



**HAL**  
open science

## Static inverse modelling of cloth

Mickaël Ly

► **To cite this version:**

Mickaël Ly. Static inverse modelling of cloth. Modeling and Simulation. Université Grenoble Alpes, 2021. English. NNT: . tel-03516830v1

**HAL Id: tel-03516830**

**<https://inria.hal.science/tel-03516830v1>**

Submitted on 21 Dec 2021 (v1), last revised 24 Jan 2022 (v3)

**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 LA  
COMMUNAUTÉ UNIVERSITÉ GRENOBLE  
ALPES**

Spécialité : **Mathématiques-informatique**

Arrêté ministériel : 25 mai 2016

Présentée par

**Mickaël Ly**

Thèse dirigée par **Florence Bertails-Descoubes**  
et codirigée par **Mélina Skouras**

préparée au sein de l'équipe-projet **Elan - Inria Grenoble Rhône-Alpes**  
dans l'École Doctorale **MSTII**

## Static inverse modelling of cloth

Thèse soutenue publiquement le **28 Septembre 2021**,  
devant le jury composé de :

**Jérôme Malick**

Directeur de Recherche, LJK, CNRS, Président

**Maud Marchal**

Professeure, IRISA, Université de Rennes, Rapportrice

**Rahul Narain**

Assistant Professor, Indian Institute of Technology, Dehli, Rapporteur

**Loïc Barthe**

Professeur, IRIT, Université Paul Sabatier, Examineur

**Bernhard Thomaszewski**

Professeur Associé, Université de Montréal, Examineur

**Florence Bertails-Descoubes**

Directrice de Recherche, Inria, LJK, Directrice de thèse

**Mélina Skouras**

Chargée de Recherche, Inria, LJK, Co-encadrante de thèse

**Charles Dapogny**

Chargé de Recherche, LJK, CNRS, Invité





# Thanks / Remerciements

Il y a tant de personnes que je souhaite remercier que ce qui suit risque d'être confus et incomplet, mais pour faire court, merci à toutes les personnes qui m'ont soutenu et qui m'ont accompagné jusqu'à là.

Mes premiers remerciements pour cette thèse vont à Florence qui m'a initié à la recherche au cours cet IRL à l'Ensimag et avec qui je n'ai cessé d'apprendre depuis lors. Un grand merci aussi à Méлина et Charles pour leur co-encadrement et avec qui j'ai également beaucoup appris. Je remercie ensuite aux rapporteurs et aux membres du jury d'avoir pris le temps de lire ce manuscrit et d'avoir évalué mon travail.

Cette thèse n'a pas été que le travail rapporté dans ce manuscrit, mais aussi une super vie au labo et autour. Un grand merci aux *Janusz* de feu BIPOP, Matteo, Nestor, Jan, Nahuel, Kirill, Achref, Alexandre R. et tout spécialement Alexandre V. ainsi qu'aux *Élans*, Thibaut, Raphaël, Victor, Haroon, Gauthier, Jean pour toutes les discussions, scientifiques ou non, les restos, les cafés et tous les autres bons moments. Merci également aux autres collaborateurs et collègues avec qui j'ai pu travailler et apprendre. Sébastien, Arnaud lors de ce super projet de validation, Damien pour ce projet de skinning - qu'on finira un jour !-, Everton ainsi que les collègues de *Dassault Systèmes* pour la formation et m'avoir initié à d'autres aspects de la simulation et enfin Jérôme pour m'avoir donné goût à l'optimisation numérique. Pour finir sur le labo, merci à Diane et Julia pour l'aide apportée à l'équipe ainsi que votre bonne humeur.

Je vais terminer cette page de remerciement avec tous les amis qui m'accompagnent depuis tant d'années. Merci aux potos des Saules et de Marie Curie d'être ou de revenir toujours dans le coin après tout ce temps, Cécile, Cédric B., Bérénice, Assma, Sabine, ainsi que Dylan, Hugo, Raphaël, Sandrine et William. Merci aux licornes de l'Ensimag, Maxence, Ilyes, Matthieu et Vincent pour nos *calls* du dimanche et le frama, toujours débordant d'énergie - ou presque. Enfin, merci à mes grenoblois (ou presque) préférés de Champo (ou pas loin), "Monsieur le Marquis" Luc, le "Canard" Florian, Estelle la licorne, Guillaume, Daphnée, Arnaud L. les co-thésards de Champo Valentine, Cédric D., Noémie, MaThieu et Eugénie (qui a enfin compris pourquoi je parlais de manches ballon) pour toutes les soirées jeux où "il fallait focus Yoshi parce qu'il est en train de gagner", et pour toutes les randos où les paysages magnifiques justifiaient les mètres de dénivelé cachés.

Et surtout, merci à mes parents pour tout.



# Contents

<b>Glossary and notations</b>	<b>3</b>
<b>Introduction</b>	<b>5</b>
<b>I Thin elastic shells, frictional contact, and application to cloth simulation</b>	<b>9</b>
<b>1 State of the art</b>	<b>11</b>
1.1 Thin elastic shell models . . . . .	12
1.1.1 Plates and shells models derived in Mechanical Engineering . . . . .	12
1.1.2 Plates and shells in Computer Graphics . . . . .	16
1.2 Contact and friction in Computer Graphics: models & simulation . . . . .	20
1.2.1 Signorini-Coulomb law . . . . .	20
1.2.2 Penalty-based/smooth methods. . . . .	23
1.2.3 Non-smooth solvers . . . . .	25
<b>2 Projective Dynamics with contact and friction</b>	<b>29</b>
2.1 The Projective Dynamics method . . . . .	30
2.1.1 Dynamics of a nodal system . . . . .	30
2.1.2 Reformulation & algorithm . . . . .	31
2.1.3 Contacts handling . . . . .	33
2.2 Projective Friction . . . . .	34
2.2.1 Splitting scheme for the contact force estimation . . . . .	34
2.2.2 Extensions of the base case. . . . .	36
2.2.3 Algorithm . . . . .	38
2.3 Evaluation and results . . . . .	38
2.3.1 Test scenarios . . . . .	38
2.3.2 Qualitative evaluation . . . . .	39
2.3.3 Convergence . . . . .	42

2.3.4	Performance	43
2.4	Discussion and conclusion	44
<b>3</b>	<b>Validation of codes in Computer Graphics</b>	<b>47</b>
3.1	Organisation of the chapter and disclaimer on the contributions (↔)	48
3.2	Validation in Computer Graphics	48
3.2.1	Motivation	48
3.2.2	Related work	49
3.3	Validation protocols	52
3.3.1	Scaling laws	52
3.3.2	The <b>Cantilever</b> test	53
3.3.3	The <b>Lateral Buckling</b> test	56
3.3.4	The <b>Stick-Slip</b> test	57
3.4	Codes tested	61
3.4.1	Plates & shells	61
3.4.2	Frictional contact	62
3.5	Evaluation and results	62
3.5.1	Our methodology	62
3.5.2	↔Results for the <b>Cantilever</b> test↔	64
3.5.3	↔Results for the <b>Lateral Buckling</b> test↔	68
3.5.4	Results for the <b>Stick-Slip</b> test	72
3.6	Discussion and conclusion	74
<b>II</b>	<b>Inverse design of shells under frictional contact, and application to inverse garment design</b>	<b>77</b>
<b>4</b>	<b>State of the art</b>	<b>79</b>
4.1	Inverse design	80
4.1.1	Inverse design in Computer Graphics	80
4.1.2	Cloth design	83
4.1.3	Inversion with contact and friction	86
4.2	Discrete developability	86
<b>5</b>	<b>Inverse elastic shell design with contact and friction</b>	<b>89</b>
5.1	Motivation	90
5.2	Overview of the algorithm	90
5.3	Mechanics	93
5.3.1	Shell model	93

5.3.2	Gravitational energy . . . . .	93
5.3.3	Frictional contact . . . . .	94
5.3.4	Equilibrium and stability . . . . .	95
5.4	Step 1: Inversion with unilateral constraints . . . . .	96
5.4.1	Least-squares formulation and draping function . . . . .	96
5.4.2	Evaluation and regularisation of the draping function . . . . .	99
5.5	Step 2: Accounting for frictional contact . . . . .	100
5.5.1	Defining the admissible forces . . . . .	100
5.5.2	Correction step . . . . .	102
5.6	Results . . . . .	103
5.6.1	Implementation details . . . . .	103
5.6.2	Framework . . . . .	103
5.6.3	Qualitative results . . . . .	104
5.6.4	Evaluation . . . . .	106
5.6.5	Convergence and performance . . . . .	113
5.7	Conclusion, limitation and discussion . . . . .	114
5.7.1	Performance . . . . .	114
5.7.2	Flaws of the second step . . . . .	115
5.7.3	Material parameters . . . . .	116
5.7.4	Toward garments inversion . . . . .	116
<b>6</b>	<b>Towards garment inversion</b> . . . . .	<b>119</b>
6.1	Including discrete developability . . . . .	120
6.1.1	Framework . . . . .	120
6.1.2	Comparison of the discrete developability criteria . . . . .	122
6.2	Modifications of the inversion algorithm . . . . .	126
6.2.1	Plates or shells ? . . . . .	126
6.2.2	Parameter identification through continuation . . . . .	128
6.2.3	Incorporating the continuation . . . . .	131
6.2.4	Full algorithm . . . . .	132
6.3	Results . . . . .	134
6.3.1	Penalisation strategy . . . . .	135
6.3.2	Adaptive coefficient . . . . .	139
6.3.3	Remeshing and hierarchical strategy . . . . .	140
6.3.4	Friction penalisation . . . . .	146
6.4	Conclusion . . . . .	147
	<b>Conclusion</b> . . . . .	<b>149</b>
	Summary of the contributions . . . . .	149



<i>CONTENTS</i>	1
Perspectives . . . . .	150
<b>A Notions of linear elasticity</b>	<b>153</b>
<b>B Fundamental forms for the mechanics of shells</b>	<b>157</b>
B.1 Continuous setting . . . . .	157
B.2 Discrete setting . . . . .	158
<b>C Reformulation of the Koiter energy</b>	<b>161</b>
<b>D Second derivatives in the implicit function theorem</b>	<b>163</b>
D.1 Framework . . . . .	163
D.2 Computation methods . . . . .	163
D.3 Application to our inverse problem . . . . .	166



## Abstract

This thesis deals with the direct simulation and inverse design of garments in the presence of frictional contact.

The shape of draped garments results from the slenderness of the fabric, which can be represented in mechanics by a thin elastic plate or shell, and from its interaction with the body through contact and dry friction. This interaction, necessary to reproduce the threshold friction occurring in such contacts, is described by a non smooth law, which, in general, makes its integration complex. In a first contribution, we modify the so-called Projective Dynamics algorithm to incorporate this dry frictional contact law in a simple way. Projective Dynamics is a popular method in Computer Graphics that quickly simulates deformable objects such as plates with moderate accuracy, yet without including frictional contact. The rationale of this algorithm is to solve the integration of the dynamics by successively calculating estimates of the shape of the object at the next timestep. We take up the same idea to incorporate a procedure for estimating the frictional contact law that robustly captures the threshold phenomenon.

In addition it is interesting to note that simulators developed in Computer Graphics, originally targeted at visual animation, have become increasingly accurate over the years. They are now being used in more "critical" applications such as architecture, robotics or medicine, which are more demanding in terms of accuracy. In collaboration with mechanicians and experimental physicists, we introduce into the Computer Graphics community a number of protocols to verify the correctness of simulators, and we present in this manuscript our contributions related to plate and shell simulators.

Finally, in a last part, we focus on garment inverse design. The interest of this process is twofold. Firstly, for computing equilibria, solving the inverse problem provides a "force-free" and possibly curved version of the input (called the rest or natural shape), whether it comes from a 3D design or a 3D capture, that allows to start the simulation with the input as the initial deformed shape. To this end, we propose an algorithm for the inverse design of clothes represented by thin shells that also accounts for dry frictional contact. Within our framework, the input shape is considered to be a mechanical equilibrium subject to gravity and contact forces. Then our algorithm computes a rest shape such that this input shape can be simulated without any sagging. Secondly, it is also appealing to use these rest shapes for a real life application to manufacture the designed garments without sagging. However, the traditional cloth fabrication process is based on patterns, that is sets of flat panels sewn together. In this regard, we present in our more prospective part our results on the adaptation of the previous algorithm to include geometric constraints, namely surface developability, in order to get flattenable rest shapes.

## Résumé

Cette thèse porte sur la simulation directe et la conception inverse de vêtements en présence de contact frottant.

La forme de vêtements portés résulte en effet à la fois de la minceur du tissu, représentable en mécanique par une plaque ou une coque mince et élastique, et de son interaction avec le corps à travers un phénomène de contact frottant solide. Cette interaction, nécessaire pour reproduire le frottement à seuil typique des interactions entre solides, est décrite par une loi non régulière, ce qui rend son intégration généralement complexe. Dans une première contribution, nous modifions l'algorithme *Projective Dynamics* afin d'y introduire simplement cette loi de contact frottant. *Projective Dynamics* est une méthode populaire en Informatique Graphique qui simule rapidement avec une précision modérée des objets déformables tels que les plaques, mais sans inclure de contact frottant. L'idée principale de cet algorithme est de résoudre l'intégration de la dynamique en calculant successivement des estimations de la forme de l'objet au pas de temps suivant. Nous reprenons la même idée afin d'y incorporer une procédure d'estimation de la loi de contact frottant qui parvient de manière robuste à capturer le phénomène de seuil.

Par ailleurs, il est intéressant de noter que les simulateurs développés en Informatique Graphique, dédiés à l'origine à l'animation, sont devenus de plus en plus précis au fil des ans. Ils sont maintenant sollicités dans des applications plus "critiques" telles que l'architecture, la robotique ou la médecine plus exigeantes en terme de justesse. Dans une collaboration avec des mécaniciens et des physiciens expérimentateurs, nous introduisons de nouveaux protocoles de validation des simulateurs graphiques et nous présentons dans ce manuscrit nos contributions relatives aux simulateurs de plaques et de coques.

Enfin, dans une dernière partie, nous nous intéressons à la conception inverse de vêtements. L'intérêt de ce procédé est double. En premier lieu, pour des simulations, résoudre le problème inverse fournit une version "sans force" et possiblement courbée de l'entrée (dite naturelle ou au repos), que celle-ci provienne d'un modèle 3D ou d'une capture 3D, qui permet d'initier la simulation avec la forme de l'entrée en tant que forme déformée initiale. En ce sens, nous proposons un algorithme pour la conception inverse de coques en présence de contact frottant. Dans notre cadre, la forme donnée en entrée est considérée comme un équilibre mécanique soumis à la gravité et aux forces de contact. Notre algorithme calcule ensuite une forme au repos telle que l'entrée puisse être simulée sans qu'elle ne s'affaisse. En second lieu, il est aussi tentant de vouloir utiliser ces formes naturelles pour une application concrète afin de confectionner lesdits vêtements sans qu'ils ne s'affaissent. Cependant, le processus classique de fabrication de vêtements est basé sur l'usage de patrons, c'est-à-dire d'ensembles de panneaux plats à coudre ensemble. Nous présentons donc dans une partie finale plus prospective nos résultats sur l'adaptation de notre algorithme précédent afin d'y incorporer des contraintes géométriques, en l'occurrence la développabilité des surfaces, afin d'obtenir des formes au repos aplatissables.

## Glossary and notations

### Mathematics

$\mathbb{N}$	Set of positive integers
$\mathbb{Z}$	Set of integers
$\mathbb{R}$	Set of reals
$[a, b]$	Interval of $\mathbb{R}$
$\llbracket a, b \rrbracket$	Interval of $\mathbb{Z}$ ; $= [a, b] \cap \mathbb{Z}$
$\mathcal{S}_n$	Set of real symmetric matrices of size $n$
$\mathcal{S}_n^+$	Set of real symmetric positive matrices of size $n$
$\mathcal{S}_n^{++}$	Set of real symmetric definite positive matrices of size $n$
$\mathbf{I}_n$	Identity matrix of size $n$
$\mathcal{S}(c, r)$	Sphere of center $c$ and radius $r$
$\nabla f$	Gradient of the function $f$ ; the Jacobian for real-valued functions
$\nabla^2 f$	Hessian of the function $f$
$Df$	Jacobian of the function $f$
$\partial_{\bullet}$	Partial derivative operator <i>w.r.t.</i> the variable $\bullet$
$\ \bullet\ $	Euclidean norm when the norm is unspecified
LCP	Linear Complementarity Problem
NCP	Non-linear Complementarity Problem
MLCP	Mixed Linear Complementarity Problem

### Geometry

- A** First fundamental form
- B** Second fundamental form

### Mechanics

$E$	Young modulus (in Pa)
$\nu$	Poisson ratio (without unit)
$\epsilon$	Strain tensor ( $\in \mathcal{S}_2/\mathcal{S}_3$ , without unit)
$\sigma$	Stress tensor ( $\in \mathcal{S}_2/\mathcal{S}_3$ , in Pa)
$\mu$	Friction coefficient ( $\in \mathbb{R}^+$ , without unit)
$\mathcal{K}_\mu(e)$	$\subset \mathbb{R}^3$ , Coulomb cone of orientation $e \in \mathbb{R}^3$ and aperture $\mu \in \mathbb{R}^+$
$\dot{\bullet}$	Time derivative of the variable $\bullet$
TOI	Time of Impact
DFCP	Discrete Frictional Contact Problem

**Algorithms/Miscellaneous**

CCD	Continuous Collision Detection
PBD	Position-Based Dynamics ( <a href="#">Müller et al., 2007</a> )
X-PBD	Extended Position-Based Dynamics ( <a href="#">Macklin et al., 2016</a> )
PD	Projective Dynamics ( <a href="#">Bouaziz et al., 2014</a> )
ADMM	Alternating Direction Method of Multipliers
PF	Projective Friction ( <a href="#">Ly et al., 2020</a> )
DOG	Discrete Orthogonal Geodesic ( <a href="#">Rabinovich et al., 2018a</a> )
DTM	Developability of Triangle Meshes ( <a href="#">Stein et al., 2018</a> )
DGC	Discrete Gaussian Curvature

# Introduction

Simulation tools, since their introduction in Computer Graphics, have become essential to artists in the movie industry to generate vivid and visually plausible animations. They have since then evolved in two different yet complementary directions. On the one hand, some researchers have focussed on improving the speed and the robustness of real-time simulators for applications such as video games or surgical training softwares. On the other hand, with the aim of producing more realistic visual effects, other simulators have been developed to handle models that are more complex and more faithful to the physics.

Yet, controlling the outcome of these simulations still remains a challenging problem. To avoid a long process of trial-error, one solution consists in providing tools to manually guide the simulation (see *e.g.* (Butts et al., 2018)). While efficient, these methods are suitable only for animation purposes as the controls interfere with the physics. Harder to tackle, yet more accurate, inverse design problems intend to automatically compute initial parameters, so that the output of the simulation is as close as possible to a given configuration (see Figure 1). Combined with physically accurate models, these methods offer applications that go beyond the virtual world with strong predictive power.



Figure 1: Derouet-Jourdan et al. (2013) ’s method prevents the hair style from sagging (left) by computing the hair rest shape, enabling the computed equilibrium to match perfectly the input (middle), while allowing further animation of the character (right).

In this thesis, our main goal is to study the inverse design of clothes. Given a 3D shape representing a garment of a given material, we interpret this *target* shape as a deformed pose of an unknown shape at rest that is submitted to its internal elastic force and external forces (gravity, friction), and we aim at computing this unknown shape.

A straightforward application of this method is to enable artists to design any garment shape they want, and then simulate them without seeing their design sag as soon as the physics are applied. But more than that, with the developments on garments 3D reconstruction, more applications in the long term can be considered such as sag-free virtual try-on of complex garments or cloth pattern computation.

The manuscript is composed of two main parts. As a prerequisite to the garment inverse design problem, we focus in the first part of this thesis on the direct simulation of clothes in frictional contact interaction.

In Chapter 1, I start by proposing a broad overview of the thin elastic plates and shells models developed in the Mechanical Engineering and the Computer Graphics communities. Then, I introduce in a short review the models and algorithms used in Computer Graphics to deal with contact and friction.

I continue by presenting in Chapter 2 our contribution regarding the simulation of garments with frictional contact (Ly et al., 2020). In this work, we modify the PROJECTIVE DYNAMICS (Bouaziz et al., 2014) framework, initially developed to produce mildly accurate but stable and efficient simulations of deformable objects, in order to incorporate frictional contact. Although the framework is not meant to yield highly physically accurate simulations, we show that our method still manages to qualitatively reproduce the dry friction behaviour and provide satisfying results, as illustrated in Figure 2.

Then, in Chapter 3, I present the work of our research group to validate the physics of numerical solvers of slender structures and frictional contact (Romero et al., 2021). We introduce several protocols inspired by theoretical and experimental results reported in the Soft Matter Physics literature, as the one depicted in Figure 3, to test and evaluate the physical accuracy of numerical simulators. Within the presentation, I take care to distinguish my contributions to the project from that of my colleagues that are presented for the sake of scientific completeness.





Figure 2: Complex dress simulated with our modified PROJECTIVE DYNAMICS framework.

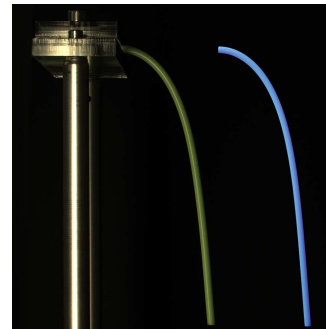


Figure 3: The **Cantilever** test evaluates the accuracy of the bending model in 1D. Fabricated rod (left) vs a simulation (right) produced by the DISCRETE ELASTIC ROD code of [Bergou et al. \(2008\)](#).

The second part of this manuscript focuses on the cloth inversion problem aforementioned.

After an introduction to inverse problems in Computer Graphics in Chapter 4, I present in the following chapter our algorithm for the inverse design of shells subject to frictional contact ([Ly et al., 2018](#)). [Casati et al. \(2016\)](#) proposed a method that works robustly in the case of pin constraints, but their extension to frictional contact was not robust. I show that by treating friction with a correction step added to their method, the resulting algorithm consistently produces rest shapes for the designed shells to be at equilibrium under gravity and friction as illustrated in Figure 4.

Finally, in a last and more prospective chapter, I present our latest modifications to our inversion algorithm. Aiming at applying our method to real garments, with as a long-term objective to automatically compute the cloth patterns, we try to introduce geometric considerations, namely the surface developability, in our method. A parameter estimation procedure is also tested. The resulting modified algorithm yields promising results although further work is required to completely achieve our goal.

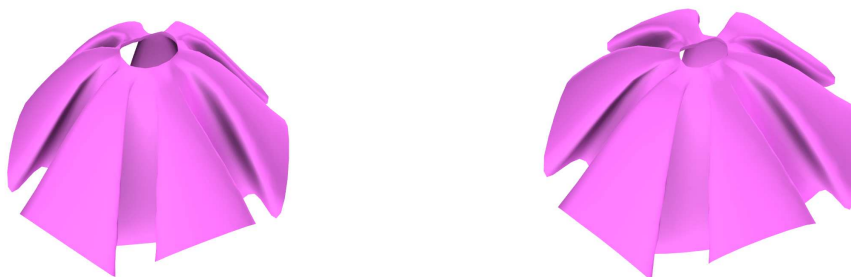


Figure 4: To preserve the design of the skirt (left), our algorithm computes a flared rest shape, tighter at the waist (right), so as to retrieve the designed shape under gravity and friction.



# Part I

Thin elastic shells, frictional contact,  
and application to cloth simulation



# Chapter 1

## State of the art

In this chapter, I present an overview of the simulation of thin elastic shells and frictional contact, as a requirement to the following chapters as well as to the second part on the inverse design of thin elastic shells subject to dry friction.

This thesis has mainly been conducted in a Computer Graphics context. Yet, as the modelling techniques, the problematics and the developed solutions of both Computer Graphics and Mechanical Engineering communities are converging on the search of efficient yet physically accurate simulators ([Bertails-Descoubes and Audoly, 2019](#)), I also propose glimpses of the related work done in the Mechanical Engineering.

In the first section, I identify and introduce three main categories of shell models: the models issued from the early work in Mechanical Engineering, the geometry-based models developed in Computer Graphics and finally the more recent models combining the work of Mechanical Engineering and Discrete Differential Geometry. Then, in the second section, I present the treatment of dry frictional contact, an early challenging problem for both communities that addresses the non-interpenetration of solid objects and their realistic interaction according to the Coulomb friction rules. Solutions to tackle this problem are quite diverse, and range from penalty based methods to variational formulations and constraints-based solvers.

## 1.1 Thin elastic shell models

### 1.1.1 Plates and shells models derived in Mechanical Engineering

This section does not pretend to cover all the work done in Mechanical Engineering, especially during the recent years, but rather depicts the general ideas leading to the derivation of the models. In this section, I focus on the Föppl–von Kármán plate model whose derivation from 3D elasticity is a good introduction to the specifics of the plate mechanics and the Koiter shell model whose discrete version will be used in our simulations. For a more complete review, the reader may refer for instance to (Caliri et al., 2016).

In the context of Mechanical Engineering, elastic plates and shells are deformable objects with one dimension, the thickness, very small compared to the others (the length and the width). The difference between a plate and a shell is that the rest (undeformed) configuration of a plate is planar whereas that of a shell is not.

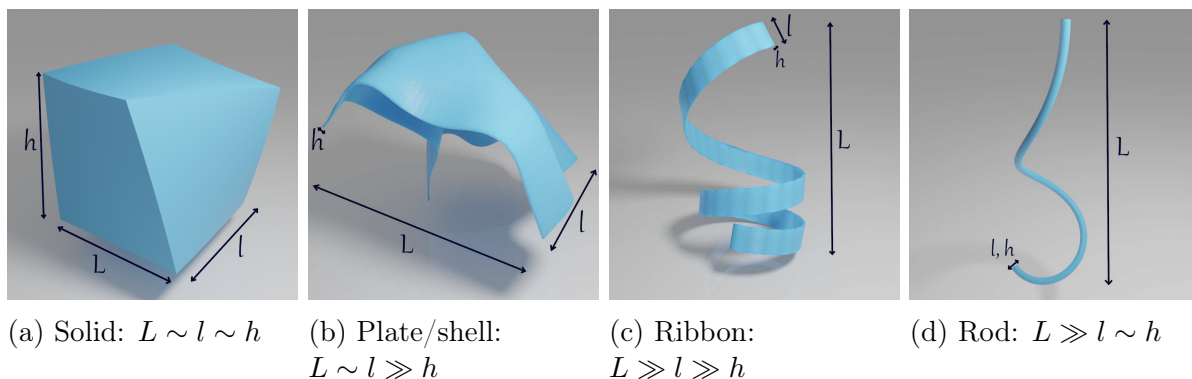


Figure 1.1: Classification of deformable objects based on their characteristic lengths.

**Asymptotic shell models.** Since elastic plates are elastic media with a small thickness, a natural approach to formulate their mechanical deformations is to take the equations of three-dimensional elasticity and to simplify them in the case where two dimensions are much bigger than one to get a reduced set of equations.

More formally, the asymptotic approach consists in writing the displacement field  $f$  of the elastic medium as an asymptotic expansion *w.r.t.* the small thickness  $h$ , and replace  $f$  in the framework of the 3D elasticity.

The goal is to obtain a set of equations prescribing the displacement field at a given thickness  $f_h$  that is convergent as  $h \rightarrow 0$  (in the sense of functional spaces). Although mathematically elegant, theorems ensuring the convergence of this method require strong conditions on the initial geometry and also on the boundary conditions, making them unpractical for more general cases. The reader may refer for instance to (Ciarlet, 2000) for a more thorough description of such asymptotic models.

**Limit models: example of the Föppl–von Kármán plate equations.** Another approach to reduce from 3D elasticity to a 2D model consists in making physical and geometrical assumptions on the behaviour of the elastic medium when the thickness is small. Such assumptions also aim at removing any explicit dependency *w.r.t.* the third dimension and obtaining a set of 2D equations.

As an example, I present in this paragraph the derivation of the Föppl–von Kármán model from 3D elasticity, following the energetic approach described in (Audoly and Pomeau, 2010). Although I will not make use of this model in the rest of this thesis, the derivation is quite didactic as an introduction to the common notations and hypotheses of plate mechanics. I also retrieve in the end an energy of a structure similar to that of other plate and shell models.

The reader is assumed to be familiar with linear elasticity, otherwise a short introduction can be found in Appendix A.

Let us consider an elastic plate of a uniform thickness  $h$  in its rest shape. As the plate is flat, we can define  $\Omega \subset \mathbb{R}^2$  such that the plate is embedded in  $\Omega \times [-\frac{h}{2}, \frac{h}{2}] \subset \mathbb{R}^3$ . We note  $f : \Omega \times [-\frac{h}{2}, \frac{h}{2}] \rightarrow \mathbb{R}^3$  its displacement field.

We also assume the elastic behaviour of the plate to be homogeneous, isotropic and Hookean, parametrised by its Young modulus  $E$  and its Poisson ratio  $\nu$ . In this case, using the Einstein summation convention, the elastic energy of the plate can be written as

$$E_{\text{el}} = \frac{1}{2} \iint_{\Omega} \int_{z=-h/2}^{h/2} \sigma_{ij} \epsilon_{ij} d\Omega dz. \quad (1.1)$$

with  $\sigma$  the stress,  $\epsilon$  the strain and  $i, j \in \{x, y, z\}$ . The goal here is to remove the dependency in  $z$  to obtain a reduced model described only by the mid-surface of the plate.

In the case of thin plates, a common hypothesis that can be made is the **Kirchhoff-Love** kinematic assumption which states that straight material segments orthogonal to the mid-surface stay straight and orthogonal to the mid-surface and have their length preserved when the plates deforms. This assumption implies that the displacement field has a specific form in function of the displacement of the mid-surface that we note  $g(x, y) = f(x, y, 0)$ , and we can explicit its components  $f_x$ ,  $f_y$  and  $f_z$  as

$$\forall (x, y) \in \Omega, \forall z \in [-h/2, h/2] : \begin{cases} f_{\alpha}(x, y, z) = g_{\alpha}(x, y) + z \underbrace{\partial_{\alpha} g(x, y)}_{\text{normal to the mid-surface}}, & \alpha \in \{x, y\} \\ f_z(x, y, z) = \underbrace{g(x, y)}_{\text{constant through the thickness}} \end{cases}. \quad (1.2)$$

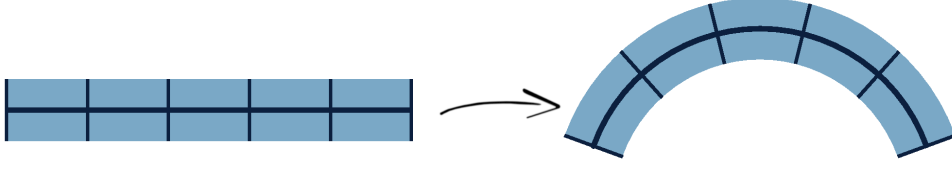


Figure 1.2: Deformation of a Kirchhoff-Love plate.

Then, under the assumption of *small displacements*, that is  $\epsilon$  is the Cauchy strain tensor, the prescribed displacement field yields  $\epsilon_{iz} = 0$ , which is then replaced in the elastic energy:

$$E_{\text{el}} = \frac{1}{2} \iint_{\Omega} \int_{z=-h/2}^{h/2} \sigma_{xx}\epsilon_{xx} + \sigma_{yy}\epsilon_{yy} + 2\sigma_{xy}\epsilon_{xy} d\Omega dz. \quad (1.3)$$

Note however that we still have some dependencies *w.r.t.*  $z$  in the remaining terms. To remove them, we first use the strain-stress relation for a 2D isotropic material (see Equation A.7), which yields

$$E_{\text{el}} = \frac{E}{2(1-\nu^2)} \iint_{\Omega} \int_{z=-h/2}^{h/2} \epsilon_{xx}^2 + \epsilon_{yy}^2 + 2\nu\epsilon_{xx}\epsilon_{yy} + 2(1-\nu)\epsilon_{xy}^2 d\Omega dz. \quad (1.4)$$

Then, we make the Cauchy strain explicit (Equation A.2) using the displacement field described in Equation 1.2. Audoly and Pomeau (2010) suggest to write the strain as

$$\epsilon(z) = \epsilon(z=0) + zD^2g \quad (1.5)$$

which can then be replaced in Equation 1.4, which gives

$$E_{\text{el}} = \frac{E}{2(1-\nu^2)} \iint_{\Omega} \int_{z=-h/2}^{h/2} \left( [\epsilon_{xx}^2 + \epsilon_{yy}^2 + 2\nu\epsilon_{xx}\epsilon_{yy} + 2(1-\nu)\epsilon_{xy}^2]_{z=0} + z(\dots) + z^2((\partial_{xx}^2g)^2 + (\partial_{yy}^2g)^2 + 2\nu\partial_{xx}^2g\partial_{yy}^2g + 2(1-\nu)(\partial_x\partial_yg)^2) \right) d\Omega dz. \quad (1.6)$$

The resulting integrand is now a quadratic form in  $z$  that can be resolved to get an energy depending only on the mid-surface. Note that we did not expand the term in  $z$  as it will be removed in the integration because of the symmetry. Finally, we obtain the formula

$$E_{\text{el}} = \underbrace{\frac{Eh}{2(1-\nu^2)} \iint_{\Omega} [\epsilon_{xx}^2 + \epsilon_{yy}^2 + 2\nu\epsilon_{xx}\epsilon_{yy} + 2(1-\nu)\epsilon_{xy}^2]_{z=0} d\Omega}_{E_s} + \underbrace{\frac{Eh^3}{24(1-\nu^2)} \iint_{\Omega} (\partial_{xx}^2g)^2 + (\partial_{yy}^2g)^2 + 2\nu\partial_{xx}^2g\partial_{yy}^2g + 2(1-\nu)(\partial_x\partial_yg)^2 d\Omega}_{E_b}. \quad (1.7)$$

The first term  $E_s$  involves only the in-plane strains, and thus can be identified as the



stretching energy. Its dependence *w.r.t.* the thickness is linear. In the second term, the first part is the Laplacian of the displacement field, while the second part is the Gaussian curvature (up to a multiplicative coefficient), allowing us to identify this term as the bending energy  $E_b$ .

So to sum up this paragraph, using the Kirchhoff-Love kinematic assumptions and the physical assumption of the small displacements enable the derivation of an energy depending only on the mid-surface of the plate.

**The Koiter shell model.** In this paragraph, I introduce the Koiter shell model following Ciarlet (2000, 2005).

Koiter (1966) also based his model on the two following assumptions:

- The Kirchhoff-Love kinematic assumptions that was presented in the previous part;
- A physical assumption based on the work of John (1965, 1971) that states that in the limit of thin shells, the stress is parallel to the mid-surface, that is, *there is no internal shearing*.

His resulting work leads to a similar energy to that of Föppl–von Kármán, but is based on the difference of metrics on the mid-surface to measure the stretching and the bending energies. Note that here, since we are considering shells, the rest state may be non-trivial, and the related quantities will be denoted by a bar over the variable name  $\bar{\bullet}$ .

As in the previous section, we consider a shell of thickness  $h$ . The deformed shape of the mid-surface is described by the function  $r : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ . Its rest shape is similarly described by  $\bar{r} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$  (to make the link with the previous section, we have  $r = \bar{r} + g$ ). If we assume that  $r$  is smooth enough ( $\mathcal{C}^2$ ) and that the tangent plane is always defined, *i.e.*  $\forall s \in \Omega, \partial_1 r(s)$  and  $\partial_2 r(s)$  are not collinear, then we can define respectively  $\mathbf{A}$  and  $\mathbf{B}$ , the first and second fundamental forms of the surface  $r$ ,

$$\mathbf{A} : s \mapsto Dr(s)^\top Dr(s) \quad \text{and} \quad \mathbf{B} : s \mapsto -Dr(s)^\top Dn(s) \quad (1.8)$$

with  $n$  the normal defined by

$$n : s \mapsto \frac{\partial_1 r(s) \times \partial_2 r(s)}{\|\partial_1 r(s) \times \partial_2 r(s)\|}. \quad (1.9)$$

Further details are given in Appendix B.1. The counterparts for the rest shape  $\bar{r}$  are noted  $\bar{\mathbf{A}}$  and  $\bar{\mathbf{B}}$ . Then, if we still assume the shell to be linearly elastic, the Koiter's shell energy is defined as

$$E_{\text{el}} = \iint_{\Omega} \frac{1}{2} \left( \frac{h}{4} \|\mathbf{A} - \bar{\mathbf{A}}\|_{\bar{\mathbf{A}}} + \frac{h^3}{12} \|\mathbf{B} - \bar{\mathbf{B}}\|_{\bar{\mathbf{A}}} \right) \sqrt{\det \mathbf{A}} \, d\Omega \quad (1.10)$$

with  $\|\bullet\|_{\bar{\mathbf{A}}}$  the matrix norm <sup>1</sup> defined by

$$\|\bullet\|_{\bar{\mathbf{A}}} = \frac{E\nu}{1-\nu^2} \left( \text{tr}(\bar{\mathbf{A}}^{-1}\bullet) \right)^2 + \frac{E}{1+\nu} \text{tr} \left( (\bar{\mathbf{A}}^{-1}\bullet)^2 \right). \quad (1.11)$$

This rewriting from (Ciarlet, 2005)'s formulation is detailed in Appendix C.

In other words, the energy measures the stretching through the difference of the first fundamental forms and the bending through the difference of the second fundamental forms under the metric induced by the parametrisation of the rest shape<sup>2</sup>.

As a side note, if we relax the Kirchhoff-Love assumption on the orthogonality of the material segments, we fall into the Cosserat type shell models where a director field attached to the shell becomes another unknown. These models are better suited for shells with a "moderate" thickness where internal shearing cannot be neglected. The Koiter model generalises to the one of Naghdi (1972) with such a relaxation.

### 1.1.2 Plates and shells in Computer Graphics

In Computer Graphics, the introduction of shells goes back to the seminal work of Terzopoulos et al. (1987), who first proposed to use physical simulation tools for animation. They formulate an energy based on the differences in the fundamental forms, which somehow resembles the Koiter model, although theirs is not based on any physical analysis. Then, they discretise the geometric quantities using standard finite differences on regular grids.

Their technique has been used in the early works on cloth simulation (Carignan et al., 1992), but has been found to be unstable due to the discretisation in space, and the implicit/explicit integration scheme where the elastic forces were evaluated explicitly.

Note that their formulation, and most of subsequent "qualitative" shell models that were derived later in Computer Graphics, depart from that of Koiter in 2 ways:

- The stretching modulus and the bending modulus introduced are two distinct parameters of the model, while they both should be derived from the thickness  $h$ , the Young modulus  $E$  and the Poisson ratio  $\nu$ . However, this choice gives more controllability on the resulting material, allowing the user to tune these parameters so as to obtain the expected visual behaviour or to approximate non-isotropic material for which there is no trivial equivalent  $E$  and  $h$ ;
- More subtle, the stretching term and the bending terms are decoupled, while in the Koiter formulation, the bending depends on the in-plane deformation. Resulting formulations are easier to evaluate, and this allows to combine differing stretching

<sup>1</sup>because  $\bar{\mathbf{A}}$  is symmetric definite positive as  $Dr$  is full-rank.

<sup>2</sup>Ciarlet (2000), states this was the "best two-dimensional" model (Chapter 7).

and bending models. Similarly to the previous point, this decoupling may also help to approximate more complex materials.

**"Qualitative" shell models.** Following work focused on formulating bending measures well-suited for triangle meshes, and are less expensive than finite element methods. These bending models are then coupled with an in-plane stretching model (mass-spring network, 2D linear elasticity...).

With the idea that bending stems from the variation of the normals along the surface, an early discrete counterpart was to measure this variation with the dihedral angle at the edge between the faces, *i.e.* variation between the (constant) face normals.

An early formulation has been given in (Baraff and Witkin, 1998), and was then popularised by the simultaneous works by Bridson et al. (2003) and Grinspun et al. (2003). The latter linked this hinge energy to a discretisation of the Willmore energy, based on the square of the mean curvature. I will come back in detail to this model right below, as this is one of the models I will use.

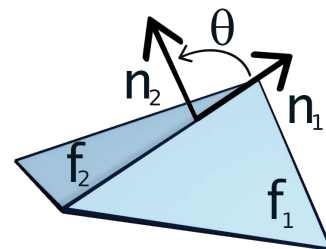


Figure 1.3: Dihedral angle between two faces.

Further works focussed on approximating the curvature in different manners. Choi and Ko (2002) estimated the curvature based on the distances within stencils of size 2 on regular grids, while Bergou et al. (2006); Wardetzky et al. (2007) approached the mean curvature of the Willmore energy using the Laplacian.

More work on the simulation of shells in Computer Graphics can be cited. However, they do not introduce new shell models *per se* but focus on improving the speed or the accuracy of current shell models and fall out of the scope of this paragraph. Among them, we can think of adaptive remeshing (Narain et al., 2012), GPU simulation (Schmitt et al., 2013; Tang et al., 2018; Li et al., 2020b) or multi-grid simulation (Xian et al., 2019; Wang et al., 2018) for improving the computation time and of energy projection (Goldenthal et al., 2007), alternative parametrisation (Weidner et al., 2018), cloth untangling (Buffet et al., 2019) and data-driven model fitting (Miguel et al., 2012; Clyde et al., 2017) for improving accuracy.

**Grinspun et al. (2003)'s discrete shells model.** The discrete shells model of Grinspun et al. (2003) is a nodal model, that is the degrees of freedom are positions representing the nodes of the surface mesh. Consider a surface triangular mesh  $\mathcal{M} = \{\mathcal{E}, \mathcal{F}\}$  of  $n_v$  vertices with  $\mathcal{E} \in \llbracket 1, n_v \rrbracket^{2n_e}$  the set of edges and  $\mathcal{F} \in \llbracket 1, n_v \rrbracket^{3n_f}$  the set of triangular faces. We denote the positions of the nodes of the deformed surface by  $x \in \mathbb{R}^{3n_v}$  and those of the rest shape by  $\bar{x} \in \mathbb{R}^{3n_v}$ .

The shell energy is composed of one term for the bending,

$$E_b = \frac{k_b}{2} \sum_{e \in \mathcal{E}_{int}} \frac{3L_e^2}{A_e} (\theta_e - \bar{\theta}_e)^2, \quad (1.12)$$

and of two terms for the in-plane deformation,

$$E_l = \frac{k_l}{2} \sum_{e \in \mathcal{E}} \frac{(L_e - \bar{L}_e)^2}{\bar{L}_e} \quad (1.13a)$$

$$\text{and } E_a = \frac{k_a}{2} \sum_{f \in \mathcal{F}} \frac{(A_f - \bar{A}_f)^2}{\bar{A}_f}, \quad (1.13b)$$

with  $L_e$  the length of the edge  $e$ ,  $A_f$  the area of the face  $f$  and  $A_e$  half of the area of the adjacent faces to the internal edge  $e$ . For the sake of readability, the dependence to  $x$  and  $\bar{x}$  are made implicit. The coefficients  $k_b$ ,  $k_l$  and  $k_a \in \mathbb{R}$  are stiffnesses controlling the material behaviour of the shell.

