les plis (vagues) se former. Sur la dernière figure nous avons rapproché les encastresments, et le ruban sort de sa configuration planaire pour passer en 3D. Les encastresments sont représentés en rouge, les différents segments sont visibles en teintes de bleu. Les plaques sont semi-transparentes, leur taille est virtuellement infinie. À noter : pour les raisons expliquées en section 6.2, il peut y avoir un peu de pénétration dans les plaques au sein d'un même segment. Sur ces figures on utilise $\ell = 1$, $w = 0.05$, $\Gamma = 10$. L'expérience utilise l'énergie de Wunderlich. Comme le suggère la figure du bas, il n'y a pas de contraintes sur η aux encastresments. Nous avons $\Delta = 0.2$ sur les quatre premières figures et $\Delta = 0.3$ sur la dernière.

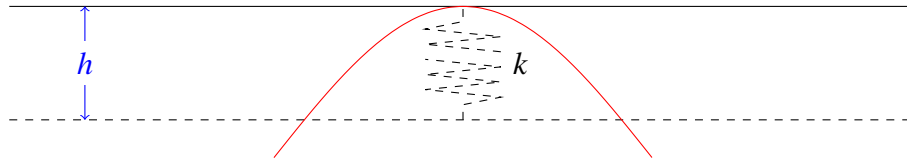


FIGURE 6.12 – Schéma de la plaque de pression – Le principe consiste à attacher une plaque à un ressort d’une raideur k . Ce ressort est virtuel, tout comme son attache, c’est pourquoi ils sont tracés en pointillés. La longueur à vide est nulle. Le ruban va donc traverser l’attache du ressort. La plaque subit deux forces, celle du ressort et celle du ruban. Ainsi, à l’équilibre la force de contact du ruban vaut kh .

contrainte est le multiplicateur de Lagrange correspondant (voir l’annexe A). Dans notre cas chaque contrainte entre le plan supérieur et un point du ruban correspond à la distance entre ce plan et ce point, c’est donc la composante orthogonale au plan de la force correspondante. Or cette force de contact entre le plan et le ruban est normale au plan. Si une contrainte n’est pas active, le multiplicateur de Lagrange associé est nul. En conclusion, en sommant les multiplicateurs de Lagrange relatifs aux contraintes du plan supérieur, on retrouve la force totale du ruban sur la plaque.

Pour vérifier la cohérence de la théorie avec la pratique, on peut utiliser une méthode alternative. On utilise une plaque de pression, voir le schéma en figure 6.12. Le principe est très simple, la plaque du haut n’est pas fixe mais reliée à un ressort. À l’équilibre, on obtient la force de contact entre le ruban et la plaque par le produit kh (voir la légende du schéma). Cette méthode a un inconvénient majeur : l’écrasement n’est pas contrôlable précisément. En effet, si on peut précisément placer l’accroche du ressort à une distance y de la plaque inférieure, on obtient l’écrasement $H = y + h$. Il y a deux approches pour contrôler H . Soit on choisit k assez grand pour que h soit négligeable. Cette option est similaire à de la pénalisation. Elle n’est pas idéale car le solveur aura plus de mal à minimiser l’énergie si les différentes énergies sont très différentes en ordre de grandeur (voir aussi l’adimensionnement en section 4.1 sur ce sujet). Soit on recherche le y nécessaire par dichotomie⁶ pour obtenir l’écrasement voulu. Cette option ralentit cependant le processus puisqu’il faut calculer plusieurs équilibres avant de tomber sur le bon.

Expérimentalement, en réalisant l’expérience de Roman sur quelques points via la méthode des plaques de pressions avec le contrôle de l’écrasement H par dichotomie sur y ou via la méthode standard, on mesure la même force. Nous resterons alors avec la première approche (plus aisée à mettre en place) où la force est mesurée par les coefficients de Lagrange.

6. Une dichotomie consiste à rechercher une valeur x dans un intervalle $[a, b]$. En se servant de propriété du problème, il suffit de faire un test sur la valeur médiane $m = \frac{a+b}{2}$ pour savoir si x est dans l’intervalle $[a, m]$ ou $[m, b]$. À chaque itération la taille de l’intervalle est divisé par deux, d’où une convergence rapide (logarithmique) de cette méthode.

Algorithme 3 : Roman, écrasement**Entrées** : Δ

```

InitialiserRuban( $\ell = 1, w = 0.2, D = 1, \nu = 0.5, \Gamma = 10, p = 100$ );
FixerVerticale( $(0 \ 1 \ 0)$ );
FixerDoubleEncastrement( $(0 \ 0 \ 0), \mathcal{I}_3, (0 \ 0 \ 1 - \Delta), \mathcal{I}_3$ );
AjoutPlaque( $y = 0 - \epsilon$ );
AppelSolveur();
alt ← CalculSommetVague();
plafond ← AjoutPlaque( $y = \text{alt}$ );
toleranceSubdivision ← 0.5;
pour  $i$  de 0 à 2.3 par incrément de 0.01 faire
     $p \leftarrow \exp -1$ ;
    si  $p < \text{toleranceSubdivision}$  alors
        toleranceSubdivision ←  $\frac{\text{toleranceSubdivision}}{2}$ ;
        SousDiviserRuban();
    BougerPlaque(plafond,  $y = p \times \text{alt}$ );
    AppelSolveur();
     $N \leftarrow \text{CalculNombreVagues}()$ ;
     $F \leftarrow \text{MesureForces}()$ ;
    EnregisterMesures( $p \times \text{alt}, N, F$ );

```

La simulation numérique se passe sans encombre. Pour $\Delta = 0.1$ on trouvera par exemple le résultat figure 6.14. Les résultats correspondent à la théorie.

Pour aller un peu plus loin, nous avons étudié l'influence de Δ sur le graphe. Nous générons les données pour $\Delta \in \{0.1, 0.11, \dots, 0.25\}$. Pris individuellement (à Δ fixé) les graphes $(HN, \frac{F(H)}{N^3})$ n'ont pas grand intérêt, car ils se ressemblent. Pour chacun, le point intéressant est celui où la force s'annule. L'abscisse de ce point HN dépend cependant de Δ . Nous définissons donc un facteur $k(\Delta)$ pour faire correspondre ces points particulier où la force F est nulle. Nous imposons aussi $k(\Delta)HN = 1$ lorsque $F(Y) = 0$. Expérimentalement nous observons qu'en multipliant aussi l'ordonnée des points par $k(\Delta)$ les données correspondant aux contacts linéiques se superposent. En figure 6.15, on trace le graphe $(k(\Delta)HN, \frac{k(\Delta)F(H)}{N^3w})$. Chaque couleur représente une valeur de Δ différente. On voit les graphes se superposer. Notons que nous avons ici choisi de diviser la force par la largeur. Cela a du sens, car la force est proportionnelle à la largeur du ruban.

Pour retrouver le facteur k on part de l'équation de l'élastica 6.3 avec $p = 0$ puisqu'on s'intéresse au point où la force s'annule. L'équation permet de relier Δ à H . À l'ordre 2, $\Delta = \frac{\pi^2}{4}H^2$. Par parité de cette fonction, l'ordre 3 est nul. On cherche alors visuellement le coefficient à l'ordre 4 et on trouve 6. En resolvant $\Delta = \frac{\pi^2}{4}H^2 + 6H^4$, on trouve $4\sqrt{3}H = \sqrt{-\pi^2 + \sqrt{\pi^4 + 384\Delta}}$. On pose alors le coefficient correspondant :

$$k(\Delta) = \frac{\sqrt{-\pi^2 + \sqrt{\pi^4 + 384\Delta}}}{H_0 \sqrt{-\pi^2 + \sqrt{\pi^4 + 384\Delta}}} \quad (6.5)$$

Ici H_0 est la hauteur de la vague pour $\Delta = 0.1$ s'il n'y a pas de plaque supérieure.

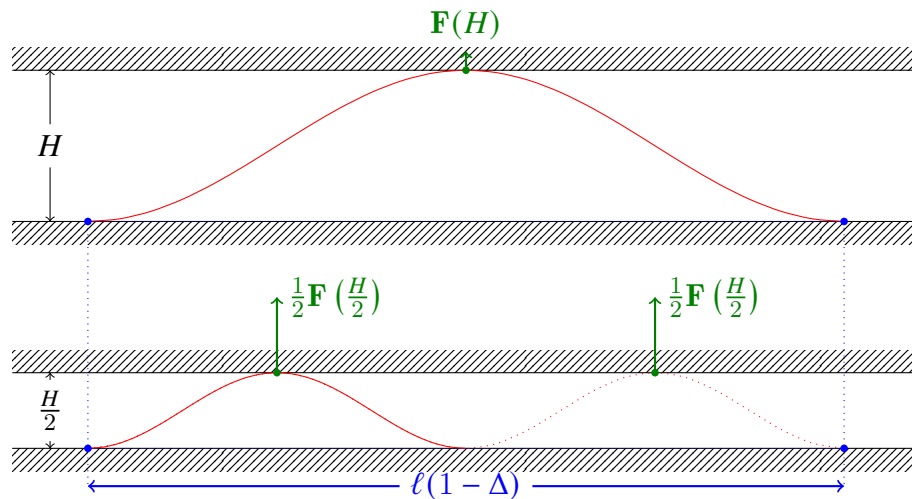


FIGURE 6.13 – Loi d'échelle de l'expérience de Roman – Ici nous choisissons $N = 2$. À partir d'une seule vague (en haut) on divise les longueurs par 2 (en bas). Ensuite on prolonge le ruban en le dupliquant (en pointillés). La largeur est commune aux deux configurations. On représente les forces que le ruban applique sur la plaque du haut en vert.

Loi d'échelle – $\Delta = 0.1$

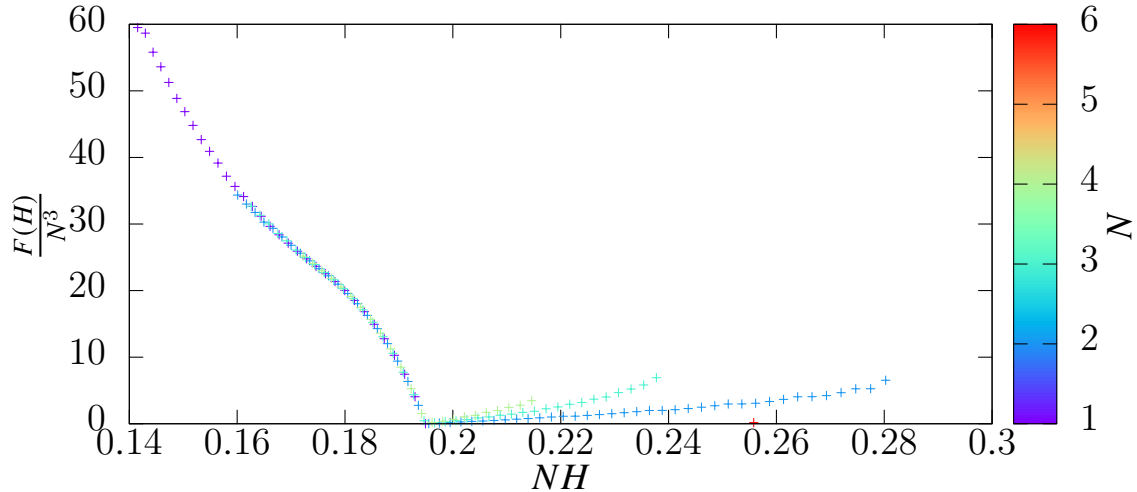


FIGURE 6.14 – Loi d'échelle pour l'expérience de Roman – On trace ici, à Δ fixé la force divisée par le nombre de vagues au cube qui doit correspondre à la force pour une vague en fonction de l'écrasement multiplié par le nombre de vagues. On vérifie ainsi que $F(H) = N^3 F(NH)$. La couleur des points correspond au nombre de vagues. On observe une correspondance sur la partie descendante ($NH < 0.195$ environ), mais pas sur l'autre partie. Cela est dû à la nature des contacts : il y a un pli en suspens (qui ne touche pas la plaque supérieure). Les points ne se superposent pas non plus pour les contacts plans, mais ceux-ci n'apparaissent pas vraiment pour 2 vagues ou plus. Ils correspondent aux points en haut à gauche.