Grinspun et al. (2003) state that the bending term is a discretisation of the Willmore energy  $\iint \kappa_{mean}^2 ds$  measuring the integral of the square of the mean curvature for a surface that deforms isometrically. They furthermore note that their discretisation is consistent *w.r.t.* changes in the topology due to the normalisation through the lengths and the areas, but not convergent under refinement. Grinspun et al. (2006) confirmed this observation, and introduced the *mid-edge* operator that we will describe more thoroughly right after.

The two in-plane terms have similar structures, but do not appear to derive from a continuous formulation. The term in length is the same as the stretching term of a 2D elastic rod discretised by finite differences. However, the extension to a surface and combined with a term on the areas, is not properly justified, although the qualitative behaviour of the stretching can be reproduced.

Nonetheless, the simplicity of this model made it popular in the Computer Graphics community (Wardetzky et al., 2007; Umetani et al., 2011). This model will also be evaluated in Chapter 3, and will be used in Chapter 5 for shell inversions, in the continuity of the work of Casati (2015).

**Mechanical models and discrete geometry.** Since (Terzopoulos et al., 1987)'s evaluation of the fundamental forms on grids using finite differences, much work has been done in discrete geometry to be able to evaluate these quantities over triangular meshes.

Consider a smooth surface  $r : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$  with enough regularity<sup>3</sup> and an approximating mesh  $\mathcal{M} = \{\mathcal{E}, \mathcal{F}\}$ .

Even if the smooth surface is parametrised, its discrete counterpart might not have a similar parametrisation. In addition, such parametrisation may not be trivial to compute.

---

<sup>3</sup>Refer to the § on Koiter's shell model, or to Appendix B.1

As such, it is more convenient to use a local parametrisation. On a triangular mesh, a natural parametrisation can be done triangle by triangle.

Let  $(x_i, x_j, x_k)$  be 3 vertices defining a face. Then a local mapping can be defined by

$$\begin{aligned} r_{ijk} : \quad \mathcal{T} &\rightarrow \mathbb{R}^3 \\ (u, v) &\mapsto x_i + u(x_j - x_i) + v(x_k - x_i), \end{aligned} \quad (1.14)$$

with  $\mathcal{T}$  the triangle of  $\mathbb{R}^2$  defined by the points  $(0, 0)$ ,  $(1, 0)$ ,  $(0, 1)$ . Under this parametrisation, the discrete fundamental forms can also be defined over the triangles (Chen et al., 2018) by the symmetric matrices

$$\mathbf{A}_{ijk} = \begin{bmatrix} \|x_j - x_i\|^2 & (x_j - x_i)^\top(x_k - x_i) \\ (x_j - x_i)^\top(x_k - x_i) & \|x_k - x_i\|^2 \end{bmatrix} \quad (1.15a)$$

$$\mathbf{B}_{ijk} = \frac{1}{2} \begin{bmatrix} (n_{jk} - n_{ik})^\top(x_i - x_j) & (n_{jk} - n_{ik})^\top(x_i - x_k) \\ (n_{jk} - n_{ik})^\top(x_i - x_k) & (n_{jk} - n_{ij})^\top(x_i - x_k) \end{bmatrix} \quad (1.15b)$$

with  $n_{ij}$  the normal at the mid-point of the edge  $ij$ , the so-called *mid-edge normal*.

**The mid-edge normal.** A simple way to evaluate the mid-edge normal, whether on the edges or at the vertices, is to compute a weighted average of the triangles normals. However, Weischedel (2012); Meek and Walton (2000) explain that these discrete normals converge "linearly" towards continuous normals as the mesh is refined towards the smooth surface.

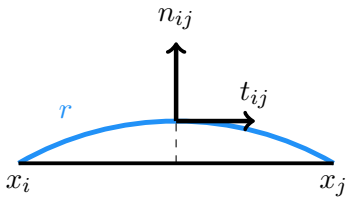


Figure 1.4: Illustration of the mid-edge normal. Reproduced from Weischedel (2012).

Introduced by Grinspun et al. (2006), a more accurate evaluation can be made with the following heuristic. Assuming that the surface is smooth enough, an edge of the mesh can be seen as a centred finite difference of a tangent to the surface at the mid-point, as depicted in Figure 1.4. That is, the convergence is "quadratic" as the mesh is refined. Then, using the fact that the normal has to be orthogonal to this tangent at the mid-point, a mid-edge normal can be defined by an angle of rotation around the edge, which therefore introduces a new degree of freedom per edge to the system.

Compared to the classical way of evaluating normals, which consists in averaging the normals of the neighbouring faces, this new formulation is heavier due to these additional degrees of freedom. However, it seems to exhibit a better convergence behaviour, as we show in section 3.5.3.

**Chen et al. (2018)’s Discrete Koiter model.** The discrete equivalents described in the previous section can be reinjected in the Koiter’s shell energy to get an equivalent defined on the triangular mesh (Chen et al., 2018):

$$E_{\text{el}} = \sum_{ijk \in \mathcal{F}} \frac{1}{4} \left( \frac{h}{4} \|\mathbf{A}_{ijk} - \bar{\mathbf{A}}_{ijk}\|_{\bar{\mathbf{A}}_{ijk}} + \frac{h^3}{12} \|\mathbf{B}_{ijk} - \bar{\mathbf{B}}_{ijk}\|_{\bar{\mathbf{A}}_{ijk}} \right) \sqrt{\det \mathbf{A}_{ijk}}, \quad (1.16)$$

with  $\|\bullet\|_{\bar{\mathbf{A}}_{ijk}}$  the matrix norm defined previously.

An implementation has been made available [online](#) by [E. Vouga](#) under the name LIBSHELL. We will use this name to refer to this code, while DISCRETE SHELL will refer to our implementation of [Grinspun et al. \(2003\)](#)’s original model. LIBSHELL implements both Saint-Venant-Kirchhoff and Neo-Hookean for the material model, and for the second fundamental form, the mid-edge normals are computed either using the average formulation (MidEdgeAverage variant) or using the additional degree of freedom (MidEdgeSin and MidEdgeTan variants). As for the original DISCRETE SHELL model, we will come back to this model in Chapters 3 and 5.

## 1.2 Contact and friction in Computer Graphics: models & simulation

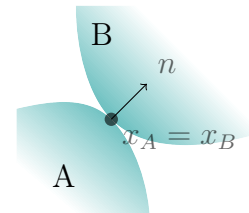
In this section, I start by recalling the Signorini-Coulomb law, widely used to model dry frictional contact at the macroscopic scale. Then I review the different methods used in Computer Graphics to handle contact and friction in simulations. Some methods are original ideas of the Computer Graphics community, while some draw from the work of the Non-smooth Mechanics community.

I will briefly introduce some non-smooth mechanics concepts through works that make the connection between the Computer Graphics and Non-smooth Mechanics communities. To have a more thorough view of non-smooth mechanics, including treatment of impacts and dry friction the reader is strongly encouraged to refer to works of this community, such as the seminal work of [Moreau \(1988, 2000\)](#) or the book of [Acary and Brogliato \(2008\)](#).

### 1.2.1 Signorini-Coulomb law

**Signorini conditions.** Physically, objects do not inter-penetrate, and obviously that is the foremost principle to follow when simulating objects in interaction. The formalisation of this principle can be done using the Signorini conditions ([Signorini, 1959](#)).

Consider two objects  $A$  and  $B \subset \mathbb{R}^3$ . Let us track the contact happening between the points  $x_A \in \partial A$  and  $x_B \in \partial B$  on the surface of  $A$  and  $B$  respectively, that is at the time of impact  $t_0$  their positions coincide :  $x_A(t_0) = x_B(t_0)$ . We also assume the objects to be smooth enough around  $x_A$  and  $x_B$  such that outer normals to the surfaces can be defined. For the sake of simplicity,  $A$  and  $B$  are also assumed to be convex.



We can define the (oriented) gap function by  $g(t) = x_B(t) - x_A(t)$ , that is the relative position of  $x_B$  w.r.t.  $x_A$ . Let us also note  $n \in \mathbb{R}^3$  the outer normal at  $x_A$  (that is oriented towards  $B$ ). The non-penetration condition between these two points writes as

$$\forall t : \underbrace{g(t)^\top n(t)}_{g_N(t)} \geq 0. \quad (1.17)$$

The convexity of the objects is used here to ensure that we have at most one contact between  $A$  and  $B$ , and that the condition above is sufficient. It can naturally be generalised to other cases by considering a restrained time interval around the time of impact, several contact points etc.

Physically, this constraint is imposed by a reaction force between the objects that is applied *if and only if* they are in contact. Let  $r$  be the reaction force applied by  $A$  on  $B$ , and  $r_N = r^\top n$  its normal component. The non-penetration along with the existence of the reaction force can be expressed by the following complementarity condition, the so-called *Signorini conditions*,

$$\forall t : 0 \leq g_N(t) \perp r_N(t) \geq 0. \quad (1.18)$$

At this stage, a lot of remarks can already be made. The first and the least important would be to notice that the formulation above seems to be dissymmetric as all the vectorial quantities are oriented from  $A$  to  $B$ . However, the condition on the positions is the same one way or another, and the Newton third law imposes the symmetry on the reaction forces.

To continue on the reaction force, the Signorini conditions only constrain the normal component  $r_N$ , leaving the tangential component  $r_T$  free. However, in the absence of other phenomena considered at the contact point, such as friction or adhesion, there is no reason for a tangential force to exist, and thus  $r_T = 0$ .

The constraint on the positions also imposes conditions on the velocity. At contact, the normal relative velocity  $u_N(t) = \frac{dg}{dt}(t)^\top n(t)$  must always be positive. Otherwise, this would implied that  $g_N(t^+) < 0$ , which violates the non-penetration constraint.

Note also that  $u_N$  can be discontinuous at the time of impact as  $u_N(t_0^-)$  may be strictly negative (the two objects are moving towards each other) and  $u_N(t_0^+)$  has to be positive. Mathematically, [Moreau \(1988\)](#) formulates that as a bounded variation of  $u_N$ , that is  $\forall t$ ,

$u_N$  has a limit value to the left  $u_N(t^-)$  and a limit value to the right  $u_N(t^+)$ .

Finally, we can note that this formulation may be complemented by an impact law. Indeed, when impacting, and elastic object bounces back due to its internal compression and expansion. However, as the object stiffens and tends towards a rigid solid, the timescale of this phenomena decreases, making the simulation of this behaviour unsuitable, or impossible for rigid bodies. In this thesis, we do not consider impact laws as our methods either apply to volumic deformable objects which do not need an impact law or to clothes where an inelastic impact is appropriate.

**Coulomb dry friction model.** A popular way to model dry friction at the macroscopic scale is the Coulomb law (Coulomb, 1781), also called the Amontons-Coulomb law as the work of Coulomb synthesises previous work on the friction, including mainly (Amontons, 1699).<sup>4</sup>

The Coulomb law states that the tangential friction force  $r_T$  is bound to the normal reaction  $r_N \geq 0$  by the conditions

$$\begin{cases} \|r_T\| \leq \mu_S r_N & \text{if } u_T = 0 & \text{(static regime)} \\ \|r_T\| = \mu_D r_N & \text{otherwise} & \text{(dynamic regime)} \end{cases} \quad (1.19)$$

with  $\mu_S \in \mathbb{R}^+$  the *static friction coefficient* and  $\mu_D \in \mathbb{R}^+$  the *dynamic friction coefficient* (often lower than  $\mu_S$ ), two coefficients depending only on the materials of the contacting objects.

Note that the principle of maximum dissipation applied to these inequalities implies that in the dynamic regime,  $r_T$  is collinear and of opposed direction to the velocity  $u_T$  so as to maximise the dissipation of the kinetic energy.

In our work, we consider a single coefficient as it was sufficient to capture the stick/slip threshold behaviour in our dynamic simulations (Chapter 2), and we did not aim for more accuracy.

We thus consider the following set of equations,

$$\begin{cases} \|r_T\| \leq \mu r_N & \text{if } u_T = 0 \\ r_T = -\mu r_N \frac{u_T}{\|u_T\|} & \text{otherwise,} \end{cases} \quad (1.20)$$

*i.e.* the reaction force  $r$  belongs to a cone oriented by the normal  $n$  and of aperture  $\mu$ , the so-called *Coulomb cone* that we note  $\mathcal{K}_\mu(n)$ .

Finally, we denote by Signorini-Coulomb law the combination of the Coulomb law described right above with the Signorini conditions

---

<sup>4</sup>Other historical contributions include the works of Da Vinci, Euler, De Bélidor & Desaguliers.



$$\begin{cases} \text{if } u_N > 0 & r = 0 \\ \text{else } u_N = 0 & \text{and } \begin{cases} \|r_T\| \leq \mu r_N & \text{if } u_T = 0 \\ r_T = -\mu r_N \frac{u_T}{\|u_T\|} & \text{otherwise} \end{cases} \end{cases} \quad (1.21)$$

which is illustrated in Figure 1.5.

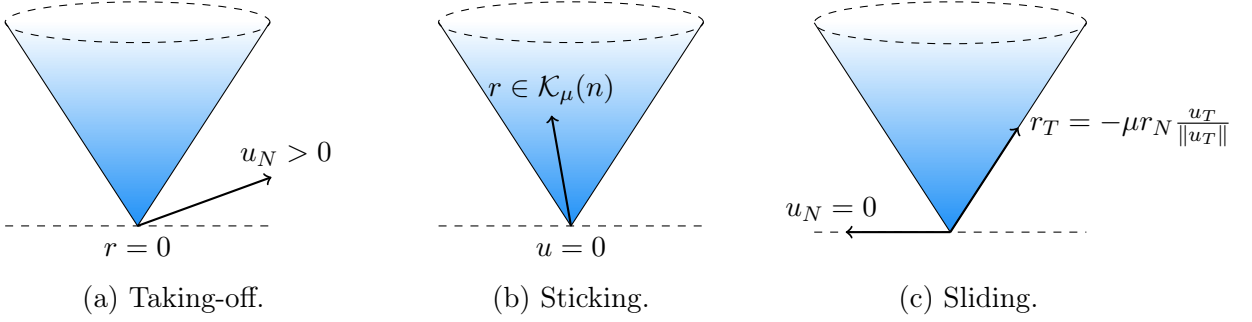


Figure 1.5: Cases of the Signorini-Coulomb law.

Looking at the formulations, we see the difficulties that arise in the simulation to apply this law. For the Signorini (contact) part, applying the complementarity 1.18 would in theory require to detect the exact time of impact realising the zero of the gap function, which is not trivial. Regarding the friction part of the equation, the force threshold and the direct dependence on the degrees of freedom through the tangential relative velocity  $u_T$  are non trivial to incorporate in a time integration scheme.

Note that in the following, we will not discuss the collision detection problem, unless it is interleaved to the method to solve the frictional contact. Instead, we will focus on how the Signorini-Coulomb problem is solved at contacts. In our algorithms, we use a simple proximity query to detect if a vertex is involved in a collision or not.

### 1.2.2 Penalty-based/smooth methods.

Due to its non-smoothness, solving the Signorini-Coulomb law is non trivial as well as computationally expensive. We review here methods that aimed at formulating simpler frictional contact problems at the cost of losing physical realism.

**Springs.** The easiest way to try to prevent penetration between objects is to add repulsion fields modelled as springs aligned in the normal direction. They are simple to add to any system, and computationally cheap and so have been used since the first simulation works such as in rock mechanics (Cundall, 1971) or in Computer Graphics (Moore and Wilhelms, 1988). Similarly, friction is introduced using tangential springs either in a simple manner to produce viscous friction, or with thresholds using previous values of the velocity and normal force to approximate the dry friction effect.

However, such simple methods do possess several drawbacks. The repulsion force is non-zero over a certain thickness rather than just at the contact and the stiffness of the spring must be adjusted to ensure the field can counter the other forces at play and to try to prevent penetration. Moreover, the higher the stiffness is, the smaller the time-step has to be, and it must also be small enough to ensure the contact is detected *w.r.t.* the width of the field and the velocities of the objects. Finally, multiple collisions are likely to generate vibrations. All in all, there are many parameters to tune, and non-penetration is not guaranteed.

**Impact zones.** Subsequent works focused on alleviating or avoiding some of these drawbacks. [Provot \(1997\)](#) proposed an iterative process to construct *impact zones* that merge neighbouring contact points into rigid zones and solved the contact problem in a specific way to avoid interpenetration within these zones. This technique has been extended by [Bridson et al. \(2002\)](#) with a prediction/correction scheme that corrects the contact-free time-steps a posteriori with explicit impulses that prevent penetration. The friction is also added a posteriori with an explicit dependence on the impulses. Finally, [Harmon et al. \(2008\)](#) also extended the impact zones method by allowing vertices to slide within the zone, reducing the stiffening effect.

**Event-driven simulation.** Traditional time-stepping methods such as Euler schemes are widely used for the simplicity of the integration scheme that approximates the continuous dynamic equation over regular time intervals. However, handling non-smooth events, such as impacts, is non-trivial as those have to be captured within a time-step and consequently integrated to the numerical scheme.

As such, *event-driven* simulations aim at performing the integration of the smooth dynamics separately from the event-handling. This allows the use of adaptive time-steps, provided that the events are first detected and then explicitly handled.

In Computer Graphics, [Harmon et al. \(2009\)](#) proposed such a method, combined with an asynchronous integration scheme that is able to treat contacts as they appear. Their method robustly handles complex contact scenarios, but is computationally expensive. More generally, event-driven schemes are robust and interesting when the number of impacts remains low, but their cost quickly increases with the number of events. I refer the reader to ([Acary and Brogliato, 2008](#), Chapter 8) for a more general explanation on event-driven schemes for Lagrangian systems.

An interesting variation that incorporates elements from event-driven methods in a time-stepping scheme has been proposed in ([Li et al., 2020a](#)). Similarly to *Interior Point* optimisation methods, they enforce the non-penetration constraints with log-barriers instead of springs. Then, in the minimisation to compute the system state at the next time-step, a *Continuous Collision Detection* (CCD) method is applied during the line-

search to compute a time of impact (TOI). This TOI serves as an upper bound, under which all applied steps are guaranteed to be intersection-free. This allows them to robustly produce simulations without any penetration. However, to fit in their framework, the friction law is smoothed, and as in [Brown et al. \(2018\)](#), that I will describe below, the tangential force is decoupled of the normal force produced by the barriers so that it can be formulated as a smooth potential that is added to the system.

### 1.2.3 Non-smooth solvers

In this section, I start by briefly explaining the work of [Moreau \(1988\)](#) for the numerical integration of the non-smooth law. I then present methods from Computer Graphics aiming at solving the non-smooth dynamical problem.

Consider a dynamical problem with  $q \in \mathbb{R}^m$  the degrees of freedom. Using a linear integration scheme (explicit Euler or linearised implicit Euler), the unconstrained discrete dynamical system can be put under the form

$$Mv + f = 0 \tag{1.22}$$

with  $M$  the inertia matrix,  $v = \frac{dq}{dt}(t^+)$  the generalised velocities at the next time-step and  $f$  the generalised forces. The work of [Moreau \(1988\)](#) allows us to add the Signorini-Coulomb law in the form of an impulse constrained along the local velocities. If we assume  $n_C$  contacts happening at positions  $x_c \in \mathbb{R}^3$ , the system reads

$$\begin{cases} Mv + f = H^\top r \\ u = Hv \\ \forall \text{ contact } c : (u_c, r_c) \in \mathcal{C}_{\mu_c} \end{cases} \tag{1.23}$$

that we will call the *Discrete Frictional Contact Problem* (DFCP) ([Cadoux, 2009](#); [Daviet, 2016](#); [Bertails-Descoubes, 2017](#)). In the system, we retrieve  $u \in \mathbb{R}^{3n_C}$  the local velocities of the contact points that are linked to the general velocities by the change of basis defined by the matrix  $H = [\partial_j x_c]_{c,j} \in \mathbb{R}^{3n_C \times m}$ . The vector  $r \in \mathbb{R}^{3n_C}$  here has the dimension of an impulse but by abuse of language, we will refer to it as the reaction force as it satisfies the same constraints. The set  $\mathcal{C}_{\mu_c}$  denotes the feasible set of velocities and forces of the Signorini-Coulomb law that we described in [Equation 1.21](#). That is, the reaction forces are a new set of unknowns and a constraint ties it to the set of unknown velocities.

Note that this system may in some cases possess no solution (see *e.g.* the Painlevé's paradox, ([Klarbring and Pang, 1998](#); [Andersson et al., 2016](#))) or multiple solutions ([Moreau, 2006](#); [Blumentals et al., 2016](#); [Agwa and Pinto da Costa, 2021](#)). But we will not discuss about this topic in this thesis, and refer the reader to the references mentioned above for more results and discussions.

**Contact problem as a LCP.** In Computer Graphics, the first non-smooth methods have been introduced by Baraff (1991, 1994). For the case of solids, he reformulated the equation of motion with the Signorini conditions as a *Linear Complementarity Problem* (LCP) of the form

$$\text{Find } f \text{ such that: } 0 \leq f \perp Af \geq 0. \quad (1.24)$$

This class of problems is well known and several methods dedicated to solve them exist (Cottle et al., 2009). He also proposed alterations to the LCP to try to introduce Coulomb friction.

Several works in Computer Graphics continued with this idea of solving the contact problem through LCPs or equivalent formulations (Raghupathi and Faure, 2006; Redon et al., 2002) with modifications to incorporate friction (Kaufman et al., 2005).

**Coulomb cone linearisation.** A popular simplification of the Signorini-Coulomb law consists in replacing the Coulomb cone  $\mathcal{K}_\mu$  with a linear approximation in the form of a pyramid, first introduced by Stewart and Trinkle (1996). The Coulomb constraint is then replaced by a set of linear constraints that read as

$$\forall t \in \text{a finite set } \{t_i \in \mathbb{R}^3 | n^\top t_i = 0, \|t_i\| = 1\} : \mu r_N - t_i^\top r_T \geq 0. \quad (1.25)$$

This linearised constraint is easier to handle, and allows the formulation of more computationally efficient methods. For instance, Silcowitz et al. (2009) reformulated the DFCP 1.23 as a simple *Non-linear Complementarity Problem* (NCP, Equation 1.24 with the right term being non-linear) that can be solved by non-smooth minimisation involving the *Fischer-Burmeister* function. Similarly, (Otaduy et al., 2009) proposed a *Mixed Linear Complementarity Problem* (MLCP, Equation 1.24 with a subset of the indices removed of the orthogonality). They solve this MLCP by converting it in a sequence of LCP solved by the *Projected Gauss-Seidel method* until convergence. Finally, Kaufman et al. (2008) also designed in this case a staggered projection scheme for interactive applications that iteratively refines the normal and the tangential components of the reaction  $r$ . Note that this linearisation naturally introduces anisotropic artefacts as the friction depends on (arbitrary) directions (Acary and Brogliato, 2008).

Another related simplification is the convexification of the constraints described in (Anitescu and Hart, 2004) and used by Mazhar et al. (2015). This allows them to solve a sub-problem using Nesterov accelerated gradient to compute the reaction force to plug back in the DFCP.

Contrary to these approaches based on simplifications of the constraints, Bertails-Descoubes et al. (2011); Daviet et al. (2011) have shown that treating the exact problem (without linearisation nor convexification) is possible for Computer Graphics problems.

**Iterative methods & functional formulations.** These methods have been mentioned in the previous paragraphs as they are able to treat the simplified non-smooth problems, but they are also suitable to solve the DFCP with the exact Signorini-Coulomb's law.

Iterative methods, such as projected Gauss-Seidel, consist in sweeping over the local problems, here the frictional contact at each contact point, until convergence. They are simple and robust, yet may fail in some conditions where the convergence is slow *e.g.* when the constraints oppose each other, and also rely on having simple local steps.

On the other hand, functional formulations like non-smooth Newton aim at solving all the contact problems at once by finding the minimum of a carefully designed function. These methods can yield better convergence to accurate results, but, because of the higher cost per iteration compared to iterative methods, may be less attractive for applications requiring low/moderate accuracy. As an example of functional method, the popular formulation of Alart and Curnier (1991) has been used by Bertails-Descoubes et al. (2011) to simulate fibre assemblies in frictional contact. More recently, Macklin et al. (2019) proposed another non-smooth fixed-point Newton method with preconditioning.

Daviet et al. (2011) also proposed a hybrid method, where the local step of the Gauss-Seidel method is solved through a small non-smooth Newton method using a modified Fischer-Burmeister function. It has been used and adapted to simulate nodal systems in the case of hair (Kaufman et al., 2014) and clothes (Li et al., 2018a).

Note that these methods are similar to that developed in Non-smooth Mechanics community as they aim at accurately solving the DFCP. However, the forementioned methods were designed to simulate deformable objects while "classic" non-smooth methods focus on the interaction between rigid bodies. The reader may refer to (Bertails-Descoubes et al., 2011) for an evaluation of the latter.

**Local/global schemes.** The local/global schemes are integration schemes that solve the dynamics by iterating over two steps called *local* and *global*. The local step is designed to be composed of many problems that can be solved in parallel. In addition, these sub-problems are constructed to be very simple to solve. The global step then gathers all the sub-results to compute a new better estimate of the system that is either used to refine the next local sub-problems, or returned if the estimation is deemed accurate enough.

As an example, in the *Projective Dynamics* method (*PD*), that I will detail more in Chapter 2, the local step consists in solving the dynamics considering "for each local force", as if it was alone *e.g.* elasticity along one edge, bending per stencil etc. Then the global step combines all these sub-results to reconstitute the dynamics of the whole system subject to all the forces.

With frictional contact, the structure may look similar to the iterative methods described above, although here the local steps may not focus solely on solving the frictional contact constraints. I present below local/global schemes that deal with contact and

friction as related works to our contribution (Ly et al., 2020).

*Position Based Dynamics* (Müller et al., 2007) and its extension *X-PBD* (Macklin et al., 2016) handled contact by formulating local problems dedicated to push vertices considering only the non-penetration criterion. However, friction was added as an *a posteriori* correction. Macklin et al. (2020) have recently proposed in a study where they compare the strengths and weaknesses of *X-PBD* and *PD* to add frictional contact in both framework through penalties.

Overby et al. (2017) used a similar idea for the *Alternating Direction Method of Multipliers* in the form of a potential valued in  $\{0, \infty\}$  to fit in this optimisation framework. They extended their method in (Brown et al., 2018) to use the normal force from the previous iteration to formulate a potential enabling the implicit computation of the tangential reaction. By doing so, they do interfere with the optimisation as the friction potential has to be redefined explicitly at each iteration. However, at convergence they correctly recover the Signorini-Coulomb law.

Finally, Daviet (2020) designed an ADMM algorithm where the local step is dedicated to the frictional contact while being simple. It is quite close to our work that I detail in Chapter 2. We modified the Projective Dynamics method broadly described above to include frictional contact in a similar local step.

## Chapter 2

# Projective Dynamics with contact and friction

In this chapter, I present our contribution originally entitled *Projective Dynamics with contact and friction* and called *Projective Friction* in the following.

I start by recalling in a first section the original Projective Dynamics method (Bouaziz et al., 2014), designed to produce fast, stable, and moderately accurate simulations of deformable objects, and which may reach higher accuracy with more time budget. However, the way contacts are treated in this algorithm leads to an increased computation time. Moreover, dry frictional contact is handled through a post-processing step. We (Ly et al., 2020) have proposed a new way to treat both contacts and dry friction within this framework that I present in a second section, along with some qualitative and quantitative evaluations.

## 2.1 The Projective Dynamics method

In this section, I describe the Projective Dynamics method in general. For more details, such as other energies in their "Projective Dynamics" form, the reader may refer to the original article (Bouaziz et al., 2014).

### 2.1.1 Dynamics of a nodal system

Consider a system described by  $n_v$  nodes (vertices), et let us note  $x \in \mathbb{R}^{3n_v}$  their positions and  $v \in \mathbb{R}^{3n_v}$  their velocities. Assuming a lumped mass model, we attach to each node a mass  $m_i \in \mathbb{R}_*^+$ . For instance, if the nodes belong to a surface mesh, the mass of each node can be taken as one third of the area of the neighbouring faces times the surface density. This system is also assumed to be submitted to a set of internal forces  $\{f_j\}_j$  and constant external forces  $f_{ext}$ . Under these conditions, the dynamics are given by the Newton second law

$$M\dot{v} = f_{ext} + \sum_j f_j(x, v, t) \quad (2.1)$$

with  $M = \begin{bmatrix} m_1 & 0 & \\ 0 & \ddots & 0 \\ & 0 & m_{n_v} \end{bmatrix}$  the mass matrix and  $\dot{v}$  the time-derivative of the velocity (the acceleration).

Then, we can use the Euler implicit scheme to discretise this equation in time with a constant time-step  $h$ . If we denote the system state at the  $n$ -th time-step with superscript  $n$ , the discrete system may take one of the following forms, depending on whether we choose the positions  $x^{n+1}$  or the velocities  $v^{n+1}$  as the unknowns (they are linked by the relation  $x^{n+1} = x^n + hv^{n+1}$ )

$$Mx^{n+1} - M \underbrace{(x^n + hv^n - h^2 M^{-1} f_{ext})}_{s^n} - h^2 \sum_j f_j \left( x^{n+1}, \frac{x^{n+1} - x^n}{h}, t^{n+1} \right) = 0 \quad (2.2a)$$

$$Mv^{n+1} - M \underbrace{(v^n - hM^{-1} f_{ext})}_{\tilde{s}^n} - h \sum_j f_j(x^n + hv^{n+1}, v^{n+1}, t^{n+1}) = 0. \quad (2.2b)$$

Finally, if we assume that all the forces derive from potentials, that is  $f_j(x) = -\frac{dW_j}{dx}(x)$ , then the equations 2.2a and 2.2b can be interpreted respectively as the first order optimality conditions of the minimisation problems

$$x^{n+1} = \operatorname{argmin}_x \frac{1}{2} \|x - s^n\|_M^2 + h^2 \sum_j W_j(x) \quad (2.3a)$$



$$v^{n+1} = \underset{v}{\operatorname{argmin}} \frac{1}{2} \|v - \tilde{s}^n\|_M^2 + \sum_j W_j(x^n + hv) \quad (2.3b)$$

with  $\|\bullet\|_M^2 = \bullet^\top M \bullet$  the norm weighted by the matrix  $M$ .

Note that in theory, both of these problems are not equivalent, since argmax or saddle points in the optimisation problem 2.3 also satisfy the optimality condition defined by 2.2. However, as we are considering small timesteps and smooth and continuous trajectories, the system should not make any jump and thus we are looking for the closest state that satisfies the critical point condition.

### 2.1.2 Reformulation & algorithm

The previous minimisation problem is complex to solve as the potentials are often highly non-linear and non-convex. Ergo, to alleviate the computational cost, the Projective Dynamics method makes some assumptions on these potentials which enable the simplification of the optimisation problem.

**Quadratic potentials.** More exactly, with the idea that each potential tries to draw the system to its rest state, each potential is assumed to have the following form

$$W_j(x) = \min_{p_j \in E_j} \frac{\lambda_j}{2} \|A_j x - B_j p_j\|^2 \quad (2.4)$$

with  $\lambda_j \in \mathbb{R}_+^*$  a weight,  $A_j$  and  $B_j$  two weighting selection matrices and  $p_j$  the projection of  $x$  on the set of rest shapes of the potential  $E_j$ .

To illustrate these definitions, let us consider the case of a spring of rest length  $l_0 \in \mathbb{R}_+^*$  between two vertices  $i_1$  and  $i_2$ . To fit in the spring energy with this framework, we can take for  $E_j$  the sphere centred in 0 and of radius  $l_0$  :  $\mathcal{S}(0, l_0)$  as depicted in Figure 2.1,  $B_j = [\mathbf{I}_3 | -\mathbf{I}_3] \in \mathbb{R}^{3 \times 6}$ , with  $\mathbf{I}_3$  the identity matrix of size 3, and  $A_j = B_j S \in \mathbb{R}^{3 \times 3n_v}$  with  $S \in \mathbb{R}^{6 \times 3n_v}$  a matrix selecting the coordinates of the vertices  $i_1$  and  $i_2$  with two blocks  $\mathbf{I}_3$ . Then, the  $p_j$  realising the minimum in Equation 2.4 is equal to  $l_0 \frac{x_{i_1} - x_{i_2}}{\|x_{i_1} - x_{i_2}\|} \in \mathcal{S}(0, l_0)$ , and we recover the potential of a spring of stiffness  $\lambda_i$ .

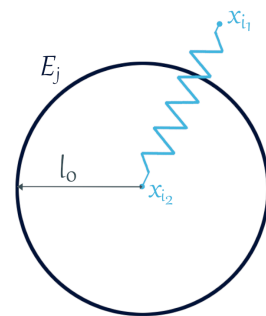


Figure 2.1: "Projective Dynamics" potential for a spring.

In practice, the spring potential described right above is the one we use for the in-plane deformation of the clothes simulated with this method. For the sake of completeness, let us also formulate the bending energy.

It is the one described in (Bouaziz et al., 2014). In the same fashion as Grinspun et al. (2003), the authors formulate their bending energy from a discretization of square of the

difference of the mean curvatures. They use the Laplace-Beltrami operator, which yields around each inner vertex  $i$  the potential

$$W_{i|bend}(x) = \min_{R \in SO(3)} \frac{\lambda_i \mathcal{A}}{2} \|C \mathcal{R}_{i|1} x - R C \mathcal{R}_{i|1} \bar{x}\|^2, \quad (2.5)$$

where  $\mathcal{R}_{i|1}$  selects the components of the one-ring edges of vertex  $i$  in the deformed and undeformed positions respectively denoted by the unbar and bar quantities,  $\mathcal{A}$  the Voronoi area around  $i$  and  $C$  contains the area-weighted cotangent weights (Botsch et al., 2010). Interestingly, the problem here is formulated so as to find a rotation  $R$  that minimizes the distortion. However, in practice, this problem has a direct solution  $R C \mathcal{R}_{i|1} \bar{x} = \|\mathcal{R}_{i|1} \bar{x}\| \frac{C \mathcal{R}_{i|1} x}{\|C \mathcal{R}_{i|1} x\|}$ .

**Local and global steps.** Assuming that the  $p_j$  are known, the considered potentials are quadratic in  $x$  while the non-linearity is hidden in the computation of the projections  $p_j$ . This gives the core idea of the method to simplify the optimisation.

The main algorithm is composed of two steps called *local* and *global* that respectively compute the projections  $p_j$  and the unknown at the next timestep (the positions  $x^{n+1}$  or the velocities  $v^{n+1}$ ). The two steps are iterated over in order to refine the results; we denote these iterations with the subscript  $k$ .

In the so-called local step, the  $p_{j|k+1}$  are computed by projecting the current estimate of the positions  $x_k^{n+1}$ . In practice, the potentials are formulated so that the projections can be efficiently computed in parallel. They are defined on small stencils (stretching per edge, shearing per face etc.) in a way such that  $p_{j|k+1}$  can be computed either by a simple close formula or an easy-to-solve minimisation, as illustrated in the previous paragraph.

Then in the global step, the minimisation 2.3 is solved with the  $p_j$  fixed, giving the next iterate  $k + 1$ . As we noted, the problem in this case is now quadratic *w.r.t.* the unknown, and thus can be solved by a linear system, in position or in velocity

$$\left( M + h^2 \sum_j \lambda_j A_j^\top A_j \right) x_{k+1}^{n+1} = M s^n + h^2 \sum_j \lambda_j A_j^\top B_j p_j \quad (2.6a)$$

$$\left( M + h^2 \sum_j \lambda_j A_j^\top A_j \right) v_{k+1}^{n+1} = M \tilde{s}^n + h \sum_j \lambda_j A_j^\top (B_j p_j - A_j x^n). \quad (2.6b)$$

Another point worth mentioning is that the left hand-side does not vary along the simulations *as long as the potentials do not change*. The matrix can therefore be pre-factorised, allowing for a quick solve of this global step.

So all in all, the Projective Dynamics iterates over these two steps designed to be efficiently solved. In the aim of having an interactive simulation, the minimisation is not solved until convergence, but the loop is only performed for a fixed number of iterations,

the so-called *local/global steps*. The algorithm is summed up in Algorithm 1, detailed for the case of the positions as the unknowns.

---

**Algorithm 1:** Projective Dynamics algorithm for computing positions at time  $t_{n+1}$ .

---

```

1 // Warm-start with an explicit computation of the next time-step
2  $x_0^{n+1} \leftarrow x^n + hv^n + h^2 M^{-1} f_{ext}$ ;
3 for  $k$  from 0 to  $max\_iter - 1$  do
4   // Local step
5   For all  $j$ , project  $x_k^{n+1}$  to obtain  $p_{j|k+1}$  using (2.4);
6   // Global step
7   Solve the linear system (2.6a) with fixed  $p_{j|k+1}$  to obtain  $x_{k+1}^{n+1}$ ;
8 end
9  $x^{n+1} \leftarrow x_{max\_iter}^{n+1}$ ;

```

---

On the topic of convergence, Bouaziz et al. (2014) showed in some examples involving only 3D elasticity that their method was faster to reach low accuracy (visual appearance) and moderate accuracy compared to the Newton’s method, but was slower to reach higher precision. Overby et al. (2017) proved that Projective Dynamics can be seen as a particular case of the ADMM optimisation algorithm, for which more theoretical results on the convergence exist.

### 2.1.3 Contacts handling

In the previous section, we presented the algorithm in the case of constant external forces and internal forces deriving from potentials, which does not account for contact forces.

To handle the non-penetration constraint, Bouaziz et al. (2014) add, when a contact is detected, a new potential of the form

$$W_c = \frac{\lambda_c}{2} \|x_i - p_{c|i}\|^2 \quad (2.7)$$

with  $\lambda_c$  the weight set to a high value, and  $p_{c|i}$  the projection of the penetrating vertex  $x_i$  on the tangent plane at the closest point on the surface of the obstacle.

This corresponds to adding a strong spring along the normal direction of the contact that aims at pulling the vertex out of the obstacle. The friction in the tangential direction (viscous or dry) is then added in a post-processing step similar to the one of PBD (Müller et al., 2007).

With a good tuning of the stiffness, this method can actually prevent visual interpenetration of the objects. However, its main drawback is that these potentials are added on demand, meaning that the Equation 2.6 may change over the iterations. This prevents the usage of the pre-factorisation of the contact-free left hand-side, slowing down the sim-

ulation even more as it is subjected to frequent impacts.

To alleviate this extra cost, subsequent strategies mainly focussed on finding efficient ways to solve the global step without relying on the pre-factorisation.

Wang (2015) proposed a parallel solver based on the Chebyshev semi-iterative method (Golub and Van Loan, 1996). Using an estimation of the spectral radius of the left hand-side, the author builds a weighted Jacobi-like solver with coefficients computed to accelerate the convergence. Although the resulting method leads to good convergence speed-ups while being parallel with little overcost compared to a standard Jacobi, it heavily relies on having a good estimation of the spectral radius to have a good speed-up without creating any divergence.

Similarly, Fratarcangeli et al. (2016) presented an algorithm to solve in parallel on the GPU the linear system. By applying a graph colouring method, they compute subsets of equations in the global system that do not share any unknown and thus can be solved in parallel using a Gauss-Seidel. Their solver produces stable and plausible simulations for time-budgets suitable for real-time animation, although the convergence speed towards higher precision has not been showcased.

Departing from parallel solvers, Komaritzan and Botsch (2018) noted that the collision potential from Equation 2.7 could in theory be set for all vertices, and the projection be the identity for contact-free vertices. However, they note that the high weight of this potential in practice dampens the convergence of the system. Thus, they proposed to use 2 pre-factorised constant matrices: one contact-free, and one with all the collision potentials. The first half of the local/global iterations is solved assuming the system is contact-free in order to have a good warm-start for the dynamics, and then the second half of the iterations is solved with the "contact-full" system and the appropriate projections to enforce non-penetration.

Our approach also differs from the ones mentioned right above as we aim at treating the contacts by adding contact forces defined by the Signorini-Coulomb law and not springs.

## 2.2 Projective Friction

The key idea of our method is to notice that the local steps actually estimates and refines the forces at play in the next time-step. Thus, we propose a new "local" step that also does the same for contact forces.

### 2.2.1 Splitting scheme for the contact force estimation

The DFPC in the case of a contact with an immobile object introduced in the previous chapter in Equation 1.23, is recalled here with slight modifications of notations

$$\begin{cases} Pv = f + H^\top r \\ u = Hv \\ \forall \text{ contact } c : (u_c, r_c) \in \mathcal{C}_{\mu_c} \end{cases} . \quad (2.8)$$

That is, to integrate the contact forces, we just need to add to the contact-free Equation 2.6b of the global step with the velocities as unknown, and add the constrained contact forces

$$\begin{cases} \underbrace{\left( M + h^2 \underbrace{\sum_j \lambda_j A_j^\top A_j}_C \right)}_P v_{k+1} = \underbrace{M \tilde{s}^n + h \sum_j \lambda_j A_j^\top (B_j p_{j|k+1} - A_j x_k^n)}_{f_{k+1}} + H^\top r \\ u_{k+1} = H v_{k+1} \\ \forall \text{ contact } c : (u_{c|k+1}, r_c) \in \mathcal{C}_{\mu_c} \end{cases} . \quad (2.9)$$

The main difficulty when integrating the Signorini-Coulomb law with an implicit scheme is that the relation between the two constrained unknowns  $v \in \mathbb{R}^{3n_v}$  and  $r \in \mathbb{R}^{3n_c}$  is not trivial.

Here, the coupling of the constraints happens because the left hand-side matrix has a non-diagonal component that we noted  $C$ . Nonetheless, we want to take advantage of the fact that the "forces"  $f_{k+1}$  are being adjusted over the local/global iterations, and thus reduces the need of an accurate estimation of  $r$  at each iteration. This goes along with the fact that the computation of  $r$  has to be fast in order to preserve the simulation speed of the base method.

The approach we adopt is to estimate  $r$  using a simpler equation. Consider the Equation 2.9 splitted in the Jacobi-like fashion

$$M v_{\mathbf{k}+1} = \underbrace{f_{k+1} - C v_{\mathbf{k}}}_{F_{k+1}} + H^\top r \quad (2.10)$$

where the non-diagonal term is explicitly evaluated using the previous iteration.

Moreover, in the case of nodal systems, the matrix  $H \in \mathbb{R}^{3n_c \times 3n_v}$ , previously defined in Section 1.2.3, that links the quantities expressed in the local contact frames (local velocity, contact force) to the quantities in the space of the general coordinates of the equation of motion is just composed of rotations. In other words, if the vertex  $i$  is the  $c$ -th contact with a local frame defined by the normal  $n_c$  and the tangents  $t_{1|c}$  and  $t_{2|c}$ , then  $H$  seen as a block matrix with block of sizes  $3 \times 3$  has at the position  $(c, i)$  the non-zero block  $R_c = [ n_c \mid t_{1|c} \mid t_{2|c} ]^\top$ .

Equation 2.10 therefore provides at every contact point a one-to-one correspondence between the velocity and the reaction force

$$m_c v_{c|k+1} = F_{c|k+1} + R_c r_c \quad (2.11)$$

that can also be rewritten as

$$m_c u_{c|k+1} = R_c^\top F_{c|k+1} + r_c. \quad (2.12)$$

Using these equations,  $r_c$  can be explicitly chosen so as to enforce the Signorini-Coulomb law (defined in Section 1.2.1):

- If the normal "force"  $F_{c|k+1}^\top n_c$  is strictly positive *i.e.* pushes away the vertex, then we can take  $r_c = 0$  and retrieve the taking-off case;
- Otherwise,  $r_c$  tries to prevent any motion with the sticking case  $r_c = -F_{c|k+1}$ ;
- But the cone constraint bounds the tangential component  $\|r_{c|t}\| \leq \mu r_{c|n}$  leading to the sliding case<sup>1</sup>.

The procedure described right above is cheap, and provides a way to estimate  $r$  that can then be reinjected in Equation 2.9 for the global solve. This algorithm, along with some extensions, is summed up in Algorithm 2.

## 2.2.2 Extensions of the base case.

In the previous section, I introduced the procedure to deal with contacts with immobile objects. This is obviously not sufficient, and so I present here how the procedure is modified to take into account more complex cases.

**Moving objects.** As presented in Section 1.2.1,  $u$  is actually the relative velocity between the two objects at the contact points. This translates in the relation between the velocities  $u$  and  $v$ . Let  $w \in \mathbb{R}^{3n_c}$  be the translation velocities of the contact frames, *i.e.* the opposite of the velocities of the contact points on the obstacle expressed in the local frames. Then the relation between the velocities is

$$u = Hv + w. \quad (2.13)$$

Equation 2.12 then becomes

$$m_c u_{c|k+1} = R_c^\top F_{c|k+1} + m_c w_c + r_c. \quad (2.14)$$

where the same estimation procedure can be carried out with the additional term.

---

<sup>1</sup>or a borderline sticking case.

**Self-contact.** In the case of a self-contact, we follow the same idea as in the previous paragraph to retrieve the relative velocity between the two colliding vertices.

Assume that vertices  $i_1$  and  $i_2$  are forming the  $c$ -th contact. By Newton third law, we know that they are respectively submitted to the reaction forces  $r_c$  and  $-r_c$ . Then, rewriting Equation 2.11 for both vertices yields

$$\begin{cases} m_{i_1} v_{i_1|k+1} = F_{i_1|k+1} + R_c r_c \\ m_{i_2} v_{i_2|k+1} = F_{i_2|k+1} - R_c r_c \end{cases} \quad (2.15)$$

To retrieve the relative velocity, we just need to subtract both equations to have

$$u_c = R_c^\top (v_{i_1|k+1} - v_{i_2|k+1}) = R_c^\top \left( \frac{1}{m_{i_1}} F_{i_1|k+1} - \frac{1}{m_{i_2}} F_{i_2|k+1} \right) + \frac{m_{i_1} + m_{i_2}}{m_{i_1} m_{i_2}} r_c \quad (2.16)$$

where the local force can be enforced as previously with appropriate scaling by  $\frac{m_{i_1} m_{i_2}}{m_{i_1} + m_{i_2}}$ .

**Vertices subject to multiple contacts.** In our current scheme, contact forces are considered as external forces which are all updated in parallel after the update of the  $p_j$ . However, in the case where nodes were subjected to multiple contacts (multi-layering case), we found this algorithm to be unstable.

Indeed, updating all the contact forces in parallel leads to each contact being aware of the other contact forces only at the previous iteration. As a result, each force would either be largely overestimated to prevent on their own non-penetration while other "neighbour" forces may also contribute or underestimated because surrounding forces were overestimated at the previous iteration. This instability leads to some unacceptable artefacts.

To overcome this issue, we sort the contact forces in several "layers" that are processed sequentially using updated information of the previous layers to avoid inconsistencies, while forces in the same layer can safely be processed in parallel. More precisely, we process the list of contacts and organise them as follows:

- Contacts between two nodes that are not involved in other contacts are safe and added to the first parallel batch (1);
- Then we process the remaining contacts by traversing our graph of contacts, and build the different contact "layers";
- Finally, for layers of self-contacts that are not in contact with an external object, we start arbitrarily from one "side" and build the layers through to the other side (Figure 2.2, right).

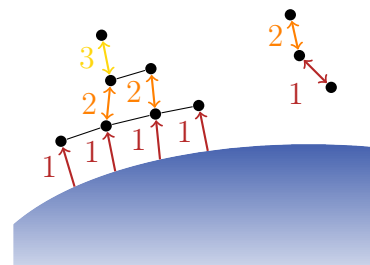


Figure 2.2: Contact ordering scheme.

### 2.2.3 Algorithm

We present in Algorithm 2 the modified Projective Dynamics algorithm to include our force estimation scheme (in purple) entitled *Projective Friction*.

As Brown et al. (2018), because of the update of the external forces, the modified algorithm does not properly solve an optimisation problem since the objective function is evolving through the iterations.

However, in practice, we observed good qualitative results that we present in the following section, along with some quantitative studies.

---

**Algorithm 2:** *Projective Friction*: Projective Dynamics algorithm, augmented with our computation of frictional contact forces.

---

```

1 // Warm-start the next time-step
2  $v_0^{n+1} \leftarrow v^n + hM^{-1}f_{ext}$ ;
3 Detect the contacts;
4 Sort self-contacts (see section 2.2.2);
5 for  $k$  from 0 to  $max\_iter - 1$  do
6     // Local step
7     For all  $j$ , project  $(x_n + hv_k^{n+1})$  to obtain  $p_{j|k+1}$  using 2.4;
8     // Contact step
9     Compute the contact forces  $r_c$  using the ordering and Equations 2.12, 2.14
       and 2.16 (Sections 2.2.1 and 2.2.2) ;
10    // Global step
11    Solve the linear system (2.9) with fixed  $p_{j|k+1}$  and  $r_c$  to obtain  $v_{k+1}^{n+1}$ ;
12 end
13  $v^{n+1} \leftarrow v_{max\_iter}^{n+1}$ ;
14  $x^{n+1} \leftarrow x^n + hv^{n+1}$ ;

```

---

## 2.3 Evaluation and results

We have reimplemented the base Projective Dynamics algorithm along with our modification in C/C++ using *OpenMP* for parallelisation. Collision detection is performed through simple proximity queries, using an acceleration structure. All the examples were run on a desktop computer featuring 4 dual-core Intel i7-5600U processors running at 2.60GHz.

### 2.3.1 Test scenarios

The results presented here focus on cloth, but Projective Dynamics and our Projective Friction are more general and can be applied to any nodal system provided the energies have the peculiar form required by the framework.



We have tested our method on five scenarios:

- **Ribbon**: a ribbon falling on an inclined plane, causing multiple layered self-contacts;
- **Square1**: a flat sheet falling on a rotating sphere, showcasing impacts and frictional contact with a moving obstacle;
- **Square3**: three stacked flat sheets falling on a rotating sphere, involving additional multi-layered contact compared to the previous example;
- **Arabesque**: the dress of a dancing character, freely available from [Li et al. \(2018a\)](#), combining impacts with a moving obstacle and stick-slip thresholding behaviour;
- **Crinoline**: a highly-detailed gown with puff sleeves and complex folds, subject to walking and turning motions.

Table 2.1 gives the configuration for each one of these scenarios. For better realism in **Arabesque** and **Crinoline**, we added air damping forces modelled explicitly as  $f_{i|k+1} = -\nu v_{i|k}$  for each non-contacting node, with  $\nu$  the damping coefficient.

Sample images of our examples are provided in Figures 2.5 and 2.4. To watch the corresponding animations, please refer to [the accompanying video](#) of the original paper.

Table 2.1: Main parameters used for our five examples.

Example	Nb of vertices	Mass density ( $\text{kg} \cdot \text{m}^{-2}$ )	Stretch. weight ( $\text{N} \cdot \text{m}^{-1}$ )	Bend. weight ( $\text{N} \cdot \text{m}$ )	Air damp. coeff. ( $\text{N} \cdot \text{s} \cdot \text{m}^{-1}$ )	Time step (ms)
<b>Ribbon</b>	5946	0.25	20	$2 \cdot 10^{-2}$	0	4
<b>Square1</b>	5996	3	200	$5 \cdot 10^{-4}$	0	5
<b>Square3</b>	17964	1.5	200	$5 \cdot 10^{-4}$	0	5
<b>Arabesque</b>	15842	0.01	4	$5 \cdot 10^{-7}$	$3 \cdot 10^{-6}$	2
<b>Crinoline</b>	53097	0.01	5	$5 \cdot 10^{-6}$	$2.5 \cdot 10^{-7}$	2

### 2.3.2 Qualitative evaluation

**Influence of the number of local/global iterations.** Figure 2.3 shows the effect of varying the number of local/global iterations for **Ribbon** and **Square1**. For both examples, simulations degrade consistently down to 10 iterations, below which **Square1** stops converging properly, while **Ribbon** continues to work consistently. Visually, no significant difference is observed above 20 iterations. In the following, we thus chose 20 iterations for these two examples along with **Square3**. For **Arabesque** and **Crinoline**, which involve higher resolution meshes and stiffer materials, we used 30 iterations.

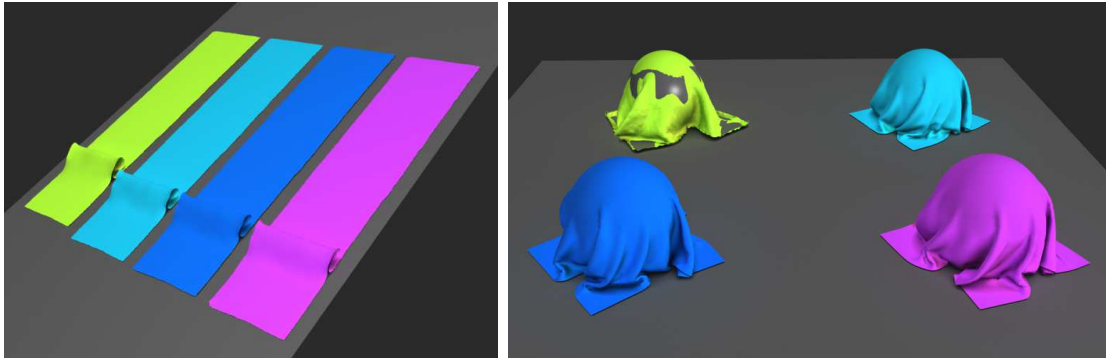


Figure 2.3: Simulating the **Ribbon** (left,  $\mu = 0.3$ ) and **Square1** (right,  $\mu = 0.1$ ) examples using a various number of iterations. In green: 5 iterations, in light blue: 10 iterations, in dark blue: 20 iterations, in purple: 30 iterations.

**Varying the friction coefficient and comparison with ARGUS.** As depicted in Figure 2.5, we have run our four first examples with both low and large values of the friction coefficient to demonstrate the macroscopic impact of sticking *vs* sliding behaviours. We have reproduced these experiments using the accurate and freely available ARGUS simulator (Li et al., 2018a), using the same timesteps and a fixed mesh resolution.

We did not try to match the material parameters between the energies of Projective Dynamics and ARGUS but we picked stretching and bending weights so as to obtain a reasonable match. Despite these differences in material, it is noteworthy that the response to frictional contact yielded by our method is similar to that generated by ARGUS, as shown the accompanying video of the paper. Moreover, in case of multi-layered contacts, ARGUS in its fixed resolution version depicts some popping artefacts (presumably due to an overly coarse mesh resolution in folding areas), whereas our method generates perfectly stable results.



Figure 2.4: Highly-detailed **Crinoline** example ( $\mu = 0.3$ , 30 iterations).

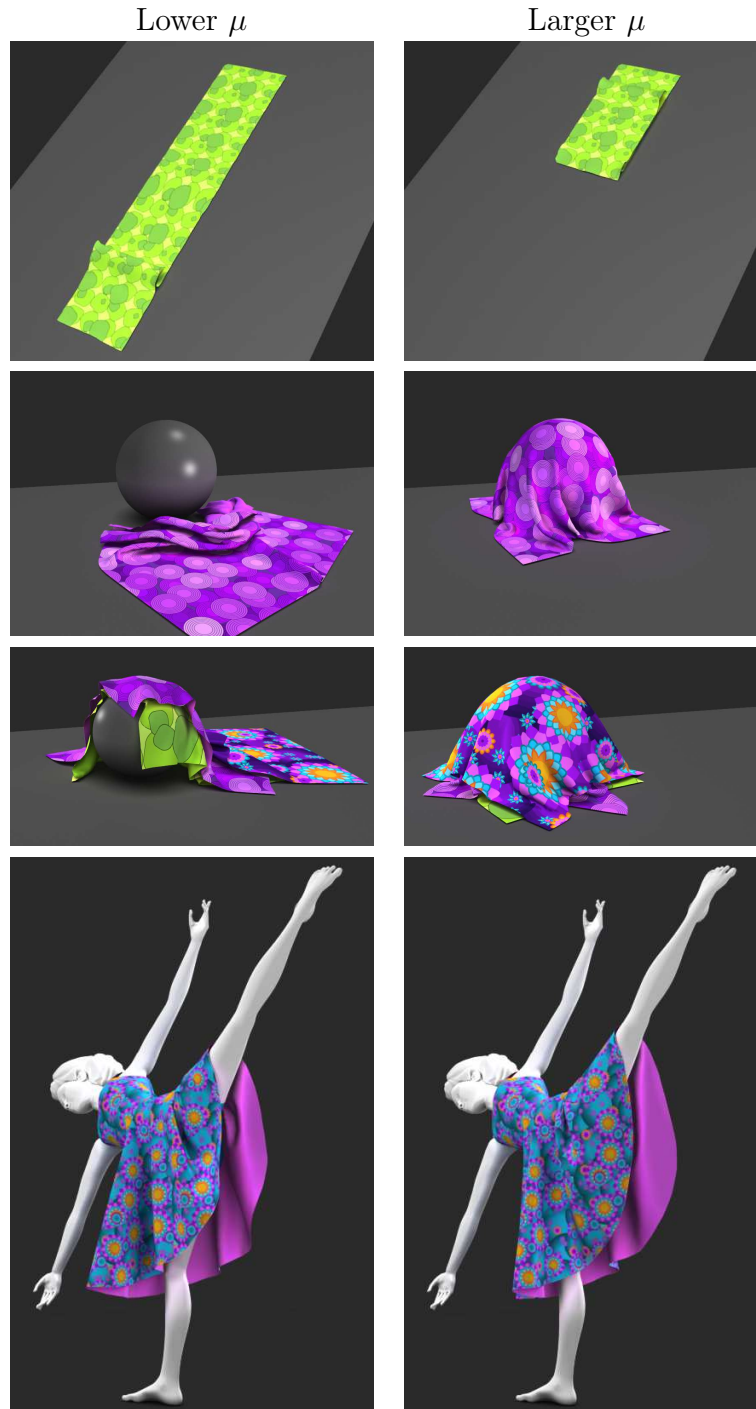


Figure 2.5: Varying the friction coefficient in our four examples **Ribbon** ( $\mu = 0.3$  and  $\mu = 0.6$ ), **Square1** and **Square3** ( $\mu = 0.1$  and  $\mu = 0.3$ ), and **Arabesque** ( $\mu = 0.0$  and  $\mu = 0.3$ ), using 20 to 30 iterations per timestep. Results are visually close to those generated by the accurate ARGUS simulator (Li et al., 2018a), while being computed 15 to 36 times faster.

### 2.3.3 Convergence

**Theoretical notes.** A full study on the convergence of our method has been deemed too complex for our purpose, as it would also need to be intertwined with results on the convergence of Projective Dynamics. We can however mention a few relevant points.

The first remark we can make is on the correctness. A fixed point of the equation 2.10 is a solution of the DFCP and conversely. It can be easily seen as the fixed point property cancels the iteration-dependant error introduced by the splitting.

Secondly, we can observe that this "splitting error" is the term  $Cv_k$  that results from the implicit integration of the internal forces. Ergo, we can think that the smaller these internal forces are, the smaller the error of our estimation scheme is and thus the better the convergence is.

**Analytical example.** As in (Li et al., 2018a), we have evaluated our method on the scenario of a falling sheet parallel to an inclined plane, illustrated in Figures 2.6a and 2.6b. In this setup, an analytical model for the dynamic of the nodes can be derived, and we show in Figures 2.6c and 2.6d that our numerical results match the analytical curves.

Moreover, we have measured a global penetration and friction error using the so-called Alart-Curnier function Alart and Curnier (1991); Bertails-Descoubes et al. (2011) and we have found out that, in this simple scenario, for each time-step the error vanishes after just one local/global iteration.

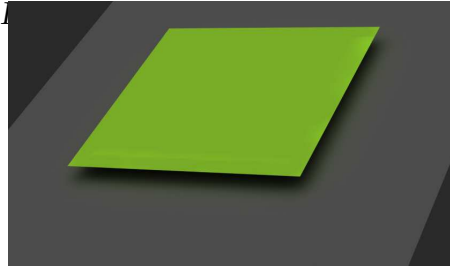
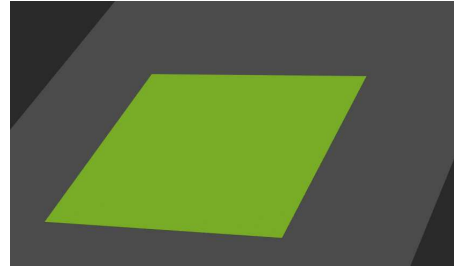
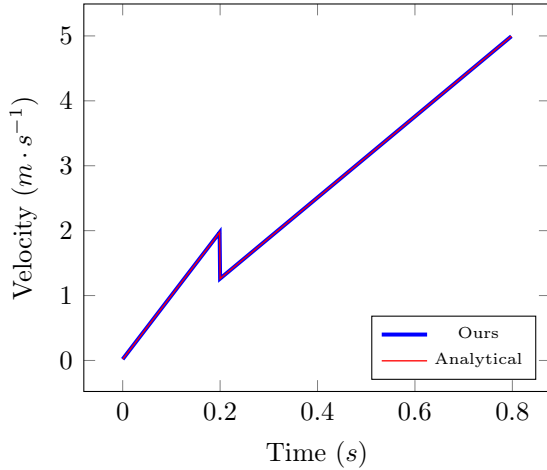
Indeed, because all the nodes have the same motion due to the initial configuration, no internal forces is applied. This leads the non-diagonal terms that we noted  $C$  on each side of Equation 2.9 to cancel each other, meaning no error is introduced by the splitting scheme 2.10 as mentioned in the previous paragraph. In other words, in this case, the integration scheme boils down to an explicit Euler, where our procedure is exact.

**General case.** In most of the cases where the terms in  $C$  are not negligible, our scheme still manages to decrease the Alart-Curnier error, yielding the visually good results presented above.

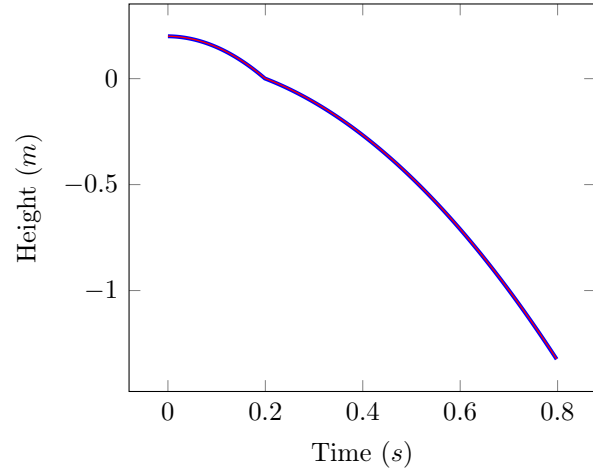
In Figure 2.7, we show the evolution of the normal and tangential errors (*i.e.* the amount of deviation from the non-penetration and Coulomb friction constraints, respectively) *w.r.t.* the number of local/global iterations for four different timesteps selected from the **Sphere1** example.

We see that both errors quickly decrease in the first iterations, before reaching a plateau at moderate precision with a much lower slope. Also note that the tangential part responsively adjusts to the bound set by the normal component in the Coulomb law.

However, we can also observe that the convergence has different behaviour depending on the moment we are looking at. To make the link with the remark on the theoretical convergence paragraph above, we can try to explain the different curves by what is hap-

(a)  $t = 0s$ .(b)  $t = 0.4s$ .

(c) Velocity of the centre of mass.



(d) Height of the centre of mass.

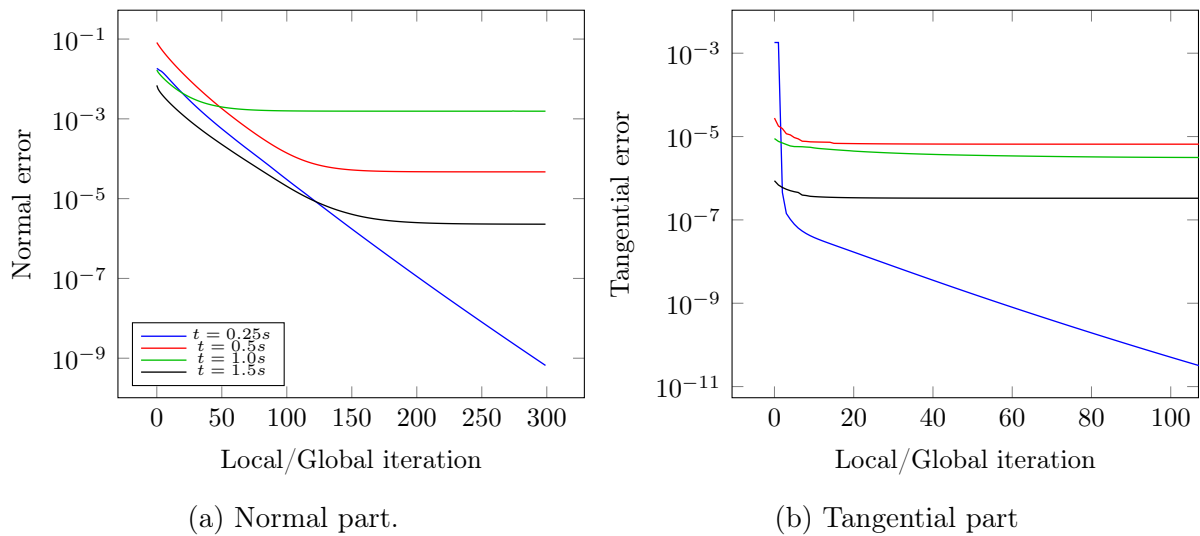
Figure 2.6: Comparison to an analytical scenario.

pening in the simulation. For the blue curve  $t = 0.25s$ , the sheet is contacting the top of the rotating sphere, but most of the sheet is still almost flat *i.e.* the sheet is subjected to little internal forces, which may explain the good behaviour. At the time of the red  $t = 0.5s$  and green curve  $t = 1.0s$ , the sheet is falling over the sphere and wrapping it, leading to "large" internal forces. Finally, for the black curve  $t = 1.5s$ , the sheet is "stabilised" on the sphere and rotates with it, so with "less" internal forces at play.

### 2.3.4 Performance

Table 2.2 shows that our method fits in well with the Projective dynamics algorithm. Indeed, compared to the native algorithm without contact, it only adds a small overhead when computing the right-hand side of the global equation and locally updating contact forces using our sorting algorithm (columns in purple). Additionally, as we do not need to modify the left-hand side of the global solve, we preserve the inherent speed of Projective dynamics.

**Comparison with penalty contact forces** We have run `Square1` by computing contacts as penalty forces, similarly to Bouaziz et al. (2014) and most follow-up papers. We have observed a speed gain  $g = \frac{t_{\text{Bouaziz}}}{t_{\text{ours}}} = 1.1$  brought by our approach, even though



(zoom on the 100 first local/global iterations).

Figure 2.7: Evolution of the mean Alart-Curnier error *w.r.t.* the local/global iterations on the **Sphere1** scenario.

ours is richer and more robust as it fully captures the Signorini-Coulomb law in a semi-implicit way. This overhead in Bouaziz et al. (2014) is due to the Cholesky factorisation which needs to be performed at each global step, and eventually represents a cost of 21% of the global step on average.

**Comparison with ARGUS** Li et al. (2018a) Table 2.2 reports the average cost of ARGUS’s timesteps for **Square1**, **Square3**, and **Arabesque** run with the same number of vertices and the same timestep values used by our method, and without adaptivity. It turns out that our method runs more than one order of magnitude faster, with a speed gain<sup>2</sup> comprised between  $\times 15$  and  $\times 36$ . We have noticed that the ARGUS solver is especially penalised when many self-contacts are involved (**Square3** example), due to their handling through vertex duplication and artificial pin constraints.

## 2.4 Discussion and conclusion

To conclude this chapter, I have presented a simple method to introduce dry frictional contact into the Projective dynamics framework in a robust yet inexpensive way. Our technique preserves the global step Cholesky factorisation that is one of the keystones of the speed of the method, and only adds a small overhead when assembling the equations and updating contact forces.

However, our method is also not devoid of flaws that may lead to interesting future work on this method.

<sup>2</sup>With adaptivity enabled in ARGUS, the speed gain lies between  $\times 6.5$  and  $\times 8$ .

Table 2.2: Performance of our solver for all our examples

Example	$\mu$	$\bar{n}_{\text{ext}}^2$	$\bar{n}_{\text{self}}^2$	$\bar{t}_{\text{rhs}}^3$	$\bar{t}_{\text{ext}}^3$	$\bar{t}_{\text{self}}^3$	$\bar{t}_{\text{solve}}^3$	$\bar{t}_i^3$	$\bar{t}_{\text{contact}}^4$	$\bar{t}_{\text{self-contact}}^4$	$\bar{t}_{\text{sorting}}^4$	$\bar{t}_p^5$	$\bar{t}_{\text{Argus}}^6$	$\bar{g}^6$
<b>Ribbon</b>	0.3 <sup>1</sup>	4210	455	1.9	0.29	0.07	1.02	3.3	3.1	38.4	0.17	109	–	–
	0.6 <sup>1</sup>	2194	1769	1.9	0.27	0.179	1.02	3.4	2.8	49.8	0.29	122	–	–
<b>Square1</b>	0.1	2273	1009	2.0	0.29	0.142	1.24	3.8	5.9	69.5	0.25	153	2244	<b>14.7</b>
	0.3	1665	301	2.1	0.28	0.071	1.23	3.7	6.6	62.6	0.16	144	4828	<b>33.5</b>
<b>Square3</b>	0.1	4034	5840	6.2	0.79	0.66	3.60	11.4	11.8	283.6	1.23	526	18713	<b>35.6</b>
	0.3	2222	5311	6.2	0.78	0.59	3.60	11.3	13.1	283.4	1.26	525	19233	<b>36.6</b>
<b>Arabesque</b>	0.0	3354	155	5.3	0.62	0.121	4.33	10.4	51.3	161.2	0.28	530	9069	<b>17.1</b>
	0.3	3673	102	5.4	0.68	0.093	4.39	10.6	58.0	152.7	0.30	543	15899	<b>29.3</b>
<b>Crinoline</b>	0.3	5977	1052	15.6	2.22	0.427	17.0	35.5	200	483	1.04	1751	–	–

<sup>1</sup> Self-friction coefficient only. In the **Ribbon** example, the friction coefficient with the inclined plane is 0.7.

<sup>2</sup> Average number of contact points with external objects ( $\bar{n}_{\text{ext}}$ ) and with the object itself ( $\bar{n}_{\text{self}}$ ).

<sup>3</sup> Average time in ms per iteration ( $\bar{t}_i$ ), including the time for assembling the right-hand side ( $\bar{t}_{\text{rhs}}$ ), the computation of the frictional contact forces with external obstacles ( $\bar{t}_{\text{ext}}$ ) and with the object itself ( $\bar{t}_{\text{self}}$ ) and the global step solve ( $\bar{t}_{\text{solve}}$ ).

<sup>4</sup> Average time in ms to detect the collisions  $\bar{t}_{\text{contact}}$ , the self-collisions  $\bar{t}_{\text{self-contact}}$  and to perform the contact sorting  $\bar{t}_{\text{sorting}}$ .

<sup>5</sup> Average time in ms per timestep ( $\bar{t}_p = \bar{t}_{\text{contact}} + \bar{t}_{\text{self-contact}} + \bar{t}_{\text{sorting}} + n_{\text{iter}} \times \bar{t}_i$ ) with  $n_{\text{iter}} = 20$  for the three first examples and  $n_{\text{iter}} = 30$  for the last two.

<sup>6</sup> Average time in ms per timestep ( $\bar{t}_{\text{Argus}}$ ) for the ARGUS solver Li et al. (2018a) used with a fixed mesh resolution, and speed gain ( $\bar{g} = \frac{\bar{t}_{\text{Argus}}}{\bar{t}_p}$ ) of our method.

First of all, Our method inherits the limitations of the base method, and in particular the lack of a simple rule to ensure convergence. In order to obtain stable simulations, the user needs in a preliminary step to adjust the number of iterations required, depending on the mesh size and the material used.

On top of that, our splitting scheme introduces an error in the estimation procedure that makes the study of the convergence tedious. Our experiments and our intuition suggest that the accuracy of our method degrades as the variation of internal forces increases. An interesting improvement of our method would therefore to alleviate this behaviour and have a more stable convergence profile.

Secondly, our algorithm only considers vertex-vertex contact, which can be insufficient for handling properly some specific scenarios, such as cloth contacting an obstacle with corners or sharp edges. However, treating a contact point that is not a node is not straightforward. In such a case, the block lines of the matrix  $H$  relating velocities of the contact points  $u$  to the degrees of freedom  $v$  do not only contain one rotation, but a linear combination of rotations. Thus, per contact point, the system is not invertible as it is with nodal contacts. Including vertex-face and edge-edge contact in our framework hence remains an open direction of research.

Orthogonally to our approach, the computation speed can still be improved, either by using parallel solvers as mentioned before (Wang, 2015; Fratarcangeli et al., 2016), mesh adaptivity Narain et al. (2012), or better detection collision schemes.





# Chapter 3

## Validation of codes in Computer Graphics

Physical simulators were introduced in Computer Graphics as a tool to help artists generate vivid motions for animation movies. Therefore, their main objectives were to target low computation time and user-control for the artists to create their target animation in as fast as possible trial and error process. The physics however only needed to be visually plausible. The simulators have since then improved in terms of complexity and accuracy, making them attractive to other applications such as robotics or architecture. Interesting results have been obtained thanks to the relatively fast yet robust simulators but the physical accuracy of the computed predictions is often not validated which may be inappropriate for critical tasks.

In collaboration with mechanicians, our research team proposed to introduce simple yet rich tests so as to encourage validation in Computer Graphics ([Romero et al., 2021](#)). In this chapter, I present my own contribution to this project, focussed on the validation of plate models. For the sake of completeness, I also present relevant work of other co-authors within this project that includes derivation of the scaling laws and other numerical experiments.

## 3.1 Organisation of the chapter and disclaimer on the contributions (⋈)

For the sake of completeness and readability, this chapters includes the work of all the co-authors of the article (Romero et al., 2021) as a whole. My own contributions to the project will be identified in the text and surrounded by the symbols ⋈ and ⋉. But more generally, this chapter can be divided into two main parts.

Sections 3.2 and 3.3 focus on the background of our work and largely draw from the corresponding sections of the original paper mainly written by the co-authors. One of the four protocols (**Bend-Twist**) is not presented here as it falls out of the scope of the manuscript. Within these sections, my contributions consist in helping reviewing the relevant work, and performing early numerical work and simulations to set-up the **Cantilever** and **Lateral Buckling** tests.

The following two Sections 3.4 and 3.5 are centred on the codes tested and their results. For this part, I introduce first the codes and the testing methodology discussed and chosen by all of the co-authors. Note that the code list presented here is restricted to a subset that is relevant to the manuscript. Next, I present the results of the three protocols. For the **Cantilever** and **Lateral Buckling** tests, the results are mainly my own contribution, with the help of Abdullah Haroon Rasheed for the ARCSIM code. For what is presented here, the **Stick-Slip** test has mainly been conducted by Victor Romero and Abdullah Haroon Rasheed.

## 3.2 Validation in Computer Graphics

### 3.2.1 Motivation

Physical simulators were developed in Computer Graphics to alleviate the tedious work of producing "environment" related animations, such as fluids for water, or, for what is our concern, slender structures subject to frictional contact such as hair or clothes. As such animations were intended for movies, their main features were the computation time and the user control, so as to provide short trial-and-error loops for the artists, while the resulting output needed only to be visually realistic (within the artistic context of the movie) Over the last decades, such simulators have successfully been used in the movie making industry (Baraff and Witkin, 1998; Daviet et al., 2011; Kaufman et al., 2014; McAdams et al., 2009; Iben et al., 2013).

However, the growing demand of visual effects and their integration into realistic-looking environments, whether virtual or coming from real shots, has increased the need for more visual realism for a better blend in. In order to attain the required visual richness, the simulators have evolved towards the integration of more complex models

able to reproduce more accurately the phenomena of the real life.

Meanwhile, these simulators have also been deemed accurate enough, yet simple and fast, for virtual prototyping, that is the fabrication of objects with specific mechanical properties predicted by the simulators. The range of covered application is large, and among them, we can cite cloth material measurement and prototyping (Wang et al., 2011; Wang, 2018; Yang et al., 2017; Bartle et al., 2016), architectural design (Konaković-Luković et al., 2018; Panetta et al., 2019; Gavriil et al., 2020; Laccone et al., 2019; Panetta et al., 2021), soft robots fabrication (Coevoet et al., 2019; Vanneste et al., 2020; Zimmermann et al., 2019), or new metamaterial design (Martínez et al., 2019; Schumacher et al., 2018; Guseinov et al., 2020). Simulations are also more and more used to generate training data for neural networks (Rasheed et al., 2020; Liang et al., 2019; Yang and Lin, 2016).

Common to the last aforementioned applications, their goal is to predict *quantitatively* the output, while image-producing applications only required *qualitative* predictions. Yet, validating the physical accuracy of simulators in Computer Graphics is rather a scarce practice, as opposed to other communities such as in Mechanical Engineering. Thus, there is often a lack of guarantee regarding the correctness of the simulated result, which might be critical for some domains like architecture or healthcare.

The goal of this project was thus to introduce simple yet rich protocols that can be easily run in order to evaluate the reliability of the simulators. The focus here is on static scenarios involving slender structures and frictional contact. Defined scenarios do not claim to exhaustively test the models (*e.g.* under other boundary conditions, other deformation modes etc.). However, the methodology we propose is general, and can be used to design other test scenarios for such structures or other physical models used in Computer Graphics (*e.g.* granular, fluids, solids).

### 3.2.2 Related work

**Extreme mechanics.** Numerical and physical validation of the numerical models has been for long a common practice in Mechanical Engineering. However, unlike in Computer Graphics where the interest is in large displacements to express vivid motions, Mechanical Engineering has considered them as undesirable or failure states and has focussed on studying the stability of the structures around small displacements. The development of validating tools suited for Computer Graphics scenarios has thus been restrained.

However, recently, a part of the Mechanical Engineering community has started to be interested in so-called *extreme mechanics* (Krieger, 2012), considered to be a promising direction of research. By studying formerly unwanted deformation behaviour and instabilities, they seek to build new mechanisms for novel applications such as actuation, structure deployment or energy harvesting (Reis, 2015; Hu and Burgueño, 2015; Ramachandran et al., 2016; Grandi et al., 2021). Yet, results that can be applied for the validation of "extreme" deformation of rods or plates and frictional contact remain scarce.

**Simple validation tests.** In simple cases analytical laws can be derived to characterise the expected deformation behaviour. For planar rods, assuming small deflections, one can solve analytically the linear Euler-Bernoulli beam equation in some classic scenarios such as the cantilever beam with clamped-free or clamped-clamped ends or the multiple point bending test (horizontal rod submitted to localised vertical upwards and downwards loads) (Timoshenko, 1953). For instance, (Martin et al., 2010) compared their unified model for rods and shells to the linear clamped-free cantilever test depicted in Figure 3.1a and (Panetta et al., 2019) validated their rod implementation using the four point bending test, illustrated in Figure 3.1b. However, these tests are no more valid for large displacements.

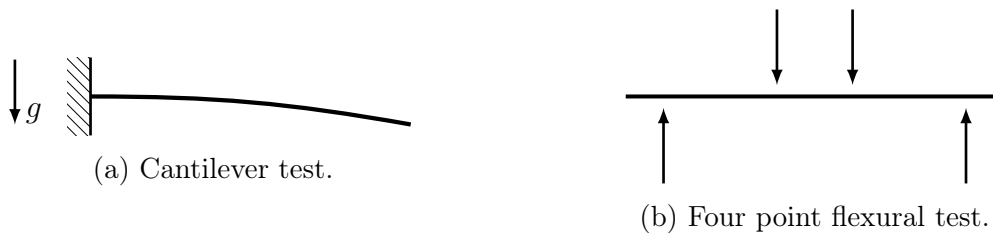
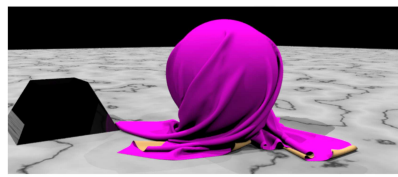


Figure 3.1: Linear rod tests.

In a similar way, Li et al. (2018a) have derived a simple analytical test for frictional contact, already presented in Section 2.3.3. Considering a falling sheet, initially parallel to an inclined plane, internal forces do not come at play and assuming immediate sliding (*i.e.* a high enough slope), they compute analytically the position and the velocity of the sheet. By doing so, they are able to evaluate the non-penetration as well as the sliding mode, but not the sticking mode and the transition between them.

**Qualitative evaluation.** Aiming to handle more and more complex scenarios, researchers in Computer Graphics have devised hard and impressive test cases to show that the *qualitative* behaviour is reproduced by their method and prove the robustness of their code in sometimes extreme cases. Over the years, this has led to the emergence of popular tests like a cloth sheet on a rotating sphere (Bridson et al., 2002), funnels (Harmon et al., 2008) or a reef knot (Harmon et al., 2009), respectively depicted in Figures 3.2a, 3.2b and 3.2c. These scenarios have been adopted and used as benchmarks in several subsequent works, but their evaluation remains qualitative.

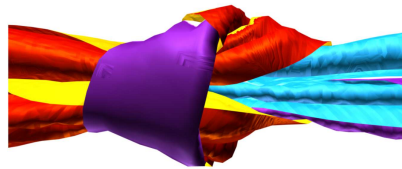
**Measurement protocols and data-driven simulation.** Another popular practice aiming to perform simulations as close to the reality as possible are the data-driven simulations. By adapting standard measurement protocols, such as for instance the ones provided by the ASTM International, or the Kawabata Evaluation System (Kawabata and Niwa, 1989) for clothes or by creating new protocols suited for the model used (Wang et al., 2011), the goal is to fit in the input parameters of the simulator so that the geometric output of the simulation matches the reality. The fitting may be done by providing a



(a) Cloth on rotating sphere, from (Bridson et al., 2002).



(b) Cloth funnel, from (Harmon et al., 2008).



(c) Reef knot, from (Harmon et al., 2009).

Figure 3.2: Some popular tests scenarios in Computer Graphics.

one-to-one correspondence between materials and their coefficients in the simulator, or a fitted function transforming physical parameters into coefficients, function that allows the interpolation between materials. The underlying model does not need to be entirely physical but should at least have enough parameters and be complex enough to fit with a good precision to the experiments used for calibration.

This strategy has been used widely on rather simple models to provide fast yet quite accurate simulations, for instance for the simulation of clothes (Wang et al., 2011; Miguel et al., 2012; Clyde et al., 2017), soft tissues (Bickel et al., 2009), inflatables (Skouras et al., 2012, 2014) or metamaterials (Bickel et al., 2010).

However, there may be no guarantee on the quality of the prediction for deformations or materials that have not been used for the fitting. Validating may help to assess the accuracy of such models, provided a link can be established between physical parameters and the coefficients of the model.

**Quantitative tests in Mechanical Engineering and in Computer Graphics.** In Mechanical Engineering, a common technique to validate a code quantitatively is to compare the result of the simulation against well-known physical experiments. For instance, for plates and shells, scenarios such as the sheared hemispherical shell or the pinched cylindrical shell has been used for long (Sze et al., 2004). Such scenarios may prove indeed challenging to reproduce, but they often rely on specific parameters to be able to compare against the reference.

In Computer Graphics, quantitative evaluations are rather scarce. To the best of our knowledge, they have been conducted in (Smith et al., 2012) for rigid body impacts, (Bergou et al., 2008) for rods and in (Bergou et al., 2010) for viscous threads. Smith et al. (2012) compare their results to a phase diagram of the literature describing the different

patterns appearing in vibrating granular materials. Bergou et al. (2010) perform a rather extensive validation of their model by reproducing several theoretical results from the literature (Ribe, 2004; Le Merrer et al., 2008; Chiu-Webster and Lister, 2006). Finally, closer to our contribution, Bergou et al. (2008) validate their model by reproducing the Michell (or Zajac)'s rod buckling instability. The **Bend-Twist** test (Miller et al., 2014) we introduced in (Romero et al., 2021), and that we will not detail here as it does not concern plates, also tests a bend-twist instability but has the advantage of having simpler boundary conditions.

### 3.3 Validation protocols

In this section, we present three of four protocols used to test the codes. As stated right before, we will not talk about the **Bend-Twist** test (Miller et al., 2014) as it is specific to rods and out of the scope of this manuscript. Please refer to the article (Romero et al., 2021) for the description and the results of this test.

#### 3.3.1 Scaling laws

As hinted in the related work, one of our goal is to provide results to compare with that do not rely on specific set of parameters. Ergo, we turned our attention to scaling laws and dimensional analysis (Buckingham, 1914), aiming to design "scale independent" validation tests that make use of reduced parameters.

A simple example is the linear pendulum. Within the assumption of small amplitude oscillations and without any dissipation, its oscillations have a periodic motion of period  $T = 2\pi\sqrt{l/g}$  with  $l$  its length and  $g$  the gravity. From this simple formula, we can already see several interesting aspects for the validation. (1) If the simulator is dimensioned (*e.g.*  $l$  is a length in meters), it can be used to test the *dependence on the scale* of the simulator *i.e.* no matter the order of magnitude of the inputs  $l$  and  $g$ , *in theory*, the same ratio  $l/g$  yields the same period  $T$ . This can be seen as a sort of *precision* of the simulator. (2) The *accuracy* is also tested: the observed period should match the theoretical value (possibly affected by the scale sensitivity). (3) The test can be performed both for dimensioned codes as described above and for dimensionless codes (*e.g.* codes that rescale to have  $l = 1$ ), to check that the adimensioning is done correctly.

In the following, the tests consist in the study of two dimensionless variables, and a curve that we call the *master curve* that either separates different regimes (**Lateral Buckling**, **Stick-Slip**) or on which the two variables must belong (**Cantilever**).

### 3.3.2 The Cantilever test

The first test we introduce is an old 2D test (Bickley, 1934) that is commonly used in the Soft Matter Physics community for measuring mechanical properties of rods, ribbons or plates (*e.g.* (Duclaux, 2006; Fargette, 2017)). This test is quite simple, as from only geometric observations, one can infer with good accuracy the elastic properties of the material. In our case, for purposes of validation, we use this measurement protocol in a reverse way: knowing the physical parameters, we test whether the simulators produce the correct geometry or not.