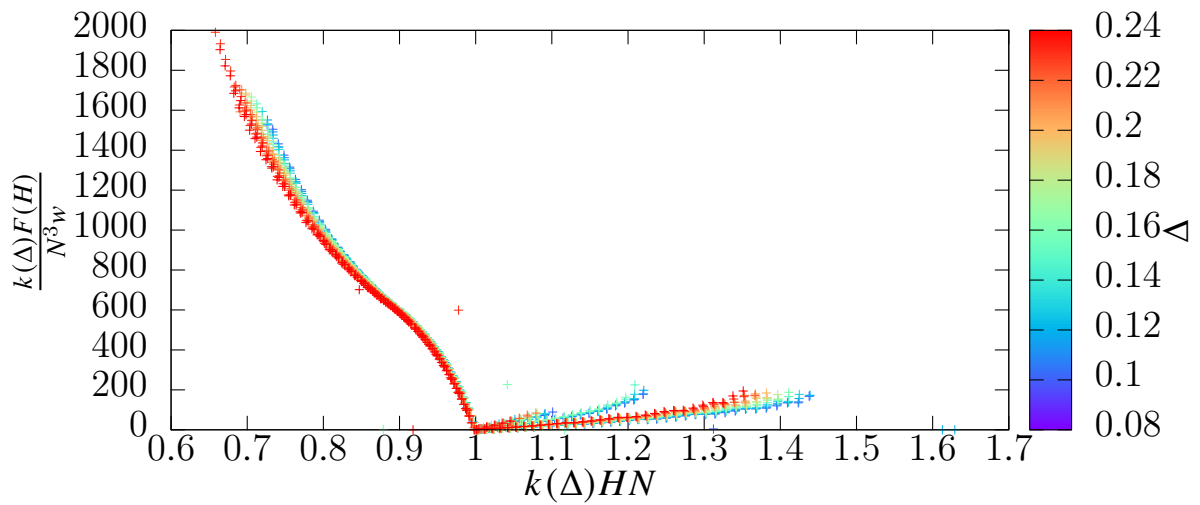
Expérience de Roman – Loi d'échelle pour Δ 

FIGURE 6.15 – Loi d'échelle pour l'expérience de Roman avec Δ . Ici on illustre une superposition des données pour plusieurs Δ en utilisant un facteur d'échelle $k(\Delta)$ adéquat défini par l'équation (6.5). Les couleurs correspondent à la valeur de Δ . La superposition a lieu pour les contacts linéiques.

Dans ce chapitre, montré comment implémenter quelques contraintes pour simuler des scénarii variés. Nous avons validé le modèle sur l'expérience de Roman. Il est alors possible d'utiliser ce modèle pour prédire le comportement d'un ruban inextensible dans des scénarii qui n'ont pas encore été étudié. Dans le prochain chapitre on verra l'importance de l'extensibilité.

Chapitre 7

Discussions

7.1 Contrainte à l'encastrement sur η

Dans la section 6.1.1, nous avons parlé d'une alternative pour réaliser le ruban de Moebius en fixant $\eta(0) = \eta(\ell) = 0$. Ce qui nous intéresse à présent, c'est l'utilité de cette contrainte au seul bord du ruban, lorsqu'il est encastré d'un seul côté, sans contact ni rien de plus. En fait, on pourrait reprendre le premier modèle (modèle chaîné) et bloquer le degré de liberté $m^{(1)}$ à 0. Fixer ainsi $\eta(0) = 0$ revient à imposer une règle orthogonale à la centerline en début de ruban. Lorsque l'on encastre physiquement le ruban entre deux pinces, le ruban est plat à cette frontière. Or le repère matériel ne suffit pas à cela, puisqu'il est ponctuel. Ainsi seule la contrainte sur η peut reproduire cet encastrement physique. Nous testons l'impact de cette contrainte sur un test que nous avons publié dans [Rom+21].

Le test de déversement latéral est initialement prévu pour une plaque élastique. La largeur w est plus grande que pour un ruban, mais l'épaisseur h reste petite devant les autres dimensions. Ainsi on doit s'attendre à quelques erreurs dues à cet aspect. Le ruban va pendre sous son propre poids. La verticale est définie par \mathbf{e}_z . Pour l'état initial (ou «graine») de la procédure de Newton, le ruban est fixé horizontalement à l'origine du repère canonique, sa normale alignée avec \mathbf{e}_y . Si le ruban est trop long, l'équilibre est instable, le ruban va basculer d'un côté sous l'influence de la pesanteur. L'énergie élastique de courbure gagnée va être en effet compensée par l'énergie de pesanteur perdue. Cela est illustré en figure 7.1. Si au contraire le ruban est court, basculer impliquerait un gain d'énergie élastique non compensé par l'énergie de pesanteur perdue. C'est ce qu'on appelle une bifurcation fourche. On définira la distance latérale comme le projeté du vecteur entre les deux bords du ruban ($\mathbf{r}(0)$ et $\mathbf{r}(\ell)$) sur \mathbf{e}_y . Pour illustrer ce propos, le graphe 7.1 affiche la distance latérale en fonction de la longueur $\ell \in [0, 1]$ sur un ruban de caractéristiques : $w = 0.02$, $\rho h = 40$, $D = 1$ et $g = 1$.

Les instabilités dues au déversement latéral sont une des raisons principales d'échec en ingénierie civile. Par exemple les poutres en acier doivent supporter un cisaillement important. L'analyse du déversement latéral remonte au XIX^{ème} siècle [Mic99].

Après l'adimensionnement, il ne reste que deux paramètres : $\frac{w}{\ell}$ et $\Gamma = \frac{\rho h g \ell^3}{D}$ (le coefficient de Poisson ν n'a pas d'influence puisque la courbure naturelle est nulle). Le calcul direct de l'équilibre n'est pas souhaitable. IPOPT calcule un équilibre, mais peut retomber sur le ruban rectiligne sans que cet équilibre soit stable (IPOPT est alors bloqué dans un point de selle). Pour cette raison, même avec un grand Γ , l'équilibre calculé peut être le ruban droit, malgré le fait que le ruban peut basculer. On doit alors suivre une stratégie du type : commencer avec un

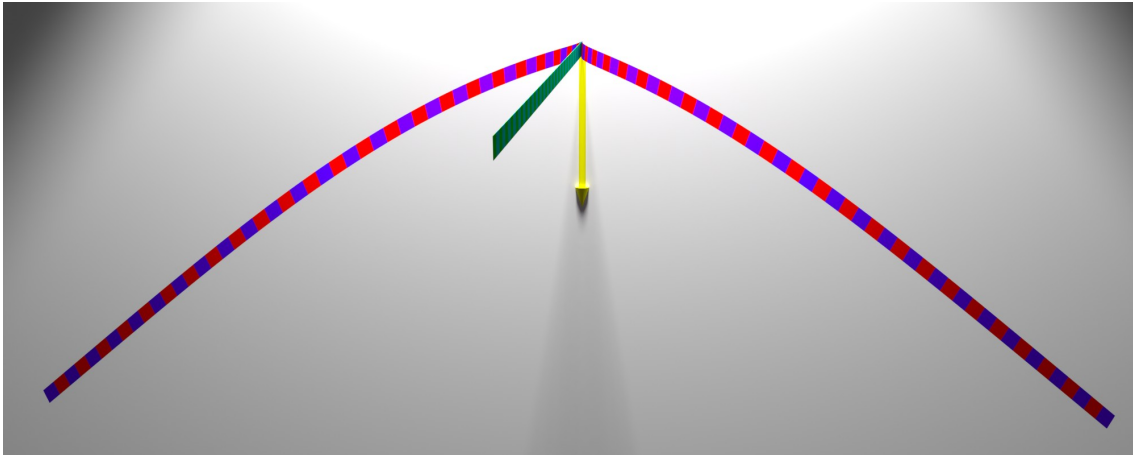


FIGURE 7.1 – Déversement latéral – Ici $\Gamma = 40$, $\frac{w}{\ell} = 0.02$. La gravité est indiquée par la flèche jaune. On observe 3 équilibres : deux en rouge/violet sont symétriques et stables, le troisième en vert est rectiligne et instable.

Déversement latéral

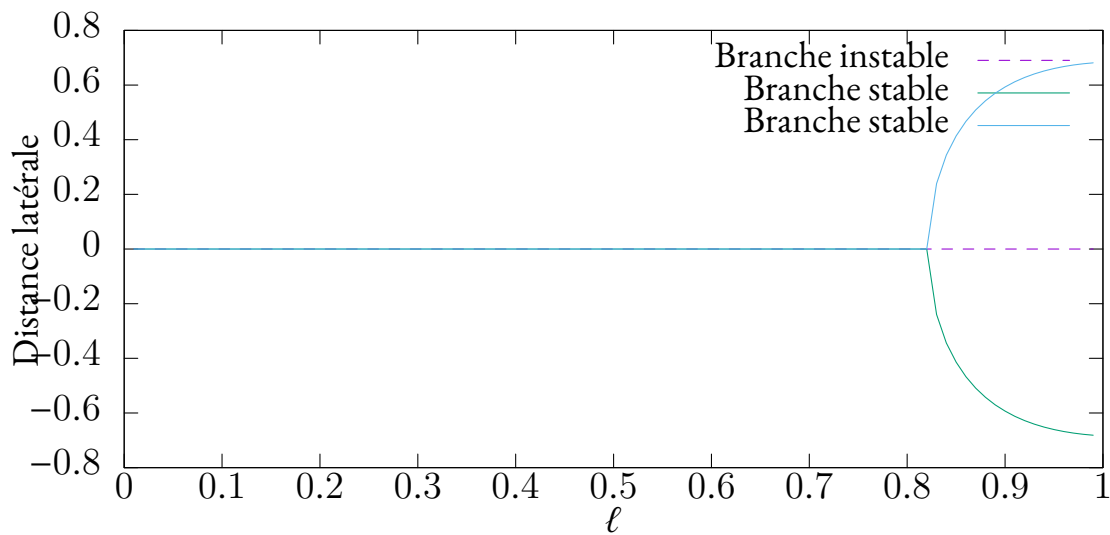


FIGURE 7.2 – Déversement latéral – On reprend le ruban de la figure 7.1. On fait varier la longueur de 0.01 à 0.99. La distance latérale est la coordonnée y du bord du ruban. On obtient une bifurcation fourche. Voir détails dans le texte, section 7.1.

encastrement horizontal, c'est-à-dire que la normale au ruban coïncide avec la verticale. Puis pivoter l'encastrement par paliers de 0.4 radians et recalculer l'équilibre. On évite ainsi l'équilibre instable où le ruban reste horizontal¹.

On va calculer dans le plan $\frac{w}{\ell}, \Gamma_c$ la frontière du déversement (c'est-à-dire le point fourche). La frontière est une fonction $\Gamma_c(\frac{w}{\ell})$. Le point initial $\Gamma_c(\frac{w}{\ell} = 0) = 18.2(1 - \nu)\sqrt{1 + \nu}$ correspond à la limite des tiges élastiques. Il est évalué grâce au modèle de Kirchhoff dans [Mic99]. Pour trouver une dépendance par rapport à $\frac{w}{\ell}$, dans [Rom+21] divers points sont évalués par le modèle coque épaisse éléments finis SR8 d'ABAQUS. Une dépendance linéaire linéaire est trouvée : $\Gamma_c(\frac{w}{\ell}) = 18.2(1 - \nu)\sqrt{1 + \nu} + 14.5\frac{w}{\ell}$. Elle servira alors de référence pour nos simulations.

Les calculs de frontières sont faits dans quatre situations : d'une part l'énergie élastique suit le modèle de Sadowsky ou de Wunderlich, d'autre part on peut contraindre ou non la règle à l'encastrement par $\eta(0) = 0$. On présente les résultats en figure 7.3. Dans aucun cas la frontière Γ_c ne correspond à la théorie. Lorsque l'on utilise Sadowsky sans la contrainte, la frontière est constante à $\Gamma = 21$ [AN21]. L'impact de la contrainte sur le résultat est très surprenant, la frontière plate pour Sadowsky devient dépendante du ratio $\frac{w}{\ell}$, et même si la frontière ne coïncide toujours pas, elle devient parallèle à la courbe théorique. On obtient une situation similaire pour le modèle de Wunderlich avec une différence : Γ_c dépend linéairement de $\frac{w}{\ell}$ même en absence de la contrainte. Ça n'est pas surprenant vu que l'énergie de Wunderlich y est sensible.

Tentons d'expliquer l'échec de notre modèle sur cette expérience. Si l'on considère que pour $\frac{w}{\ell} = 0.1$, on a $w \ll \ell$, on peut considérer que la définition de ruban est applicable². Or les résultats entre notre modèle et la référence ne correspondent pas pour $\frac{w}{\ell} = 0.1$. Nous mettons alors en cause l'hypothèse d'inextensibilité du ruban pour expliquer la différence. En effet, d'une part, nous avons validé notre modèle sur de nombreux exemples et la courbe de référence a été calculée avec un modèle de coques extensible. D'autre part, la nécessité d'extensibilité est expliquée dans [AN21].

7.2 L'expérience de décalage latéral des encastremets

Dans cette section, on présente une expérience reprise de l'article [Hua+20], elle-même reprise de [YH19] dans un cas plus simple. Pour cette expérience comme pour la précédente, on utilise la contrainte $\eta = 0$ aux encastremets. Aussi, au même titre que pour le ruban de Moebius, la gravité est négligée : $\Gamma = 0$.

Le ruban est d'abord encasté aux deux extrémités. L'encastrement au début fixe la position à l'origine du repère canonique et le repère matériel à $(\mathbf{e}_y \ \mathbf{e}_z \ \mathbf{e}_x)$. En fin de ruban, l'encastrement fixe le repère matériel de la même manière et il contraint la position à $(\ell - \Delta\ell \ \Delta W \ 0)^T$. On illustre cela en figure 7.4. On commence par fixer $\Delta\ell$. Nous prenons comme dans l'article [Hua+20] $\frac{\Delta\ell}{\ell} = \frac{1}{2}$. L'expérience consiste à mesurer à l'équilibre la composante z adimensionnée du point au milieu du ruban en fonction de ΔW . Nous définissons cette valeur comme étant l'altitude $H = \left| \frac{1}{\ell} \left\langle \mathbf{r} \left(\frac{\ell}{2} \right), \mathbf{e}_z \right\rangle \right|$. La configuration étant symétrique par rapport au plan $z = 0$, pour chaque équilibre, il y aura un équilibre symétrique correspondant. C'est pour cela qu'on utilise une valeur absolue. On utilisera les paramètres suivants : $\frac{w}{\ell} = \frac{1}{20}$, $p = 100$ segments. Sans perte de généralité et comme précédemment, on fixe $\ell = 1$ et $D = 1$.

1. Cela pourrait expliquer en partie la différence entre la courbe théorique et les données en figure 7.3.

2. Pour affirmer que w est négligeable devant ℓ , on préférerait qu'il y ait au moins deux ordres de grandeur. Ici, il n'y en a qu'un seul.

Déversement latéral

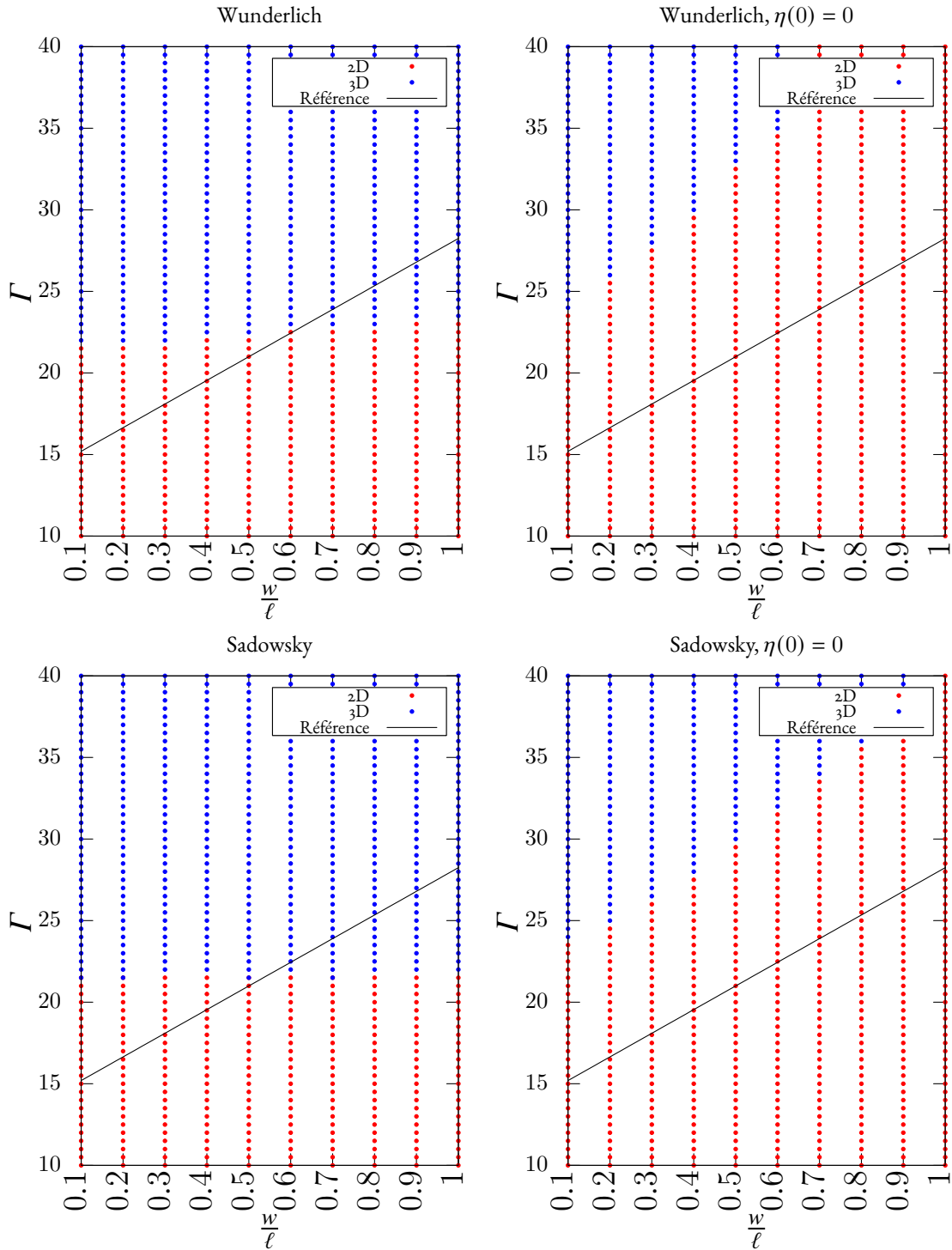


FIGURE 7.3 – Calcul de la nature des équilibres pour le déversement latéral. 2D indique un point où le calcul de l'équilibre du ruban résulte en une configuration rectiligne, 3D indique un point où le calcul de l'équilibre du ruban résulte en une configuration où ce dernier a basculé. La frontière entre les deux zones correspond à Γ_c . Se référer au texte pour les détails de calcul.

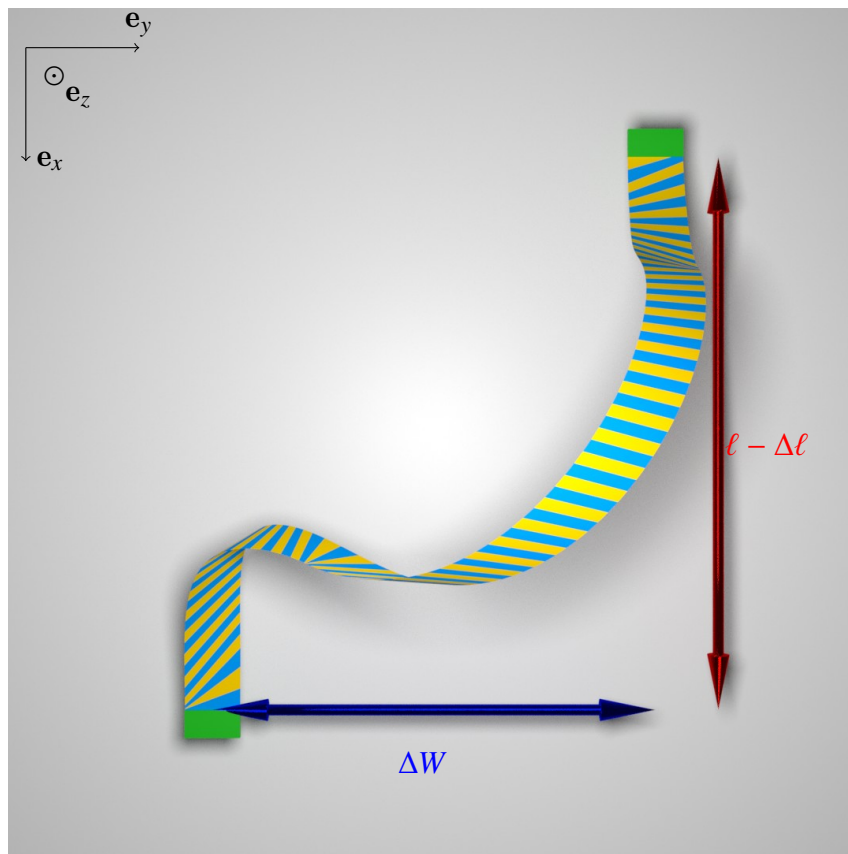


FIGURE 7.4 – L'expérience de décalage latéral des encastrements – Le ruban est encasté aux deux bords dans les pinces vertes. Ces encastrements sont dans le même plan et parallèles, c'est-à-dire pour la position $r_z = 0$, et pour le repère matériel $\mathbf{t}_y = \mathbf{e}_z$, $\mathbf{t}_z = \mathbf{e}_x$. L'expérience ignore la gravité. Les deux paramètres sont alors $\Delta \ell$ et ΔW . $\Delta \ell$ est fixé au début de l'expérience et ΔW est augmenté, puis diminué, ce qui déplace les encastrements latéralement l'un par rapport à l'autre.

Pour notre expérience, on commence par calculer un premier état d'équilibre. Si l'on choisit $\Delta W = 0$ on aura un problème de symétrie dû au fait que dans ce cas $\eta = 0$. On prend alors $\Delta W = 0.01$. Pour obtenir cet état initial on va tourner les encastresments autour de \mathbf{e}_y . On notera les deux angles de rotation correspondants θ_1 et θ_2 . En même temps, on fait varier $\Delta \ell$. L'algorithme 4 détaille les étapes pour calculer ce premier équilibre. Rappelons que la graine passée au solveur correspond à la déformation actuelle du ruban. Ainsi, on a juste besoin de la spécifier avant le premier appel au solveur (à IPOPT).

Algorithme 4 : Initialisation pour l'expérience de décalage latéral des encastresments.

```

InitialiserRuban( $\ell = 1, w = 0.05, D = 1, \Gamma = 0, p = 100$ );
AjoutContraintes( $\eta(0) = 0, \eta(\ell) = 0$ );
FixerDoubleEncastrement( $\Delta W = 0.01, \Delta \ell = 0.8, \theta_1 = \frac{\pi}{4} \text{ rad}, \theta_2 = -\frac{\pi}{4} \text{ rad}$ );
FixerGraine( $\omega(s) = 10^{-5} + 10^{-4}s, \eta(s) = 10^{-5} + 10^{-4}s$ );
AppelSolveur();
ModifierEncastrement( $\theta_1 = 0$ );
AppelSolveur();
ModifierEncastrement( $\theta_2 = -\frac{\pi}{20}$ );
AppelSolveur();
ModifierEncastrement( $\theta_2 = 0$ );
AppelSolveur();
pour  $\Delta L$  dans  $\{0.7, 0.6, 0.5\}$  faire
    ModifierEncastrement( $\Delta \ell \leftarrow \Delta L$ );
    AppelSolveur();

```

Pour calculer la courbe qui nous sert de référence, les auteurs de [Hua+20] utilisent un modèle de plaques nodal. La courbe $H(\Delta W)$ se scinde en trois parties nommées U , US et S en référence à la forme du ruban dans ces zones. La frontière de ces zones se caractérise par un changement de pente, c'est-à-dire un point non différentiable. Dans la première partie U , le ruban forme une arche. En son milieu, la normale reste \mathbf{e}_y . Ensuite dans la partie US , le ruban bascule, l'arche s'effondre. Finalement en partie S le ruban reste assez proche du plan $z = 0$ et a une forme en "S".

Dans nos expériences, en observant les courbures, pour la partie "U", on constate que le graphe des courbures est symétrique ($\kappa(s) = \kappa(\ell - s)$). Puis, quand ΔW augmente, on transite vers des solutions antisymétriques ($\kappa(s) = -\kappa(\ell - s)$). Sans atteindre une antisymétrie parfaite, on s'en rapproche pour $\Delta W > 0.6$. On illustre cette transformation sur la figure 7.5.

Dès nos premiers tests, on rencontre des difficultés majeures, en particulier une hystérèse apparaît. C'est-à-dire que les résultats obtenus sont différents suivants que l'on fasse croître ou diminuer ΔW . On retrouve le graphe en figure 7.6. Pour calculer ce graphe, on fait varier ΔW par pas de 0.005 entre 0.01 et 0.6 dans le sens croissant puis décroissant. On a utilisé l'énergie de Wunderlich avec la contrainte $\eta(0) = \eta(\ell) = 0$. La bascule entre U et US survient pour ΔW plus élevé dans le sens ΔW croissant, dans le sens décroissant elle correspond bien à la référence [Hua+20]. La frontière entre US et S ne correspond jamais à la référence.