**Description.** The **Cantilever** test consists in a slender, uniform, and naturally straight rod or ribbon, clamped horizontally and deformed in 2D solely by the gravity, as depicted in the inset figure. For a rod with a circular cross-section of radius  $r$ , the parameters at play are the material density  $\rho$ , the length  $L$ , the area of the cross-section  $A = \pi r^2$ , the gravity acceleration  $g$  and the bending rigidity  $EI$ , with  $E$  the Young modulus and  $I = \frac{\pi}{4}r^4$  the second moment of area.

Considering only the physical parameters mentioned above, one can construct a characteristic length that measures the resistance to bending against the weight, namely the gravito-bending length  $L_{gb} = \sqrt[3]{EI/(\rho Ag)}$ . Thus, to compare the deformation regime of the rod of length  $L$ , one can use the following dimensionless ratio  $\Gamma = (L/L_{gb})^3$  which yields

$$\Gamma = \frac{\rho Ag L^3}{EI}. \quad (3.1)$$

Note that if the cross section is rectangular of width  $w$  and thickness  $h$ , the second moment of area is  $I = \frac{wh^3}{12}$  and  $A$  naturally is equal to  $wh$ . Moreover, for ribbons and plates, when  $w \gg h$ , one must also consider the Poisson ratio  $\nu$ . In the previous formula,  $E$  is replaced by  $E^* = E/(1 - \nu^2)$  (Shield, 1992). As a result,  $\Gamma$  is replaced by  $\Gamma^* = (1 - \nu^2)\Gamma$ , that can also be expressed as

$$\Gamma^* = \frac{\rho A^* g L^3}{D w}, \quad (3.2)$$

where

$$D = Eh^3/(12 [1 - \nu^2]) \quad (3.3)$$

is the bending rigidity of the plate.

We show in the next paragraph that the deformation of the rod (or ribbon or plate) is determined *solely* by the value of this dimensionless ratio. Low values indicate a bending dominated regime, and so a small deflection whereas larger values indicate a weight dom-



Figure 3.3: The **Cantilever** test.

inated regime, leading to large deflection. As a measure of the geometric deformation, we take the aspect ratio  $H/W$  of the shape at equilibrium (Figure 3.4, left). By performing both measures on fabricated rods and ribbons and simulations, we show that this aspect ratio is a good way to measure the quality of the output as the mapping between  $\Gamma$  and  $H/W$  is unique, and strictly monotonic (Figure 3.4, right). There is no need to measure curvatures distributions, the shape of the cantilever is determined by this aspect ratio.

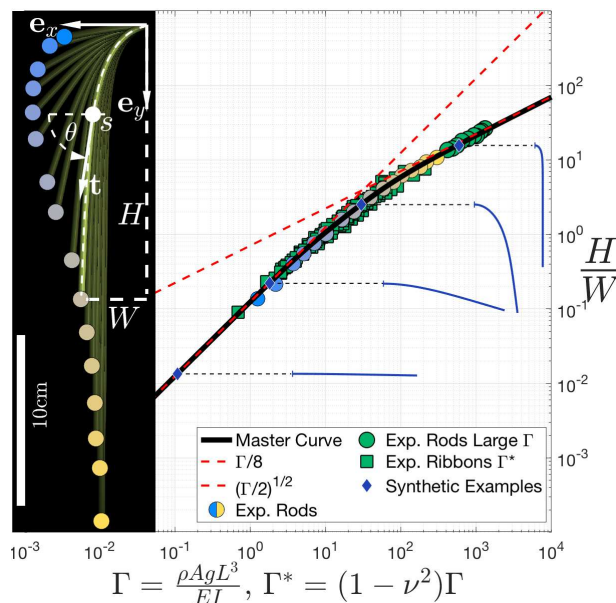


Figure 3.4: **Master curve and experimental validation** for the **Cantilever** test. *Left*: snapshots of experimental rods under gravity, with 16 different  $\Gamma$  values. *Right*: The experimental data (circles for rods; squares for ribbons) are in perfect agreement with the master curve (in black). The blue to yellow circles correspond to the experimental rods depicted on the left, the colour providing a one-to-one correspondence between shapes and points of the graph. Four synthetic rods, simulated with DISCRETE ELASTIC ROD (Bergou et al., 2008), are shown for illustration purposes.

**Master curve.** To compute the master curve, in our case the function mapping  $\Gamma$  to  $H/W$ , we can solve the boundary value problem describing the positions along the rod at equilibrium using the planar elastica equations - *i.e.* the Kirchhoff equations for 2D inextensible rods (Landau and Lipshitz, 1959).

Consider the rod to be parametrised by the curvilinear abscissa  $s$ , and note  $F$  the internal force and  $M$  the moment. The equilibrium reads as

$$\forall s \in [0, L] : \begin{cases} \frac{dF}{ds} + f_{ext} = 0 \\ \frac{dM}{ds} + t \times F = 0 \end{cases}$$



with  $t$  the unit tangent to the rod. The constitutive law links the bending moment  $M$  and the curvature  $\kappa = \frac{d\theta}{ds}$ , with  $\theta$  the angle between the horizontal and the tangent (see Figure 3.4, left):  $M = EI \frac{d\theta}{ds}$ . Finally, the only external force is the gravity  $f_{ext} = \rho A g e_y$ . Introducing the dimensionless variables  $\bar{s} = s/L$ ,  $\bar{x} = x/L$  and  $\bar{y} = y/L$ , we can put all of this together to have the boundary value problem

$$\left\{ \begin{array}{l} \frac{EI}{L^2} \frac{d^2\theta}{d\bar{s}^2} + \rho A g L (1 - \bar{s}) \cos \theta = 0 \\ \theta(0) = 0 \text{ and } \frac{d\theta}{d\bar{s}}(1) = 0 \\ \bar{x}(0) = 0 \text{ and } \bar{y}(0) = 0 \\ \frac{d\bar{x}}{d\bar{s}} = \cos \theta \text{ and } \frac{d\bar{y}}{d\bar{s}} = \sin \theta \end{array} \right\} \begin{array}{l} \text{Clamped-free boundary conditions} \\ \text{Kinematics.} \end{array} \quad (3.4)$$

We see that the first equation can actually be rewritten as

$$\frac{d^2\theta}{d\bar{s}^2} + \Gamma(1 - \bar{s}) \cos \theta = 0$$

where  $\Gamma$  is the only reduced parameter to control the rescaled boundary value problem, ergo justifying its usage.

Note that this law is derived in the case of an inextensible rod. However, even in the presence of stretching the main deformation mode in this experiment is the bending, leading to very little stretching and thus a correct evaluation of the bending.  $\curvearrowright$  We check this claim by simulating a simple finite-difference rod model allowing for both bending and stretching and we measure that the in-plane elongation remains indeed negligible ( $\approx 1\%$  at  $\Gamma \approx 10^4$ ).  $\curvearrowleft$

$\curvearrowleft$  With Sébastien Neukirch and Victor Romero, we try to solve numerically Equation 3.4 with shooting methods (Ascher et al., 1995). By doing so, we are able to obtain correct cantilevers until  $\Gamma \lesssim 200$ . However, for higher values, in our experiments, the shooting methods yields solutions that satisfy the boundary conditions, but are totally unphysical as depicted in the inset figure.  $\curvearrowright$  To solve the problem that becomes stiffer as  $\Gamma$  grows, we use the collocation-based continuation package AUTO07p (Doedel et al., 1991) that is robust enough to compute the full curve over 8 decades, plotted in Figure 3.4.

Besides, there exists asymptotic analytical expressions for small and large values of  $\Gamma$ . Their are defined by  $H/W = \Gamma/8$  when  $\Gamma \ll 1$  (Gere, 2004) and  $H/W = \sqrt{\Gamma/2}$  when  $\Gamma \gg 1$  and are also plotted in Figure 3.4.

**Experimental validation.** We explain here rather briefly the experimental validation process as it is out of the scope of this manu-

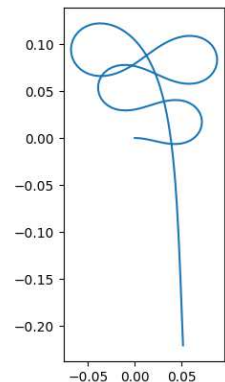


Figure 3.5: A result of a shooting method for  $\Gamma = 1000$ .

script. We refer the reader to the article, and the part of its supplementary document dedicated the fabrication for more details.

To confirm by ourselves the theory, we fabricate rods and ribbons with a controlled geometry using a polymer. The mass density  $\rho$  can be measured, and the only uncertain parameter is the Young modulus  $E$ , given in an approximate range by the polymer supplier. After using this protocol in its measurement form, we find a single value of  $E$  that makes all of the data match perfectly the theoretical curve, as depicted in Figure 3.4.

### 3.3.3 The Lateral Buckling test

The second test we introduce here can be seen as an extension in 3D of the previous test as we seek to evaluate the correctness of the impact of the width on the bending. This test is ergo dedicated to elastic ribbons and plates.

**Description.** We consider a uniform plate of length  $L$ , width  $w$  and thickness  $h$  with  $h \ll w \lesssim L$  hanging under the gravity like in the **Cantilever** test. However, the clamp here along the width is vertical and not horizontal. At equilibrium, the flat configuration in the vertical plane is either stable or unstable, making the plate buckle sideways as illustrated in Figure 3.6.

To test the simulators accuracy, we wish to test whether they actually recover the transition between these two regimes or not. However, although the study of buckling instabilities has been for long of interest in civil engineering (Michell, 1899) we did not find any study interested in building the boundary between these regimes, using moreover a reduced number of dimensionless parameters. The closest work has been done by Reissner (1995) who reduced the set of 2D equations describing the lateral buckling of a plate with all the weight at its end (and not distributed along the plate) to a set of 1D equations. We therefore build up a new master curve for this problem.

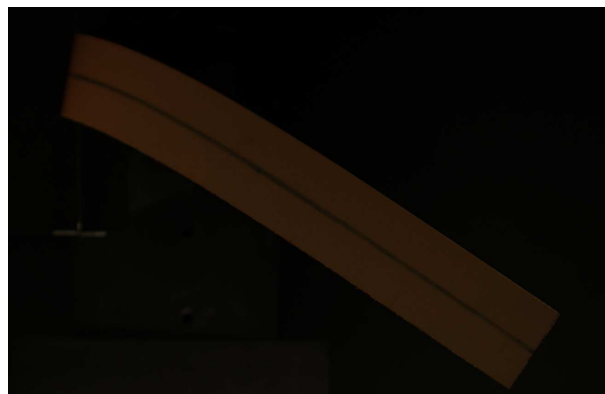


Figure 3.6: The **Lateral Buckling** test.

**Master curve.** From preliminary experiments made using paper sheets, we observe that when we increase the length  $L$  and thus  $\Gamma^*$ , the "ratio" between the gravity and the bending resistance along the length, while keeping the other dimensions fixed, there exists a limit value above which the flat configuration is always unstable. Similarly, decreasing the width of the paper strips also leads to the buckling.

◀ Then with Arnaud Lazarus and Sébastien Neukirch, following early numerical simulations using ©ABAQUS (Dassault Systèmes), FENICSSHELL (Hale et al., 2018) and LIBSHELL yield the following observations. First, the thickness  $h$  seems to have little impact on the results. This effect is easily explained as we are in the limit of thin plates. Next, we observe that for a given Poisson ratio  $\nu$  and length  $L$  the frontier between the buckling region on the graph  $(w, \Gamma^*)$  seems almost straight. Thus, we aim at finding this master curve  $\Gamma_C^*(w/L; \nu)$  in the form of an affine function. Note that the first parameter has been made dimensionless in order to define a proper scaling law. ▶

We compute this curve numerically with the ©ABAQUS software using high-order elements. For thin plates ( $h/L = 6 \times 10^{-4}$ ) of varying width, we find the minimal  $\Gamma_C^*(w/L)$ <sup>1</sup> above which the stiffness matrix becomes singular.

The resulting curve is plotted in Figure 3.7 (in black) with  $\nu = 0.35$  to compare against our fabricated plates. We can observe that when  $w/L \rightarrow 0$ , we recover the limit value that can be derived from the Kirchhoff rod theory:  $\Gamma_{c|kirchhoff}^* = 18.2(1 - \nu)\sqrt{1 + \nu}$  (Michell, 1899). Also, we see that  $\Gamma_C^*$  is almost linear *w.r.t.*  $w/L$  in the considered range  $0 < w/L \leq 1$ <sup>2</sup>, matching some of observations of Reissner (1995).

Finally, we consider the following approximated formula to define our master curve for this test

$$\Gamma_C^*(w/L; \nu) \simeq 18.2(1 - \nu)\sqrt{1 + \nu} + 14.5 \frac{w}{L}, \quad (3.5)$$

plotted in Figure 3.7.

**Experimental validation.** The master curve is compared against experimental results in Figure 3.7 produced using five plates with different widths  $w$  and thicknesses  $h$ , and whose length  $L$  of the suspended part is modified by adjusting the clamp position. As we are looking for an instability, this experiment is very sensitive for instance to the flatness of the fabricated plate or to the clamp orientation. Nonetheless, the observed data are in good agreement with the theory.

### 3.3.4 The Stick-Slip test

The **Stick-Slip** test originally comes from (Sano et al., 2017). It has been successfully used by Rasheed et al. (2020) to evaluate the ARGUS solver (Li et al., 2018a) before using

<sup>1</sup>This was done in practice by computing the critical value of the gravity.

<sup>2</sup>Above this range, we observe non linearities that we have not studied.

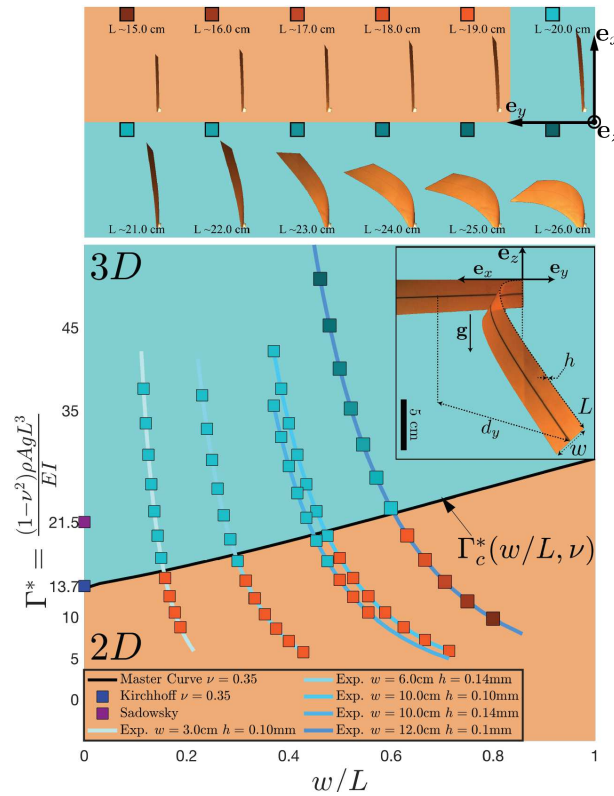


Figure 3.7: **Master curve** ( $\nu = 0.35$ ) and **experimental validation** for the **Lateral Buckling** test. *Top*: our experimental plates under gravity, with varying aspect ratio  $w/L$ . The background has been coloured to indicate whether the plate lies in 2D (orange) or has buckled in 3D (turquoise). *Bottom*: phase diagram where the computed master curve separates the 2D regime (in orange) from the 3D buckled regime (in turquoise). Five series of experiments with different values of  $w$  and  $h$  are presented. The points are colour-coded to depict the configuration observed (2D/3D). The darker set corresponds to the experiments represented in the top row.

it to generate data for their training set. We rebuild the mathematical formulation of the master curve of Sano et al. (2017) and verify it with a more elaborated set-up before using it to test static friction solvers presented in Section 1.2.1

**Description.** In this test, a strip of length  $L$ , rigid enough to neglect gravity ( $\Gamma \ll 1$ ) is clamped downwards and pushed quasi-statically against a solid substrate as illustrated in the inset figure. The vertical displacement of the clamped end is noted  $\delta_y$ . The free end of the strip is in contact with the substrate and so is submitted to a normal repulsion force  $P$  and a tangential friction force  $Q$ .

As depicted in the phase diagram in Figure 3.9 (left), there exist three different regimes. Under the assumption of negligible gravity as mentioned above, the phase diagram depends only the static friction coefficient  $\mu$  between the strip and the substrate and the vertical strain  $\epsilon_y = \delta_y/L$ . In the stick phase, only the free end of the strip is in contact and the friction forces are enough to hold this free end while the strip buckles. Conversely, in the slip phase, the internal forces of the strip have overcome the friction forces and the free end has slipped away from its initial location.

Finally, in the extended contact phase, the friction forces are strong enough to hold the free end until the buckling extends the contact to a portion of the strip.

The boundary of these phases are the master curve that interests us and that we compute below, and the lines  $\epsilon_y = \epsilon_{y,c} \simeq 0.33$  and  $\mu = \mu_c \simeq 0.36$ . All of these three curves meet at the triple point  $(\epsilon_{y,c}, \mu_c)$ .

**Master curve.** Like for the **Cantilever** experiment, the master curve can be computed using a boundary value problem. Let the strip be parametrised by the curvilinear abscissa  $s$  going downwards, and let us note  $\bar{s} = s/L$ ,  $\bar{x} = x/L$  and  $\bar{y} = y/L$  the dimensionless parametrisation and coordinates and  $\theta$  the angle between the tangent and the horizontal as shown in Figure 3.9 (left). The only external force is  $(P, Q)$  at the end, left unconstrained. All of this gives the following boundary value problem.

$$\left\{ \begin{array}{l} \frac{d^2\theta}{d\bar{s}^2} = \frac{PL^2}{EI} \sin\theta + \frac{QL^2}{EI} \cos\theta \\ \bar{x}(0) = 0, \bar{y}(0) = 0 \text{ and } \theta(0) = \pi/2 \\ \bar{x}(1) = 0, \bar{y}(1) = 1 - \epsilon_y^* \text{ and } \frac{d\theta}{d\bar{s}}(1) = 0 \\ \frac{d\bar{x}}{d\bar{s}} = \sin\theta \text{ and } \frac{d\bar{y}}{d\bar{s}} = \cos\theta \end{array} \right. \begin{array}{l} \text{Clamped end} \\ \text{Immobile free end} \\ \text{Kinematics.} \end{array} \quad (3.6)$$

The resulting  $Q$  and  $P$  can be used to compute the value of the friction coefficient



Figure 3.8: The **Stick-Slip** experiment.

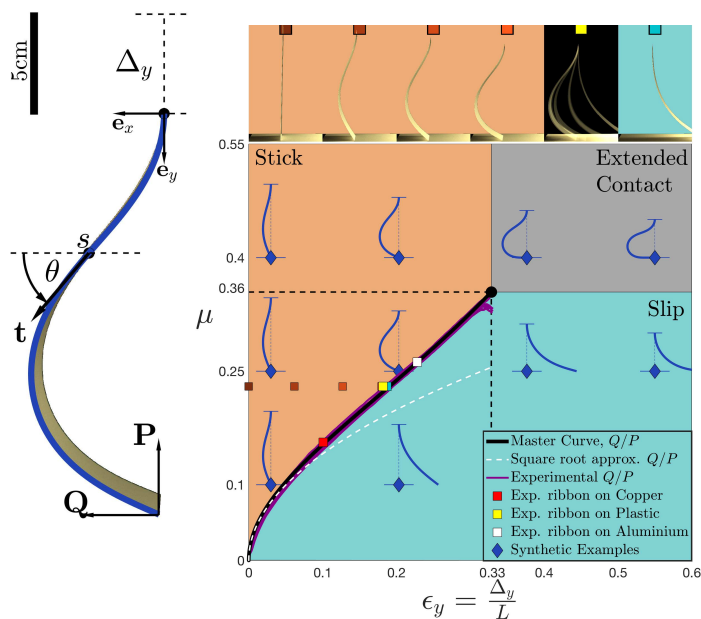


Figure 3.9: **Master curve and experimental validation** for the **Stick-Slip** test. *Left*: Experimental snapshot of a **Stick-Slip** test with a high friction coefficient, for  $\epsilon_y = 0.2$ . In blue is superposed the solution of Equation 3.6 with the same parameters. *Top-Right*: several snapshots of the experiment. The background has been coloured to indicate whether the rod is sticking (orange) or slipping (turquoise after slippage, black during the dynamic transition). *Bottom-Right*: phase diagram where the master curve (in black) separates the sticking regime (in orange) from the slipping regime (in turquoise). The extended contact region (in gray) is not used in our validation protocol. The numerical (in black) and experimental (in purple) master curves  $Q/P$  are plotted. Ten synthetic rods (in dark blue), simulated with SUPER-HELIX 2D (Bertails et al., 2006) coupled with SO-BOGUS (Daviet et al., 2011), are depicted on the phase diagram for illustration purposes.

required to have this static equilibrium by simply computing  $\mu = Q/P$ .

This problem can also be solved analytically up to the second order in  $\theta$ , yielding

$$\frac{Q}{P} \simeq 0.445\sqrt{\epsilon_y}. \quad (3.7)$$

However, as shown in Figure 3.9 (right), this approximation plotted in white is not sufficient for our purpose. Instead, we solve the problem numerically using shooting methods and plot the computed curve.

**Experimental validation.** To validate this master curve, we attach a polymer strip to a motor that is quasi-statically pushing it downwards against a substrate with a controlled vertical strain. Two load cells attached to the motor and the substrate respectively measure the normal and tangential force. We used different lengths for the strip and different

substrates.

With smooth substrates (copper, plastic, aluminium), the samples slip, and at slippage, the measured ratio  $Q/P$  and vertical strain  $\delta_y$  match the theoretical curve.

Using a rough substrate (grit paper), ensuring that  $\mu > \mu_c$ , performing the experiment allows to reconstruct experimentally the whole master curve until the sample enters in the extended contact phase. This experimental master curve is plotted in purple in Figure 3.9.

## 3.4 Codes tested

In this section, we list some of the codes tested that are relevant to us. As mentioned previously, we do not mention the rods and ribbons codes as they are out of scope of the manuscript.

### 3.4.1 Plates & shells

The codes presented below are based on models already mostly described in 1.1.2.

As DISCRETE SHELL (Grinspun et al., 2006) is one of the model used in our inversion algorithm (see Section 5.3.1), we present the results of this code regarding the aforementioned tests. We use a home-made implementation that was done using symbolic calculus and automatic differentiation, making us quite confident that we correctly implemented the original model. To test only the bending model, we choose to report to combine the bending energy of DISCRETE SHELL with the StVK membrane model of LIBSHELL. The results are reported under the name DISCRETE SHELL (+ LIBSHELL).

For the sake of completeness, we also present the results of Bridson et al. (2003)'s model that is very close to the previous one. In the original paper, Bridson et al. (2003) uses the sines of the dihedral angles, easier to compute, while Grinspun et al. (2006) uses directly the dihedral angles. However the implementation of (Bridson et al., 2003)'s model in ARGUS (Li et al., 2018a) (that is essentially ARCSIM (Narain et al., 2012) when frictional contact is not used) that we test also uses the dihedral angles. Nonetheless, for small deflections, this should not affect our results. The main resulting difference between both models is that Bridson et al. (2003)'s bending energy must be multiplied by 12 to get that of Grinspun et al. (2006). For the sake of completeness, we also test a variant of ARCSIM with that scaling factor. The corresponding code is noted DISCRETE SHELL + ARCSIM. Besides, ARCSIM by default adaptively remeshes during the simulation to improve the performance. The code with this option disabled is called ARCSIM NON ADAPTIVE.

Then, we also include in our tests LIBSHELL (Chen et al., 2018), that is implementing the discrete Koiter shell energy. The code is [open-source and available online](#), and we also use it for the inversion. For reasons that will appear clear in the results section, we also test Bridson et al. (2003) and Grinspun et al. (2006)'s bending energies with the

LIBSHELL’s in-plane stretching energy. These codes are respectively noted DISCRETE SHELL (+ LIBSHELL) and BRIDSON 2003 (+ LIBSHELL).

Finally, we also test our implementation of PROJECTIVE DYNAMICS (Bouaziz et al., 2014), already described in Section 2.1.2. The in-plane energy is defined by springs, although stretching should be negligible, and the bending energy is defined by a discretisation of the square of the differences between mean curvatures using the Laplace-Beltrami operator.

### 3.4.2 Frictional contact

For the frictional contact, we will present the results of the SO-BOGUS solver (Daviet et al., 2011), couple with the SUPER-HELIX 2D rod model. As introduced in Section 1.2.3, SO-BOGUS is a constrained-based solver aiming to solve accurately the DFCP described in Equation 1.23 in a Gauss-Seidel-like fashion. In practice, this is the solver we use coupled to a plate model to perform the direct simulations of clothes to check the output of our inversion algorithm (see Section 5.6.2).

We also test it in its ARGUS version (Li et al., 2018a), where it is coupled to ARCSIM (Narain et al., 2012). The version without the adaptive remeshing is also tested under the name ARGUS NON ADAPTIVE.

Lastly, we test our *Projective Friction* model (Ly et al., 2020) presented in Chapter 2.

## 3.5 Evaluation and results

### 3.5.1 Our methodology

The codes tested are diverse. Some provide static equilibrium computation while some are only dynamics, and in the original study, some codes are using dimensioned parameters while others are dimensionless. Therefore, even if the theoretical results are quite general thanks to dimensionlessness, the codes must be in practice also compared evaluated consistently.

The general approach adopted is (1) for each protocol, define a range of physical parameters on which the simulator has to match the theoretical result and then (2) for each code, find an optimal set of solver parameters that allows the code to pass the test (if it exists). We detail below this approach. Note that for (1), the range can be adapted if one is targetting an application with specific scenarios (material, characteristic dimensions and displacements known for instance).

**Passing a test.** Aiming at validating the accuracy, one has to decide the precision that is requested from the codes. As for the physical parameters range, it may be adjusted depending on the criticality of a potential underlying application. In our case, we asked



for the computed curves to visually matches the theoretical curve (*i.e.* close in the sense of Hausdorff) but all over the parameter space. The requested precision to reach an **OK** is therefore rather loose, but failing to match even a small portion of the curve results in a **KO**.

**Solver calibration.** For each test, we perform a large number of simulations by sampling a wide parameter space. The resulting simulations ergo range from simple ones (small displacements) to complex ones (large displacements), seemingly requiring different tunings of the solver. However, as we aimed to keep the benchmarking process simple, we did not want to tune each simulation. Instead, for each code, we asked for a common set of solver parameters to be used for the whole test. In other words, between the reported results of a same code within a same test, the numerical configuration is the same and only the physical parameter vary. In practice, tuning for hard cases also allowed easier cases to pass except for PROJECTIVE DYNAMICS in the **Cantilever** test.

To get the **OK** stamp, the solver tuning also needs to be consistent. This means that when we refine the configuration in ways that should decrease the error (*e.g.* more elements in the input, decreasing the timestep etc.), the output should be consistent accordingly to the previous value: if it was matching the master curve, then with these finer parameters, it should also match the curve.

The computation time is only remotely considered in our study. We only tested solver parameters that allow the experiments to be conducted in a "reasonable" time (a few days maximum for the whole test). The configuration for an **OK** code might not be practical for some applications given the resulting computation time.

**Fairness.** Properly and objectively testing a code is not an easy task, especially if one is not familiar with the underlying model or the options. This may lead to a tedious tuning process and might lead to a **KO** due to a wrong usage of the code.

To try to avoid that, all the testers are rather experienced programmers. We spent some time on each code to understand and compare with the written publication. Overall, we could also check that the observed results and limitations were consistent with the original claims. Finally, we investigated the **KO** to try to understand the causes of failure and give a potential fix.

With this approach, we are rather confident that there is no false positive, given the requirement of the tests and all the preliminary simulations done to prepare for the displayed results. However, the possibility of a false negative (wrongly **KO**-ed) cannot be excluded, and can be shown in a future benchmark.

**Reproducibility.** To be able to judge the fairness of our tests and to check or enrich the results list, the benchmark process needs to be reproducible. We refer the reader to the supplemental documents of the original publication (Romero et al., 2021) that

extensively cover and detail several aspects of our study. In the "Supplementary", we give convergence graphs for the simulation, all the details of the experimental fabrication and validation, and show some of the inputs used. In the "Recipe Manual", we detail the algorithms used in practice, as well as some general recommendations. [The data master curves are also available.](#)

### 3.5.2 <Results for the Cantilever test>

<For this test, we use two meshes of lengths  $L = 0.5m$  and  $L = 0.8m$ , both of width  $0.3m$ . The surfaces are uniformly meshed using 3 resolutions: Res  $-$  has around 55 vertices along the length, Res 0 around 120 and Res  $+$  around 240.

$\nu$  is fixed to 0.3 and the  $\Gamma^*$  space is sampled using the two lengths and by varying  $E$  between  $10\text{ GPa}$  and  $10\text{ kPa}$ ,  $\rho$  in  $\{12.87\text{ kg}\cdot\text{m}^{-3}, 1.287\text{ kg}\cdot\text{m}^{-3}, 1287\text{ kg}\cdot\text{m}^{-3}\}$  and  $h$  in  $\{1\text{ mm}, 2.5\text{ mm}\}$ . In practice, we process by continuation by decreasing  $E$  with all other parameters fixed to go from small displacements to larger ones.

The results are gathered in Figure 3.10. We detail below the results for each of the codes.

**LIBSHELL.** As mentioned in Section 1.1.2, the LIBSHELL code has two in-plane models (Saint-Venant-Kirchhoff and Neo-Hookean) and three ways to compute the normals for the bending (MidEdgeAverage, MidEdgeSin, MidEdgeTan). This model deriving from a physical model, plugging in the physical parameters is straightforward.

By performing first the test with the standard resolution on the 6 combinations, we can see in Figure 3.11 that the in-plane model has no influence.

Moreover, we can see in the right column of Figure 3.11 that despite MidEdgeSin and MidEdgeTan being of "higher order" for the normal computation, MidEdgeAverage seems to perform slightly better in this 1D experiment, although they all agree fairly well. The difference is even more striking if we compare the results at the resolution Res  $-$ , depicted in Figure 3.12 However, raising the resolution to Res  $+$  yields a good agreement for the three of them (see Figure 3.13), **OK**-ing this model.

**DISCRETE SHELL (+ LIBSHELL).** For this code, the link between the physical parameters and the hinge energy described in Section 1.1.2 is not as straightforward. However, following (Tamstorf and Grinspun, 2013), we take as bending coefficient  $k_b$  in Equation 1.12 the bending modulus  $D$  of a plate defined in Equation 3.3<sup>3</sup>

As we can see in Figure 3.14, at the default resolution Res 0, the computed curve is far from the master curve. However, with the increased resolution Res  $+$ , this model also passes the test and gets an **OK**.

---

<sup>3</sup>Note that we wrote our bending energy with a factor  $1/2$ , while Grinspun et al. (2003) do not. Ergo taking  $k_b = D$  is consistent within our framework *w.r.t.* (Tamstorf and Grinspun, 2013).

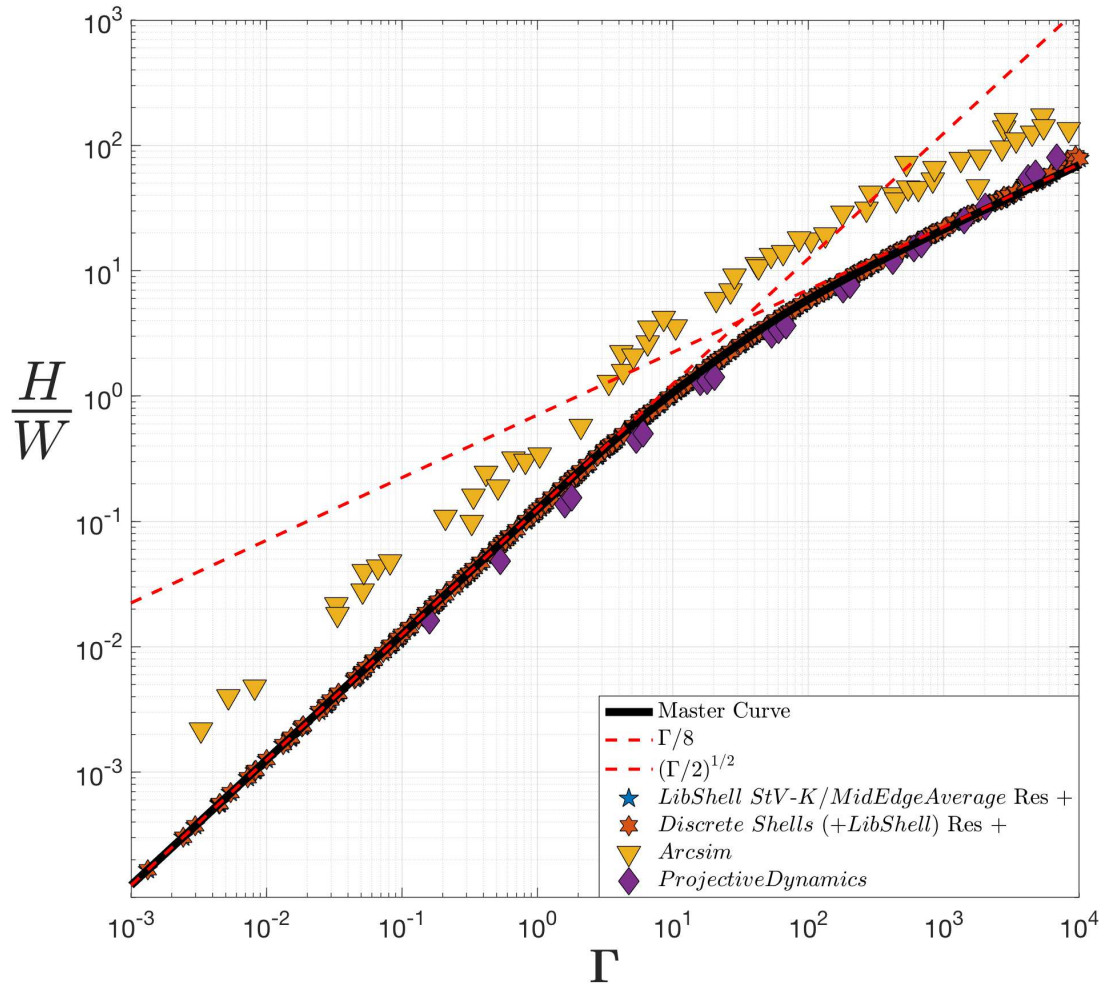


Figure 3.10: Results on the **Cantilever** test. LIBSHELL and DISCRETE SHELL are perfectly aligned with the master curve.

Interestingly, Grinspun et al. (2006) report that the hinge energy of DISCRETE SHELL does not converge in the general case, except in the case of triangular meshes with equilateral triangles and "when used to compute the mean curvature". The meshes used were close to this ideal case, and the experiment configuration implies a zero-curvature along the width, and so the curvature along the length is twice the mean curvature, which probably explains these good results.

ARCSIM. ➤ As mentioned at the beginning of the chapter, experiments using the ARCSIM code have been conducted by Abdullah Haroon Rasheed. ⚡ For ARCSIM, the input format is more complex as it is based on the work of Wang et al. (2011) who approximate the anisotropic behaviour of cloth by interpolating between orthotropic matrices. Ergo, in our case, we need as inputs duplicates of an isotropic elasticity matrix, defined

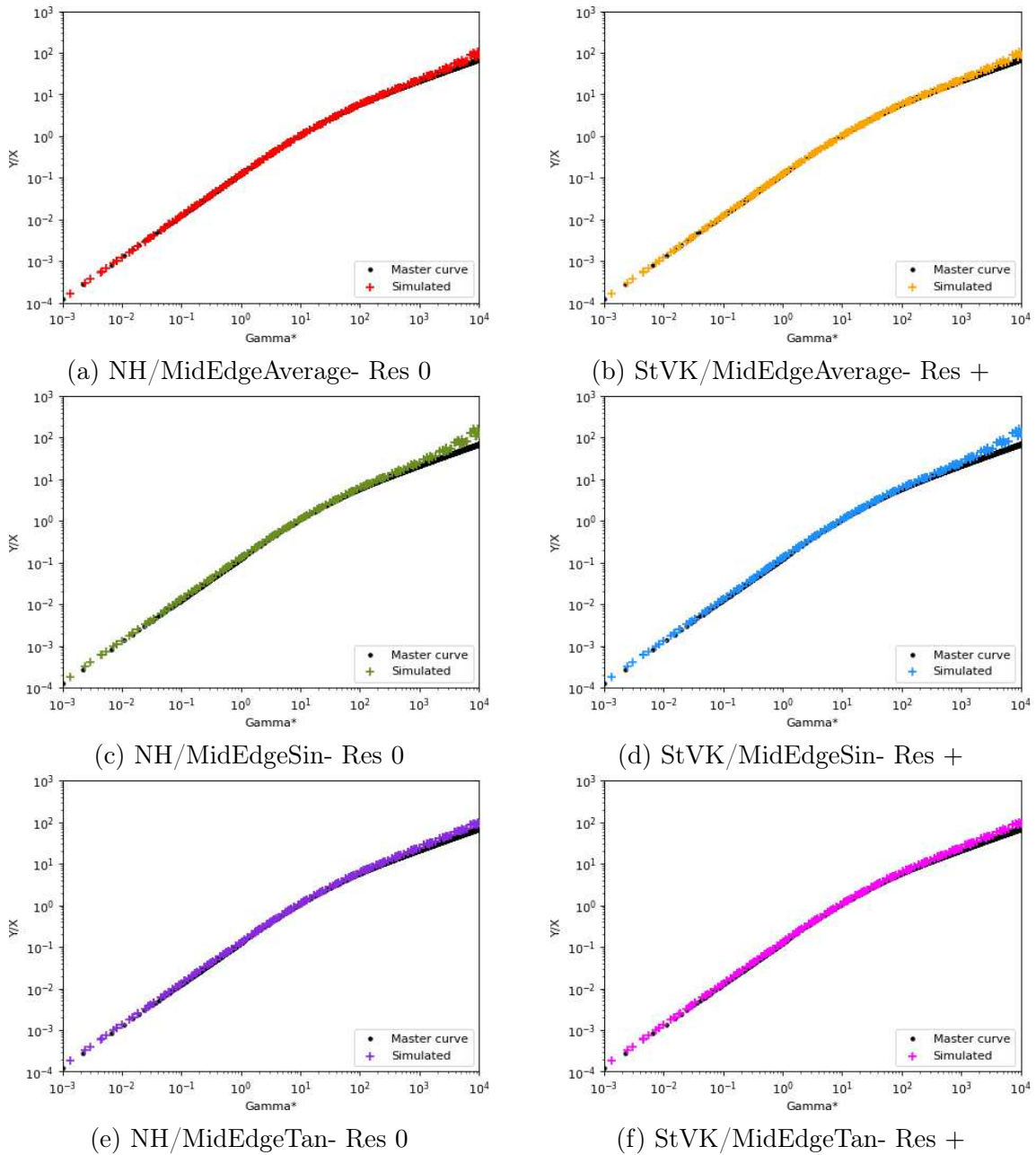
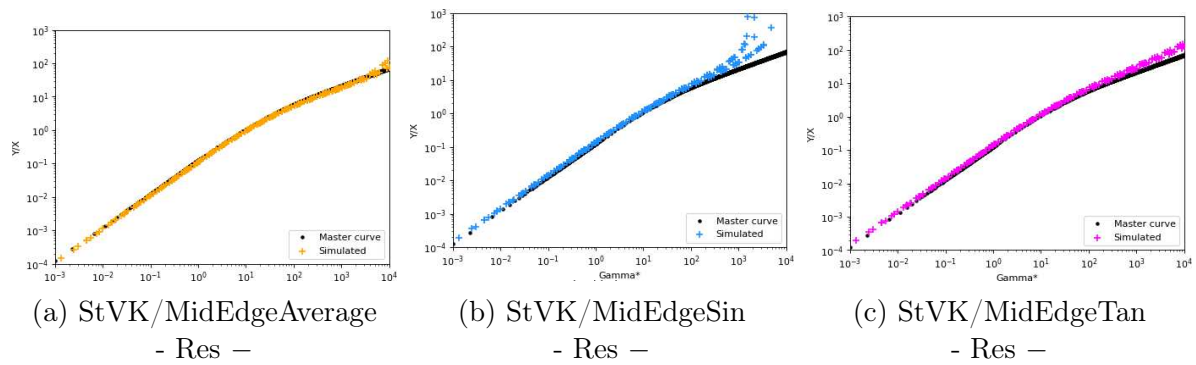
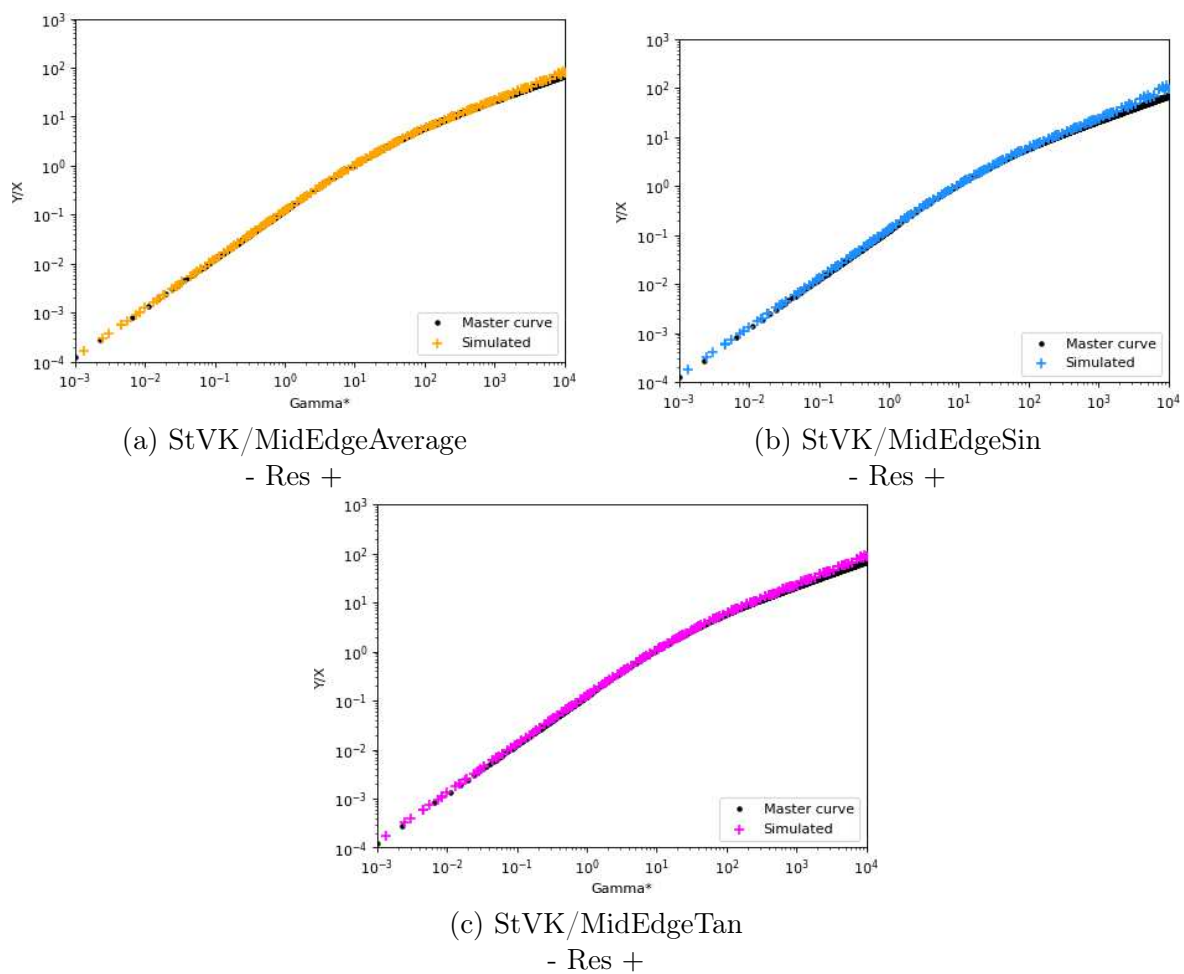


Figure 3.11: **Cantilever** test on the 6 variants of LIBSHELL at resolution Res 0. Each graph took about 1 or 2  $h$  to compute. The in-plane model has no influence and MidEdgeAverage seems to perform slightly better.

in Appendix A and duplicates of a diagonal matrix containing the bending coefficient<sup>4</sup>.

At the light of the previous result on DISCRETE SHELL, taking the same bending coefficient should lead to a shifted curve. However, the data we obtain are shifted *and*

<sup>4</sup>Similar our us, the ARCSIM implementation also has an "extra" 1/2 factor

Figure 3.12: **Cantilever** test on 3 StVK variants of LIBSHELL at resolution Res -.Figure 3.13: **Cantilever** test on 3 StVK variants of LIBSHELL at resolution Res +. Each graph took about 7 or 8 h to compute. All yield a good agreement with the master curve.

*scattered* as shown in Figure 3.15. Removing the adaptive remeshing does help a little to reduce the spreading, but does not remove it completely. In contrast, improving the resolution does not help as we can see by comparing Figures 3.15b and 3.15c. To confirm

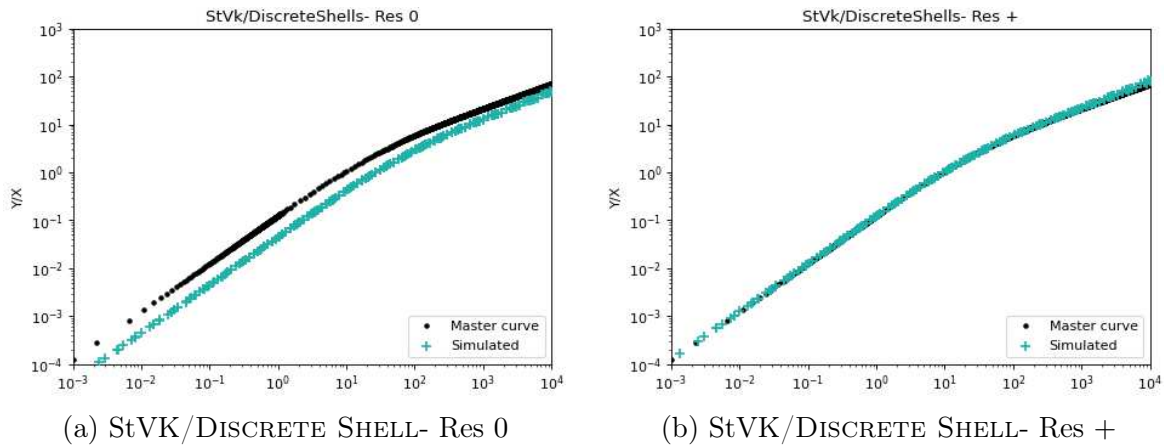


Figure 3.14: **Cantilever** test on DISCRETE SHELL (+ LIBSHELL) at resolutions Res 0 and Res +. The Res + resolution has a good agreement with the master curve.

that there should only be a shift, we test [Bridson et al. \(2003\)](#)'s bending model in BRIDSON 2003 (+ LIBSHELL), yielding Figure 3.15d.

As we have not found any way to fix it, we are forced to give this code a **KO**. Correcting this spreading could be of interest as this code is quite popular, and would allow people to use different materials from the ones fitted by [Wang et al. \(2011\)](#).

PROJECTIVE DYNAMICS. As for the previous codes, there is no evident link between the physical parameters and the bending weight given in Section (?). Considering the similarity of origin between the bend energy and that of ([Grinspun et al., 2003](#)), we use the bending modulus  $D$ . After manually fitting a scaling factor, we find that  $\lambda_{\text{bend}} = 0.25D$  yields a rather good agreement.

However, as depicted in Figure 3.16, the results obtained are quite scarce as we did not find solver parameters that would enable the simulation to converge on all the parameter space. Indeed, it seems that the different scaling between the weights with the stretching weight being much bigger than that of the bending yields a poor convergence for the local-global approach.

Nonetheless, the code produces fairly good results in the range  $0.1 < \Gamma < 500$ , beyond which it crosses the curve and depart from it. We therefore attribute this code a **KO**, although we recall that PROJECTIVE DYNAMICS was designed for real-time animation and not accurate simulation. >

### 3.5.3 <Results for the Lateral Buckling test >

<Similarly to the **Cantilever** test, the **Lateral Buckling** test also means to evaluate the accuracy of the bending. However, this test is more demanding as it includes 3D effects through the width. In addition, the evaluation is not on the correctness of the

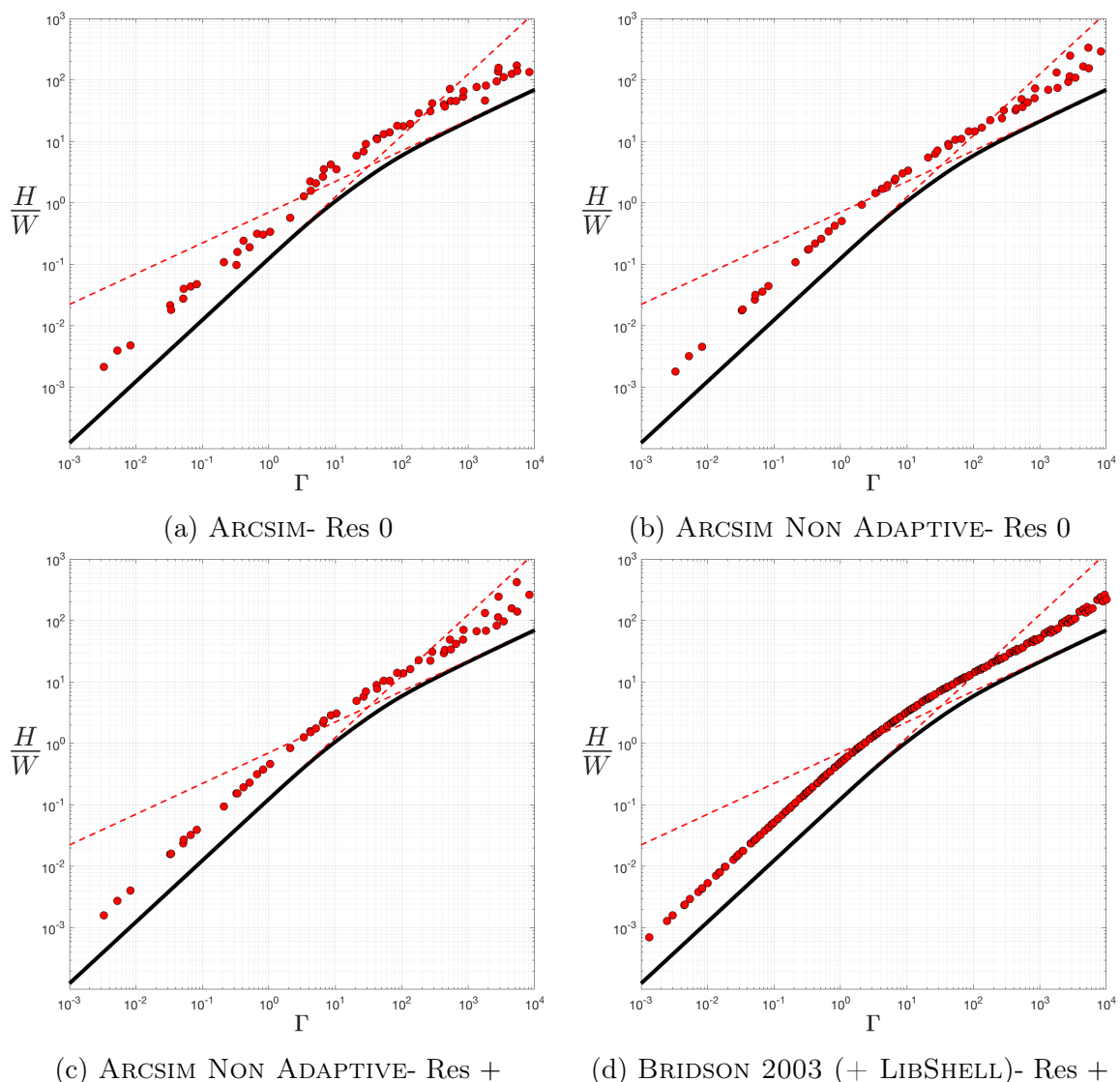


Figure 3.15: **Cantilever** test on ARCSIM, and a reimplemention using LIBSHELL. The results with the base code are shifted and scattered.

shape, but on the reproduction of pitchfork instability at the right value. Thus, we only retain for this test the codes that got an **OK** at the **Cantilever** test: DISCRETE SHELL (+ LIBSHELL) and LIBSHELL. Like un the previous test, the elastic law has little impact here, so we use the StVK material.

In our simulations, we use 10 meshes of length  $L = 1 m$  and of width  $w$  varying between  $0.1 m$  and  $1 m$ . As previously, we have different meshes resolution Res 0, Res + and even Res ++ that have respectively around 50, 75 and 100 vertices along the length and  $50 \times w/L$ ,  $75 \times w/L$  and  $100 \times w/L$  along the width.

To perform this test, we set the thickness  $h$  to  $1 mm$  and the Poisson ratio  $\nu$  to 0.35 (the one of the polymer used for the experimental part). Then the Young modulus  $E$  is

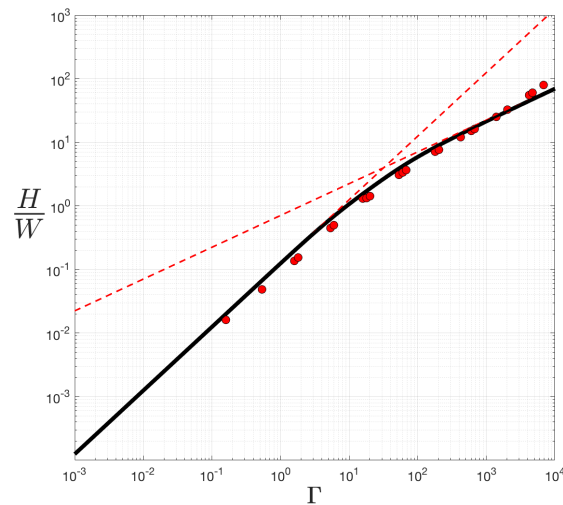


Figure 3.16: **Cantilever** test on PROJECTIVE DYNAMICS at resolution Res 0.

adjusted so as the bending modulus  $D$  is 1, and  $\Gamma^*$  is controlled by adjusting the mass density  $\rho$ .

In order to make sure to capture the bifurcation, we start by computing the equilibrium at  $\Gamma_{max}^* = 40$  with a clamp slightly tilted from the vertical, ensuring the transverse buckling. Then the clamp is set back to the vertical, and we track the evolution of the equilibriums as we decrease  $\Gamma^*$  to  $\Gamma_{min}^* = 10$ .

DISCRETE SHELL (+ LIBSHELL). As we can see in Figure 3.17a, with the protocol described right above, the plate goes directly to the flat configuration as soon as the clamp is vertical. We have to raise  $\Gamma_{max}^* = 80$  to see that the bifurcation threshold seems to be linear *w.r.t.* the width but much larger than the expected one. However, increasing the resolution from Res 0 (Figure 3.17b) to Res + (Figure 3.17c) and further yields very little difference on the threshold, indicating that we cannot retrieve the correct value.

The **KO** here can be explained following the discussion on the results of DISCRETE SHELL for the **Cantilever** test. The curvature here is not along a single direction, which leads out of the favourable case described by Grinspun et al. (2006).

LIBSHELL. All 3 variants of LIBSHELL pass relatively well this test as we can see in Figure 3.18. However, unlike in the **Cantilever** test where MidEdgeAverage was performing better, here it is the two "second-order" model MidEdgeSin and MidEdgeTan that converge the faster. They match perfectly the curve at resolution Res + (see respectively Figures 3.18e and 3.18g), while MidEdgeAverage start to match at resolution Res ++ (see Figure 3.18c). We recall however that the master curve is a linear approximation. Nonetheless, LIBSHELL gets an **OK** thanks mainly to 2 of its variants.



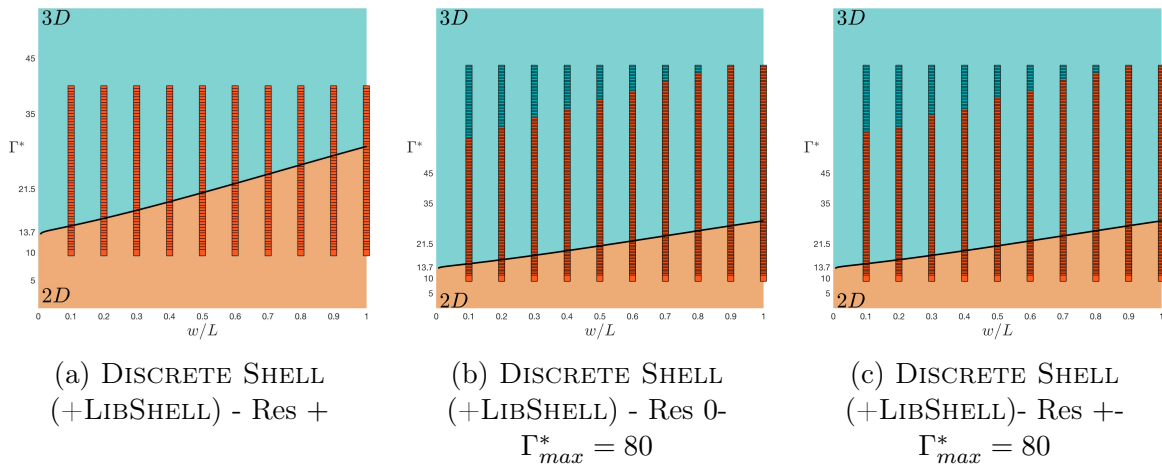


Figure 3.17: **Lateral Buckling** test on DISCRETE SHELL (+ LIBSHELL). The colour of the dots represent whether the configuration is 3D (turquoise) or 2D (orange). The computed threshold is much bigger than predicted.

**Rotation experiment.** Considering the difficulty of capturing a bifurcation, one could also think of an experiment similar to **Cantilever** where we observe the geometry.

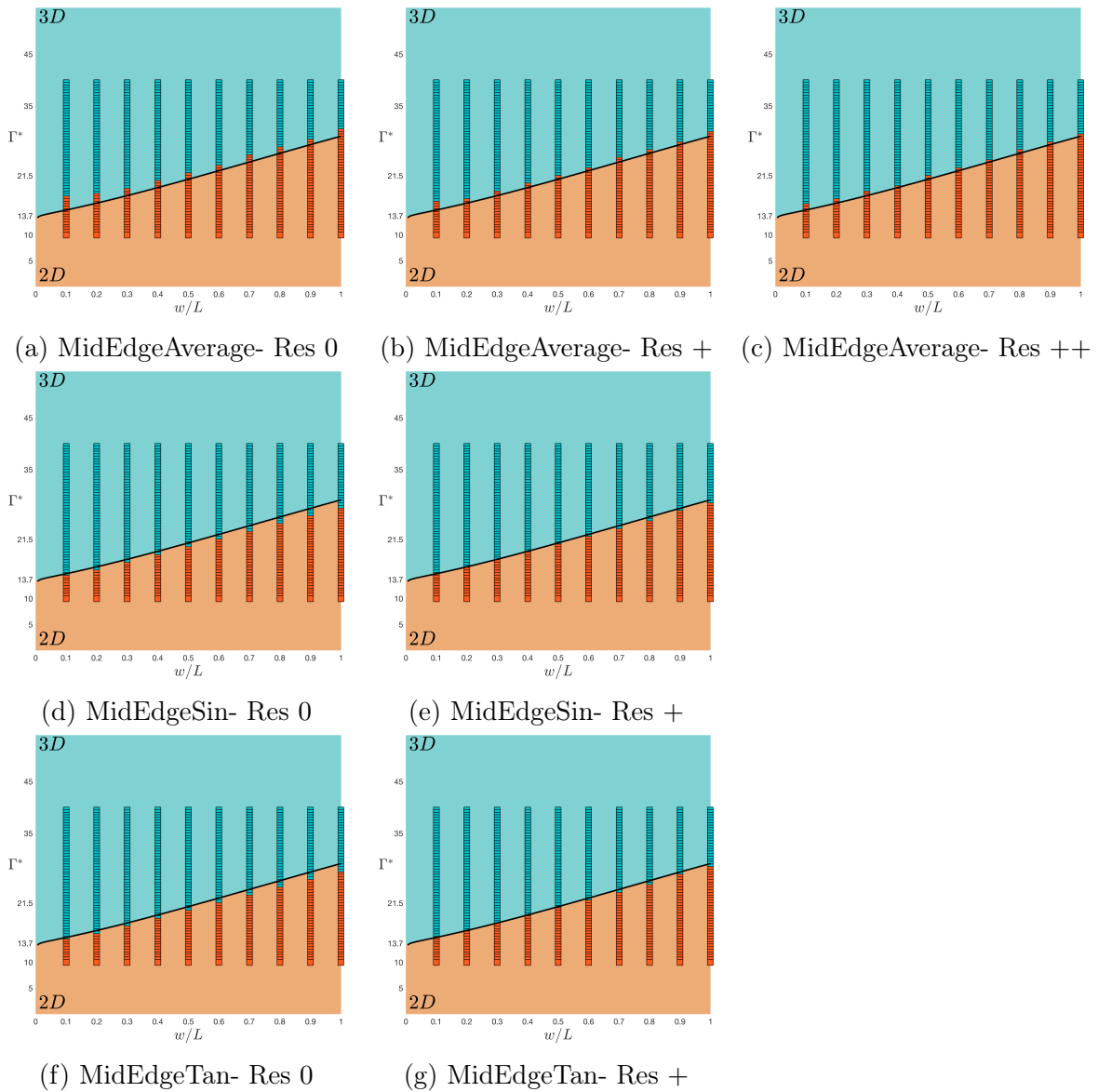
➤ This *rotation* variant has originally been designed by Sébastien Neukirch. As we have not derived a master curve in this case, we compare the results of the simulations are compared against the experimental data measured by Victor Romero. ➤ By noting  $\theta$  the angle between the vertical and the clamp, this scenario consists in measuring the lateral displacement of the free end at equilibrium with a clamp going from the horizontal position  $\theta = \pi/2$  (**Cantilever**), to the vertical position  $\alpha = 0$  (**Lateral Buckling**). The results of the simulations at resolution Res 0 are plotted in Figure 3.19.

To reproduce the experimental set-up, we take  $h = 0.1 \text{ mm}$ ,  $\rho = 1410 \text{ kg} \cdot \text{m}^{-3}$ ,  $\nu = 0.35$  and  $E = (1 - \nu^2)7.75 \text{ GPa}$ <sup>5</sup>. We have two strips of  $\Gamma^* \approx 10.5172$  and  $\Gamma^* \approx 42.1353$ , that yield respectively  $L \approx 0.17 \text{ m}$  and  $L \approx 0.27 \text{ m}$ . Note that both of them, we are "far" from the instability values which are  $\Gamma_C^* = 22.27$  for the first strip and  $\Gamma_C^* = 19.12$  for the second one. Thus, at  $\theta = 0$ , the shorter strip should have a 0 lateral displacement while the longer one should buckle.

MidEdgeAverage has a good agreement, albeit slightly off and MidEdgeSin and MidEdgeTan have a very good agreement with the data. DISCRETE SHELL (+ LIBSHELL) still exhibits a very stiff behaviour at resolution Res 0, but works surprisingly well at resolution Res +.

This is probably due to the bending modulus in this case  $D = Eh^3/(12(1 - \nu^2)) \approx 6.5 \times 10^{-4}$  being much lower than in the original protocol where it is set to 1. Although DISCRETE SHELL indeed directly relies on the bending modulus, LIBSHELL's bending coefficient also has a dependence in  $Eh^3$ . Therefore, both models should be affected the

<sup>5</sup>7.75 GPa is the measured value  $E^*$  thanks to the **Cantilever** setup.

Figure 3.18: **Lateral Buckling** test on LIBSHELL.

same way by the bending modulus, while it appears they are not. ➤

### 3.5.4 Results for the Stick-Slip test

For this test, one must be careful, as in the experimental setup, to push the strip very slowly to ensure the quasi-static regime and not introduce any inertia effect. Moreover, the test also requires a regime where  $\Gamma \ll 1$  *i.e.* the gravity is negligible *w.r.t.* the internal forces.

In practice, we use a mesh of length  $L = 0.2m$  and of width  $w = 0.01m$ , with around 50 vertices along the length. For the other parameters, we take  $h = 1mm$   $\rho = 10^2 kg \cdot m^{-3}$ ,

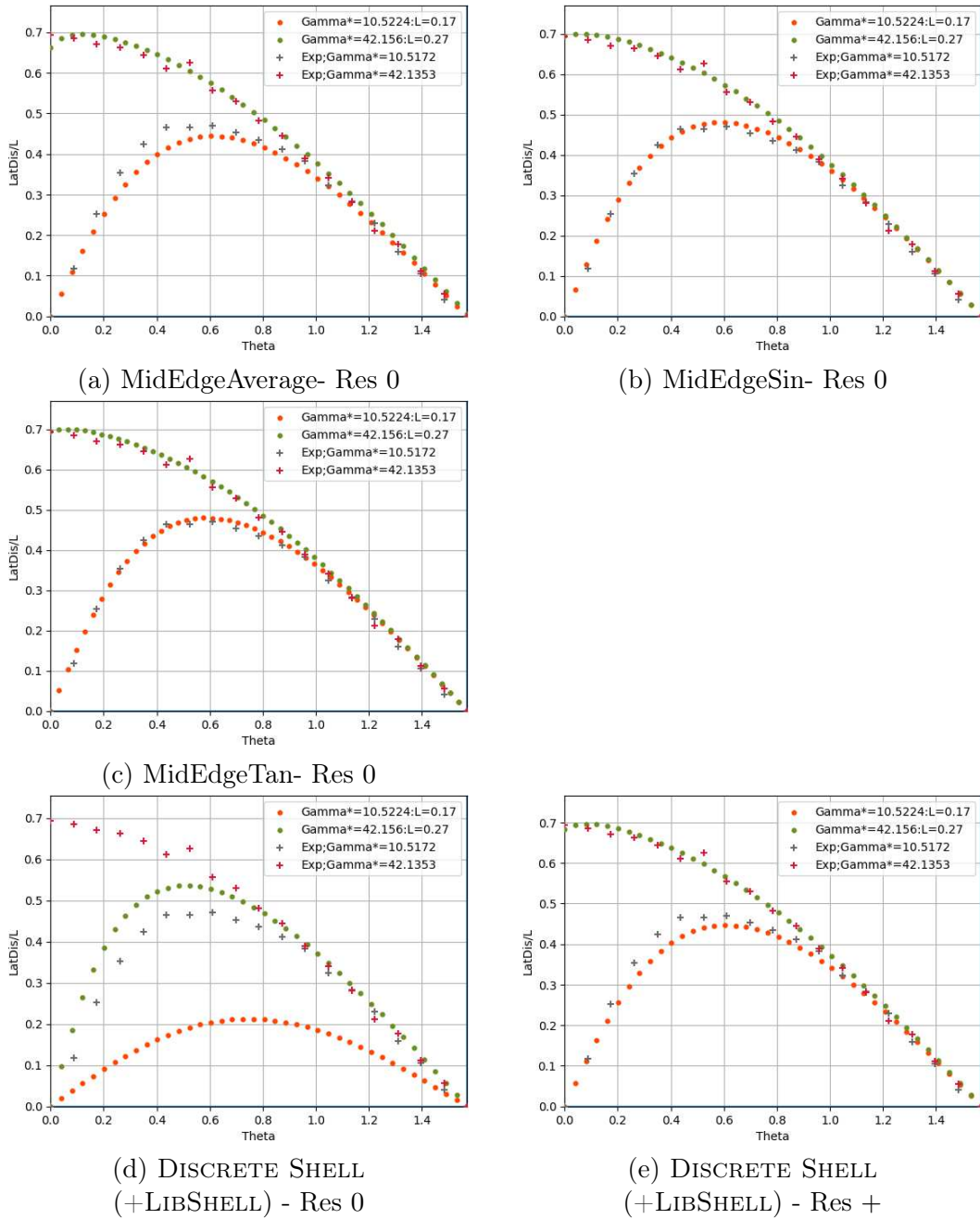


Figure 3.19: Rotation variant of the **Lateral Buckling** test. Simulations are represented by dots and experimental data by crosses.

$EI/L^2 = 10N$ ,  $\nu = 0.3$  and  $g = 0m \cdot s^{-2}$  which yields  $\Gamma^* = 0 \ll 1$ .

SO-BOGUS **and** ARGUS. As written before, we test the implicit solver SO-BOGUS coupled "raw" with the rod model SUPER-HELIX 2D (Bertails et al., 2006) ( $\approx$  a 2D version of the original implementation in (Daviet et al., 2011)), and in ARGUS (Li et al., 2018a).

We can see in Figure 3.20a that the coupling is passing very well the test and get an **OK**. However, to reach this precision, the timestep has to be set very low to  $0.5\text{ ms}$ . In Figure 3.21, we note that the higher the friction coefficient is, the lower the timestep should be to accurately capture the stick/slip limit.

For ARGUS, we encounter a phenomenon previously seen on the **Cantilever** test: the adaptive remeshing seems to mess up with the physics, as depicted in Figure 3.20b and already noticed by (Rasheed et al., 2020). As soon as the remeshing is dropped, we retrieve in Figure 3.20c the **OK** results previously obtained by the coupling SO-BOGUS and SUPER-HELIX 2D, although we loose one big advantage of the ARGUS code that gets a **KO**.

Note that although ARCSIM did not pass the **Cantilever** test, we see that it does not affect this test. This illustrates the property that this graph is independent of the material as long as gravity is negligible (Sano et al., 2017).

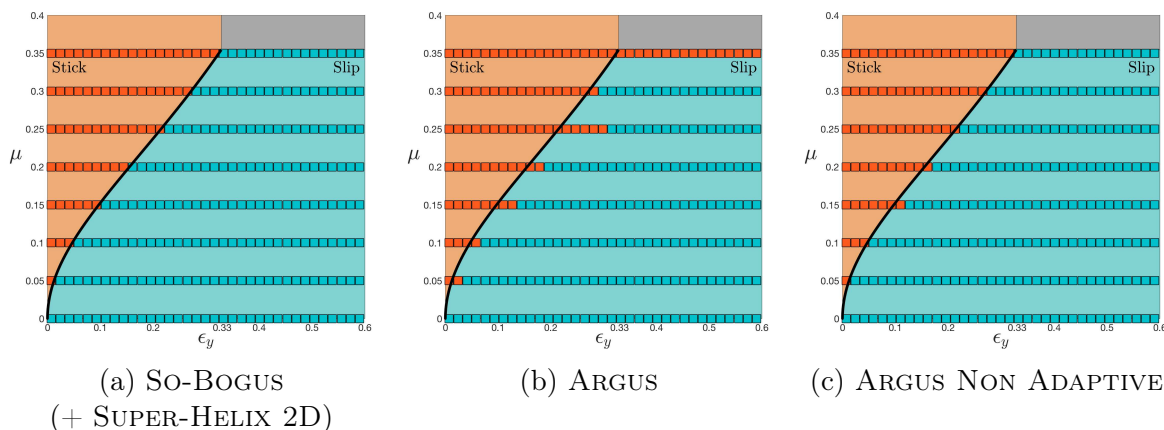


Figure 3.20: **Stick-Slip** test for SO-BOGUS and ARGUS.

**PROJECTIVE FRICTION.** For this code, we try to catch as much precision as possible by taking a timestep of  $0.1\text{ ms}$  and up to 500 local/global iterations per timestep. We see that the algorithm, although based on an explicit estimation of the frictional contact forces, actually manages to catch fairly well the stick/slip boundary for low friction coefficients. However, for  $\mu \geq 0.25$ , it generates excessive sticking, due to a poor convergence behaviour as seen in Section 2.3.3.

## 3.6 Discussion and conclusion

In conclusion, I presented three tests among the four of the original paper that aim at assessing the physical accuracy of simulators of slender structures and static frictional contact. Inspired by measurement protocols and known results of the Soft Matter Physics community, our research team has designed tests that we believe relatively easy to setup,

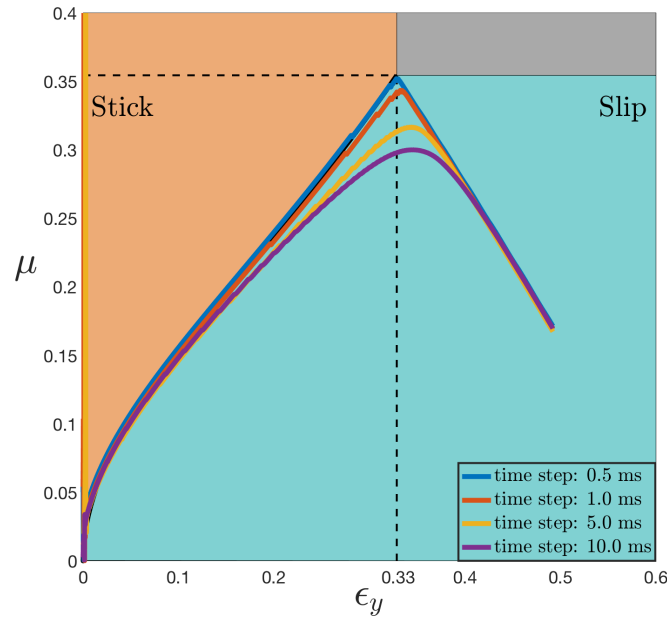
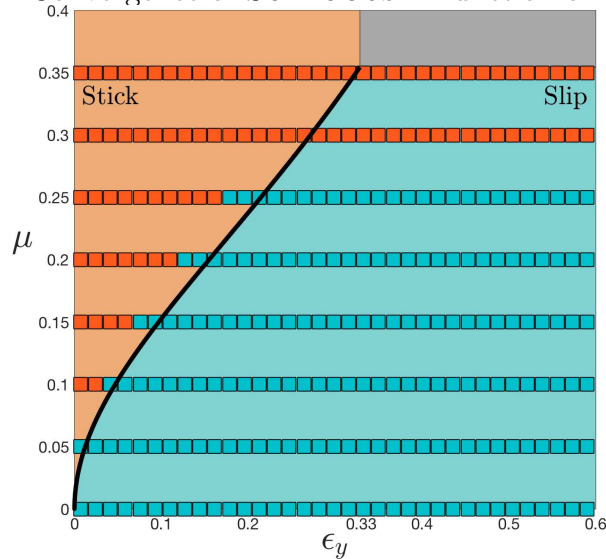


Figure 3.21: Convergence of SO-BOGUS in function of the timestep.

Figure 3.22: **Stick-Slip** test for PROJECTIVE FRICTION.

yet rich thanks to the dimensionless parameters and scaling laws to compare against. Moreover, such scaling laws can also be used to fix currently used simulators or to fit simulators with non-physical inputs instead of using data-driven methods. Besides, although our framework is mainly targetting for simulators designed to be accurate. However, for faster simulators more dedicated to animation, running such tests could still be informative, and perhaps reveal some potentials for quick prototyping.

Note that in this contribution, we address the problem of *validation* which is to check the physical accuracy of a simulator with an underlying model. However, the other aspect of simulators we only evoked is the *verification* which is about verifying that the numerical

solving is correct. For instance, in our simulations, we quickly check the convergence *w.r.t.* numerical parameters, such as the number of elements, the timestep etc. But we did not conduct a full study. Other numerical aspects were also not discussed, such as the *objectivity i.e.* the invariance of the numerics *w.r.t.* rigid motion (Crisfield and Jelenić, 1998) or, better known in Computer Graphics, the *non-locking i.e.* the convergence of the model *w.r.t.* some physical parameters (*e.g.* when the thickness of a plate tends towards 0, or when the Poisson ratio tends towards 0.5) (Arnold and Brezzi, 1997).

Moreover, as noted in the introduction, although we tried to address richer cases than those commonly used, for instance with large displacements, our tests are far from being sufficient to assess the richness of the motions used in Computer Graphics. Building more complex scenarios is an interesting but challenging future work.

Nonetheless, we hope that this study will encourage researchers in Computer Graphics to develop similar validation protocols and to better assess their model for perhaps diffusing them in other communities, such as in (Brun et al., 2012; Isvoranu et al., 2019; Gaume et al., 2019).

## Part II

Inverse design of shells under frictional contact, and application to inverse garment design





# Chapter 4

## State of the art

In this chapter, I review the works in the Computer Graphics community related to our topic of inverse plate and shell design, with the application of inverse cloth design in mind.

In a first section, I present the inverse design problems addressed in Computer Graphics. Indeed, while the first inverse problems involved "basic" elements such as fibres and shells, the recent work has started to tackle problems involving more complex structures especially in the fields of 3D printing and architecture design. Given the large range of covered problems, I do not claim to be exhaustive and rather give a broad overview with a focus on thin structures, and more specifically on shells and clothes.

Then in a second section, I present some works on the developability of surfaces. The characterisation and design of developable surfaces has for long drawn the interest of various scientific communities from geometry to engineering ([Hilbert and Cohn-Vossen, 1952](#); [Nolan, 1971](#)) and more recently physics ([Liu et al., 2007](#)) However, here, I focus on the literature about the discrete developability in the search of a practical characterisation criterion to incorporate in our inversion process.

## 4.1 Inverse design

In computational mechanical engineering *inverse problems* can be defined as the reverse process of a *direct* problem such as for instance a the computation of a mechanical equilibrium. Considering a model assumed to describe some phenomenon, instead of computing the outcome from some inputs, the goal is to compute from an observed state what unknown input could have produced this output.

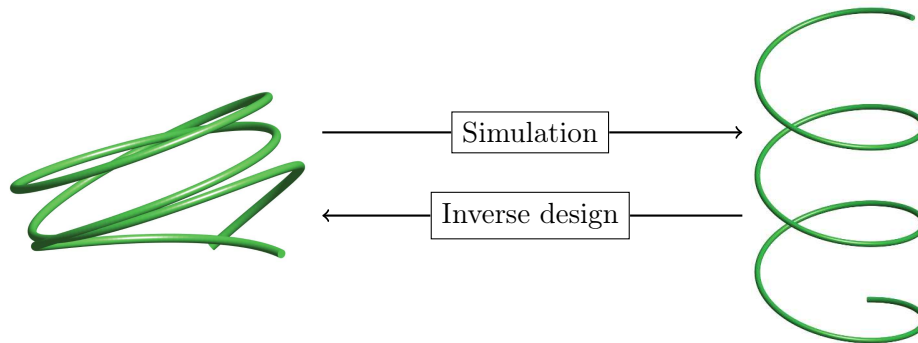


Figure 4.1: Example of an inverse design problem: [Romero et al. \(2018\)](#) look for the unknown rod rest shape that, suspended under gravity, gives a helix shape at equilibrium.

Following the paradigms described in [Beck and Woodbury \(1998\)](#), inverse problems across the many fields of engineering can be classified into two categories. On the one hand, *inverse measurement problems* consist in searching for unknown material parameters. As an example, one could think of seismology where the soil composition is reconstructed from the observed seismic waves, or material characterisation in mechanics where the properties of a sample are inferred by observing the deformations under some loads. On the other hand, *inverse design problems* aim at retrieving an unknown geometry. For instance, one can think of aeronautics where the shape of aerofoils are optimised to obtain some aerodynamics properties, or, closer to our work, the inverse suspended rod design illustrated in [Figure 4.1](#).

In the following, we will restrict ourselves to the inverse design problems that have been addressed in the Computer Graphics community. Although it might appear narrow, we will see that it actually covers a large range of problems as hinted in the [Section 3.2.1](#), and thus we will focus more on the works involving thin elastic structures and frictional contact.

### 4.1.1 Inverse design in Computer Graphics

The original motivation for inverse design in Computer Graphics was to provide sag-free simulations for animation. Indeed, physics-based simulation has proved to be a powerful tool to help artists generate realistic and vivid animations. However, directly simulating

from an artist's design may result in a sagged outcome, as the designed shape represents the final appearance, and thus includes the enviroing physics such as gravity and contact, as depicted in Figure 4.2.

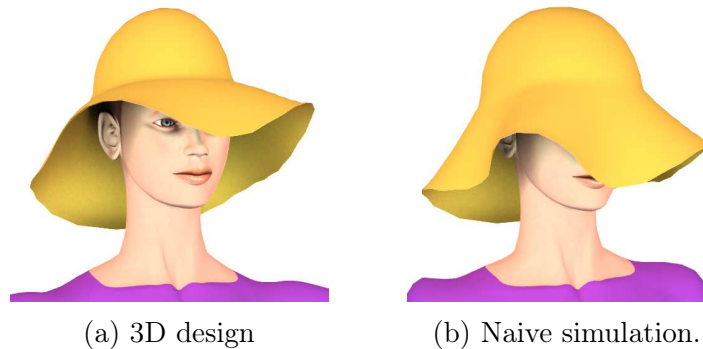


Figure 4.2: Naive simulation of a 3D design leads to sagging.

Subsequent work is therefore needed to compensate *i.e.* to "remove" the physics from the target shape in order to preserve the intended design in the simulation. Inverse problems aim at computing such "force-free" shape, called *rest shape* or *natural shape*, through techniques specialised depending on the type of structures addressed.

Moreover, with the increase of accuracy of simulators, able to closely match reality (see Section 3.2.1), inverse problems are also developed for virtual prototyping applications, where the aim is to fabricate the rest shape to reach the target design.

Twigg and Kačić-Alesić (2011) gave perhaps the most general method that consists in minimising the total forces applied by optimising the rest shape - in their case, described by springs rest lengths as they considered mass-spring systems. Although simple, their method may fail to make the forces vanish, and thus to preserve the intended design.

**Thin elastic rods.** In Computer Graphics, the earliest work goes back to (Hadap, 2006) in the context of inextensible thin elastic rods to facilitate hair posing. As the author formulated his strand model as a sequence of rigid segments attached by joints, he was able to use inverse dynamics methods developed in robotics (Featherstone, 1987) to compute the joint forces required to preserve the intended design under gravity.

Then, based on the super-helix 2D inextensible rod model (Bertails et al., 2006), Derouet-Jourdan and Bertails-Descoubes (2010) noticed that, thanks to the curvature-based formulation, finding a rest shape that guarantees a stable equilibrium against gravity boils down to solving a linear system. The authors managed to provide sufficient conditions on the material properties for the existence of the solution. In more recent work, Bertails-Descoubes et al. (2018); Romero et al. (2018) proved the uniqueness of the natural shape for continuous Kirchhoff rods hanging under gravity, up to the choice of the material parameters. They have also extended their work in the 3D discrete case,

including dry friction to better account for the fibres interaction in hair (Derouet-Jourdan et al., 2013).

More recently, Hafner and Bickel (2021) formulated a stability criterion when gravity is negligible for 2D clamped-clamped rods based solely on their geometry. Then they adapted this criterion for an inverse design process of structures made of strips subject to gravity that takes the form of a linear problem.

**Thin elastic shells.** To the best of our knowledge, there exists no reduced parametrisation such as the ones of inextensible rods models aforementioned and thus we can not reuse the associated simple inverse formulations.

For the case of rubber balloons, that are elastic membranes subject to pressure forces, Skouras et al. (2012) used an Augmented Lagrangian method that was able to deal with large stretching deformations to optimise the deflated rest shapes. In a subsequent work (Skouras et al., 2014), the authors extended their method to optimise the shape of the panels composing balloons made of a quasi-inextensible material. However, they had to use a relaxed formulation for the elastic energy density to handle compression while keeping the number of nodes tractable. While this technique works well for in-plane deformations as they drew from the tension field theory, it is however not easily exploitable in other contexts.

Panetta et al. (2021) have recently addressed a related inverse problem where the shell deformation is controlled by the inflation of air channels wandering through the surface, and so they optimise the positions of these channels. Their method shares a similar structure to ours, that is, the difference between the target and the simulated shape is retro-propagated to the rest shape thanks to a gradient-based optimisation.

**Volumetric objects.** For volumetric objects, the deformation behaviour is controlled by three dimensional elasticity. Finite-elements methods with linear elasticity models such as Saint-Venant-Kirchhoff or Neo-Hookean can thus be used to provide a quite simple relationship between the displacement and the stress.

Skouras et al. (2013) adapted their previous Augmented Lagrangian method for the computational design of actuated deformable objects. To reach a set of target equilibrium positions, they optimise the actuators locations and also the material distribution inside the objects.

For the case of suspended objects, (Chen et al., 2014) proceeded by continuation on the gravity. As they managed to reformulate the Neo-Hookean constitutive law as a quadratic form with the help of auxiliary variables, they could use an asymptotic numerical expansion to track their solution during the continuation and reach faster convergence speed than a classical Newton-Raphson method.

Ulu et al. (2019) aimed at minimising the weight of hollow objects while keeping them resistant to a prescribed set of loadings. To do so, they used a level-set-like representation

that allowed them to optimise locally the thickness of the object.

The methods described are efficient to deal with the deformation of solids but can not be applied to the case of shells which exhibit “higher-frequency” deformations through the folds, making the inverse problem harder to solve.

**Assemblies & meta-material design.** Recently, the Computer Graphics community has cast its interest on the design and the fabrication of more complex structures for architecture and additive manufacturing.

Here, we do not mean to be exhaustive as the range of materials and applications addressed is wide and starts to depart from our purpose. It includes, and is not limited to, rods structures (Pérez et al., 2015), weaved ribbons (Ren et al., 2021), deployable surfaces using auxetic materials (Konaković-Luković et al., 2018; Chen et al., 2021) or printed microstructures (Zhu et al., 2017).

For more complete reviews, the reader may refer to the following recent states of the art reports on the computation of assemblies of rigid elements (Wang et al., 2021), on additive manufacturing (Attene et al., 2018) and more generally on fabrication-oriented design (Bermano et al., 2017; Bickel et al., 2018).

However, it is worth noting that most of the physics-based inverse design processes, including our method, follow the same pattern. Starting from an initial guess for the rest configuration, a corresponding deformed shape is computed thanks to a procedure involving physical and/or geometric considerations. The rest configuration is then optimised to reduce the gap between the target shape and the deformed shape often using a gradient-based minimisation.

However, each algorithm is then specialised based on its geometric features and its physical constraints. In our case, we specifically address the case of shells that we want to be in a *stable* static equilibrium with the presence of *frictional contact*.