Pour faire coller nos résultats au mieux à la référence, nous avons testé plusieurs pistes. Tout d'abord ajouter de la gravité au problème. Ensuite nous voyons des événements de Wunderlich apparaître. Augmenter le nombre de segments pour mieux les capturer serait peut-être une solution. Nous testons alors séparément de monter le nombre de segments à $p = 500$ et de fixer $\Gamma = 1$ ou $\Gamma = 10$. Ces solutions ne fonctionnent pas, c'est-à-dire que le graphe obtenu change,

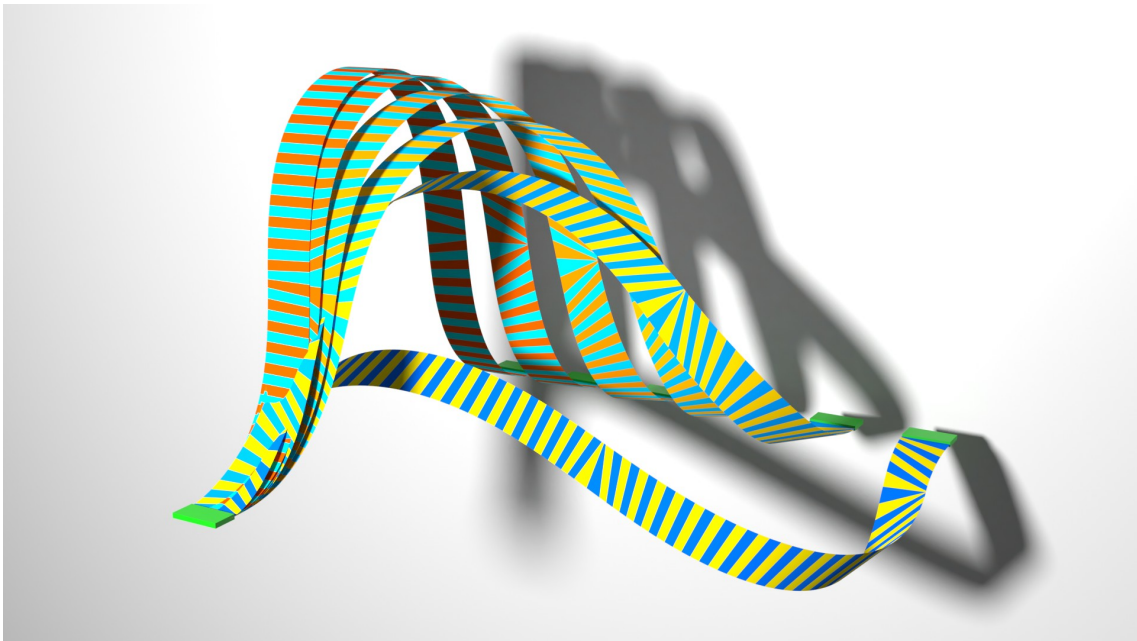


FIGURE 7.5 – Instantanés de l'expérience de décalage latéral des encastresments – Les encastresments sont en vert et les segments des rubans sont visibles via une alternance de couleurs. Les rubans représentés correspondent à $\Delta W \in \{0.01, 0.1, 0.2, 0.3, 0.4, 0.5\}$. On voit le ruban basculer à $\Delta W = 0.4$. Notre modèle a été utilisé avec l'énergie de Wunderlich pour capturer ses équilibres.

Expérience de décalage latéral des encastrements

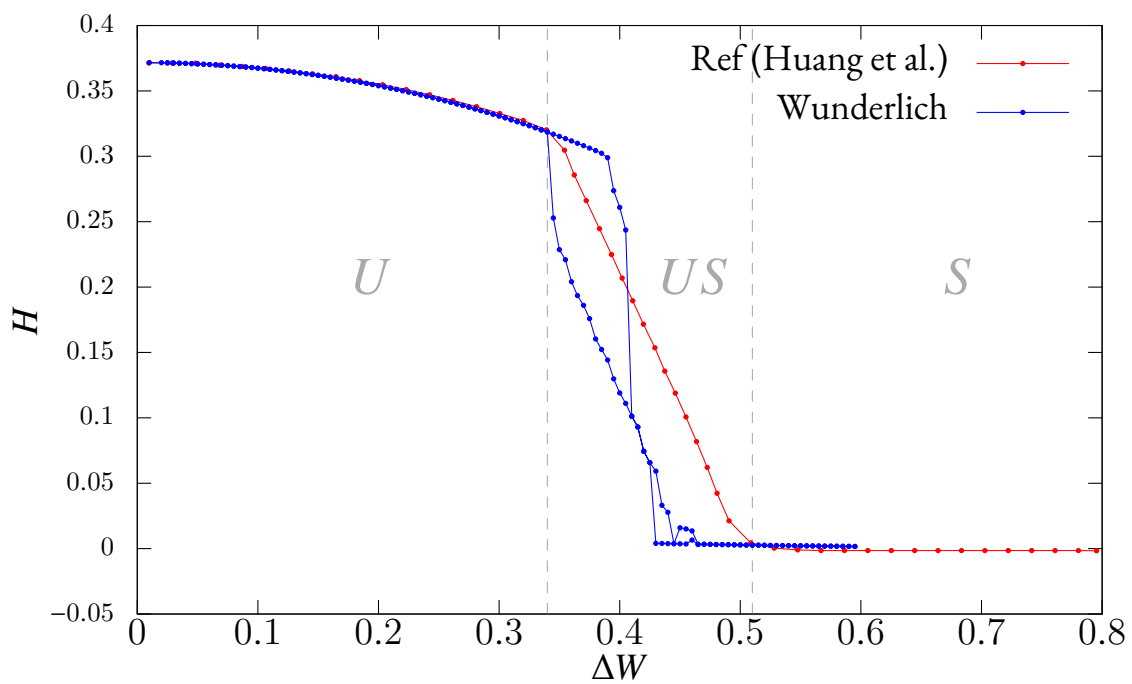


FIGURE 7.6 – Résultat de l'expérience de décalage latéral des encastrements – Le graphique est généré en augmentant, puis en diminuant ΔW , on repère alors une hystérese, le modèle ne parvient pas à capturer le comportement de référence. On indique en gris les trois parties U , US et S calculées par rapport à la référence.

mais ne se rapproche pas significativement de la courbe de référence. S'il y a une hystérèse, alors à ΔW fixé, l'équilibre n'est pas unique (en excluant en plus la symétrie d'une solution par le plan $z = 0$). Peut-être qu'il faut alors mieux suivre les équilibres, en diminuant les incréments de ΔW . Là non plus, nous n'observons rien de concluant.

Vu les échecs pour améliorer les résultats, nous pensons à une limitation du modèle. En effet, pour la courbe de référence les simulations ont été faites avec un modèle extensible. Il est donc raisonnable de penser, vu que notre modèle ne gère pas l'extensibilité que la différence vient de là.

7.3 L'extensibilité du ruban

Dans les deux dernières sections, nous avons réalisé deux expériences numériques. Les résultats obtenus ne collent pas aux courbes de références. Pour expliquer cet échec, nous avons suspecté l'inextensibilité intrinsèque du modèle. Nous allons étudier ici l'importance de l'(in)extensibilité en nous basant sur FENICS-SHELL [Hal+18]. Rappelons que nous avons comparé notre modèle au modèle de coques de Naghdi en section 4.3. Les résultats correspondent quand l'épaisseur de la coque tend vers zéro. Aussi, le modèle de Naghdi couplé à l'utilisation de FENICS-SHELL est validé sur l'expérience de déversement latéral, comme montré dans [Rom+21].

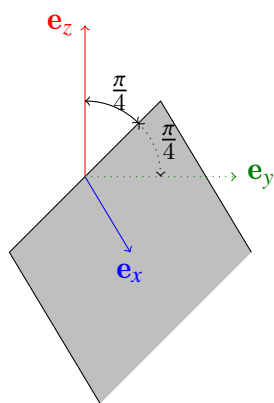


FIGURE 7.7 – Schéma de l'encastrement du ruban

L'expérience que l'on mène est simple. On fixe la verticale à $\mathbf{e}_g = \mathbf{e}_z$, le ruban est encasté d'un côté dans le plan $x = 0$ avec un angle de $\frac{\pi}{4}$ rad par rapport à \mathbf{e}_z et sa tangente à l'origine est \mathbf{e}_x (voir figure 7.7). On fixe $\nu = 0.4$, $h = 10^{-3}$, $D = 1$ et $\ell = 1$. On va faire varier w dans l'ensemble $\{0.03, 0.05, 0.1, 0.2, 0.3, 0.5\}$ et Γ dans $\{10, 100, 1000\}$. FENICS-SHELL a besoin d'un maillage pour représenter le ruban. On choisit une grille rectangulaire régulière dont les éléments sont des triangles rectangles de largeur 0.01. Pour l'encastrement on fixe les triangles en coin de ruban (et non toute la rangée). Il n'est pas possible de résoudre le problème en une fois dans FENICS-SHELL. À la place, il faut progressivement augmenter Γ , on choisit une augmentation linéaire d'unité en unité.

Les équilibres calculés par FENICS-SHELL sont présentés en figure 7.8. L'effet de la gravité est bien visible : le ruban se courbe plus si Γ est grand. Le fait notable est la tendance du ruban à se bomber à l'encastrement. Il n'y a rien de faux puisque l'encastrement ne se fait qu'aux deux coins et non sur tout son bord. C'est tout de même un premier indice de la présence d'extension (on ne parle pas ici de possibilité d'extension due au modèle, mais bien de sa présence dans la configuration à l'équilibre).

Nous quantifions la présence de non-isométrie entre la configuration déformée et la configuration à plat par les énergies d'extension et de cisaillement. Ces deux énergies ensemble constituent l'énergie de membrane. (Elles étaient nulles pour notre modèle de ruban puisque celui-ci impose l'inextensibilité.) On va d'abord présenter la répartition des différentes énergies. En figure 7.9 on présente six graphes, deux pour chaque valeur de Γ . On présente d'un côté la part de chaque type d'énergie, c'est-à-dire le ratio $\frac{e}{E_{tot}}$ où e est l'énergie considérée et E_{tot} l'énergie totale, somme de l'énergie de pesanteur et de l'énergie élastique, et de l'autre côté on refait la même chose mais seulement sur l'énergie élastique. On remarque une influence modérée de la largeur du ruban sur la répartition. Plus Γ est élevé, plus la part de l'énergie de pesanteur est

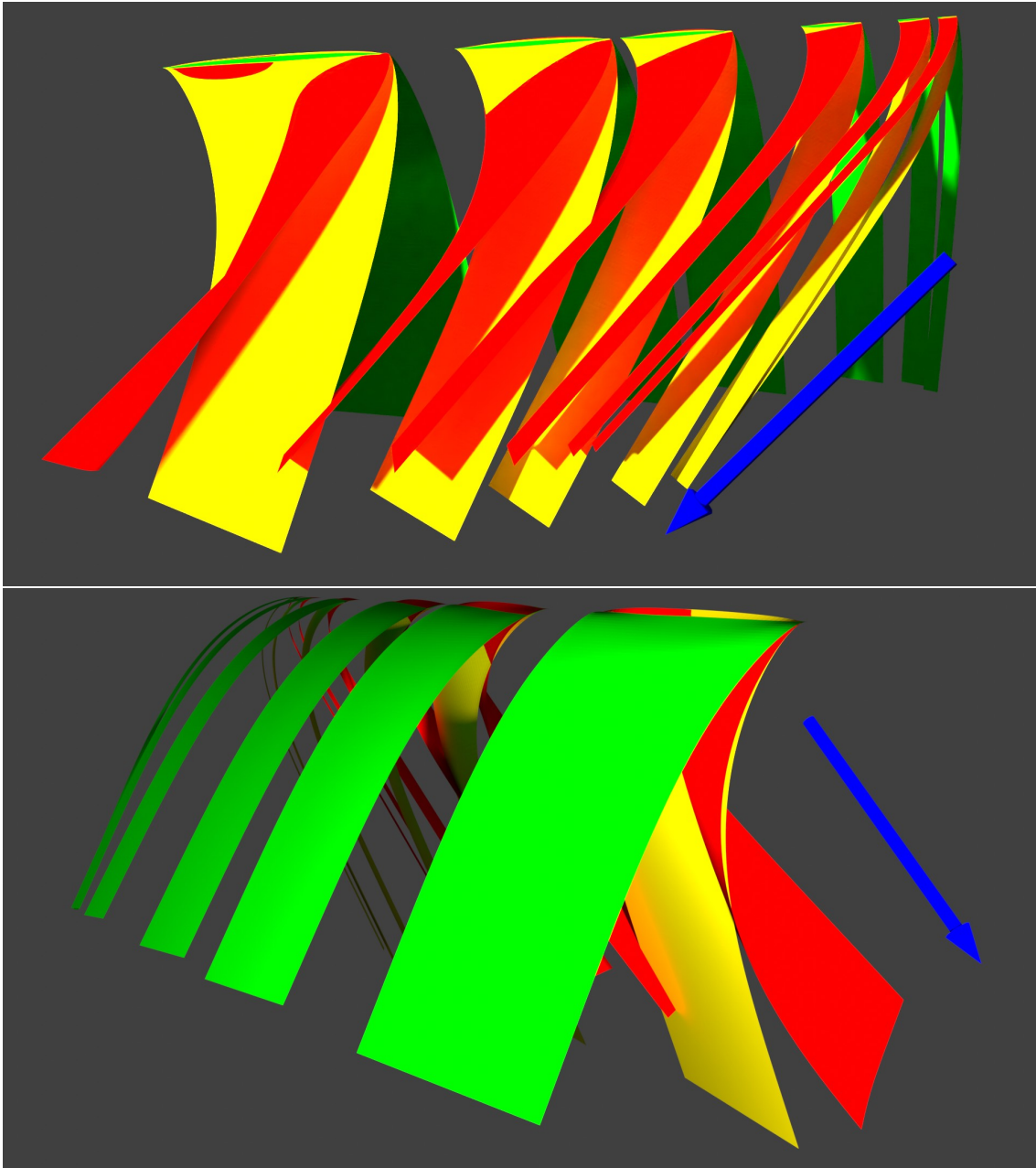


FIGURE 7.8 – Deux points de vue des équilibres calculés par FENICS-SHELL pour l'expérience de la section 7.3. En vert l'équilibre est calculé pour $\Gamma = 10$, en jaune pour $\Gamma = 100$ et en rouge pour $\Gamma = 1000$. Les encastresments des rubans de même largeur sont superposés. La gravité est indiquée par la flèche bleue.