### 4.1.2 Cloth design

**Pattern adjustment.** In real life, the traditional way for making garments require to design 2D patterns, *i.e.* fabric patches made from cut flat panels that are then sewn together to create the garment. Early works in cloth simulation mimicked this process in order to dress virtual characters before animating them (Carignan et al., 1992). To improve the garment modelling process, Volino et al. (1995) designed an interactive environment where the user can simultaneously edit the patterns and visualise the 3D resulting shape under gravity on a virtual character thanks to a fast draping simulator. This interactive physics-based tailoring process has become the standard workflow and have been integrated in many commercial software packages such as *Marvelous Designer* (2010). The process has also been the object of further work in academia. Umetani et al. (2011) improved the process speed and also allowed the pattern to be adjusted by performing

dragging operations on the 3D shape. [Berthouzoz et al. \(2013\)](#) proposed a tool that parses scanned patterns schematics to automatically segment the patterns and infers the stitching process.

In the vein of the 3D to 2D edition introduced by [Umetani et al. \(2011\)](#), [Bartle et al. \(2016\)](#) have also developed a tool that adjusts the cloth pattern after cutting, lengthening or merging operations done by the user in the 3D space. In their approach, the pattern adjustment is done by an iterative gradient-free fixed point procedure. At each iteration, the current shape is computed using a cloth simulator. Then for each triangle of the cloth mesh, the inverse of the deformation difference between the deformed shape and the target garment is applied to its counterpart on the pattern. Finally, the resulting set of disconnected triangles is embedded again in a 2D mesh using the ARAP (As Rigid As Possible) algorithm ([Liu et al., 2008](#)). Their method is fast enough to allow interactive edition and is also blind to the cloth simulator used, provided that the mesh topology is unchanged. However, it converges only if the natural shape of the garment results from a contraction of the target. Although this assumption is reasonable in the case of a detailed garment pose, we consider it too restrictive to deal with arbitrary surfaces coming from 3D design or reconstruction.

Another limitation of [Bartle et al. \(2016\)](#)'s method is the difficulty to add small details such as folds. [Li et al. \(2018b\)](#) depart from this approach of trying to reach a precise target shape designed by the user. Instead, they proposed to the user to draw strokes representing the folds to guide the adjustment process, in the limits of the chosen material parameters. They reused [Bartle et al. \(2016\)](#)'s gradient-free method, using for the distance metric a "design energy" to compare with the strokes.

Concurrently, [Wang \(2018\)](#) has also relaxed the constraint to reach a precise 3D target as his goal was to provide a tool to adjust designed garments to other bodies sizes. He thus optimised the patterns using a "fitting" metric with a technique similar to ours, evaluating equilibria and modifying the patterns using a gradient-based optimisation scheme. His method is very efficient, thanks to careful implementation choices and the reduced patterns parametrization that only consider their boundaries. However, in his approach contact is dealt with using penalties, and friction is not taken into account.

Finally, [Yang et al. \(2018b\)](#) proposed a method for the recovery and adjustment of garment patterns by combining machine learning and shape optimisation. First, their trained neural network recovers from a single picture the garments types and their patterns from a library and estimates the underlying body. Then, they alternatively optimise the material parameters and parameters controlling the geometry of the pattern. On the one hand, the material is optimised by minimising the distance between the average fold curvature estimated from the picture and the average discrete curvature of the garment simulated by ARCSIM ([Narain et al., 2012](#)) On the other hand, the sizing of the pattern is optimised in a gradient-free minimisation (Particle Swarm Optimisation) aiming at matching the 2D silhouette of garment on the picture and that of the garment also

simulated by ARCSIM. Finally, they optimise the body position by also using a the gradient free PSO algorithm. Although impressive, their method is limited by the clothes and bodies dataset and the pattern library. Moreover, to use the gradient-free optimisation, they resorted to a non-negligible number of assumptions to limit the size of the solution space for the parameters.

Although working using patterns seems to be a natural way to deal with garments, we find it too restrictive. Indeed, such approach only allows to work with well-known garments for which patterns exist, or requires expert knowledge on the design of patterns for more complex shapes. As we aim to apply our method to fanciful garments designed for the animation, for which patterns may not exist, we depart from the pattern-based approach and chose to work with shells.

**Free-form cloth modelling.** Another approach to model clothes is to depart from patterns and directly sculpt the final 3D shape. Currently, two families of methods are available: geometric design and automatic capture.

On the one hand, geometry editing tools allow the artist to directly sculpt the garment around a virtual character (see *e.g.* (Porumbescu et al., 2005), or modelling software packages such as ?). Further dedicated tools, such as sketch interfaces (Igarashi and Hughes, 2003; Turquin et al., 2007; Li et al., 2017) may also ease the design process and the final result can be transferred to other characters using geometric design transfer methods (Brouet et al., 2012; de Goes et al., 2020).

On the other hand, 3D reconstruction from images or videos has considerably improved over the years. From the reconstruction of static poses (see *e.g.* (White et al., 2007; Bradley et al., 2008)), recent techniques are able to capture the dynamic geometry with folds and wrinkles, and also to propose a segmentation of the worn garment from the body underneath (see *e.g.* (Pons-Moll et al., 2017; Leroy et al., 2018)).

The advantage of the geometric sculpting methods is that they allow the user to design shapes freely and to focus solely on the 3D shape, while capture is limited to existing clothes. The user can thus create any type of garment, whether classic or fancier, without considering the patterns that may require more and more expert knowledge as the complexity of the garment increases.

However, all of these methods produce a geometry that is oblivious of any physical consideration. The result therefore cannot be easily interpreted mechanically as the deformed shape of an underlying rest shape subject to boundary conditions and/or some motion. The naive way to simulate such garments then consists in simply plugging the designed shape as its own rest shape, causing a sagging motion under gravity that may diverge from the initial design and ruin all the modelling efforts. Our method is suited to remedy to such situations.

### 4.1.3 Inversion with contact and friction

Most of the inverse design methods described above consider clamped boundary conditions, such as hanging structures, while disregarding contact and friction.

[Bartle et al. \(2016\)](#)'s method includes indirectly the treatment of any type of contact as it blindly relies on the plugged-in cloth simulator for the simulation of contacts. Indeed, the gradient-free optimisation of the rest shape relies only on geometric considerations no matter how the deformed shape is computed. However, this method suffers from the drawbacks described in the previous section, converging only when the natural shape is a contraction of the deformed shape.

In their design transfer, [Brouet et al. \(2012\)](#) constrained the geometry of the cloth to be outside the body using inequality constraints. [Wang \(2018\)](#) used penalty functions depending on the distance to the body to prevent penetration while keeping a smooth potential. More recently, [Geilinger et al. \(2020\)](#) used a smooth formulation that also included a regularised law for the friction. This allowed them to keep the trajectories differentiable for their control framework.

Finally, to the best of our knowledge, only [Derouet-Jourdan et al. \(2013\)](#) has dealt with the Coulomb friction law (described in Section 1.2.1) in an inversion process. Under some reasonable assumptions on the natural shape, they have shown that the inversion of static inextensible rods with dry friction can be cast in a form that is similar to the forward problem, and thus can be solved using the direct solvers of the literature ([Daviet et al., 2011](#)). However, the guarantee of stability previously provided in ([Derouet-Jourdan and Bertails-Descoubes, 2010](#)) is lost.

## 4.2 Discrete developability

In the continuous setting, a developable surface is defined as a surface with zero Gaussian curvature everywhere. In other words, it is a 3D surface that is constructed by isometrically bending and gluing a 2D panel. The design of such surfaces has ergo been for long of interest in the several fields of industrial manufacturing ([Ferris, 1968](#); [Tang and Wang, 2005](#); [Chalfant and Maekawa, 1998](#)) as they can be used to design surfaces made of materials that hardly stretch such as metal sheets for instance.

Here, I do not aim at reviewing all the richness of the work done in this domain, ranging from reconstruction from boundaries ([Frey, 2002](#); [Rose et al., 2007](#)), design and simulation using the rulings-based definition ([Solomon et al., 2012](#); [Tang et al., 2016](#); [Charrondière et al., 2020](#)), to folding and origami design ([Jiang et al., 2019](#); [Demaine and Tachi, 2017](#)).

I rather propose to focus on the characterisation of the developability of meshes, namely *discrete developability*, as the goal is to incorporate such criterion in our mesh-based inversion algorithm.



The classic formulation of discrete developability comes from the discretisation of the Gauss-Bonnet formula. As methods to discretise on surfaces may differ, mainly in the weight computation, one of the possible resulting definition of the discrete Gaussian curvature for triangle meshes is

$$\kappa_G(v) = \frac{2\pi - \sum_{v \in f} \alpha_v^f}{A} \quad (4.1)$$

where  $\alpha_v^f$  is the angle of the face  $f$  incident to the vertex  $v$  (see inset figure) and  $A$  is one third of the area of the surrounding faces. This yields the developability criterion

$$\forall \text{ interior vertex } v : 2\pi - \sum_{v \in f} \alpha_v^f = 0. \quad (4.2)$$

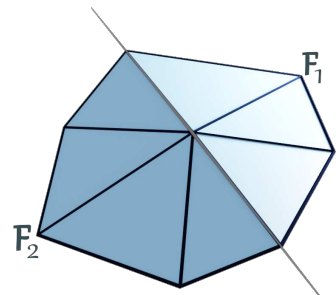
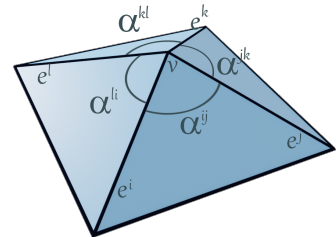
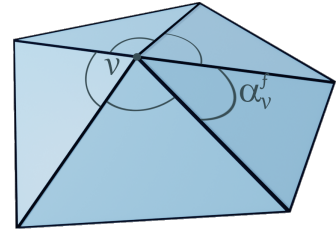
This definition is rather simple, but does not include any notion of smoothness. A surface meeting only this criterion may suffer from crumpling artefacts, as for instance in the Schwarz lantern (Wardetzky, 2007).

Rabinovich et al. (2018a) have proposed a similar criterion for quad meshes. They prove that they only need the four  $\alpha^f$  around each interior vertex to be equal, as depicted in the inset figure, defining what they called *Discrete Orthogonal Geodesic* meshes. Assuming that the mesh is not degenerated, the definition may be relaxed by requesting the equality of the cosines, which can be computed easily using scalar products,

$$\begin{cases} e_i^\top e_j \|e_k\| - e_j^\top e_k \|e_i\| = 0 \\ e_j^\top e_k \|e_l\| - e_k^\top e_l \|e_j\| = 0 \\ e_k^\top e_l \|e_i\| - e_l^\top e_i \|e_k\| = 0. \end{cases} \quad (4.3)$$

Their formulation also has no guarantee on the smoothness and may lead to spikes-like artefacts, but its simplicity allows the authors to track and deform their DOG surfaces with interactive speeds (Rabinovich et al., 2018b; Wolf et al., 2021).

Lastly, Stein et al. (2018) proposed a criterion for a triangle mesh to be both piecewise smooth and developable. They note that each vertex is either a boundary vertex of a developable patch, a *seam vertex*, or within a patch. The criterion for the discrete developability is that each "interior" vertex is a *hinge*, meaning that the surrounding faces can be partitioned into two sets of faces, in which all the faces have the same normal. Mathematically, they provided two definitions. The first one takes the form of a combinatorial problem, searching



the two sets of faces denoted  $F_p$ ,

$$\pi(P) = \sum_{p \in \{1,2\}} \frac{1}{|F_p|} \sum_{f \in F_p} \|n_f - \bar{n}_p\|^2 = 0, \quad (4.4)$$

with  $n_f$  the normal of the face  $f$ , and  $\bar{n}_p$  the average normal within the set  $F_p$ . The second one has the form of a minimisation problem to find the hinge direction that should be orthogonal to all the normals,

$$\lambda = \min_{u \in \mathcal{S}(0,1)} \sum_{v \in f} \alpha_v^f (u^\top n_f)^2, \quad (4.5)$$

with  $\alpha_v^f$  defined as previously, used as a weight. Although more complex, their definition enforces smoothness of the surface and naturally includes boundaries of patches.

# Chapter 5

## Inverse elastic shell design with contact and friction

I present in this chapter our method for the inverse design of elastic shell subject to gravity and dry frictional contact.

After presenting an overview of the final algorithm in Section 5.2, I introduce in Sections 5.3 and 5.4 the initial method of Casati (2015); Casati et al. (2016) for the inversion in the case of unilateral constraints, that we have enhanced by accounting for mass variation. This algorithm works relatively well, but has no straightforward extension that robustly deals with dry frictional contact.

Thus, we propose to add a second step to enforce the Coulomb law that is presented in Section 5.5 (Ly et al., 2018). With this correction step, I show in our results in Section 5.6 that the resulting inversion process successfully produces rest shapes that make use of the friction forces to prevent sagging.

## 5.1 Motivation

In the previous chapter, we have reviewed the recent works on the inverse design methods in Computer Graphics. Algorithms that specifically target garments mainly focus on adjusting a known cloth pattern in order to reflect some target modifications such as the addition of new folds or grading. Our goal is to be able to treat any kind of surface, including fanciful garments without the knowledge of such patterns that may be unknown in the literature and hard to design. Consequently, we turn our attention to shell and plates inversion methods.

As briefly summarised at the end of Section 4.1.1, physics-based inversion methods are often based on a direct step that computes the deformed shape from a current estimate of the unknown shape and/or parameters that is combined with an inverse step in order to refine the unknowns. My work is in the continuity of the shell inversion algorithm of Casati (2015); Casati et al. (2016), that did not handle frictional contact robustly.

Because we consider that friction is an essential component in the appearance of garments, we do not choose to include only non-penetration constraints as most of previous works did, but strove to treat frictional contact during the inversion. This has led us to the inverse shell design algorithm presented in the following section.

## 5.2 Overview of the algorithm

Let us consider a target input surface that we wish to simulate without any sagging. As presented in Section 4.1.1, the goal is therefore to compute a corresponding rest shape, that is a "force-free" version of the input that prevents the sagging.

In my approach that follows Casati (2015)'s, the target shape is considered to be an elastic shell subject to gravity and to frictional contact with an external body. Frictional contact is either approximated as pin constraints or modelled realistically using unilateral constraints with Coulomb conditions (see Section 1.2.1). In this configuration, we would like the target to be a static stable equilibrium by finding an appropriate rest shape. The stability of the equilibrium is a desirable property for us, as we do not want any small perturbation (*e.g.* numerical error) to cause the object to depart from its target shape.

We assume that the target geometry is represented by a triangular mesh with  $n$  vertices whose positions are denoted by  $x_t \in \mathbb{R}^{3n}$ . Similarly, we note respectively  $x$  and  $\bar{x}$  both also in  $\mathbb{R}^{3n}$  the physical deformed shape and the rest shape. The material parameters, that are composed of the stretching, shearing and bending stiffnesses, the surface density and the friction coefficient, are considered to be fixed.

Ideally, if we note  $f_p$  the total conservative forces (internal + gravity) and  $f_c$  the

contact forces, the problem we would like to solve would take the form

$$\text{Find } \bar{x} \text{ such that } \begin{cases} f_p(x_t, \bar{x}) + f_c = 0 & (x_t \text{ is an equilibrium}) \\ x_t \text{ is stable} \end{cases} \quad (5.1)$$

with  $f_c$  belonging to  $\mathbb{R}^{3n_c}$  in the case of unilateral contact or subject to the Coulomb law (c.f. Section 1.2.1) in the case of dry frictional contact.

Although appealing, this formulation exhibits several drawbacks. The target shape might be noisy, for instance if it comes from a 3D reconstruction, or be physically unreachable. In either case, instead of concluding that there is no solution for the exact match, one would prefer to find a solution that approximates the input. This motivates the rewriting of the problem in a least-squares form

$$\min_{x, \bar{x}} \frac{1}{2} \|x - x_t\|^2 \quad \text{s.t. } (x, \bar{x}) \text{ is in a stable equilibrium.} \quad (5.2)$$

The goal now is to formulate mathematically the stable equilibrium condition to be able to enforce this constraint. Although the equilibrium condition with dry friction can be expressed as the contact forces should belong to Coulomb cones, the stability condition is more complex. With pin constraints, the stability depends on the eigenvalues of the hessian of the energy and in the case of frictional contact, the condition is not clearly defined.

To solve this problem, I devise a two-step algorithm by breaking down the previous problem into two subproblems.

In a first step detailed in Section 5.4, I present the algorithm of Casati (2015) to solve the problem 5.2 assuming that all contacts are unilateral. Within this assumption, the stability constraint can be directly integrated in an evaluation process of the so-called *draping function*.

Casati (2015) also noticed that the problem needs to be regularised as long as the current estimate of the solution is far from being optimum. However, such regularisation takes the form of unphysical forces. To come back to the true physical problem, the inversion process, composed of this first step and the second step presented below, are iteratively solved by subsequently decreasing values of this regularisation factor until it is not needed anymore.

Then, in Section 5.5, I present a second step that I formulated to handle frictional contact. This correction step takes the form of an optimisation problem that aims at rectifying the violation of the Coulomb conical constraints. The first and second step are illustrated in Figure 5.1 and the full algorithm is summarised in Algorithm 3.

---

**Algorithm 3:** Robust inversion algorithm.

---

**Data:** Target equilibrium pose  $x_t$ , initial regularisation factor  $\lambda_0$ , regularisation reduction factor  $\alpha \in ]0; 1[$

**Result:** A pair  $(x, \bar{x})$  consisting of a stable equilibrium pose and a natural pose with  $x$  as close as possible to  $x_t$

// No a priori knowledge of the natural shape

1  $k \leftarrow 0$  ;

2  $\bar{x}_0 \leftarrow x_t$  ;

3 **while**  $\lambda_k > 0$  **do**

4     **if**  $\lambda_k \leq \varepsilon_\lambda$  **then**

    // Last iteration with no regularisation

5      $\lambda_k \leftarrow 0$ ;

6     **end**

    // Step 1 (EvalObjective is defined in Algorithm 4 of Section 5.4) (Casati, 2015)

7      $(x_{k+\frac{1}{2}}, \bar{x}_{k+\frac{1}{2}}) \leftarrow \text{BFGS\_min}(\text{EvalObjective}(x_t, \lambda_k, \bullet), \bar{x}_{init} = \bar{x}_k)$  ;

    // Step 2 ( $G_K$  is defined in Section 5.5)

8      $(x_{k+1}, \bar{x}_{k+1}) \leftarrow \text{BFGS\_min}(G_K(F(x_{k+\frac{1}{2}}, \bullet)), \bar{x}_{init} = \bar{x}_{k+\frac{1}{2}})$ ;

    // Decrease  $\lambda$

9      $\lambda_{k+1} \leftarrow \alpha \lambda_k$  ;

10     $k \leftarrow k + 1$  ;

11 **end**

12 **return**  $(x_k, \bar{x}_k)$ ;

---

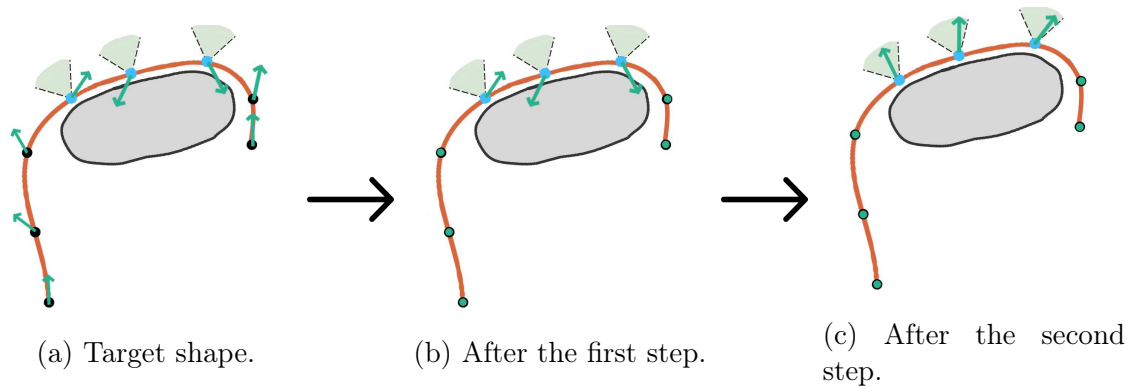


Figure 5.1: Illustration of one pass of our two-step process. The green arrows depict the conservative force (including the regularisation) being exerted at the free vertices (in black) and at the vertices in contact (in blue).

## 5.3 Mechanics

In this section, I present the mechanical model used for the inversion by (Casati, 2015) and that I have taken up. In that framework, I have modified the mass computation procedure to handle the variation of mass during inversion (see Section 5.3.2).

### 5.3.1 Shell model

In this chapter, the thin elastic shell model used to represent the clothes is the DISCRETE SHELL model of Grinspun et al. (2003) that I presented in Section 1.1.2. The shell internal energy is noted  $E_{\text{int}}$ .

Note that, in our study, we did not consider the term implying the face area (Equation 1.13b). However, we verified that adding this term does not affect our framework, nor the observed results. More generally, we can use any shell model, provided we can compute the energy  $E_{\text{int}}$  and the required derivatives. For instance, in the next chapter, we also use the LIBSHELL code.

### 5.3.2 Gravitational energy

We use a simple lumped mass model, where the mass of vertex  $i$  can be computed by

$$m_i(\bar{x}) = \sum_{\substack{f \in \mathcal{F} \\ i \in f}} \frac{1}{3} \sigma \bar{A}_f, \quad (5.3)$$

with  $\sigma$  the surface density,  $\mathcal{F}$  the set of the triangular faces of the mesh and  $\bar{A}_f$  the area of the face  $f$  of the natural shape  $\bar{x}$ .

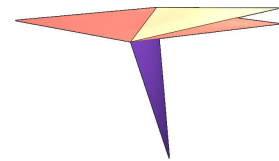
With  $\mathbf{g} = g e_z$  the gravity acceleration and  $e_z$  the unit upward vector, the gravitational

energy is therefore

$$E_g(x, \bar{x}) = - \sum_{i \in \mathcal{V}} g m_i(\bar{x}) x_i^\top e_z. \quad (5.4)$$

We have to note here that the mass, and therefore the gravitational energy have a dependence *w.r.t.* the unknown rest shape. In his original contribution, [Casati \(2015\)](#) did not acknowledge for the fact that the mass should change during the inversion as the natural shape evolves. I modified the algorithm to take the mass recomputation and the corresponding derivatives into account.

To illustrate the difference between the two treatments, let us consider the case presented in the inset figure. The initial rest shape in orange is composed of two triangular faces. One face is fixed, and the remaining vertex is left hanging, yielding the deformed shape in purple. Computing the mass only at the initialisation by using the deformed shape as an initial guess for the rest shape results in the free vertex to have a larger mass due to the stretching. The inversion process then yields the rest shape in yellow, where an extra curvature is needed to counterbalance the extra weight and thus preventing the recovery of the initial flat shape. Correctly recomputing the mass along the process, along with using the corresponding derivatives of the energies, enable to retrieve exactly the flat shape. Note that retrieving the original natural shape is possible here as the problem is very simple. This is not the case in more complex examples as presented in [Section 5.6.3](#).



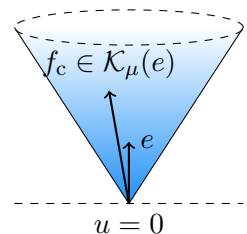
### 5.3.3 Frictional contact

For the frictional contact law, we use the Signorini-Coulomb model described in [Section 1.2.1](#), which offers a good compromise between realism and simplicity.

More specifically, as we are considering shapes at equilibrium, we are interested in the sticking case of this law. Let us consider a contact between a vertex and an obstacle, and note  $e \in \mathbb{R}^3$  the unit normal at the contact point oriented outward the obstacle and  $\mu \in \mathbb{R}_*^+$  the friction coefficient. Then the contact force  $f_c$  has to be in the Coulomb cone  $\mathcal{K}_\mu(e) \subset \mathbb{R}^3$  defined by

$$\|f_{c|T}\| \leq \mu f_{c|N}, \quad (5.5)$$

where  $f_{c|N} = f_c^\top e \in \mathbb{R}$  and  $f_{c|T} = f_c - f_{c|N}e \in \mathbb{R}^3$  are respectively the normal and tangential components of the contact force.





### 5.3.4 Equilibrium and stability

In our approach, the conservative forces applied to the shell are the internal forces and the gravity, yielding the following potential energy

$$E_p(x, \bar{x}) = E_{\text{int}}(x, \bar{x}) + E_g(x, \bar{x}) \quad (5.6)$$

and conservative forces  $f_p(x, \bar{x}) = -\nabla_x E_p(x, \bar{x})$ . Note that other conservative forces can also be included.

First, let us consider the case of pin constraints. Without loss of generality, we can assume that the vertices are ordered in a way such that we have  $x = [x_F | x_C] \in \mathbb{R}^{3n}$  with  $x_F \in \mathbb{R}^{3n_F}$  the components of the  $n_F$  free vertices and  $x_C \in \mathbb{R}^{3n_C}$  the components of the  $n_C = n - n_F$  vertices in contact.

The static equilibrium is then simply the balance of the forces at the free nodes,

$$-\nabla_{x_F} E_p(x, \bar{x}) = 0. \quad (5.7)$$

The equilibrium is *asymptotically* stable if the Hessian of the energy  $\nabla_{x_F}^2 E_p(x, \bar{x})$  w.r.t. the free components is a definite positive matrix. The asymptotic stability is an interesting property as it implies that in the presence of small perturbations, the deformed shape will return to the same equilibrium.

Now if we consider frictional contact, the force balance also needs to be done at the contact points using the unknown constrained contact forces

$$\begin{cases} -\nabla_{x_i} E_p(x, \bar{x}) = 0 & \text{if } i \text{ is free} \\ -\nabla_{x_i} E_p(x, \bar{x}) + f_{c|i} = 0, f_{c|i} \in \mathcal{K}_{\mu_i}(e_i) & \text{if } i \text{ is in contact.} \end{cases} \quad (5.8)$$

We note  $K_i = \{0\}$  the vertex  $i$  is free and  $K_i = \mathcal{K}_{\mu_i}(e_i)$  otherwise. Writing  $\mathcal{K} = \prod_{i \in \mathcal{V}} K_i$ , we get the compact expression for the equilibrium

$$\nabla_x E_p(x, \bar{x}) \in \mathcal{K}. \quad (5.9)$$

Regarding the stability condition, it is not as clear as in the previous case. [Basseville and Léger \(2006\)](#) studied and derived conditions on the *asymptotic* and *Lyapunov* stability in a very simple system composed of a single 2D solid and two springs on an inclined plane. The *Lyapunov* stability is in a sense less strong than the asymptotic one as it only requires the small perturbations to yield bounded displacements. [Leine and van de Wouw \(2008\)](#) proposed a much more general approach, although less practical to use as they rely on differential inclusions. We choose for our problem a condition that aims to enforce a kind of *Lyapunov* stability and which reads for the contact points as

$$\nabla_x E_p(x, \bar{x}) \in \text{Int}(\mathcal{K}_\mu(e)) \quad (5.10)$$

with  $\text{Int}(\mathcal{K}_\mu(e))$  the interior of the set  $\mathcal{K}_\mu(e)$ . With this definition, we hope that small perturbations will still keep the forces inside the cone and not generate any motion. In practice, as detailed in Section 5.4, this constraint is enforced by requesting the force to belong to a Coulomb cone of smaller aperture  $\mu' < \mu$ .

## 5.4 Step 1: Inversion with unilateral constraints

I explain in this section the first step of our inversion process that treats all contacts as pin constraints.

For the sake of readability, the notation  $x_F/x_C$  is dropped but recall that the contact points  $x_C$  of the deformed shape  $x$  are fixed and thus not degrees of freedom. In practice, the fixed points are handled by setting the corresponding quantities equal to zero.

For now, the components of the contacting points on the natural shape  $\bar{x}$ , namely  $\bar{x}_C$ , are fixed during this step. The motivation behind this is that the first step should not deal with the contact points, and thus should perturb as little as possible the forces at the contact points possibly fixed by a previous second step by not displacing the contact points. This assumption is discussed in Section 5.7.4.

### 5.4.1 Least-squares formulation and draping function

Following the previous section, the inversion problem of this first step with pin constraints is

$$\min_{x_{\frac{1}{2}}, \bar{x}_{\frac{1}{2}}} \frac{1}{2} \|x - x_t\|^2 \quad \text{s.t.} \quad \begin{cases} \nabla_{x_F} E_p(x_{\frac{1}{2}}, \bar{x}_{\frac{1}{2}}) = 0 \\ \nabla_{x_F}^2 E_p(x_{\frac{1}{2}}, \bar{x}_{\frac{1}{2}}) \succ 0 \end{cases} . \quad (5.11)$$

As discussed before, we have an optimisation problem with highly non-linear constraints. However, we will see below that these constraints can be enforced by a careful evaluation procedure.

**Draping function.** Let  $(x^*, \bar{x}^*) \in \mathbb{R}^{3n} \times \mathbb{R}^{3n}$  be a deformed and rest pose pair satisfying the constraints for a stable equilibrium and let us note  $F : (x, \bar{x}) \rightarrow \nabla E_p(x, \bar{x})$ .

We have  $F(x^*, \bar{x}^*) = 0$  and  $D_x F(x^*, \bar{x}^*) \succ 0$ . Then, by the implicit function theorem, there exists a neighbourhood  $V$  of  $x^*$ , a neighbourhood  $\bar{V}$  of  $\bar{x}^*$  and a function  $\Phi : \bar{V} \rightarrow V$  such that

$$\forall x \in V, \forall \bar{x} \in \bar{V} : F(x, \bar{x}) = 0 \Leftrightarrow x = \Phi(\bar{x}). \quad (5.12)$$

Moreover, the theorem states that  $\forall x \in V, \forall \bar{x} \in \bar{V}$ ,  $D_x F(x, \bar{x})$  is invertible,  $\Phi$  is differen-

tiable on  $\bar{V}$  and

$$\forall \bar{x} \in \bar{V} : D\Phi(\bar{x}) = -(D_x F(\Phi(\bar{x}), \bar{x}))^{-1} (D_{\bar{x}} F(\Phi(\bar{x}), \bar{x})). \quad (5.13)$$

In other words, there exists locally an injective differentiable mapping, that we call the *draping function*, between the rest shape and the deformed shape. However, such function can obviously not be defined globally, as, in general, different initial states with the same rest shape may lead to different equilibria (see Figure 5.2).

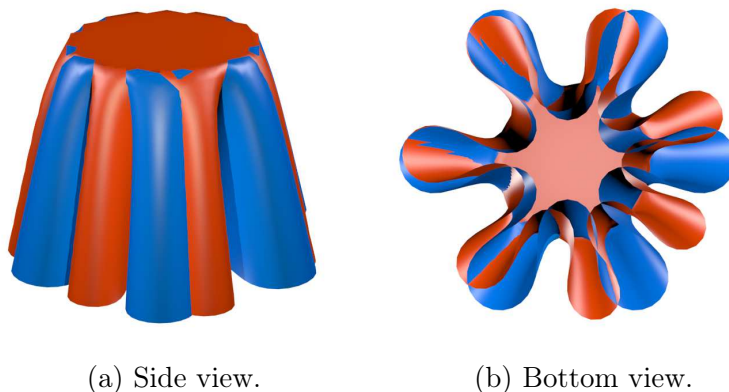


Figure 5.2: Two different equilibria for a tablecloth with the same flat rest shape.

Note that this method may also be found in the literature under the name of "sensitivity analysis". We retrieve the same expression of the differential (the "sensitivity matrix") by searching for a condition to the first order on a small displacement  $(dx, d\bar{x})$  such that  $(x^* + dx, \bar{x}^* + d\bar{x})$  still satisfies the constraints.

**Least-square minimisation.** With the existence of this function that incorporates all of the constraints, whose practical evaluation will be detailed in the following section, the inverse problem 5.11 can be cast as a much simpler unconstrained minimisation problem

$$\min_{\bar{x}} \frac{1}{2} \underbrace{\|\Phi(\bar{x}) - x_t\|^2}_{J(\bar{x})}. \quad (5.14)$$

To solve this problem, the method that works best for us among the ones we have tested is the BFGS approach. The gradient descent method yields a too slow descent, even on simple cases. While a full Newton method, detailed in Section D, is also possible and allows indeed the convergence in much less iterations than BFGS, the evaluation of the Hessian is too costly as it relies on the evaluation of the third derivatives of the energies and several inversions of linear systems. Finally, the Gauss-Newton method may also have a good convergence rate with a relatively lower cost per iteration than the previous method. However, in our experiments, it only works consistently when the initial guess

is not too far from the solution, and otherwise yields relatively slow convergence.

**Adjoint method.** To use the BFGS method to solve the minimisation problem 5.14, we have to compute the gradient of  $J$ ,

$$\nabla J(\bar{x}) = D\Phi(\bar{x})^\top (\Phi(\bar{x}) - x_t). \quad (5.15)$$

Although we have an explicit expression for  $D\Phi(\bar{x})$  from Equation 5.13, we do not need to compute this dense matrix, but only its product with  $(\Phi(\bar{x}) - x_t)$ . If we replace its expression in Equation 5.15, we have

$$\nabla J(\bar{x}) = -(D_{\bar{x}}F(\Phi(\bar{x}), \bar{x}))^\top (D_xF(\Phi(\bar{x}), \bar{x}))^{-\top} (\Phi(\bar{x}) - x_t). \quad (5.16)$$

The adjoint method consists in splitting this computation into two steps

$$\begin{cases} (D_xF(x, \bar{x}))^\top p & = x - x_t \\ \nabla J(\bar{x}) & = -(D_{\bar{x}}F(x, \bar{x}))^\top p \end{cases} \quad (5.17)$$

with  $p$  the adjoint state and  $x = \Phi(\bar{x})$  evaluated only once. If we replace  $F$  with its original expression, we obtain

$$\begin{cases} (\nabla_{x^2}^2 E_p(x, \bar{x})) p & = x - x_t \\ \nabla J(\bar{x}) & = -(\nabla_{x\bar{x}}^2 E_p(x, \bar{x}))^\top p. \end{cases} \quad (5.18)$$

The evaluation of the objective function and its gradient, including the regularisation parameter discussed hereafter is described in Algorithm 4.

---

**Algorithm 4:** Evaluation of the objective function  $J$  and of its gradient  $\nabla J$  (under regularized energy)

---

**Data:** Target equilibrium pose  $x_t$ , energy regularisation factor  $\lambda$ , current natural pose  $\bar{x}$

**Result:** The value and the gradient of the objective function  $J$  at  $\bar{x}$ , with energy regularisation factor  $\lambda$

```

1 Procedure EvalObjective( $x_t, \lambda, \bar{x}$ )
   // Draping is defined in Algorithm 5
3    $x \leftarrow \text{Draping}(x_t, \lambda, \bar{x});$ 
5    $objective \leftarrow \|x - x_t\|^2;$ 
7    $p \leftarrow \text{Linear\_solver}(\nabla_{x^2}^2 E_p(x, \bar{x}), x - x_t);$ 
9    $gradient \leftarrow -(\nabla_{x\bar{x}}^2 E_p(x, \bar{x}))^\top p;$ 
11  return  $x, objective, gradient;$ 

```

---

### 5.4.2 Evaluation and regularisation of the draping function

In the previous section, I explained the optimisation process assuming the existence of the draping function  $\Phi$ . I explain here how this function is evaluated in practice.

**Naive draping.** As mentioned earlier, the draping function is defined locally with the implicit function theorem using the fact that a stable equilibrium is a minimum of the potential energy. As such, the natural way to evaluate it is to solve the minimisation problem

$$\Phi(\bar{x}) = \underset{x}{\operatorname{argmin}} E_p(x, \bar{x}) \quad \text{with } x_{\text{init}} = x_t. \quad (5.19)$$

Unfortunately, this formulation is unstable. Indeed, if  $x_t$  is not close to be an equilibrium with the given  $\bar{x}$ , then this procedure is likely to return an equilibrium  $x$  very far from  $x_t$  and thus may trap the optimisation in a local minimum far from the target. Moreover, the size of the neighbourhood on which the function is defined is unknown, and thus in practice we may observe that two close natural shapes yield very different equilibria, perturbing the gradient-based optimisation.

**Regularised energy.** The goal is therefore to modify the formulation to have consistency between subsequent evaluations. As we aim at finding a deformed equilibrium shape that is close to the target, we modify the potential energy by adding an attraction term to the target, yielding

$$E_p^\lambda(x, \bar{x}) = E_p(x, \bar{x}) + \frac{\lambda}{2} \|x - x_t\|^2 \quad (5.20)$$

with  $\lambda \geq 0$  our regularisation factor, and we note  $\Phi^\lambda$  the corresponding draping function using this modified energy.

We see that this term actually convexifies the energy, and so leads to a simpler minimisation problem for the draping by "erasing" the local minima around  $x_t$ . Intuitively, this also extends the local range of definition of the implicit function and thus provides a better consistency between the evaluations in a close neighbourhood. To the limit  $\lambda \rightarrow +\infty$ ,  $\Phi^\lambda$  is a constant function, and we remove any dependence to  $\bar{x}$  and to the physics of our problem.

With this regularised energy, we can now relatively safely evaluate our draping function. In practice, the minimisation 5.19 is done using the Newton-CG method (Nocedal and Wright, 2006, Section 6.2), which has the advantage compared to a naive Newton method of always providing a descent direction. The draping procedure is summarised in Algorithm 5. Overall, in our tests, for each evaluation, less than 10 iterations were needed to perform the draping operation.

Also note that if we find  $\bar{x}$  such that  $\Phi^\lambda(\bar{x}) = x_t$ , the penalty term has no effect and we have found a solution to the inverse problem. Otherwise, if  $x$  is only approaching  $x_t$ , these fictive forces have to be removed to yield a physically correct solution. To this end

---

**Algorithm 5:** Draping procedure  $\Phi$ 

---

**Data:** Target equilibrium pose  $x_t$ , regularization factor  $\lambda$ , current natural pose  $\bar{x}$ **Result:**  $x = \Phi(\bar{x})$ , a local minimum of the shell potential energy, this energy being evaluated at  $\bar{x}$  and  $\lambda$ -regularized towards  $x_t$ .

```

1 Procedure Draping( $x_t, \lambda, \bar{x}$ )
3    $x \leftarrow \text{Newton-CG}(E_p^\lambda(\bullet, \bar{x}), x_{\text{init}} = x_t);$ 
5   return  $x;$ 

```

---

and as presented in Algorithm 3, we decrease  $\lambda$  after solving the two steps with a given regularisation factor and then solve again the two steps. The regularisation factor  $\lambda$  is set to 0 below a given threshold.

## 5.5 Step 2: Accounting for frictional contact

In the previous step, I described the method of Casati (2015) to solve the inverse problem with fixed contact points. However, this step alone is not sufficient to deal with frictional contact as depicted in Figure 5.3.

In this figure, the arrows represent the contact forces at the waist of a character required to hold the **Gored skirt**. We see in Figure 5.3a that inverting by considering only fixed points yields artificial *adhesive* contact forces, which have no meaning in a fabric/skin interaction. Conversely, we can see in Figure 5.3b that with our full algorithm dealing with frictional contact, the resulting contact forces are correctly pointing outwards the body of the character. I describe in this section how we actually correct the contact forces by adjusting the rest shape accordingly through a correction step that I have added to the initial algorithm.

For the sake of readability, the  $\lambda$  indicating the regularised quantities is dropped. Recall however that this step is called using the same regularisation coefficient  $\lambda$ . Also, without any loss of generality, we assume that the friction coefficient is the same at all the contact points and is noted  $\mu$ . As we will see below, this assumption can easily be removed as the correction is defined per contact point.

### 5.5.1 Defining the admissible forces

After solving the first step, we get a pair  $(x_{\frac{1}{2}}, \bar{x}_{\frac{1}{2}})$  that is a static equilibrium under bilateral constraints. We now wish to turn this pair into an equilibrium under frictional contact. However, adjusting the forces at the contact points by moving the nodes will affect the forces at the surrounding vertices. Thus, to preserve the equilibrium at the free points, the formulation of this correction step has to take into account the whole mesh.

As presented in Equation 5.10, the static equilibrium condition under frictional contact

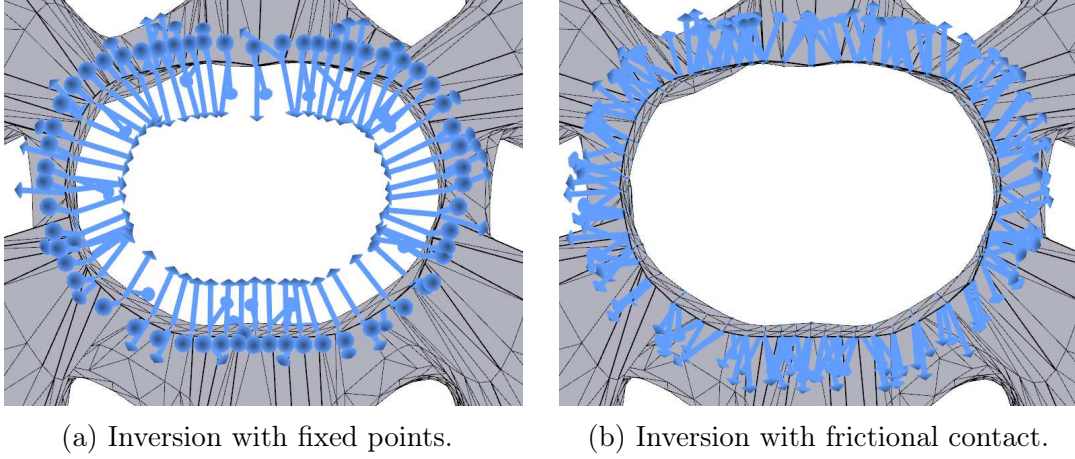


Figure 5.3: Comparison of the yielded interaction forces in the bilateral case (left) and in the unilateral case with dry friction (right) at the waist of **Gored skirt** (top view, normalized vectors).

can be formulated as an inclusion. To enforce it, we would like to turn this inclusion constraint into a functional constraint by defining a function  $G_K$  such that

$$\forall f \in \mathbb{R}^{3n} : f \in K \Leftrightarrow G_K(y) = 0. \quad (5.21)$$

We define  $G_K$  by first assuming that it acts per force *i.e.*

$$\forall i \llbracket 1, n \rrbracket : [G_K(y)]_i = G_{K_i}(y_i) \text{ with } y_i \in \mathbb{R}^3. \quad (5.22)$$

Then, a natural choice for the vertices not in contact is to simply take  $G_{K_i} \equiv I$ , as we would like to keep the forces null.

Regarding the contact points,  $G_{K_i}$  has to be null in the Coulomb cone  $\mathcal{K}_\mu(e_i)$ . Note that this also implies that its differential in the interior of the cone is also null, meaning that we do not favour any particular orientation on the contact force as long as it is in the cone.

Consider  $g : \mathbb{R} \rightarrow \mathbb{R}^+$  such that

$$\forall t \in \mathbb{R} : g(t) \begin{cases} = 0 & \text{if } t \leq 0 \\ > 0 & \text{otherwise.} \end{cases} \quad (5.23)$$

We see that  $g$  will act as a sort of distance function to our admissible set. In practice, we take  $g : t \rightarrow -t$  on  $\mathbb{R}^-$ . Smoother formulation such as  $g : t \rightarrow t^2$  are also possible but we found out that higher order terms would dampen the convergence as  $t \rightarrow 0^-$ .

With this function  $g$ , we can take for a given contact point  $i$

$$\forall f_i \in \mathbb{R}^3 : G_{K_i}(f_i) = g(f_{i|N}) + g(\mu f_{i|N} - \|f_{i|T}\|) \quad (5.24)$$

with  $f_{i|N} = f_i^\top e_i$  and  $f_{i|T} = f_i - f_{i|N}e_i$  respectively the normal and tangential parts as seen previously. It is easy to check that

$$\begin{aligned} \forall f_i \in \mathbb{R}^3 \quad G_{K_i}(f_i) = 0 &\iff f_{i|N} \geq 0 \text{ and } \mu f_{i|N} \geq \|f_{i|T}\| \\ &\iff f_i \in K(e_i, \mu). \end{aligned} \quad (5.25)$$

Note that many definitions of  $G_{K_i}$  are possible. We tried for instance using the distance to the orthogonal projection to the cone (see (Cadoux, 2009, Appendix E)), but did not observe any noticeable difference regarding the behaviour of our algorithm.

### 5.5.2 Correction step

Ideally, we would like to project the forces on the manifold  $G_K = 0$ , but this problem is far from being easy as the relation between the natural shape  $\bar{x}$  and the contact force  $\nabla E_p(x_{\frac{1}{2}}, \bar{x})$  is complex.

Instead, we solve the following minimisation problem

$$\min_{\bar{x}} \frac{1}{2} \|G_K(F(x_{\frac{1}{2}}, \bar{x}))\|^2 \quad (5.26)$$

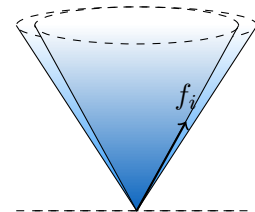
using again the BFGS method. The objective function is this time continuous but non-smooth, however in practice we did not observe any related issue.

Note that this formulation has the disadvantage that if the minimum reached is not zero, the resulting shape is not a static equilibrium. Moreover, this problem is complex as the objective function is highly non linear.

However, in practice, in scenarios where we expected a solution to be found (*e.g.* by shrinking the natural shape for the elastic force to make the cloth stick to the body), the minimisation problem correctly reaches zero. The key to the success is the proper warmstart by the first step that provides a configuration "almost" solving the problem. The second step then "only" has to fix the contact forces by moving the contact points, and propagating the displacements to the rest of the mesh while keeping it on the equilibrium manifold.

In addition, the formulation 5.26 does not enforce the stability for the contact-free zones. It is therefore inherited from the first step through the warmstart and this phenomenon is illustrated in Section 5.6.4. Regarding the contact zones, as announced in Section 5.3.4, we enforce a kind of stability by requesting the problem to be solved with a slightly lower friction coefficient.

With this trick, even if our minimisation moves the forces at the border of the smaller cone, they will be strictly inside the real cone as illustrated in the inset figure. In practice, we use  $\mu' = 0.975\mu$ ,





## 5.6 Results

### 5.6.1 Implementation details

As presented in Algorithm 3, we alternate over the two steps described in Sections 5.4 and 5.5 using the regularised energy 5.20.

In practice, we start with  $\lambda = 1000$ , and its value is halved after each pass on the two steps. When it reaches the threshold  $\epsilon_\lambda = 0.1$ , it is set to 0 to ensure that no unphysical force remains.

Concerning the optimisation, the first step stops when  $\|\delta\bar{x}\|_\infty \leq 1.e-15$ ,  $\|\nabla J(\bar{x})\| \leq 1.e-15$ , or when 2000 iterations are reached. The second step stops when  $\|\delta\bar{x}\|_\infty \leq 1.e-15$ ,  $\|\nabla G(\bar{x})\| \leq 1.e-15$ , or when 5000 iterations are reached. The choice of these parameters is further discussed in Section 5.7.

Our method was implemented in C/C++ with a single-threaded architecture, on a PC featuring 4 dual-core Intel i7-5600U processors running at 2.60GHz.

### 5.6.2 Framework

We have tested our method on ten different examples listed in Table 5.1 along with the material parameters used. Most of the target shapes to invert represent pieces of clothing in frictional contact with the body of a character, with the exception of the **Book Page** also in frictional contact with an open book and **Bitriangle**, the synthetic example already presented in Section 5.3.2 that uses fixed points. The examples are diverse through their design, either realistic or stylized in a cartoon-like fashion and through their origin: two of them are synthetic examples generated by simulation, one comes from a 3D capture by (Pons-Moll et al., 2017), and the rest were designed using the *Autodesk 3ds Max* modelling software by our infographist Laurence Boissieux. All of this show that our method is fairly versatile and can be applied to any type of input surface.

For each of our examples, we start by choosing material parameters (mass density, stiffness parameters & friction coefficient) so as to observe noticeable bending and stretching – our method does not target inextensible surfaces. As written in Section 5.5, choosing a uniform friction coefficient is not a limitation of our method, but only a simplification of the implementation.

Then, we run our inversion algorithm to compute a natural shape  $\bar{x}$  and a deformed shape  $x$ . To check the correctness of our results, we perform a direct simulation initialised with this pair using the SO-BOGUS solver (Daviet et al., 2011), that is not used in our inversion process and only shares the property of using the Coulomb law to model friction. If the computed pair  $(x, \bar{x})$  corresponds indeed to a stable equilibrium configuration, then during the simulation, the deformed shape should not move as long as the contacting obstacle does not. Moreover, we also observe that the resulting animation induced for

instance by wind forces or body motion does not exhibit any artefacts. We detail in the following sections the results. Please also refer to the accompanying video of the original publication (Ly et al., 2018) to watch the animations.

Table 5.1: Material parameters for our inversion examples.

Example	Source	$\sigma$ ( $kg \cdot m^{-2}$ )	$k_L$ ( $N$ )	$k_B$ ( $N \cdot m$ )	$\mu$
<b>Bitriangle</b>	Simulation	0.5	50.0	$5.0e^{-2}$	--
<b>Synthetic Skirt</b>	Simulation	0.1	3.0	$5.0e^{-3}$	0.6
<b>Book Page</b>	3D Design	10.	5.0	$2.0e^{-3}$	0.4
<b>Beret</b>	3D Design	0.25	$5.0e^{-2}$	$5.0e^{-4}$	0.3
<b>Floppy Hat</b>	3D Design	2.5	5.0	$5.0e^{-4}$	0.6
<b>Top</b>	3D Design	0.5	0.2	$1.0e^{-1}$	0.2
<b>Saroual</b>	3D Design	0.05	1.0	$1.0e^{-2}$	0.2
<b>Puff Sleeve</b>	3D Design	1.0	0.1	$1.0e^{-3}$	0.7
<b>Gored Skirt</b>	3D Design	0.05	0.8	$5.0e^{-3}$	0.7
<b>Clothcap Shirt</b>	Capture	1.0	3.0	$5.0e^{-4}$	0.2

### 5.6.3 Qualitative results

**Target generated by simulation.** The **Bitriangle** example is the most simple one, and consists in a square sheet composed of two triangular faces. One face is fixed and one vertex is left hanging under gravity. As seen in Section 5.3.2, our algorithm (only the first step here) recovers the initial square from the deformed shape.

The **Synthetic Skirt** case is another simple example generated by simulating a flat torus falling onto a cone. The torus sags under gravity and its inner radius expands as it slides on the cone until the elastic and frictional contact forces balance gravity. We then take the resulting deformed shape, rendered in Figure 5.4a, as an input for our inversion algorithm using the same material parameters.

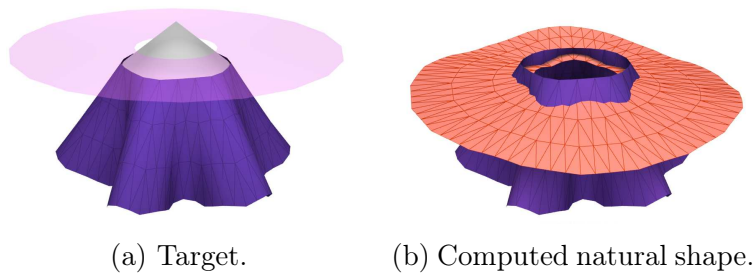


Figure 5.4: The **Synthetic Skirt** target (a - in purple) was generated by simulation from a flat torus (a - in semi-transparent pink). During inversion, the natural shape flattens and shrinks around the waist (b - in orange).

The result of the inversion is presented in Figure 5.4b. As expected, the natural shape flattens and tightens at the waist in order to generate bending and stretching forces

to counterbalance gravity. We observe here that our method manage to converge to a natural shape that is geometrically far from the input target, showing that our method is not limited to local adjustments on the natural shape. Moreover, we can also note that, although our algorithm converges to a high precision to a natural shape satisfying the static equilibrium problem, we do not recover the initial torus, illustrating the *non-uniqueness* of the solution to our inverse problem. This point is further discussed in Section 5.7.4.

**Target manually designed** Among all of our examples, seven were freely designed by an Laurence Boissieux using the *Autodesk 3ds Max* software to create the target surfaces, the supporting obstacles and the body motions for the subsequent simulation. The target are depicted in Figures 5.5 and 5.6 in the left column.

For these cases, the only input we have is the geometry. As mentioned in Section 5.6.2, we choose the material parameters to obtain an interesting looking material behaviour that produces a nice animation. However, we had to choose the parameters for which a solution to the inverse problem exist. For instance, the hardest example of our suite is the **Saroual** (Figure 5.6, second row), which consists in cartoonish baggy pants. For this example, we had to use relatively a stiff material to be able to compute static equilibriums and solve the problem. The choice of material parameters is further discussed in Sections 5.6.4 and 5.7.3.

The first example, **Book Page**, represents a page forming an arch on top of other pages of an open book (see Figure 5.5). The page was actually modelled by cutting and deforming a cylinder without any physical consideration. When simulated naively, the book page does not hold and slides over the book. However, when it is simulated using the natural shape computed by our algorithm, the page holds still, and we can then add a wind motion to gently turn the page while it rubs against the book.



Figure 5.5: The **Book Page** example, simulated naively, and then nicely animated after being inverted by our algorithm.

We then present two hats, namely **Beret** and **Floppy Hat**, each represented by a single shell posed on a character head through contact and friction. Without inversion, **Floppy Hat** sags and completely covers the face of the character, while **Beret** simply falls away due to insufficient friction forces and a deflation of the hat. In contrast, with the help of our inversion process, both hats have their design preserved in the simulation.

**Floppy Hat** can then be animated, resulting into a convincing flip-flopping motion when the head is moving, while **Beret** can be blown away of the head of the character by applying a wind.

Finally, we perform the same procedure on the four pieces of clothing **Top & Saroual**, **Puff Sleeve**, and **Gored Skirt** to preserve the design for the subsequent simulation with the character moving. All of the resulting natural shapes are depicted in Figure 5.9. Note the generation of the **Gored Skirt** from pattern design would have been a challenging task, while our shell-based approach allows its free-form design.

**Target automatically reconstructed from capture** Automatic reconstruction and segmentation from 3D captures or images is a current active topic as it provides interesting applications for the fashion industry (Pons-Moll et al., 2017; Yang et al., 2018a,b; Leroy et al., 2018).

To test our algorithm, Pons-Moll et al. (2017) have provided us with the meshes a character wearing a shirt reconstructed by their method. We have applied our algorithm to the shirt lying on the body at the initial pose (see Figure 5.7 (a)) and used the reconstructed body motion for the subsequent simulation.

Even though the contacting regions between the garment and the body meshes do not have as much precision as our freely designed examples, our method still converges to a plausible natural shape, visible in Figure 5.9, and prevents sagging at the start of the simulation. For this more realistic example, we chose material parameters to have a less stretchy behaviour than for the cartoonish examples. However, as we did not try to fit them accurately, we see by comparing Figures 5.7 (c) and 5.7 (d) that the simulation and the actual capture naturally do not match. We discuss further the estimation of material parameters in Section 5.7.3.

## 5.6.4 Evaluation

**Existence of a solution.** *A priori*, the existence of a solution depends on the material parameter chosen, although we did not prove this statement. In practice, we rely on the intuition obtained by observing the behaviour of the material when simulated to guess if the problem is likely to have a solution or not.

In the case of fixed contact points, we can actually say a bit more. The first intuition that we can have is that the stiffer the material is, the more likely there exists a natural shape and a deformed shape matching exactly the target. Indeed, if the surface is totally rigid, the solution is the input itself.

To confirm this intuition, we have performed a simple test on the **Bitriangle** example. We have launched multiple inversions of our synthetic deformed shape by varying the stretching coefficient  $k_L$  and bending coefficient  $k_B$ , with the surface density  $\sigma$  constant (and the shearing coefficient  $k_A$  equal to zero), and we have measured the distance

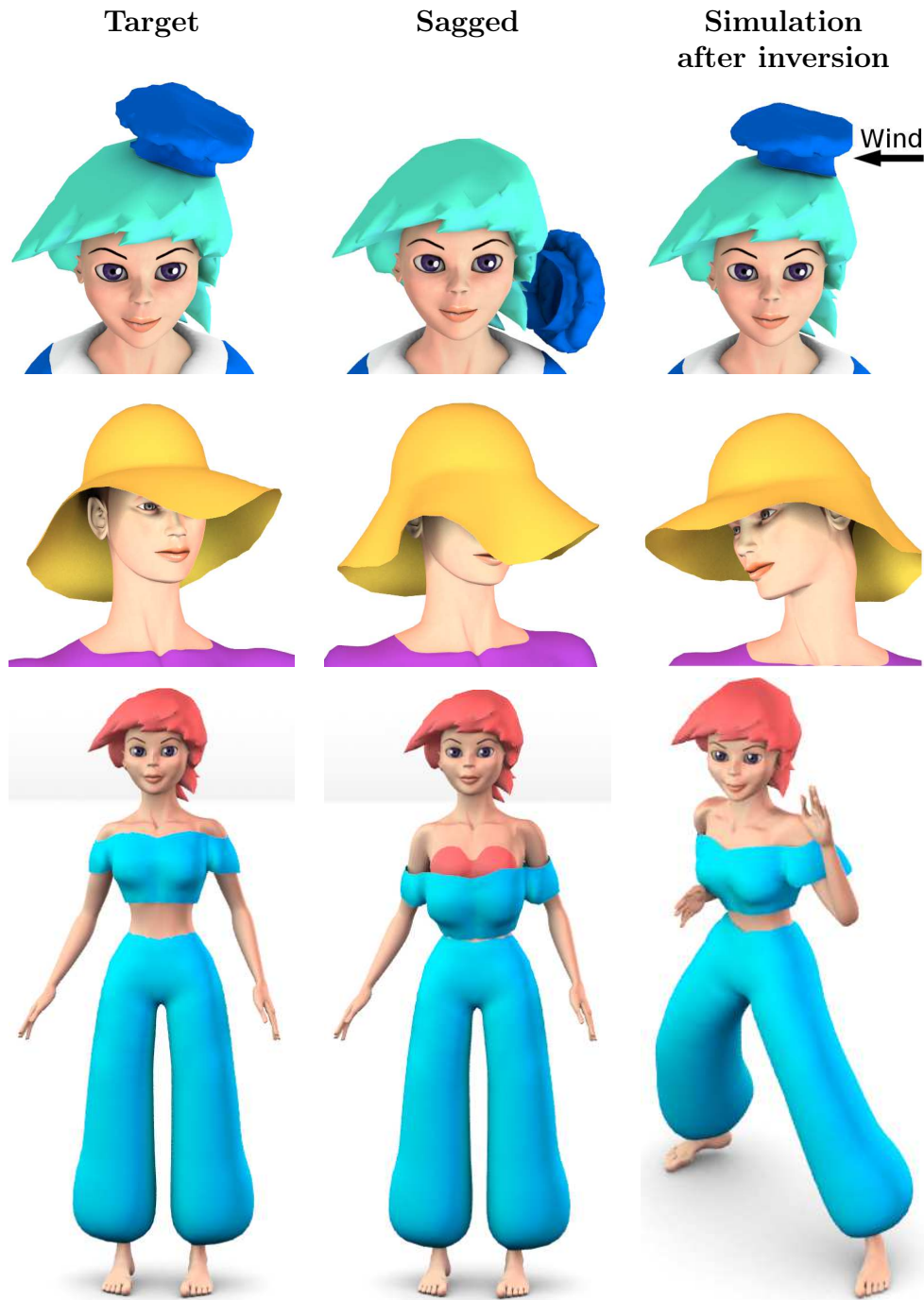


Figure 5.6: (1/2) Five cartoonish examples, namely **Beret**, **Top & Saroual**, **Puff Sleeve**, and **GoredSkirt**, inverted by our method and consistently animated. Note that in **Puff Sleeve**, the sleeve is not attached to the dress and properly fits around the arm only due to friction.

between the deformed equilibrium shape and the target shape. In the results displayed in Figure 5.8, we see that a material too soft either in bending (top left corner) or in stretching (bottom right corner) fails to yield a solution with a perfect match.



Figure 5.6: (2/2)

However, there should always be a solution in the least square sense. Indeed, consider the material parameters fixed and that one vertex of  $\bar{x}$  is fixed to remove its invariance to translations. Assume also that the draping function  $\Phi$  is defined globally and is continuous. Then we can see that the function  $J$  of the first step is coercive:  $\lim_{\|\bar{x}\| \rightarrow +\infty} J(\bar{x}) \rightarrow +\infty$ , thus there exists a finite minimum to this function that is attained at a finite value of  $\bar{x}$ . Morally, we can have the same result if we remove the assumptions on  $\Phi$ .

However, with frictional contact, the existence of a solution, even approximate, seems case dependent as the mesh can simply always slide or fall away. A similar issue is tackled in the so-called variational Signorini problem (see *e.g.* (Capatina, 2014)), where the unknown is not the geometry of the elastic volume at rest, but the distribution of  $\mu$  over its surface. However, a non-empty subset of the surface is always clamped to ensure the existence of a solution in the least square sense as explained above.

**Measure of deformation.** In Figure 5.9, we show the computed natural shapes for all of our examples. For each of them, we display the relative stretching and bending *w.r.t.*

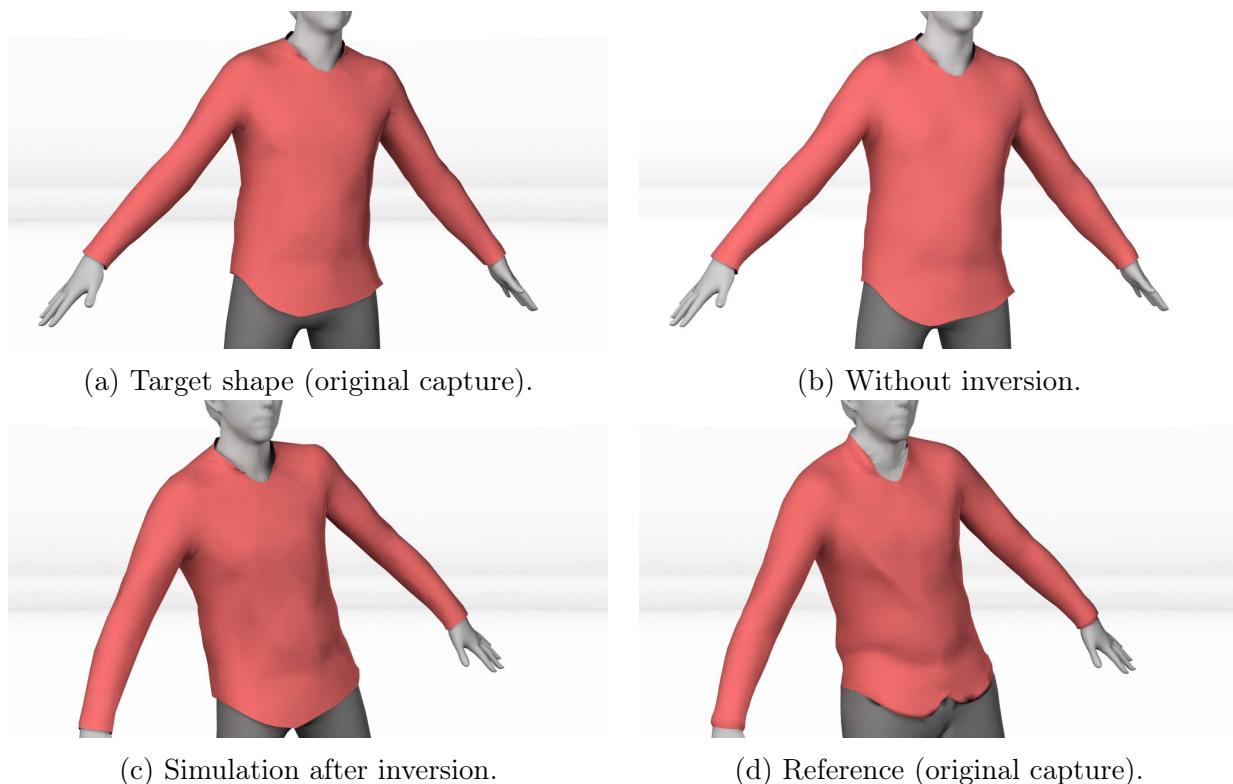


Figure 5.7: Inverse design of a real shirt, **ClothCap Shirt**, captured and segmented by Pons-Moll et al. (2017).

the target (*i.e.* our initial guess for the natural shape) to show the transformation applied by our method.

The relative stretching is computed as  $\frac{\bar{L}_e - L_e}{L_e}$ . When negative (depicted in blue), it means that the natural shape has been compressed compared to the target. Conversely, positive values (in red) indicate a dilatation. We can observe, as one could have expected, that the compression regime is predominant so as to compensate the sagging due to the gravity and to tighten the garments in order to generate larger sticking forces. However, dilatation is also present, mainly on **Beret** and **Puff Sleeve**, preventing these examples to be treated with methods limited to compression such as Bartle et al. (2016).

The relative bending is also computed per edge as  $\|\bar{\theta}_e - \theta_e\|$  and depicted in a white to red scale. We see, mainly on **Synthetic Skirt**, **Book Page** and **Floppy Hat**, that the natural curvature mostly opposes the sagging in order to preserve the local shape of the target.

**Role of the inversion with the draping function (first step).** As one may have noticed, solving the problem described in Section 5.5.2 yields a static equilibrium if the objective  $G_K(F(\bar{x})) = 0$  is reached. However, as presented in Section 5.2, we are not interested only in finding a static equilibrium configuration but we also want it to be stable.

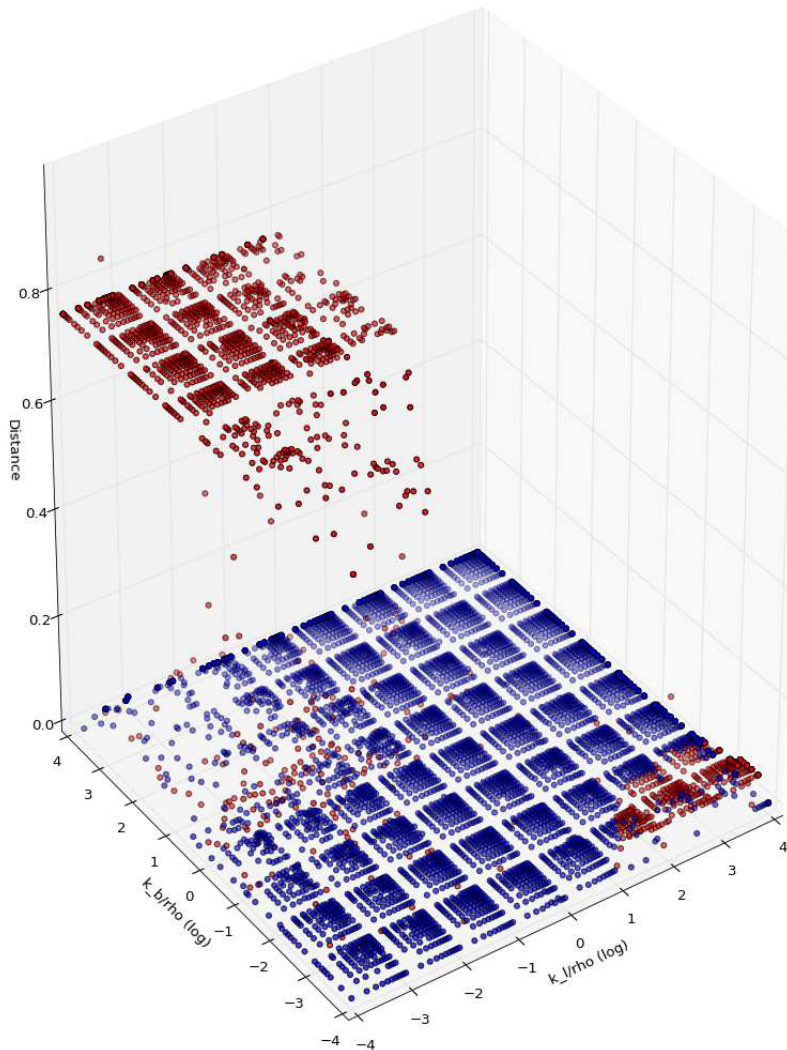


Figure 5.8: Distance between the target and the equilibrium shape on the **Bitriangle** example depending on the ratios  $k_L/\sigma$  and  $k_B/\sigma$ . The points in red denote a distance between the equilibrium  $x$  and the target  $x_t$  greater than  $10^{-2}m$ . The axis of the parameters are displayed in log scale.

Indeed, if the deformed shape  $x$  is an unstable equilibrium, any small perturbation (some noise at the beginning of the resimulation for instance) can shift it to another shape, therefore ruining our efforts to preserve the target.

We have seen in the first step that this stability constraint in the contact-free zones is enforced thanks to the draping procedure. As stated in Section 5.5.2, initialising the second step using the result of the first step enables the former to only have to perform a correction at the contact points and to

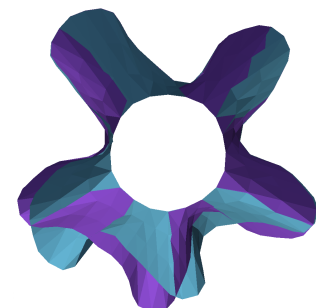


Figure 5.10: Target shape (purple), stable equilibrium (blue)



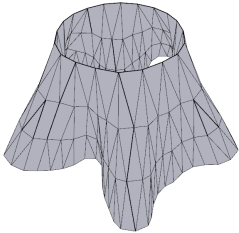
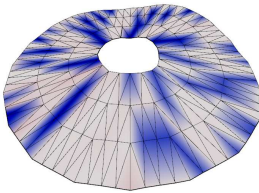
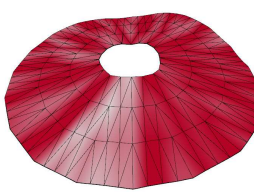
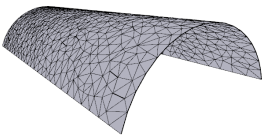
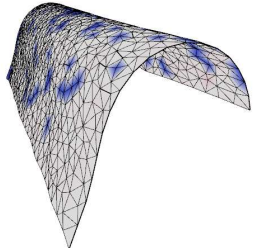
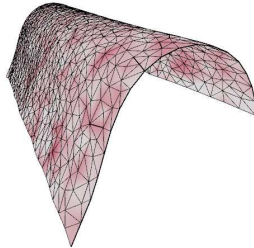
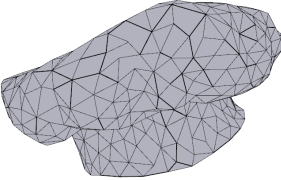
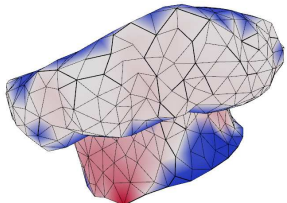
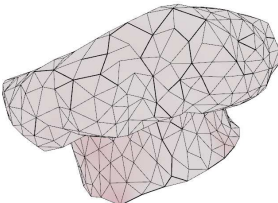
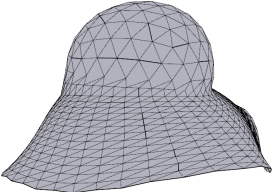
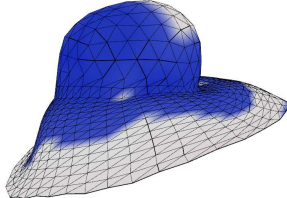
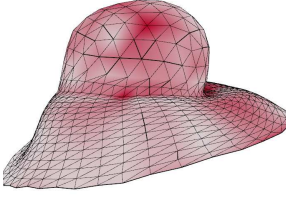
	Target	Natural Shape	
		Relative stretch	Relative bend
Synthetic Skirt			
Book Page			
Beret			
Floppy Hat			

Figure 5.9: For each of our examples, we display the target (first column) and the natural shape found by our algorithm (second and third columns). The relative stretching of the natural shape *w.r.t.* the target is depicted in colour in the second column in a blue (compression) to red (dilatation) scale. The same is done for the bending on a white to red colour scale indicating the difference of local curvature.

propagate this correction along the rest of the shell. The free points, starting from a stable equilibrium, are unlikely to shift to an unstable configuration and in practice we have never observed such a case.

To illustrate our point, we have solved the **Synthetic skirt** using on the one hand our full algorithm and on the other side only the second step. Both methods converge to a  $\bar{x}$  such that  $x_t$  is at equilibrium. However, when we introduce a small perturbation (a variation of magnitude  $10^{-2}$  in the value of the gravity), the target simulated using the natural shape computed using only the second step shifts to another equilibrium depicted in Figure 5.10 in blue, whereas the natural shape computed with both steps preserves the

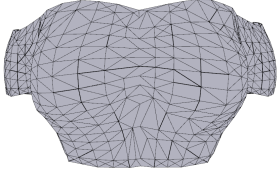
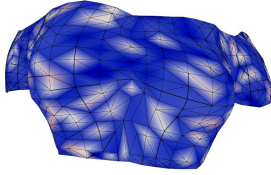
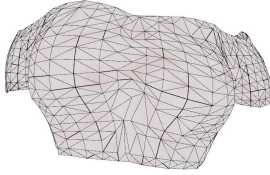


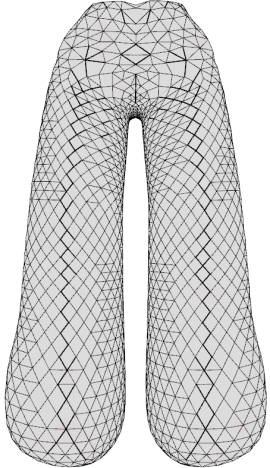
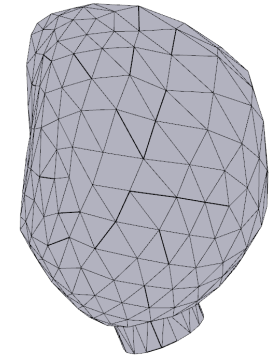
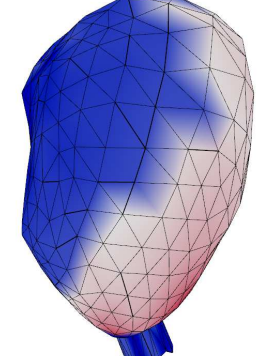
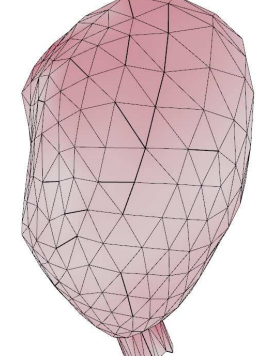
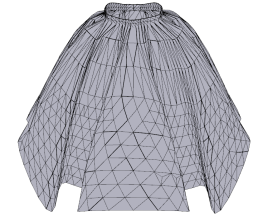





	Target	Natural Shape	
		Relative stretch	Relative bend
Top			
Saroual			
Puff Sleeve			
Gored Skirt			
ClothCap Shirt			

Figure 5.9: (Cont.)

initial target.

**Role of frictional contact during inversion.** As illustrated previously in Figure 5.3 depicting the waist of the **Gored Skirt**, inverting using only the first step is insufficient to deal with frictional contact as the resulting contact forces do not satisfy Coulomb’s law.

Resimulating using only fixed points may limit the range of possible animations, or lead to unrealistic motions. This is the case for the **Book Page**, presented in Figure 5.5 where we want the page to hold before we flip it using wind forces. Inverting and simulating with the fixed points only indeed yields a static stable equilibrium, but the page is unable to flip as two edges are fixed. Moreover, using the shape inverted with fixed points to simulate with frictional contact causes the page to sag as the contact forces are not correctly accounted for. In this particular case, using a larger friction coefficient makes it work, but it requires manual parameter tweaking. Moreover, it might not be sufficient in the general case as, like for the **Gored Skirt**, the contact forces might be *adhesive*, where sliding and sagging occur no matter the friction coefficient.

In contrast, using both steps for the **Book Page** yields a static equilibrium subject to friction forces with the wanted friction coefficient. Then the page can be animated nicely.

### 5.6.5 Convergence and performance

Example	$n_v$ <sup>1</sup>	$n_c$ <sup>1</sup>	$t_{exec}^1(\%)$ <sup>2</sup>	$\bar{n}_{iter}$ <sup>3</sup>	$t_{exec}^2(\%)$ <sup>4</sup>	$\bar{n}_{iter}^2$ <sup>5</sup>	$t_{exec}(min)$ <sup>6</sup>	$\Phi(\bar{x})$ <sup>7</sup>	$G(\bar{x})$ <sup>7</sup>
<b>Synthetic Skirt</b>	200	19	75	598	25	157	4	0	$1.5e^{-23}$
<b>Book Page</b>	817	143	46	75	54	2338	187	$1.5e^{-11}$	$1.2e^{-23}$
<b>Puff Sleeve</b>	402	67	68	509	32	389	14	0	$7.4e^{-23}$
<b>Beret</b>	846	69	48	883	52	3055	97	0	$4.5e^{-25}$
<b>Soft Hat</b>	876	178	51	500	49	723	50	0	$2.3e^{-22}$
<b>Top</b>	900	116	82	472	18	1800	382	$1.4e^{-19}$	$1.3e^{-19}$
<b>Saroual</b>	2500	61	26	858	74	3432	950	0	$4.5e^{-22}$
<b>Gored Skirt</b>	1255	168	47	1115	53	3149	427	$6.1e^{-22}$	$7.1e^{-22}$
<b>ClothCap Shirt</b>	2550	931	20	633	80	2751	750	0	$4.0e^{-23}$

<sup>1</sup> Number of vertices ( $n_p$ ) and contact points ( $n_c$ )

<sup>2</sup> Percentage of the computation time spent in Step 1

<sup>3</sup> Average number of iterations per  $\lambda$  in Step 1

<sup>4</sup> Percentage of the computation time spent in Step 2

<sup>5</sup> Average number of iterations per  $\lambda$  in Step 2

<sup>6</sup> Total computation time

<sup>7</sup> Final error  $\Phi(\bar{x})$  and constraint error  $G(\bar{x})$

Table 5.2: Performance of our inversion algorithm for all our examples

As depicted in Table 5.2, our inversion algorithm converges well for all of our examples

while applied onto a range of different material parameters. As intuited in Section 5.6.4, the stiffer the material is, the easier the problem is, and so the lower the computation time is. In contrast, softer materials yield harder problems to solve, and sometimes make our method fail as the draping function  $\Phi$  is not evaluated properly. In practice, when this happens (typically for **Saroual** or **Gored Skirt**), we simply choose a stiffer material and rerun the inversion. However, this manual intervention may lead to a tedious trial-and-error process that we want to avoid. We discuss further in Section 5.7.3 how we would like to lighten this process.

We can note that among our two steps, the first one is in most examples more costly than the second one, which can be explained by the draping function being a costly procedure. Another obvious observation is that both steps are affected by the number of vertices, and their repartition between free and contacting vertices. The more vertices there are in one category, the slower the convergence rate of the corresponding step is. Reducing the overall cost of our algorithm is mandatory to scale it up to larger meshes, and we discuss this issue in Section 5.7.1.

## 5.7 Conclusion, limitation and discussion

In this chapter, I have presented our algorithm to invert elastic shells subject to gravity and dry friction. Our method is able to match accurately the input target as demonstrated in our examples. However, our system does have some limitations that are discussed in this section.

### 5.7.1 Performance

As one may see in Table 5.2, the biggest limitation of our method is currently the computation time. In our examples that use relatively coarse meshes, computing the natural shape takes up to several hours.

From our observations, in general, the first step has a fairly good convergence rate but has a high cost per iteration. Indeed, the draping procedure is evaluated several times per iteration for the linesearch and is costly to evaluate as it requires several Newton-CG steps. Wang (2018) has proposed to use an adaptive tolerance to increase the precision at which the equilibria are computed as the optimisation advances. In practice, we had troubles to make this technique work consistently within our framework. As discussed in Section 5.4.1, the Newton or Gauss-Newton techniques can increase the convergence rate, but have a large cost per iteration (see D). A promising approach would be to find a good and inexpensive preconditioner.

The objective function of the second step is lighter to evaluate, but requires in general a lot of iterations to converge. We suspect that it is due to its formulation which consists in multiple interdependent local problems (while the draping is more "global") and that

the optimisation problem also has to navigate close to the manifold of the stable static free zones. For this step, we also tried higher order optimisation methods and did not find them to increase the convergence rate substantially.

Finally, the regularisation parameter  $\lambda$  also controls the convergence rate, mainly in the first step. As presented in Section 5.4.2, it allows to convexify the energy to remove local minima that may be fatal to our method, but, by doing so, it also dampens the dependence *w.r.t.* the natural shape, and thus the convergence. In the limit  $\lambda \rightarrow +\infty$ , almost all  $\bar{x}$  are a close solution and the minimisation does not need to advance. It is reflected in the adjoint system 5.18 where the first equation is regularised by  $\lambda \mathbf{I}$ . Finding a good value of  $\lambda$  that is just enough to carve a path to an equilibrium close to the target while not overdamping the convergence would be a nice addition to our method.

### 5.7.2 Flaws of the second step

When we defined the correction step in Section 5.5, we acknowledged that finding a residual that would not be close to 0 at this stage would not yield a static equilibrium at the end. However, as discussed in Section 5.6.4, in the absence of theoretical guarantees on the existence of a solution, we have to rely on our intuition to judge the performance of this step. In all of our examples, heuristics such as reducing the weight or increasing the friction coefficient consistently leads to finding solutions, but we do have no exact threshold values as it is always case-dependent.

Besides, a careful reader may have noticed that the deformed shape  $x$  is fixed in the correction step as we are trying to adjust the natural shape. For the contact points, the reason is that we did want to keep the Coulomb cones constant. Otherwise, allowing the contact points of the deformed shape to move would imply the introduction of a non-penetration constraint to keep them outside the obstacle and moreover to acknowledge the dependence between the position of each vertex and the corresponding contact normal  $e_i$ , which would have drastically increased the complexity of the problem. Regarding the free points, they have been handled in the first step and are unlikely to decide whether suitable contact forces exist. Ergo, we did not choose to add them as degrees of freedom, although they would be correctly constrained to be static by  $G$ .

A more compact, and theoretically safer formulation that combines both steps would be

$$\min_{\bar{x}} \frac{1}{2} \|\Phi(\bar{x}) - x_t\|^2 \text{ s.t. } G_{\text{contact}}(\Phi(\bar{x}), \bar{x}) = 0. \quad (5.27)$$

We test this formulation by introducing the constraint as a quadratic penalty, and find that, although it is indeed able to compute a solution to the inverse problem with frictional contact, it also brings out the worst of the two steps in terms of computation time. The draping function  $\Phi$  evaluates  $x$  oblivious to the constraint on the contact forces  $G_{\text{contact}}$  and the introduction of the constraint leads to a slower convergence rate and thus more evaluation of  $\Phi$ . In the end, we do not find this formulation more practical and robust

than our two-step process.

### 5.7.3 Material parameters

In our approach, we currently leave the choice of the material properties to the user, which appears to us as beneficial in the context of an animation process, where an artist may want to tune the parameters to get the desired behaviour. As discussed in Section 5.6.4, we have no way to give the user accurate bounds on the range of feasible parameter, as it was done in the case of isolated fibres (Derouet-Jourdan and Bertails-Descoubes, 2010), but we have checked that following some intuitions, such as choosing stiffer materials for complex shape, allow to find solutions where the target is a stable equilibrium. However, this may make the inversion process tedious and cumbersome, especially if the user wants a soft material.

In the next chapter, Section 6.2.2, we explore one avenue that could alleviate this issue by exploring the material parameter space and propose in the end the user to choose between several appropriate values.

Besides, in the context of 3D capture of real garments, an interesting extension of our method would be to use different poses to try to infer the material parameters along with the common natural shape. Such an approach would be complementary to the capture process, and provide a ready-to-simulate garment.

Very close to that end, a recent and promising avenue lies in differentiable physics. The core idea is to build a simulator of a physical process such that the output is a differentiable function of some parameters. These parameters can then be inferred from observed situations or be used in control problems through either a neural network which has learnt the "simulation function" or a numerical optimisation scheme also relying on the differentiable function. These methods have been successfully used to control simple fluid equations (Holl et al., 2020), soft robots manipulating deformable objects (Hu et al., 2019; Geilinger et al., 2020), inverse rendering (Loubet et al., 2019) and, closer to our work, cloth material optimisation using dynamic sequences (Liang et al., 2019). However, in our case, using such differentiable simulators would transform our static inverse problem into a dynamic inverse problem and additional work would be required to handle the static case.

### 5.7.4 Toward garments inversion

As the focus of the method is animation, we chose to interpret the designed surfaces as shells that offer more expressiveness than flat patterns. Indeed, thanks to the natural curvature of shells, we managed to handle examples such as the **Book Page** for which no flat inverse shape exists, or **Gored Skirt** for which the modelling through flat patterns would have been complex. Then in our subsequent simulations, we did not observe arte-

facts due to this natural curvature even in the case of **Synthetic Skirt** where we expected a more flat rest shape. In this example, the natural curvature is small compared to that of the deformed shape, and thus its impact is no visible. However, this approach may be problematic when dealing with soft garments exhibiting folds and wrinkles as explained below.

In all of our examples, the folds are part of the intended design and are localised in contact-free zones; they do not stem from contact interactions, which happen at tight parts of the mesh. Yet, recall that in Section 5.4, we fixed the contact points of the deformed shape *and* of the rest shape. With this constraint, the first step is allowed to only displace the contact-free vertices, and thus perturbs at most the forces at the boundary of the contacting zones. However, with badly placed contact points, the natural shape may be unable to unfold.

Consider the example in Figure 5.11a, where we add to the **Synthetic Skirt** example a ground that is slightly contacting the bottom of the mesh. Inverting with the contact points fixed on the natural shape during the first step prevents it to lift, and another less intuitive solution depicted in Figure 5.11b is found.