élevée, ce qui n'a rien de surprenant. La partie intéressante est la présence d'extensibilité. On ne considère que l'énergie élastique pour cela. La part de l'énergie de membrane n'est jamais négligeable, toujours supérieure à 1% suivant les expériences, et même dans les cas où la largeur est faible $\frac{w}{l} \leq \frac{1}{10}$ et la pesanteur pas trop forte $\Gamma \leq 100$, on dépasse les 7%. En comparaison sur l'expérience de la poutre console, dans [Rom+21], l'énergie de membrane est à $10^{-5}\%$ pour $\Gamma = 10$, 0.05% pour $\Gamma = 100$ et 1% pour $\Gamma = 1000$. Cela montre bien que la déflexion joue un rôle pour le besoin d'extensibilité.

Pour avoir un meilleur aperçu de l'impact de ces énergies d'extension et de cisaillement, nous regardons les endroits sur le ruban où chaque type d'énergie prédomine. Nous traçons cette répartition en figure 7.10 : pour chaque point on indique la part d'énergie de flexion, de cisaillement et d'extension. Attention cette répartition est donc locale. Pour $\Gamma = 10$ ou $\Gamma = 100$, l'extension est localisée à l'encastrement, sur l'un des bords et en fin de ruban³. Pour $\Gamma = 1000$ de l'extension va apparaître sur une bonne partie du ruban, la zone est d'autant plus grande que le ruban est étroit.

Cette expérience montre bien une présence d'extensibilité non négligeable dans certains cas. Ainsi, avant d'utiliser notre modèle de ruban inextensible, il faut bien prendre en compte cet aspect. Pour ajouter de l'extensibilité à notre modèle sans le changer totalement, on peut s'inspirer de [AN21] qui propose de rajouter un peu d'extensibilité aux rubans.

3. Attention sur la toute fin du ruban, l'énergie peut être localement nulle. Pour éviter cela on rajoute un epsilon à l'énergie élastique, il peut donc y avoir des artefacts à cet endroit. Cependant, on voit clairement se dessiner une zone pour par exemple $\Gamma = 100$ quelque soit le ratio $\frac{w}{l}$.

Répartition de l'énergie

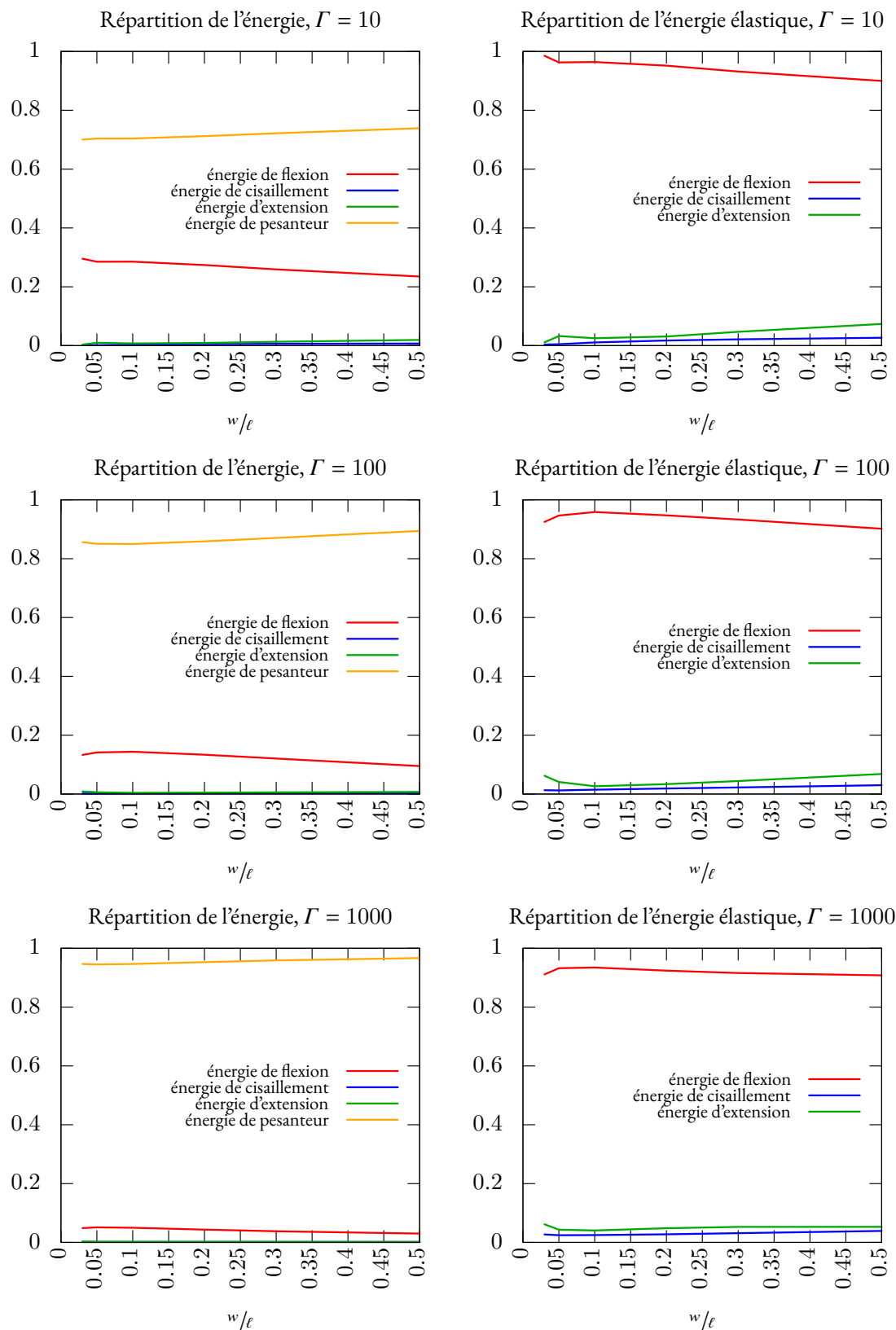


FIGURE 7.9 – Répartition des énergies du ruban selon la largeur du ruban w et Γ pour l'expérience décrite en 7.3 (la somme des courbes de chaque graphe vaut donc 1). Dans l'énergie élastique, on distingue l'énergie de flexion, de cisaillement et d'extension. Si l'hypothèse d'inextensibilité est vérifiée l'énergie de flexion devrait frôler les 100% de l'énergie élastique. Cela n'est pas le cas, même à faible largeur $\frac{w}{\ell} < 0.1$.

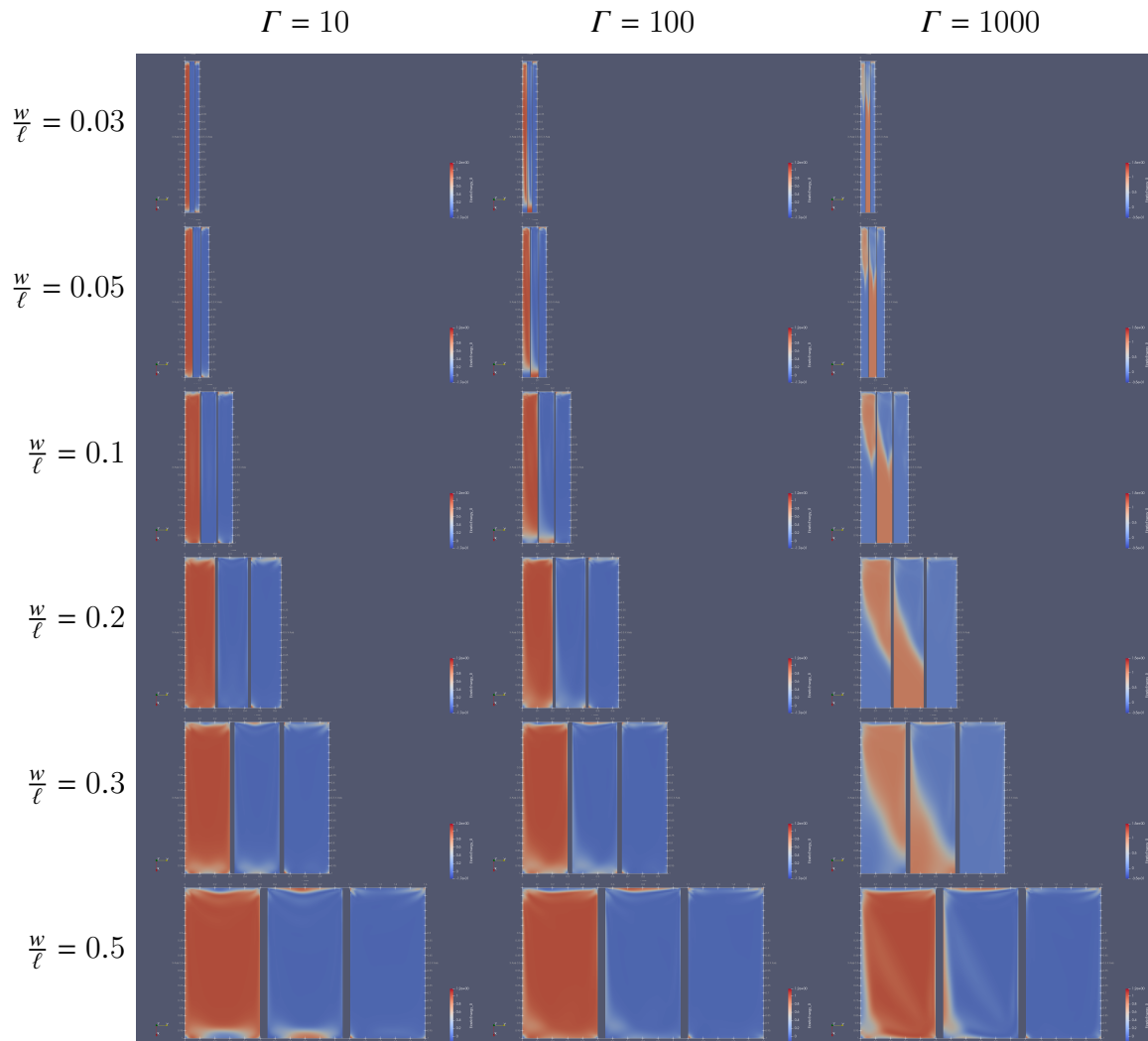


FIGURE 7.10 – Répartition locale de l'énergie élastique entre les énergies de flexion, de cisaillement et d'extension pour l'expérience décrite en 7.3. Pour Γ et $\frac{w}{\ell}$ donnés, on trace trois fois le ruban à l'équilibre aplati. Sur le premier (à gauche), on représente en chaque point le rapport entre l'énergie de flexion et l'énergie élastique totale, sur le second on fait de même pour l'énergie d'extension et sur le troisième avec l'énergie de cisaillement. La couleur rouge indique une valeur de 1, bleue de 0. L'encastrement est en haut sur les figures. Pour éviter les erreurs numériques de division par zéro, un epsilon (10^{-8}) est ajouté à l'énergie élastique. Pour cette raison, en fin de ruban des valeurs artefacts peuvent apparaître. **Attention** : étant une répartition locale, deux points d'une même couleur n'auront pas la même énergie.

Conclusion et perspectives

Tout au long de ce travail, nous avons établi un modèle de ruban rectangulaire inextensible. Par définition un ruban doit respecter les caractéristiques géométriques $h \ll w \ll \ell$. L'inextensibilité rend la surface du ruban développable. Ainsi, l'équation de Darboux et les propriétés des surfaces développables permettent de décrire l'évolution de la centerline et du repère matériel en fonction de deux scalaires le long du ruban : la courbure normale ω et la direction de la règle η . En spécifiant un point d'attache et une orientation d'attache, on obtient un système de Cauchy. Nous discrétisons alors ω et η par des fonctions linéaires par morceaux. Ce choix ne nous permet pas de résoudre le système de Cauchy directement, il faut passer par une technique d'intégration numérique. Nous avons choisi le développement de fonctions en séries entières qui permet alors de résoudre le problème. Le modèle constitué se base ainsi sur des éléments en courbure et torsion polynomiales, de degré 1 et 2 respectivement.

Nous avons validé l'implémentation de notre modèle grâce à la comparaison de celui-ci à deux implémentations des équations fortes du ruban par méthodes de tir (via MATHEMATICA et AUTO). Le ruban étant une structure à mi-chemin entre les coques et les tiges, nous comparons donc notre modèle avec des modèles de coques et de tiges. Grâce à une comparaison au modèle de tiges des super-clothoïdes nous validons le besoin d'un modèle spécifique aux rubans, car les deux modèles ne produisent pas les mêmes équilibres. Nous avons même essayé un modèle de tige incorporant du cisaillement, mais cela n'est toujours pas suffisant. La comparaison au modèle de coques de Naghdi est aussi intéressante. Tout d'abord notre modèle est un modèle limite où l'épaisseur est négligeable. Au contraire, pour le modèle de coques, nous devons nous approcher de la limite $h \rightarrow 0$, pour que les résultats concordent. De plus, notre modèle ne nécessite que quelques secondes pour calculer l'équilibre quand le code de coques en nécessite environ mille fois plus. Cela montre l'un des intérêts de notre modèle.

En enrichissant le modèle, nous avons ouvert la voie à de nouvelles possibilités : encastrement du ruban des deux côtés, rubans circulaires ou de Moebius, collision entre ruban et plans. Nous réussissons avec succès à simuler le ruban de Moebius et à reproduire l'expérience de Roman. Malheureusement dans le chapitre 7, nous avons vu deux expériences où notre modèle échoue à reproduire des courbes théoriques connues calculées pour un modèle de plaques élastiques. La différence pourrait s'expliquer par l'hypothèse d'inextensibilité de notre modèle de ruban. De ce fait, nous avons étudié la présence d'extensibilité avec FENICS-SHELL. Nous constatons qu'elle n'est pas anecdotique, même lorsque la largeur est faible $\frac{w}{\ell} = 0.03$. On peut aussi dans cette idée citer [AN21] où les auteurs ajoutent la possibilité d'extension même si la centerline reste inextensible.

En conclusion, notre modèle démontre une très bonne efficacité : il est rapide, il calcule des équilibres précis, et il permet de simuler des scénarii très variés. Cependant, c'est un modèle limite où l'on a fait une forte hypothèse d'inextensibilité. Il faudra alors relâcher cette hypothèse dans le futur. On peut commencer par s'inspirer de [AN21] pour changer l'énergie élastique.

Même si la surface n'est plus développable, la forme de la centerline restera correcte. Il faudra juste adapter le calcul de la surface à partir de celle-ci.

Un autre point d'amélioration concerne la gestion du contact, en partant de ce qui a été fait au chapitre 6. Il serait intéressant d'essayer de passer vers un modèle pseudo-dynamique pour mieux gérer le contact, cela permettrait d'ajouter une contrainte de contact quand il y en a besoin, à un endroit arbitraire de la surface du ruban. Ainsi on pourrait traiter l'auto-contact et simuler un plectonème de rubans et étendre ainsi tous les résultats établis sur les plectonèmes des tiges au cas des rubans.

Annexe A

Sur les multiplicateurs de Lagrange

Ici on répond à une question commune : si une contrainte est une distance pourquoi le multiplicateur de Lagrange correspond-il à la force ?

Prenons un système quelconque contenant un point x . Ce système est caractérisé par une énergie E , mais on définit une contrainte $u = x - p$ où p est un point fixe externe. La condition de Karush-Kuhn-Tucker à l'ordre 1 donne $\nabla E + \lambda \nabla u = 0$. Si le point fait partie des degrés de liberté, on aura $\nabla_x E + \lambda \nabla_x u = 0$. $\nabla_x E$ est la force du système en x due à l'énergie. À la contrainte u est aussi associée une force. Étant à l'équilibre $\lambda \nabla_x u$ peut alors être vu comme la dérivée du travail de cette force. Ainsi λ correspond à la force de la contrainte.

Que ce passe-t-il si x ne fait pas partie des degrés de liberté ? λ est en fait invariant si l'on change la paramétrisation du système. C'est ce qu'on va prouver dans les sections suivantes. Ainsi, en appliquant cette propriété, il suffit de dire qu'il existe une paramétrisation du système où x fait parti des degrés de liberté pour affirmer que λ est bien une force.

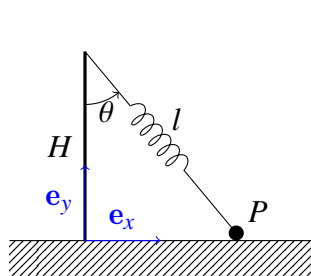


FIGURE A.1 – Schéma de l'exemple

Prenons un exemple concret illustré en figure A.1 : Un mat de hauteur H est planté dans le sol. En haut du mat, au point T est attaché un ressort de longueur l et longueur à vide l_0 et raideur k par un pivot. Il forme alors un angle θ avec le pivot. On attache à son autre côté P , une masse m . La masse va donc tomber au sol. Cet exemple est volontairement très simple. On peut le paramétriser de deux manières. Soit par la position de P . On obtient alors la contrainte (C_y) de non-pénétration dans le sol $P_y = 0$ avec le multiplicateur de Lagrange associé λ_y . À l'équilibre, la composante normale de la force du sol sur la masse sera alors λ_y . Le Lagrangien est calculé par $L = mgP_y + \frac{1}{2}k(l - l_0)^2 + \lambda_y P_y$. On calcule alors la

dérivée du lagrangien selon P_x et P_y : $\frac{dL}{dP_x} = k \frac{P_x}{l} (l - l_0)$, $\frac{dL}{dP_y} = mg + k \frac{H - P_y}{l} (l - l_0) + \lambda_y$. Comme on résout pour un gradient valant 0, nous obtenons donc $l = l_0$, puis $\lambda_y = -mg$. Ce résultat est bien cohérent avec le théorème des actions réciproques. Alternativement on peut choisir d'utiliser les paramètres l et θ , on obtiendra alors la contrainte de non-pénétration dans le sol (C_θ) : $l \cos(\theta) - H = 0$ le multiplicateur de Lagrange associé λ_θ . On calcule le lagrangien $L = mg(H - l \cos(\theta)) + \frac{1}{2}k(l - l_0)^2 + \lambda_\theta(l \cos(\theta) - H)$. En dérivant suivant θ on obtient $\frac{dL}{d\theta} = mgl \sin(\theta) + \lambda_\theta l \sin(\theta)$. En résolvant pour un gradient nul, on obtient $\lambda_\theta = -mg$ comme pour l'autre paramétrisation.

Problème version bijective : Soit un système ayant un ensemble de contraintes C . Ce système est paramétré de deux manières différentes. À l'équilibre, le lagrangien calculé pour les

contraintes C est le même pour les deux paramétrisations.

Problème version injective : Soit un système \mathcal{S}_1 ayant un ensemble de contraintes C . Soit un système \mathcal{S}_2 plus général que le premier système. Supposons que ce système soit doté d'un ensemble de contraintes K tel que si ces dernières sont respectées alors les configurations de \mathcal{S}_2 sont les mêmes que les configurations de \mathcal{S}_1 . Alors peu importe la paramétrisation des deux systèmes, lorsque l'équilibre est atteint le langrangien des contraintes C est le même pour les deux systèmes.

Pour illustrer le cas injectif, on peut prendre l'exemple des modèles de ruban. Le modèle chaîné correspond au système \mathcal{S}_1 , et le modèle mixte au système \mathcal{S}_2 . En effet, le modèle mixte est plus général car les degrés de liberté permettent de contrôler les segments indépendamment les uns des autres. L'ensemble K correspond alors aux contraintes connectant les différents segments et le premier segment à l'encastrement.

A.1 Cas bijectif

Définissons le problème mathématique. Soit \mathcal{E}_1 et \mathcal{E}_2 deux ouverts d'un espace vectoriel sur \mathbb{R} , $\phi : \mathcal{E}_1 \mapsto \mathcal{E}_2$ une fonction bijective, $o_1 : \mathcal{E}_1 \mapsto \mathbb{R}$ et $C_1 : \mathcal{E}_1 \mapsto \mathbb{R}^p$ dont la jacobienne est injective. On pose $o_2 = o_1 \circ \phi^{-1}$ ($o_2 : \mathcal{E}_2 \mapsto \mathbb{R}$) et $C_2 = C_1 \circ \phi^{-1}$ ($C_2 : \mathcal{E}_2 \mapsto \mathbb{R}^p$). On définit les deux problèmes suivants :

$$(\mathbf{P}_1) : \begin{array}{l} \min_{x \in \mathcal{E}_1} o_1(x) \\ \text{tel que } C_1(x) = 0 \end{array} \quad \text{et} \quad (\mathbf{P}_2) : \begin{array}{l} \min_{x \in \mathcal{E}_2} o_2(x) \\ \text{tel que } C_2(x) = 0 \end{array} .$$

Pour interpréter les choses, le choix des degrés de liberté ou paramètres du modèle va donner l'ensemble \mathcal{E}_1 . Un point de cet espace est une affectation des valeurs aux paramètres. Mais un choix alternatif mène à l'ensemble \mathcal{E}_2 . Dans ce cas là, il est facile de passer d'un choix à l'autre en utilisant la fonction ϕ . C_1, C_2 sont les vecteurs contenant les contraintes. Nous expliquons dans la section 5.1 pourquoi l'hypothèse de l'injectivité du jacobien de C_1 est importante. Si ce n'est pas le cas au moins une contrainte est redondante avec d'autres. Lors de la minimisation du problème (\mathbf{P}_1) le multiplicateur de Lagrange ne sera alors pas défini.

Trivialement, on a le résultat :

$$\forall x \in \mathcal{E}_1, x \text{ solution de } (\mathbf{P}_1) \Leftrightarrow \phi(x) \text{ solution de } (\mathbf{P}_2)$$

Pour démontrer ce résultat, prenons $x^* \in \mathcal{E}_1$ solution de (\mathbf{P}_1) . On a $C_2(\phi(x^*)) = C_1 \circ \phi^{-1}(\phi(x^*)) = C_1(x^*)$. Supposons qu'on ait $y \in \mathcal{E}_2$ solution de (\mathbf{P}_2) avec $o_2(y) < o_2(\phi(x^*))$ alors, $o_1 \circ \phi^{-1}(y) = o_2(y) < o_2(\phi(x^*)) = o_1 \circ \phi^{-1}(\phi(x^*)) = o_1(x^*)$, autrement dit x^* ne peut pas être solution de (\mathbf{P}_1) . Donc $\phi(x^*)$ est bien solution de (\mathbf{P}_2) . Par symétrie et bijectivité de ϕ l'implication inverse est vraie aussi.

Passons maintenant au résultat principal. Soit x^* solution de (\mathbf{P}_1) . Soit $\lambda \in \mathbb{R}^p$, on montre :

λ multiplicateur de Lagrange associé à x^* pour $(\mathbf{P}_1) \Leftrightarrow \lambda$ multiplicateur de Lagrange associé à x^* pour (\mathbf{P}_2) .

Démonstration : Soit λ_1 le multiplicateur de Lagrange associé à x^* pour (\mathbf{P}_1) et λ_2 celui pour (\mathbf{P}_2) . La condition d'optimalité du premier ordre sur le Lagrangien (KKT) pour (\mathbf{P}_1) s'écrit $\nabla_x \mathcal{L}_1(x^*, \lambda_1) = \nabla_x o_1(x^*) + \mathcal{J}_x C_1^T(x^*) \lambda_1 = 0$, où l'opérateur $\mathcal{J} \cdot$ note le jacobien. De même,

$\nabla_x \mathcal{L}_2(\phi(x^*), \lambda_2) = \nabla_y o_2(\phi(x^*)) + \mathcal{J}_y C_2^\top(\phi(x^*)) \lambda_2 = 0$. En appliquant les différentes définitions, on obtient $\mathcal{J}_y \phi^{-1}(\phi(x^*)) \nabla_x o_1(\phi^{-1} \circ \phi(x^*)) + \mathcal{J}_y \phi^{-1}(\phi(x^*)) \mathcal{J}_x C_1^\top(\phi^{-1} \phi(x^*)) \lambda_2 = 0$. Notons de $\mathcal{J} \phi$ est inversible car ϕ est bijective. On peut ainsi simplifier l'expression en $\nabla_x o_1(x^*) + \mathcal{J}_x C_1^\top(x^*) \lambda_2 = 0$. Par hypothèse, cette équation en fonction de λ_2 n'a qu'une solution (injectivité de la jacobienne). Ainsi $\lambda_1 = \lambda_2$.

Note : dans la pratique on cherche un minimum locale. Pour traiter ce cas théoriquement, on peut simplement restreindre \mathcal{E}_1 à un ouvert contenant un unique minimum local. \mathcal{E}_2 contiendra le minimum correspondant. En réappliquant ce qui a été dit précédemment, le multiplicateur de Lagrange correspondant à ces minimums est bien le même.

A.2 Cas injectif

Soit $\mathcal{E}_1 \subseteq \mathbb{R}^n$ et $\mathcal{E}_2 \subseteq \mathbb{R}^m$ deux ouverts avec $n < m$, $o_1, o_2 : \mathcal{E}_1 \mapsto \mathbb{R}$, $C_1 : \mathcal{E}_1 \mapsto \mathbb{R}^p$, $C_2 : \mathcal{E}_2 \mapsto \mathbb{R}^q$, $C_2^{(L)} : \mathcal{E}_2 \mapsto \mathbb{R}^d$ avec $C_1, C_2, C_2^{(L)}$ de jacobienne injective. On définit les deux problèmes suivants :

$$(\mathbf{P1}) : \begin{array}{l} \min_{x \in \mathcal{E}_1} o_1(x) \\ \text{tel que } C_1(x) = 0 \end{array} \quad \text{et} \quad (\mathbf{P2}) : \begin{array}{l} \min_{x \in \mathcal{E}_2} o_2(x) \\ \text{tel que } C_2(x) = 0 \\ \text{tel que } C_2^{(L)}(x) = 0 \end{array} .$$

On suppose que les deux problèmes sont reliés par l'existence d'une fonction $f : \mathcal{E}_1 \mapsto \mathcal{E}_2$ telle que

1. f est injective
2. $\forall x \in \mathcal{E}_1 C_2(f(x)) = C_1(x)$ (Les contraintes coïncident)
3. $\forall y \in \mathcal{E}_2 C_2^{(L)}(y) = 0 \Leftrightarrow y \in f(\mathcal{E}_1)$ (La contrainte $C^{(L)}$ ne s'annule que sur l'image de f .)
4. si x est solution de **(P1)**, alors $f(x)$ est solution de **(P2)**.
5. si y est solution de **(P2)**, alors il existe x une solution de **(P1)** avec $y = f(x)$. (x est unique par injectivité de f .)

On a ajouté le vecteur contraintes C_2^L , qui correspond à la restriction de l'espace des solutions pour contrebalancer l'enrichissement de l'espace \mathcal{E}_2 .

Soit $x^* \in \mathcal{E}_1$, si $f(x^*)$ est solution de **(P2)** avec λ_2 le multiplicateur de Lagrange associé à la contrainte C_2 et λ_1 le multiplicateur de Lagrange associé à x^* dans **(P1)**, alors $\lambda_1 = \lambda_2$.

Démonstration : On définit un espace $\mathcal{E}^D \subseteq \mathbb{R}^{m-n}$, puis on définit $\mathcal{E}^A = \mathcal{E}_1 \times \mathcal{E}^D$. On introduit deux fonctions de projections $P^A : \mathcal{E}_1 \rightarrow \mathcal{E}^A$ et $P : \mathcal{E}^A \rightarrow \mathcal{E}_1$. On étend f en une fonction $f_A : \mathcal{E}^A \rightarrow \mathcal{E}_2$ telle que $f_A((x, 0)) = f(x)$ et f_A est bijective¹. On pose alors un nouveau problème «équivalent» aux deux autres dans le domaine \mathcal{E}^A .

$$(\mathbf{PA}) : \begin{array}{l} \min_{x \in \mathcal{E}^A} o_A(x) \\ \text{tel que } C_A(x) = 0 \\ C_A^{(L)}(x) = 0 \end{array} \quad \text{avec} \quad \begin{array}{l} o_A((x, y)) = o_2(f_A^{-1}((x, y))) \\ C_A((x, y)) = C_1(x) \\ C_A^{(L)} = C^{(L)}(f_A^{-1}(x, y)) \end{array} .$$

1. On supposera l'existence d'une telle fonction. Dans l'exemple du ruban f^A est l'identité et f est la restriction de tout les paramètres aux variables $b^{(1)}, m^{(1)}, a^{(i)}, n^{(i)}$ pour $i \in \llbracket 1, p \rrbracket$.

Par définition $C_A^{(L)}$ ne s'annule que si $y = 0$. Aussi $\forall x \in \mathcal{E}_1$ $o_A((x, 0)) = o_1(x)$, d'où $\nabla_x o_A((x, 0)) = \nabla_x o_1(x)$. On dit que **(PA)** est équivalent à **(P1)** car si $x \in \mathcal{E}_1$ est solution de **(P1)**, alors $P^A(x)$ est solution de **(PA)**, et inversement si $z \in \mathcal{E}_1$ est solution de **(PA)**, alors $P(z)$ est solution de **(P1)**. De même on dit que **(PA)** est équivalent à **(P2)** car pour tout $z \in \mathcal{E}^A$, z est solution de **(PA)** si et seulement si $f_A(z)$ est solution de **(P2)**.

Soit $x^* \in \mathcal{E}_1$, avec $f(x^*)$ solution de **(P2)**. Notons λ_1 le multiplicateur de Lagrange au point x^* pour le problème **(P1)**, λ_2 le multiplicateur de Lagrange associé au vecteur contrainte C_2 au point $f(x^*)$ pour le problème **(P2)**, $\lambda_2^{(L)}$ le multiplicateur de Lagrange associé au vecteur contrainte $C_2^{(L)}$ au point $f(x^*)$ pour le problème **(P2)** et de même on définit $\lambda_A, \lambda_A^{(L)}$ pour **(PA)**.

Le cas bijectif traité précédemment permet d'affirmer que $\lambda_A = \lambda_2$ et $\lambda_A^{(L)} = \lambda_2^{(L)}$.

On note $y^* = P^A(x^*)$. Le lagrangien de **(PA)** est $\mathcal{L}(y, \lambda, \lambda^{(L)}) = o_A(y) + \lambda C_A(y) + \lambda^{(L)} C_A^{(L)}(y)$. Son gradient s'annule en y^* (condition de KKT). On va projeter le gradient sur \mathcal{E}_1 .

$P(\nabla_y \mathcal{L}(y^*, \lambda_A, \lambda_A^{(L)})) = P(\nabla_y o_A(y^*)) + P(\mathcal{J}_y C_A(y^*)) \lambda_A + P(\mathcal{J}_y C_A^{(L)}(y^*)) \lambda_A^{(L)} = \nabla_x o_1(x^*) + \mathcal{J} C_1^T(x^*) \lambda_A + 0 \lambda_A^{(L)}$. On retrouve $0 = \nabla_x o_1(x^*) + \mathcal{J} C_1^T(x^*) \lambda_A$. Comme dans le cas bijectif, on retrouve $\lambda_A = \lambda_1$.

Table des figures

1	Illustration des rubans dans notre environnement	7
1.1	Modèle nodal vs. modèle en courbure	14
1.2	Les trois types d'énergie élastique d'une coque ou plaque.	16
1.3	Surfaces réglées et développables.	17
2.1	Le ruban de MOEBIUS	19
2.2	Schéma d'un ruban	20
2.3	Paramétrisation du ruban	21
2.4	Paramétrisation du ruban aplati	22
2.5	Point singulier sur un ruban aplati	22
2.6	Points singuliers	24
3.1	Un segment sur la configuration aplatie	30
3.2	Courbure et continuité	30
3.3	Division d'un ruban en segments et sous-segments	33
3.4	Fonction H : erreur numérique et développement limité	34
3.5	Cas tests pour mesurer l'efficacité des algorithmes de calcul d'équilibre	39
4.1	Comparaison de notre modèle au modèle de tir	43
4.2	Comparaison de notre modèle au modèle de Naghdi avec FENICS-SHELL	44
4.3	Temps de convergence pour FENICS-SHELL selon l'épaisseur	45
4.4	Comparaison de notre modèle au modèle des tiges	47
4.5	Comparaison des modèles de tiges d'Antman, Kirchhoff et du modèle de ruban avec l'énergie de Sadowsky – Graphique	48
4.6	Comparaison des modèles de tiges d'Antman, Kirchhoff et du modèle de ruban avec l'énergie de Sadowsky – Une configuration d'équilibre en 3D	49
4.7	Expérience de la poutre console – Graphique	50
4.8	Expérience de la poutre console – Comparaison expérimentale	50
5.1	Une expérience simple de double encastrement	54
5.2	Efficacité du modèle mixte par rapport au modèle chaîné	60
5.3	Illustration de l'équilibre du ruban décrit dans le « Scénario encasté libre »	60
5.4	Efficacité du modèle mixte et chaîné dans un scénario bi-encasté	61
5.5	Illustration de l'équilibre du ruban décrit dans le « Scénario bi-encasté »	61
5.6	Cinq scenarii pour tester l'encastement	63
6.1	Schéma d'un ruban circulaire et d'un ruban de Moebius	66
6.2	Raccord sur Moebius – Modèles 3D	67

6.3	Raccord sur Moebius – Graphiques courbure	68
6.4	Division des segments autour de l'événement de Wunderlich	71
6.5	Tracé paramétrique de la courbure du ruban de Moebius pour différentes largeurs.	72
6.6	Tracé de la taille de l'événement de Wunderlich en fonction de la largeur du ruban.	73
6.7	Un plectonème de ruban	74
6.8	Distance signée d'un point à un plan	74
6.9	Rubans de Moebius posés sur un plan	76
6.10	Schéma de l'expérience de Roman	76
6.11	Configurations 3D de l'expérience de Roman	78
6.12	Schéma de la plaque de pression	79
6.13	Loi d'échelle de l'expérience de Roman – Schéma	81
6.14	Loi d'échelle de l'expérience de Roman – Graphique	81
6.15	Expérience de Roman – Loi d'échelle pour Δ	82
7.1	Déversement latéral – Configuration 3D	84
7.2	Déversement latéral – Graphique distance latérale, longueur du ruban	84
7.3	Déversement latéral – Stabilité	86
7.4	L'expérience de décalage latéral des encastresments – Schéma	87
7.5	L'expérience de décalage latéral des encastresments – Configurations 3D	89
7.6	L'expérience de décalage latéral des encastresments – Graphique	90
7.7	Schéma de l'encastrement du ruban	91
7.8	Expérience FENICS-SHELL– Configurations 3D	92
7.9	Expérience FENICS-SHELL– Répartition des énergies du ruban	94
7.10	Expérience FENICS-SHELL– Répartition locale des énergies du ruban	95
A.1	Schéma de l'exemple	99

Liste des Algorithmes

1	Algorithme générique de minimisation	36
2	Algorithme calcul du Moebius	69
3	Roman, écrasement	80
4	Initialisation pour l'expérience de décalage latéral des encastements.	88

Bibliographie

- [Kir59] G. KIRCHHOFF. « Ueber das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes. » In : *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1859.56 (1^{er} jan. 1859), p. 285-313. DOI : [10.1515/crll.1859.56.285](https://doi.org/10.1515/crll.1859.56.285).
- [Lov88] Augustus Edward Hough LOVE. « The small free vibrations and deformation of a thin elastic shell ». In : *Philosophical Transactions of the Royal Society of London. (A.)* 179 (31 déc. 1888), p. 491-546. DOI : [10.1098/rsta.1888.0016](https://doi.org/10.1098/rsta.1888.0016).
- [Mic99] A. G. M. MICHELL. « Elastic stability of long beams under transverse forces ». In : *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 48.292 (1899), p. 298-309.
- [Sad29] Michael SADOWSKY. « Die Differentialgleichungen des MÖBIUSschen Bandes ». In : *Jahresbericht der Deutschen Mathematiker-Vereinigung* (1929), p. 49-51.
- [Wun62] W. WUNDERLICH. « Über ein abwickelbares Möbiusband ». In : *Monatshefte für Mathematik* 66.3 (juin 1962), p. 276-289. DOI : [10.1007/BF01299052](https://doi.org/10.1007/BF01299052).
- [Nag63] P. M. NAGHDI. « Foundations of elastic shell theory ». In : *Progress in Solid Mechanics*. Sous la dir. d'I. N. SNEDDON et R. HILL. T. 4. North-Holland, 1963, p. 1-90.
- [Bro70] C. G. BROYDEN. « The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations ». In : *IMA Journal of Applied Mathematics* 6.1 (1970), p. 76-90. DOI : [10.1093/imamat/6.1.76](https://doi.org/10.1093/imamat/6.1.76).
- [Ant74] S. S. ANTMAN. « Kirchhoff's problem for nonlinearly elastic rods ». In : *Quarterly of Applied Mathematics* 32.3 (1974), p. 221-240.
- [Sim85] J. C. SIMO. « A finite strain beam formulation. The three-dimensional dynamic problem. Part I ». In : *Computer Methods in Applied Mechanics and Engineering* 49.1 (1985), p. 55-70.
- [DKK91a] Eusebius DOEDEL, Herbert B. KELLER et Jean Pierre KERNEVEZ. « Numerical Analysis and Control of Bifurcation Problems (I) Bifurcation in Finite Dimensions ». In : *International Journal of Bifurcation and Chaos* 1.3 (1991), p. 493-520.
- [DKK91b] Eusebius DOEDEL, Herbert B. KELLER et Jean Pierre KERNEVEZ. « Numerical Analysis And Control of Bifurcation Problems (II) : Bifurcation In Infinite Dimensions ». In : *International Journal of Bifurcation and Chaos* 01.4 (1991), p. 745-772.

- [BB94] Marco BORRI et Carlo BOTTASSO. « An intrinsic beam model based on a helicoidal approximation—Part I : Formulation ». In : *International Journal for Numerical Methods in Engineering* 37.13 (1994), p. 2267-2289. DOI : [10.1002/nme.1620371308](https://doi.org/10.1002/nme.1620371308).
- [SF96] Meng SUN et Eugene FIUME. « A technique for constructing developable surfaces ». In : *Proceedings of the 1996 Conference on Graphics Interface Toronto* (1996). Avec la coll. de Richard BARTELS et Wayne A. DAVIS, 10 pages, 295.57 kB. DOI : [10.20380/GI1996.21](https://doi.org/10.20380/GI1996.21).
- [CJ98] M. A. CRISFIELD et G. JELENIĆ. « Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation ». In : *Proc. Royal Society of London, Series A* 455.1983 (1998), p. 1125-1147.
- [Neh99] M. NEHER. « An enclosure method for the solution of linear ODEs with polynomial coefficients ». In : *Numerical Functional Analysis and Optimization* 20 (1999), p. 779-803.
- [RP99] B ROMAN et A POCHEAU. « Buckling cascade of thin plates : Forms, constraints and similarity ». In : *Europhysics Letters (EPL)* 46.5 (1^{er} juin 1999), p. 602-608. DOI : [10.1209/epl/i1999-00306-3](https://doi.org/10.1209/epl/i1999-00306-3).
- [Rom00] B. ROMAN. « De l'élastica aux plaques plissées ». Thèse de doct. Université de Provence (Aix-Marseille I), 2000. URL : <https://blog.espci.fr/benoitroman/files/2019/05/Elastica-plaques-plissees-these-Roman.pdf>.
- [Iva01] Elena A IVANOVA. « On use of a new method of solution of Darboux problem for solution of the problem of motion of a ball on a Routh plane ». In : *XXVIII Summer School Nonlinear oscillations in Mechanical Systems. Actual Problems in Mechanics* (2001), p. 361-367.
- [Pai02] D. PAI. « Strands : Interactive Simulation of Thin Solids using Cosserat Models ». In : *Computer Graphics Forum (Proc. Eurographics'02)* 21.3 (2002), p. 347-352.
- [Luro5] A. I. LURIE. *Theory of Elasticity*. Foundations of Engineering Mechanics. Springer, 2005.
- [Ber+06] F. BERTAILS et al. « Super-Helices for Predicting the Dynamics of Natural Hair ». In : *ACM Transactions on Graphics* 25 (2006), p. 1180-1187. DOI : [10.1145/1141911.1142012](https://doi.org/10.1145/1141911.1142012).
- [GNo6] Alain GORIELY et Sébastien NEUKIRCH. « Mechanics of Climbing and Attachment in Twining Plants ». In : *Physical Review Letters* 97.18 (1^{er} nov. 2006), p. 184302. DOI : [10.1103/PhysRevLett.97.184302](https://doi.org/10.1103/PhysRevLett.97.184302).
- [NW06] J. NOCEDAL et S. WRIGHT. *Numerical Optimization*. 2nd edition. New York, NY, USA : Springer, 2006.
- [WBo6] Andreas WÄCHTER et Lorenz T. BIEGLER. « On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming ». In : *Mathematical Programming* 106.1 (2006), p. 25-57. DOI : [10.1007/s10107-004-0559-y](https://doi.org/10.1007/s10107-004-0559-y).
- [BW07] Pengbo BO et Wenping WANG. « Geodesic-Controlled Developable Surfaces for Modeling Paper Bending ». In : *Computer Graphics Forum* 26.3 (sept. 2007), p. 365-374. DOI : [10.1111/j.1467-8659.2007.01059.x](https://doi.org/10.1111/j.1467-8659.2007.01059.x).

- [SBT07] J. SPILLMANN, M. BECKER et M. TESCHNER. « Non-iterative Computation of Contact Forces for Deformable Objects ». In : *Journal of WSCG* 15 (2007), p. 33-40.
- [SH07] E. L. STAROSTIN et G. H. M. van der HEIJDEN. « The shape of a Möbius strip ». In : *Nature Materials* 6 (juill. 2007), p. 563. DOI : [10.1038/nmat1929](https://doi.org/10.1038/nmat1929).
- [Ber+08] M. BERGOU et al. « Discrete elastic rods ». In : *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'08)* 27.3 (2008), p. 1-12. DOI : <http://doi.acm.org/10.1145/1360612.1360662>.
- [GPL08] S. GOYAL, N. PERKINS et C. LEE. « Non-linear dynamic intertwining of rods with self-contact ». In : *International Journal of Non-Linear Mechanics* 43.1 (2008), p. 65-73. DOI : [10.1016/j.ijnonlinmec.2007.10.004](https://doi.org/10.1016/j.ijnonlinmec.2007.10.004).
- [Bero9] F. BERTAILS. « Linear Time Super-Helices ». In : *Computer Graphics Forum (Proc. Eurographics'09)* 28.2 (avr. 2009). DOI : [10.1111/j.1467-8659.2009.01381.x](https://doi.org/10.1111/j.1467-8659.2009.01381.x).
- [AP10] B. AUDOLY et Y. POMEAU. *Elasticity and Geometry : From hair curls to the nonlinear response of shells*. Oxford University Press, 2010.
- [LLA11] Holger LANG, Joachim LINN et Martin ARNOLD. « Multi-body dynamics simulation of geometrically exact Cosserat rods ». In : *Multibody System Dynamics* 25.3 (mars 2011), p. 285-312. DOI : [10.1007/s11044-010-9223-x](https://doi.org/10.1007/s11044-010-9223-x).
- [DO12] Eusebius J. DOEDEL et Bart E. OLDEMAN. *AUTO-07P : Continuation and bifurcation software for ordinary differential equations*. Concordia University, 2012.
- [Sol+12] Justin SOLOMON et al. « Flexible Developable Surfaces ». In : *Computer Graphics Forum* 31.5 (août 2012), p. 1567-1576. DOI : [10.1111/j.1467-8659.2012.03162.x](https://doi.org/10.1111/j.1467-8659.2012.03162.x).
- [CB13] R. CASATI et F. BERTAILS-DESCOUBES. « Super Space Clothoids ». In : *ACM Transactions on Graphics* 32.4 (2013), p. 48. DOI : [10.1145/2461912.2461962](https://doi.org/10.1145/2461912.2461962).
- [Cir+14] Gabriel CIRIO et al. « Yarn-level simulation of woven cloth ». In : *ACM Transactions on Graphics* 33.6 (19 nov. 2014), p. 1-11. DOI : [10.1145/2661229.2661279](https://doi.org/10.1145/2661229.2661279).
- [SCB14] V. SONNEVILLE, A. CARDONA et O. BRÜLS. « Geometrically exact beam finite element formulated on the special Euclidean group SE(3) ». In : *Computer Methods in Applied Mechanics and Engineering* 268 (2014), p. 451-474. DOI : <http://dx.doi.org/10.1016/j.cma.2013.10.008>.
- [Cas15] Romain CASATI. « Quelques contributions à la modélisation numérique de structures élastiques pour l'informatique graphique ». Thèse de doct. Université Grenoble Alpes, 2015. URL : <http://www.theses.fr/2015GREAM053/document>.
- [DA15] Marcelo A. DIAS et Basile AUDOLY. « “Wunderlich, Meet Kirchhoff” : A General and Unified Description of Elastic Ribbons and Thin Rods ». In : *Journal of Elasticity* 119.1 (avr. 2015), p. 49-66. DOI : [10.1007/s10659-014-9487-0](https://doi.org/10.1007/s10659-014-9487-0).
- [HF15] Denis F. HINZ et Eliot FRIED. « Translation of Michael Sadowsky's Paper “The Differential Equations of the Möbius Band” ». In : *Journal of Elasticity* 119.1 (2015), p. 19-22.

- [Sch+15] Camille SCHRECK et al. « Non-smooth developable geometry for interactively animating paper crumpling ». In : *ACM Transactions on Graphics* 35.1 (2015), 10 :1-10 :18. DOI : [10.1145/2829948](https://doi.org/10.1145/2829948).
- [She+15] Zhongwei SHEN et al. « Geometrically Exact Simulation of Inextensible Ribbon ». In : *Computer Graphics Forum* 34.7 (2015), p. 145-154. DOI : [10.1111/cgf.12753](https://doi.org/10.1111/cgf.12753).
- [Dud+16] Levi H. DUDTE et al. « Programming curvature using origami tessellations ». In : *Nature Materials* 15.5 (1^{er} mai 2016), p. 583-588. DOI : [10.1038/nmat4540](https://doi.org/10.1038/nmat4540).
- [SRH17] Camille SCHRECK, Damien ROHMER et Stefanie HAHMANN. « Interactive paper tearing ». In : *Computer Graphics Forum* 36.2 (2017), p. 95-106. DOI : [10.1111/cgf.13110](https://doi.org/10.1111/cgf.13110).
- [Van17] Aleš VANĚK. « Lightweight Grid Shell Pavilion - Design, Manufacture and Erection of Full Scale Grid Shell Prototypes ». In : *Selected Scientific Papers - Journal of Civil Engineering* 11.2 (2017), p. 115-128. DOI : [doi:10.1515/sspjce-2016-0024](https://doi.org/10.1515/sspjce-2016-0024).
- [Hal+18] Jack S. HALE et al. « Simple and extensible plate and shell finite element models through automatic code generation tools ». In : *Computers & Structures* 209 (2018), p. 163-181.
- [MGN18] Derek E. MOULTON, Paul GRANDGEORGE et Sébastien NEUKIRCH. « Stable elastic knots with no self-contact ». In : *Journal of the Mechanics and Physics of Solids* 116 (2018), p. 33-53. DOI : [10.1016/j.jmps.2018.03.019](https://doi.org/10.1016/j.jmps.2018.03.019).
- [RHS18] Michael RABINOVICH, Tim HOFFMANN et Olga SORKINE-HORNUNG. « Discrete Geodesic Nets for Modeling Developable Surfaces ». In : *ACM Transactions on Graphics* 37.2 (3 juill. 2018), p. 1-17. DOI : [10.1145/3180494](https://doi.org/10.1145/3180494).
- [Ren+18] F. RENDA et al. « Discrete Cosserat Approach for Multisection Soft Manipulator Dynamics ». In : *IEEE Transactions on Robotics* 34.6 (déc. 2018), p. 1518-1533. DOI : [10.1109/TR0.2018.2868815](https://doi.org/10.1109/TR0.2018.2868815).
- [SGC18] Oded STEIN, Eitan GRINSPUN et Keenan CRANE. « Developability of triangle meshes ». In : *ACM Transactions on Graphics* 37.4 (10 août 2018), p. 1-14. ISSN : 0730-0301, 1557-7368. DOI : [10.1145/3197517.3201303](https://doi.org/10.1145/3197517.3201303).
- [MH19] Alexander MOORE et Timothy HEALEY. « Computation of elastic equilibria of complete Möbius bands and their stability ». In : *Mathematics and Mechanics of Solids* 24.4 (2019), p. 939-967. DOI : [10.1177/1081286518761789](https://doi.org/10.1177/1081286518761789).
- [YH19] T. YU et J.A. HANNA. « Bifurcations of buckled, clamped anisotropic rods and thin bands under lateral end translations ». In : *Journal of the Mechanics and Physics of Solids* 122 (jan. 2019), p. 657-685. DOI : [10.1016/j.jmps.2018.01.015](https://doi.org/10.1016/j.jmps.2018.01.015).
- [Cha+20] Raphaël CHARRONDIÈRE et al. « Numerical modeling of inextensible elastic ribbons with curvature-based elements ». In : *Computer Methods in Applied Mechanics and Engineering* 364 (juin 2020), p. 112922. DOI : [10.1016/j.cma.2020.112922](https://doi.org/10.1016/j.cma.2020.112922).
- [Hua+20] Weicheng HUANG et al. « Shear induced supercritical pitchfork bifurcation of pre-buckled bands, from narrow strips to wide plates ». In : *Journal of the Mechanics and Physics of Solids* 145 (déc. 2020), p. 104168. DOI : [10.1016/j.jmps.2020.104168](https://doi.org/10.1016/j.jmps.2020.104168).

- [AN21] Basile AUDOLY et Sébastien NEUKIRCH. « A one-dimensional model for elastic ribbons : A little stretching makes a big difference ». In : *Journal of the Mechanics and Physics of Solids* 153 (août 2021), p. 104457. DOI : [10.1016/j.jmps.2021.104457](https://doi.org/10.1016/j.jmps.2021.104457).
- [Rom+21] Victor ROMERO et al. « Physical validation of simulators in Computer Graphics : A new framework dedicated to slender elastic structures and frictional contact ». In : *ACM Transactions on Graphics* (août 2021), p. 1-18. URL : <https://hal.inria.fr/hal-03217459>.
- [WCS21] Thomas WOLF, Victor CORNILLÈRE et Olga SORKINE-HORNUNG. « Physically-based Book Simulation with Freeform Developable Surfaces ». In : *Computer Graphics Forum (proceedings of EUROGRAPHICS 2021)* 40.2 (2021).
- [HA22] G van der HEIJDEN et B. AUDOLY. « preprint ». In : (2022).