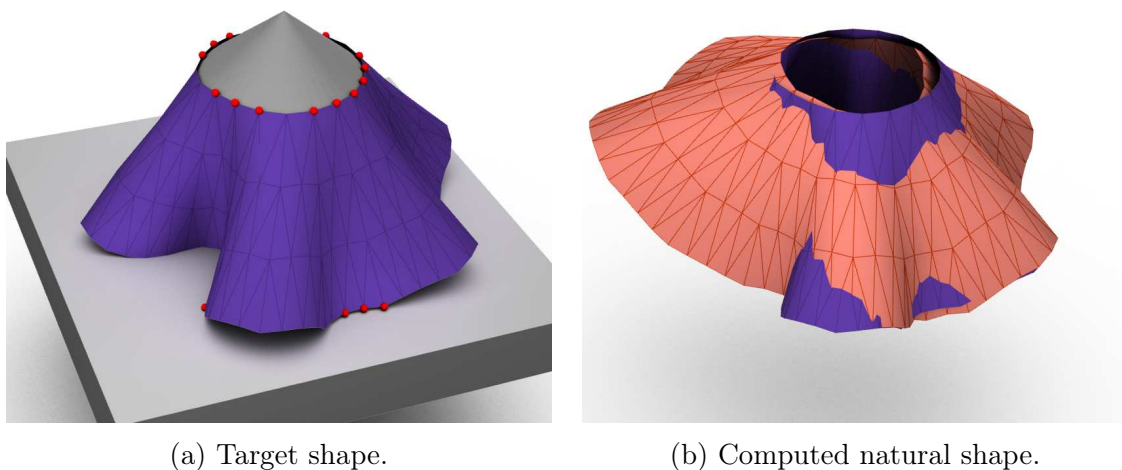


Figure 5.11: This example is close to **Synthetic Skirt**, but with additional contacting points in red (a). Fixing their components on the natural shape during the first step prevents it to unfold, and leads to less flatten shape than in the original example (b - in orange).

Allowing the first step to use the full  $\bar{x}$  and not only the free components solve the issue in this example. However, the problem remain in the case of large contact zones where their physics is totally discarded in the first step. The interior of such regions have then no reason to be moved and surrounding folds remain as depicted in Figure 5.12. In this example, a large portion of the dress is in contact with the character and the first step finds a natural shape rendered in Figure 5.12b that exhibits the same folds as the deformed garment and the resulting animation is unrealistic.

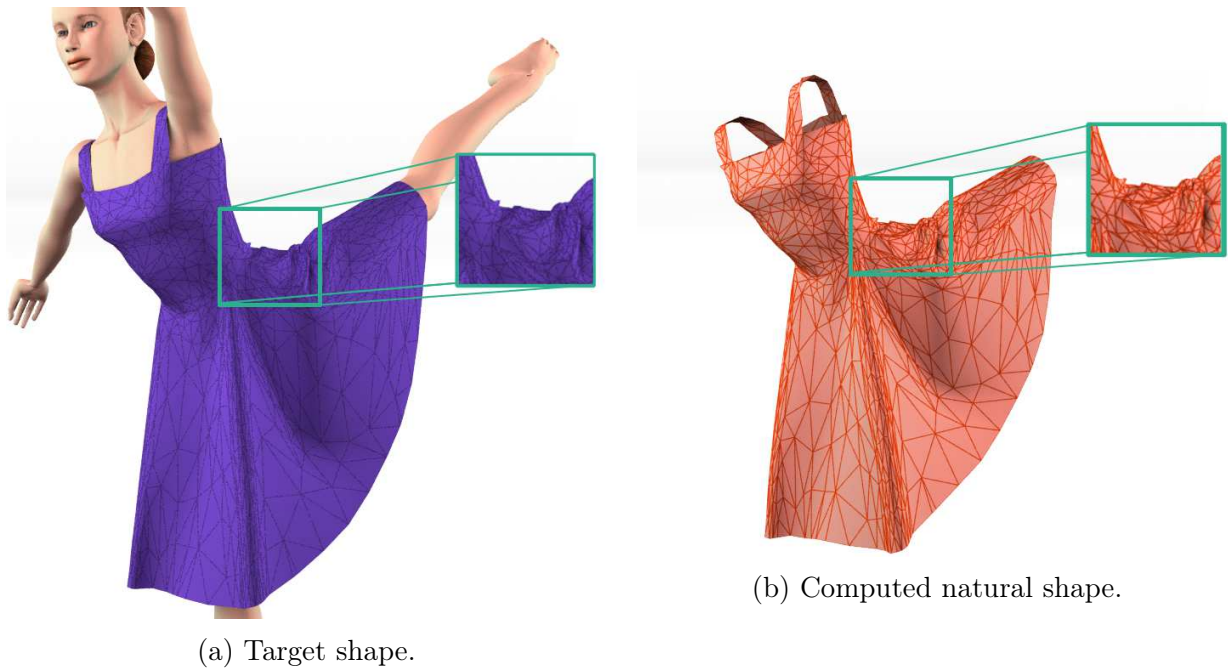


Figure 5.12: Inverting a frame of the **Arabesque** of (Li et al., 2018a). Because the contact zones are ignored by the first step of our method, our algorithm does not unfold the dress and thus leaves wrinkles on the garment natural shape.

The latter case is a typical example where one would rather like to have the tissue flatten during inversion. This request is even stronger if the final application is to fabricate real cloth from flat patterns: in such a case the recovered natural shape must be made of developable patches. We follow this avenue in the following chapter, where our modifications to the algorithm to try to favour developable rest shapes are presented.



# Chapter 6

## Towards garment inversion

In order to apply our inversion method to real garments, we have identified two issues that we would like to tackle. First, aiming at computing cloth patterns, we would like in a first step to be able to yield developable natural shapes. Then, we also would like to be able to estimate the material parameters of the garment to faithfully invert, simulate and/or fabricate it.

Within this chapter, I present the modifications done to the inversion algorithm to the ends aforementioned. As this work is prospective, we are not able to deal with complex examples although we will identify the pros and cons of our approach.

## 6.1 Including discrete developability

### 6.1.1 Framework

**Single developable patch.** As forementioned, the goal is to obtain a developable natural shape, that is a surface that could be isometrically flattenable without self intersecting and possibly with the help of arbitrary cuts. For instance, a cylinder complies with that definition as, with any cut that turns it into a surface of topological genus zero (*i.e.* with no holes), the resulting surface is flattenable.

Yet, as one may know, even the simplest garment is composed of several patches sewn together with possibly some seams that introduce singularities in the curvature. In such a case, the resulting surface is said to be piecewise developable. Handling such examples would therefore require the localisation of such singularities.

For a first approach, we decided to consider simple examples where the natural shape can be assumed to be developable as a whole without the introduction of singularities. Note that this does not limit us to flat panels as some surfaces of higher genus, such as the aforementioned cylinder or cones, are more complex shapes that can be addressed within the same framework.

To tackle natural shapes with singularities, future work would involve the development of strategies to either adjust user-drawn seams (Skouras et al., 2012) or cone singularities (Soliman et al., 2018) or to automatically make the seams emerge (Stein et al., 2018).

**Developability and inverse physics.** A naive approach to solve our problem would be to apply one of the algorithms of the discrete geometry literature to flatten or make the natural shape developable. We show in the example detailed in Figure 6.1 that this approach does not work.

We consider again the **Synthetic Skirt** presented previously in Figure 5.4. We apply the algorithm summarised in Algorithm 6 that consists in alternatively applying the shell inversion algorithm 3 presented in the previous chapter and the algorithm of Stein et al. (2018) using the authors' code to make it piecewise developable. The resulting shapes are depicted in Figure 6.1.

Overall, the process does not appear to converge. We can observe that the shape gets more and more distorted because each stage solves its own problem, whether physic or geometrical, and breaks what the other stage has computed earlier.

On this example, we can note that the original torus cannot be retrieved unless the waist is correctly tightened. However, there are many solutions that satisfy the static equilibrium condition and once the inversion stage has found a solution that is unlikely to be developable, it has no reason to further adjust the waist based on geometric considerations.

Such competing problems between physics and geometry encouraged us to incorporate

---

**Algorithm 6:** Naive algorithm to compute a developable natural shape.

---

**Data:** Target equilibrium pose  $x_t$ , initial regularisation factor  $\lambda_0$ , regularisation reduction factor  $\alpha \in ]0; 1[$

**Result:** A pair  $(x, \bar{x})$  consisting of a stable equilibrium pose and a developable natural pose with  $x$  as close as possible to  $x_t$

```

1 while there is no convergence do
2    $(x_{k+1}, \bar{x}_{k+\frac{1}{2}}) \leftarrow$  Apply the shell inversion algorithm 3 to the target  $x_t$  with
    $\bar{x}_{init} = \bar{x}_k$ , the parameters  $\lambda, \alpha$ ;
3    $\bar{x}_{k+1} \leftarrow$  Apply the algorithm of Stein et al. (2018) on  $\bar{x}_{k+\frac{1}{2}}$ ;
4    $k \leftarrow k + 1$ ;
5 end
6 return  $(x_k, \bar{x}_k)$ ;

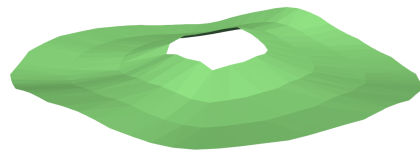
```

---

the developability constraint directly in the physical inverse problem.



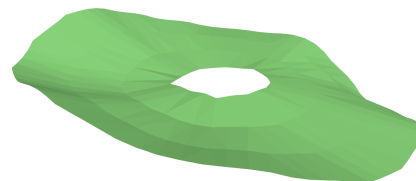
(a) Target shape (in blue) generated by dropping the flat torus (in red) on the cone.



(b) Natural shape computed by our algorithm.



(c) Shape after applying the algorithm of Stein et al. (2018) on the shape of Figure 6.1b.



(d) Natural shape after inverting using the shape of Figure 6.1c as a warmstart.

Figure 6.1: Alternating between our inversion algorithm and the algorithm of Stein et al. (2018) to make the surface developable does not allow us to retrieve the initial torus. The process does not converge.

**Soft constraint *v.s.* hard constraint.** Aiming at obtaining a developable natural shape, the developability criterion, that I will discuss in next section, has to be treated

as a hard constraint to truly enforce the geometric property.

Yet, in our numerical experiments, optimisation methods that enforce too "strictly" the constraint such as interior point methods are unable to descend the objective of the inverse problem. We suspect that navigating on the manifold of developable surfaces is too restrictive for the inverse problem that is already highly non-linear.

The algorithm that works best for us is the Augmented Lagrangian Method (ALM) that starts by treating the constraint as a quadratic penalty, allowing the optimisation problem to be treated with standard optimisation methods. In practice, we continue using the BFGS algorithm. The weight of the penalty is then progressively increased and a Lagrangian multiplier is updated to enforce the constraint without the need to raise the weight to infinity. This strategy decreases both the main objective and the constraint in a smooth manner before progressively focussing on the constraint.

### 6.1.2 Comparison of the discrete developability criteria

In this section, I compare the three formulations of the discrete developability introduced in Section 4.2 in order to select the one to incorporate in our inversion process.

We introduce the **Hanging sheet** case that consists of a square sheet hanging under gravity with its four corners fixed as depicted Figure 6.2. The purpose of this example is to have a simple case to test the inversion process with the developability constraint and without the Coulomb constraint, allowing us to deal with a single optimisation problem.

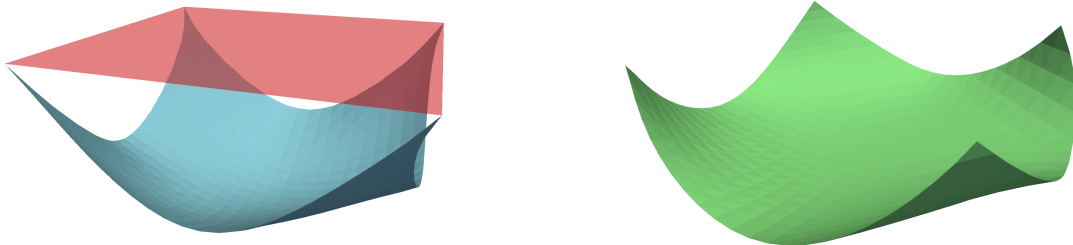
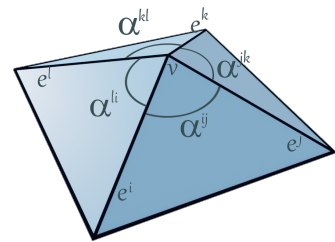


Figure 6.2: **Hanging sheet** example generated with DISCRETE SHELL. The original rest shape is in red, and the deformed equilibrium shape in blue. The natural shape computed by our algorithm is the curved shape in green.

**Discrete Orthogonal Geodesic (DOG) nets** (Rabinovich et al., 2018a). The first criterion we discuss is the DOG criterion of Rabinovich et al. (2018a). The condition for a *quad* mesh to be developable is that for any interior point, the four adjacent angles are equal.

However, for a given surface, this criterion reads as if the surface is developable then *there exists* a quad parametrisation

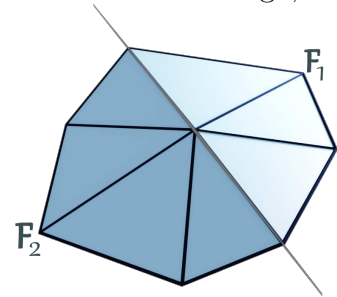


that satisfy the DOG condition. This turns out to be impractical for us, as the natural shape is evolving through the inversion process and its final ‘developable parametrisation’ is unknown.

To illustrate our point, consider the **Hanging sheet** example generated from a regular grid (Figure 6.3a) and a grid where three rows have been slightly shifted (Figure 6.3b, highlighted in red). We apply our inversion algorithm on both of them, incorporating the DOG constraint using the ALM as stated in the previous section, and we obtain respectively the results depicted in Figures 6.3c and 6.3d.

In the case of the regular grid, we correctly retrieve the initial flat square. However, with the slightly perturbed parametrisation, we find a curved natural shape and the associated equilibrium  $x$  does not match the intended deformed shape. The flat configuration cannot be reached as its parametrisation does not satisfy the DOG constraint. Ergo, we discard this formulation for our inversion problem.

**Developability of Triangle Meshes (DTM)** (Stein et al., 2018). Next, we discuss the criterion for the *Developability of Triangle Meshes* (DTM) of Stein et al. (2018). The criterion introduced in their paper allows the characterisation of piecewise developable surfaces, which can be of great interest for our application to cloth patterns. However, their numerical formulation is rather complex, resorting either to combinatorial or to eigenvalue computation. We also tried to use the [author’s code available online](#) on the deformed shape of **Hanging sheet** (without any consideration of the physics), but we often get an error in the linesearch, preventing an accurate convergence.

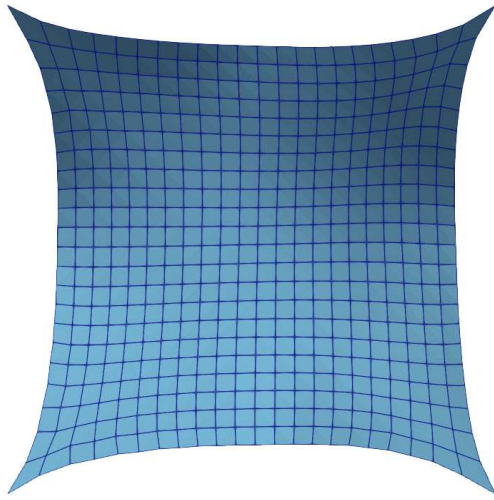


To reimplement their criterion, we have derived another functional formulation. Recall that the DTM criterion for the piecewise developability consists in saying that for each interior vertex, there are at most two different face normals among the adjacent faces as depicted in the inset figure. We reformulate this idea by stating that the cross-product between two consecutive faces normals of the one-ring neighbourhood is either zero (the two normals belong to the same class of the partition) or define the hinge direction (in grey on the figure) that is orthogonal to all the normals. Finally, we arrive to this numerical criterion ,

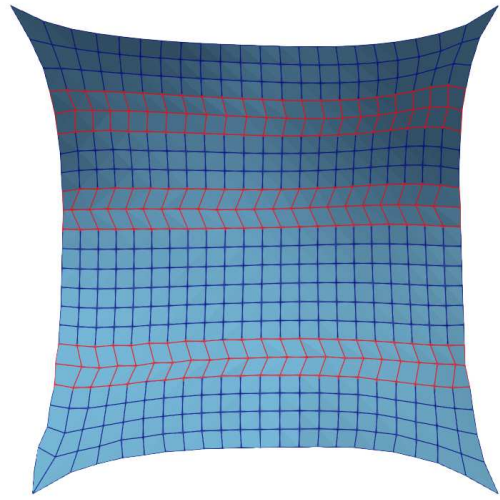
$$C_{DTM|i}(\bar{x}) = \sum_{\substack{i \in f, f_i, f_{i+1} \\ f \neq f_i, f_{i+1}}} \left( \frac{n_{f_i} \times n_{f_{i+1}}}{\|n_{f_i} \times n_{f_{i+1}}\|} \cdot n_f \right)^2 \quad (6.1)$$

where  $n_f$  is the normal of the face  $f$ . When the vertex  $i$  is an hinge, either the cross-product or the scalar product is zero.

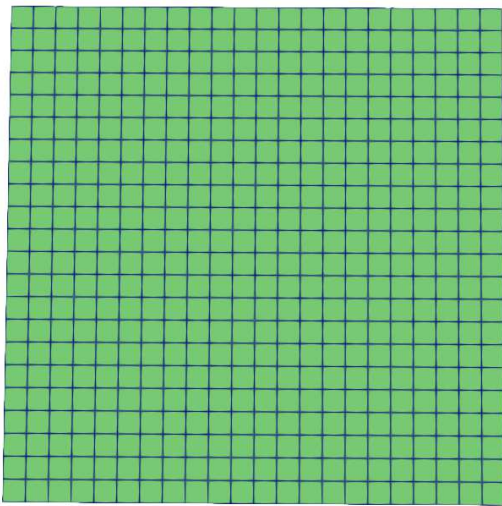
We test this formulation again on the **Hanging sheet**, first outside the inversion algorithm. The convergence behaviour appears to improve as we are able to run the optimisation smoothly. Yet, it seems that our formulation favours too much flat config-



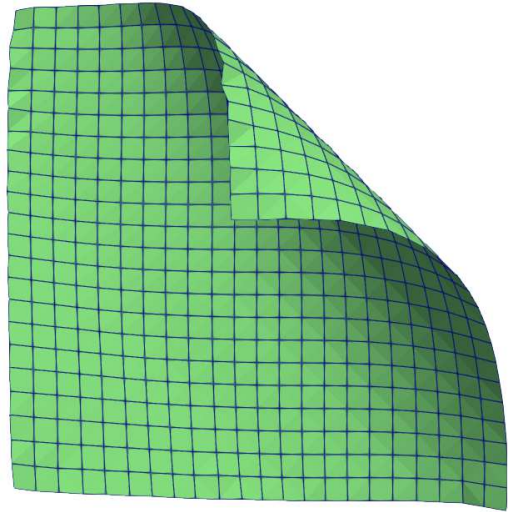
(a) Target shape (regular grid).



(b) Target shape (perturbed grid).



(c) Computed natural shape (regular grid).



(d) Computed natural shape (perturbed grid).

Figure 6.3: Inverting the **Hanging sheet** with the DOG constraint. With the regular grid (left column), we retrieve the initial flat square. In the other case, the DOG constraint prevents to reach the square, and furthermore the associated deformed shape  $x$  does not match the target.

urations compared to simply smooth regions and generates an uneven repartition of the vertices positions. The resulting mesh is displayed in Figure 6.4.

Further combining this formulation with the inversion process yields a poor convergence rate, as this over flattening behaviour seems to be opposed to the physical considerations.

After comparing to the next formulation, we have also decided to discard this one as

it appears too complex for our single patch objective. However, this criterion may be an interesting avenue for future work involving shapes composed of several patches.

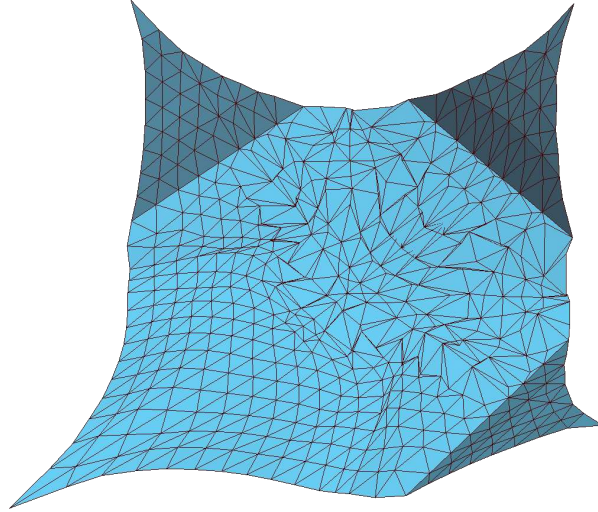
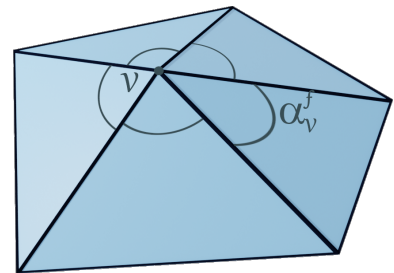


Figure 6.4: The deformed shape of the **Hanging sheet** after being made developable by our reformulation of (Stein et al., 2018). Flatness is favoured against smooth surfaces and the preservation of the vertices repartition.

**Discrete Gaussian Curvature (DGC).** Finally, the discrete developability criterion we take on is the Discrete Gaussian Curvature. In its simplest form, it requires for all interior vertices  $i$  to have the adjacent angles sum up to  $2\pi$ . Additional weighting based on the triangles areas or the cotangent weights can be used if we wish to integrate this quantity over the mesh or to simply acknowledge for a non-uniform discretisation. As a first approach, we consider the meshes uniform enough to not include such weighting. We will remove this assumption later in Section 6.3.3 when we will use meshes with non-uniform sampling, but for now the numerical criterion is



$$C_{DGC|i}(\bar{x}) = 2\pi - \sum_{\substack{f \in \mathcal{F} \\ i \in f}} \alpha_i^f. \quad (6.2)$$

Using this formulation, we are able to solve perfectly two **Hanging sheet** cases with the regular grid and the perturbed grid, retrieving each time the flat square as depicted in Figure 6.5.

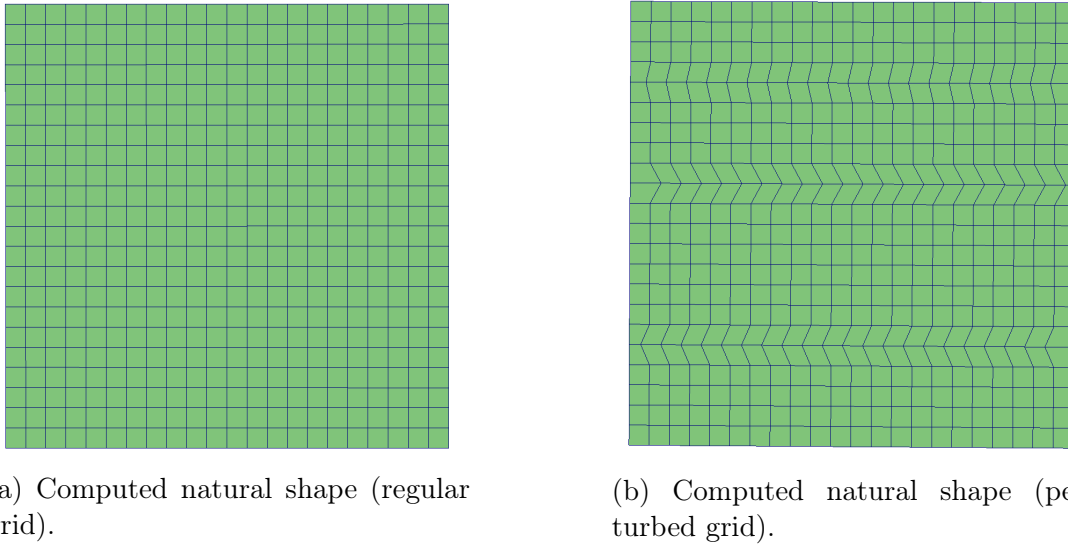


Figure 6.5: Inverting the **Hanging sheet** with the DGC constraint. With both parametrisation, we manage to retrieve the initial flat square.

## 6.2 Modifications of the inversion algorithm

In the previous section, I have selected the geometric criterion to be incorporated to our inversion process. Now, I describe the modifications brought to the algorithm presented in the previous chapter in the objective of an application to real garment inversion.

### 6.2.1 Plates or shells ?

Since we have mentioned the final objective of reconstructing cloth patterns, we have argued in favour of having a developable rest shape in order to flatten it.

However, a shell is in part defined by its natural curvature. Ergo, flattening would change its natural curvature to zero and hence the resulting bending forces that hold the static equilibrium.

If we consider that this difference in bending to be non-negligible, then a solution would be to remove any dependence to the natural curvature in the energy. For instance, we can assume it always to be zero no matter the real curvature of the natural shape ( $\bar{\theta} = 0$  in 1.12 for the DISCRETE SHELL model for instance). As such, any natural shape found by solving the inversion problem can be isometrically deformed without introducing any modification to the bending energy and so a developable natural shape can be flattened.

Yet, this approach makes the inversion problem more ill-posed as the solutions become invariant by isometry. In practice, we have observed that solving the inversion problem indeed yields a slower convergence rate and is prone to get stuck into local minima that are far from the target shape. This is in part due to the fact that some vertices such



as the corners can still move even if the length of the adjacent edges is fixed, preventing the existence of a strict local minimum. But most importantly, as the curvature is not considered anymore, the natural shape crumples and may form local minima as the vertices become unable to move to reach the true optimal length.

As an example, the **Hanging sheet** example inverted with our method and this plate-like energy without the developability constraint yields the crumpled natural shape depicted in Figure 6.6. Nonetheless, this example is simple enough so as this crumpled shape is isometric to the flat square (the maximum relative length difference is  $\approx 4 \cdot 10^{-8}$ ) and so the associated equilibrium shape matches exactly the target

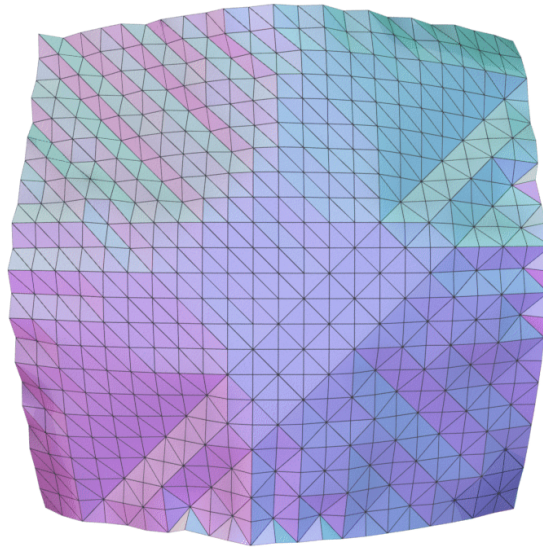


Figure 6.6: Natural shape computed by our algorithm using the plate-like energy. The face colours depend on the face normal, and show that the mesh is slightly crumpled.

To avoid the crumpling and the undefiniteness of the solution, geometric regularisers can be used. The developability criterion helps regularise the problem as it constraints the Gaussian curvature  $\kappa_G$ , and thus one of the two degrees of freedom in curvature. To enforce the smoothness of the surface, we can also add a regulariser on the mean curvature  $\bar{\kappa}$ , whose discrete form is

$$\frac{1}{2} \sum_{e \in \mathcal{E}_{int}} L_e \theta_e \quad (6.3)$$

with  $L_e$  the length of the edge  $e$  and  $\theta_e$  the dihedral angle at the edge  $e$  as defined in Section 1.1.2 (Cohen-Steiner and Morvan, 2003; Pellis et al., 2019).

In order to minimise the variations in curvature, we will use the following formulation

$$C_{smooth}(\bar{x}) = \frac{1}{2} \sum_{e \in \mathcal{E}_{int}} \bar{\theta}_e^2. \quad (6.4)$$

As for the DGC, the weighting term is omitted for now and will be reintroduced later on

in Section 6.3.3 where we consider non-uniform meshes.

With these additional geometric terms, inverting the **Hanging sheet** leads to the initial flat square as this shape is realising the global minimum of the physical objective and the smoothness regulariser while satisfying the constraint.

The other option is to consider that the error introduced by flattening the shell is negligible. Although this may not be the case for the DISCRETE SHELL model, where the stretching and bending stiffnesses are two unrelated coefficient, this assumption appears to be more valid for physics-based plate models such as the Koiter energy implemented by LIBSHELL.

In these plate models, the stretching stiffness has a linear dependence to the thickness  $h$ , while the bending stiffness has a dependence to the cube to  $h$  and moreover  $h$  is small so as to satisfy the thin plate assumption. Ergo, the flattening introduces a perturbation of magnitude  $\approx h^3\kappa$  within the energy, which is relatively small compared to the other quantities.

As the flattening step is not tackled in this thesis, the impact of that assumption remains to assess. If the bending error proves to be non-negligible regarding the stability, a solution would be to use the result of the more well-posed inversion problem with the shell model to warmstart the problem using the plate-like energy in order to improve the convergence rate. For the following experiments, we choose to stick to the shell version of the energy.

## 6.2.2 Parameter identification through continuation

Another problem that we have mentioned in the previous chapter is the choice of the material parameters. Apart from their intuition and the heuristic of the almost fully rigid shell, the user has no choice but to try some material parameters and see if the algorithm manages to solve the inverse problem. In the following, we show that this heuristic can actually be leveraged in order to numerically seek for the limit of the possible material parameters in the case of pinned contact points.

**Continuation on the weight.** Consider therefore the case where the target shape is a thin elastic shell with some vertices fixed. In the limit of rigidity, the target shape is also a valid natural shape. Within our framework, this case is equivalent to the situation where *the weight is negligible*.

As such, we devise a continuation strategy where we solve subsequent problems by increasing the weight. This method allows to explore one direction of the parameter space until the inversion algorithm fails, while also enhancing the convergence rate of each problem that can be warmstarted by the solution of the previous and simpler problem. In

practice, we increase each time the surface density  $\sigma$  or the volumic mass  $\rho$ .

We apply this strategy to the **Hanging sheet** example generated by DISCRETE SHELL using  $\sigma_{ref} = 0.25 \text{ kg} \cdot \text{m}^{-2}$ ,  $k_L = 5.0 \text{ N}$ ,  $k_A = 2.0 \text{ N} \cdot \text{m}^{-1}$  and  $k_B = 10^{-2} \text{ N} \cdot \text{m}$ . We start from  $\sigma = 0$ , where the solution is trivial and increase the mass 30 times to reach the reference value  $\sigma_{ref}$  before going further.

Unsurprisingly, we are able to find rest and equilibrium shapes that exactly solve the problem until  $\sigma = \sigma_{ref}$ . We are able to push further and find solutions matching exactly the target until  $\sigma \lesssim 0.32 \text{ kg} \cdot \text{m}^{-2}$ . After this value, we only find approximating equilibriums.

Some of the obtained natural shapes are presented in Figure 6.7. We can see that, as expected, the natural shape contracts and curves, mainly at the corners, in order to compensate the increasing weight (Figures 6.7a, 6.7b and 6.7c). For  $\sigma > \sigma_{ref}$ , the centre of the shape also starts to buckle.  $\sigma = 0.3167 \text{ kg} \cdot \text{m}^{-2}$  which is the last value for which we exactly match the target.

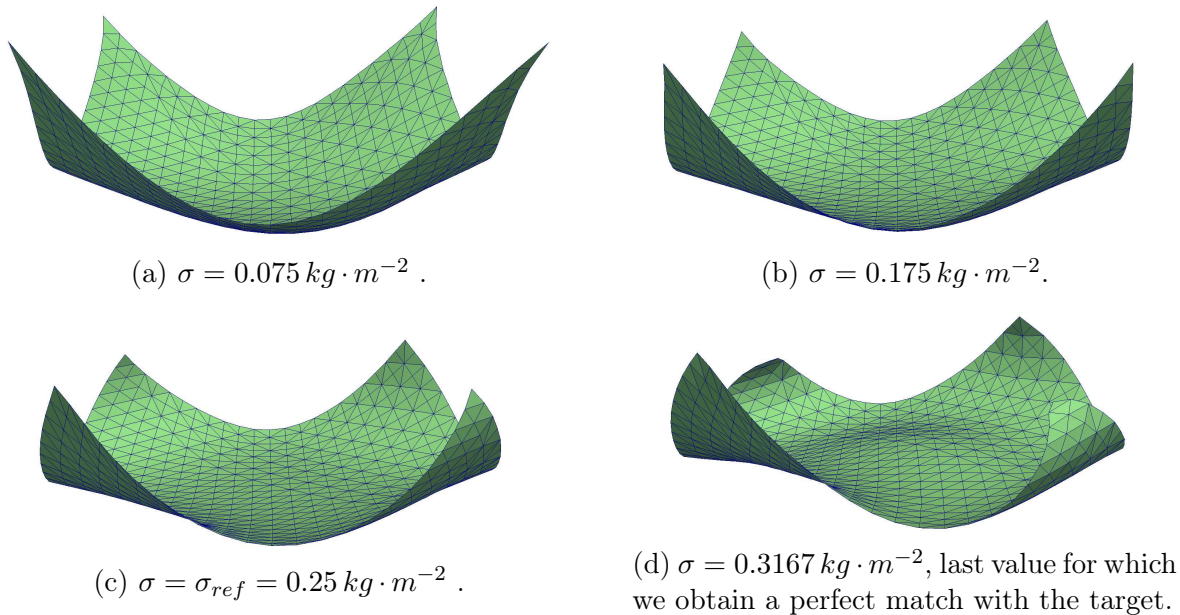


Figure 6.7: Inverting the **Hanging sheet** without the developability constraint and with a continuation on the surface density  $\sigma$ .

From a practical point of view we continue to solve the optimisation problem using the BFGS algorithm like in the previous chapter. The Gauss-Newton method is also an effective method here and proves to be efficient in terms of convergence rate as each subproblem is nicely warmstarted, unlike in Section 5.4.1 where the warmstart was lacking in order to yield better results than BFGS. Note however that the Gauss-Newton method requires to compute the Jacobian of the implicit function, which, as detailed in Section D, requires to solve  $n + 1$  linear systems.

With this exploration strategy, we can therefore remove one parameter ( $\sigma$ ) from the model, which leaves as free parameters the ratios  $k_L/\sigma$ ,  $k_A/\sigma$  and  $k_B/\sigma$ . Moreover, this continuation on one parameter becomes more interesting when the other parameters can be determined using other considerations.

**Determination of real material parameters.** Consider the case where we want to invert a shell made of a real material from which we have a sample. If we assume the fabric to be isometric and uniform, then it can for instance be modelled by the Koiter shell energy that we treat using the LIBSHELL code and the free parameters are the volumic mass  $\rho$ , the Young modulus  $E$ , the Poisson ratio  $\nu$  and the thickness (assumed to be uniform)  $h$ .

From the sample,  $\rho$  and  $h$  can be relatively easily measured. Then, by using the **Cantilever** measurement protocol, we can simply recover  $E^* = E/(1-\nu^2)$  from geometric observations, which leaves only one unknown parameter to recover. To determine the last parameter however, one would need to resort for instance to traction experiments and recover the stress-strain curve to get  $E$ . Yet, with our continuation strategy, this free parameter can be left to be determined and will be numerically estimated by our procedure.

By considering that  $E$  is unknown, it is tempting to perform the continuation method by starting with  $E \approx +\infty$  to be in the situation where the gravity is negligible, and then progressively decrease  $E$  but this approach is numerically very ill-posed as it yields a very stiff material, which is unsuitable the computations. However, as we noted four paragraphs above, the true free parameters are the ratios between the parameters. As such, we can fix a reasonable and arbitrary  $E^0$ . Then our continuation procedure will yield a corresponding value  $\rho^0$ , which can in turn be used to recover the true  $E = \frac{E^0 \rho}{\rho^0}$ .

**Adding the LIBSHELL code.** Because we aim at applying our algorithm to real garments, we turn our attention to the LIBSHELL code as the model parameters are directly identifiable from real physical parameters.

Furthermore, its membrane model stems from the well-known linear elasticity and we showed in Sections 3.5.2 and 3.5.3 that its bending model appear more accurate than that of DISCRETE SHELL.

As stated in the Section 5.3.1, our algorithm also works with this model, albeit in practice we observe a slightly slower convergence rate in general for the original inversion problem 5.14 as the difference in order of magnitude between the stretching stiffness and the bending stiffness is bigger here ( $\propto h$  *v.s.*  $\propto h^3$ ) than with the coefficients we selected in our experiments with DISCRETE SHELL (see *e.g.* Table 5.1).

In the results section of this chapter (Section 6.3), we present results using both DISCRETE SHELL and LIBSHELL codes.

**Penalised energy.** Recall that we introduced in Section 5.4.2 a regularisation term to the energy in order to ease the inversion process. Now, with the continuation strategy, one could argue that this penalisation is not required anymore as the continuation provides a good warmstart.

From our observations, this statement is half-true as indeed the solution with the previous  $\rho$  may be a good warmstart, although sometimes not good enough if the jump  $\delta\rho$  is "too big". Thus, we choose to keep this penalisation term with a low value of  $\lambda$  in order to make the continuation process more robust. This decision is further motivated as different discussions in Sections 6.2.1 and 6.2.3 advocate for a final optimisation problem where all the relaxations are removed and a "true" problem is solved with a proper warmstart thanks to the regularised problems.

### 6.2.3 Incorporating the continuation

**Continuation *v.s.* plate-like energy.** We discussed the possibility in Section 6.2.1, of removing the dependence to the natural curvature. As one could have noticed, this strategy is incompatible with the continuation method described in this section.

Indeed, even in the absence of external forces ( $\rho = 0$  in our case), the target shape is not its own natural shape with the plate-like energy as internal bending forces will tend to unfold it. This observation ergo advocates for the second strategy regarding the bending error discussed, that is deal with the inversion with a shell model and then potentially account for the error introduced by the flattening.

**Developability constraint or penalisation.** Previously in Section 6.1, we showed that the developability constraint can be strictly enforced using for instance ALM to solve the optimisation problem.

However, favouring the geometric constraint may ruin the continuation method that relies on close iterates of the subsequent physical problems. In other words, fully enforcing the developability could yield a solution that is a bad warmstart for the next value of  $\rho$  and make this stage fail.

As there needs not be a solution satisfying exactly the physics and the geometry for each value of the mass, we depart from the idea of a constraint and rather formulate the developability as a penalty. With this approach, we aim at making emerge a best value of  $\rho$  that may realise or not the zero of both the physical and geometric terms. In the latter case, we can then enforce the constraint at the chosen mass value in order to get an approximated equilibrium shape and a developable natural shape.

The only requirement to this strategy is to make sure that the penalty gently encourages the shape to be developable without either being predominant or conversely dominated by the inversion term. To this end, we use the following heuristic. Recall that we noted  $J(\bar{x}) = \frac{1}{2} \|\phi(\bar{x}) - x_t\|^2$  the least-square inversion term (Equation 5.14) and

$C_{DGC}$  the developability term (Equation 6.2). The objective we consider is

$$\min_{\bar{x}} J(\bar{x}) + \alpha_{DGC} C_{DGC}(\bar{x}) \quad (6.5)$$

with  $\alpha_{DGC} \in \mathbb{R}_+$  the penalisation coefficient. Other regularisers such as  $C_{smooth}$  are neglected here. We would like to adjust  $\alpha$  such that the optimisation is steadily and slowly minimising the  $C_{DGC}$  term. We translate this condition as

$$\nabla(J + \alpha_{DGC} C_{DGC})^\top (\nabla C_{DGC}) > 0 \quad (6.6)$$

that is the gradient of the total objective is also a descent direction for the developability term. This yields

$$\alpha_{DGC} > -\frac{\nabla J^\top \nabla C_{DGC}}{\nabla C_{DGC}^\top \nabla C_{DGC}}. \quad (6.7)$$

In practice, we compute this coefficient at the beginning and every few (10) iterations and only take increasing values (*i.e.* the max between the previous and the new value of  $\alpha$ ). Also, to prevent this geometric term to break the continuation by overpowering the physics, we halve  $\alpha_{DGC}$  and stop its update for the remaining  $\rho$  if  $J(\bar{x})$  has grown since the last update. We consider that we have pushed the shape enough towards the geometric term, and now let  $J$  be optimised in order to warmstart the next iteration of the continuation.

## 6.2.4 Full algorithm

All in all, the new proposed strategy that includes both the developability and a continuation on the parameters is summed up in Algorithm 7.

As stated in the previous sections, the final step (Lines 13-16) has not been treated in this thesis and is left as further work. Yet, we can already discuss the *best* criterion on Line 14. In our tests where we applied the continuation part to synthetic examples, or cases that stemmed from perturbed synthetic examples, we always have a good intuition of the value of  $\rho$  that should be found. Numerically, we observed that the best value realised a concomitant minimal value for both  $J^\lambda$  and  $C_{DGC}$  over all the  $\rho$  iterations. However, for more complex cases, one would have to favour the iteration with the smallest of the two as the corresponding physic term is the less likely to change in the final problem.

Regarding the *successfulness* criterion of Line 6, in practice we stop the continuation process when  $J^\lambda$  and  $C_{DGC}$  barely descend, which means that the value of  $\rho^k$  starts to get too high and the problem becomes too difficult.

---

**Algorithm 7:** Algorithm proposition for the inverse shell design with developability and parameter estimation.

---

**Data:** Target equilibrium pose  $x_t$ , small regularisation factor  $\lambda(\in [1, 5])$  Young modulus  $E$ , Poisson ratio  $\nu$ , thickness  $h$ , mass density increments  $\delta\rho$ , maximum mass value  $\rho^{max}$  potential small regularisation parameter

$\alpha_{smooth}$

**Result:** A pair  $(x, \bar{x})$  consisting of a stable equilibrium pose and a developable natural pose with  $x$  as close as possible to  $x_t$  and the corresponding volumic mass  $\rho$ .

```

// Warm-start of the continuation
1  $k \leftarrow 0$  ;
2  $\bar{x}_0 \leftarrow x_t$  ;
3  $\rho^0 \leftarrow \delta\rho$ ;
4 while  $\rho^k < \rho^{max}$  do
    // Regularised problem
5    $(x_k, \bar{x}_k) \leftarrow \text{BFGS\_min}(J^\lambda(\bar{x}) + \alpha_{DGC}C_{DGC}(\bar{x}) + \alpha_{smooth}C_{smooth}(\bar{x}), \bar{x}_{init} = \bar{x}_k)$ 
    with  $\alpha_{DGC}$  updated every  $n_{DGC} = 10$  iterations using Equation 6.7;
6   if the optimisation was succesfull then
7      $\rho^{k+1} = \rho^k + \delta\rho$ ;
8      $k \leftarrow k + 1$ ;
9   else
10    break;
11  end
12 end
    // Final problem
13  $\lambda \leftarrow 0$ ;
14  $(\rho, x_{best}, \bar{x}_{best}) \leftarrow$  the best solution from the continuation;
15 Switch to the plate-like model ? (Section 6.2.1);
16  $(x, \bar{x}) \leftarrow \text{ALM\_min}(J(\bar{x}) + \alpha_{smooth}C_{smooth}(\bar{x}) \text{ s.t. } C_{DGC}(\bar{x}) = 0, \bar{x}_{init} = \bar{x}_{best})$ ;
17 return  $(\rho, x, \bar{x})$ ;

```

---

### 6.3 Results

To test the algorithm described above, we use two examples: the **Hanging sheet** already introduced in Figure 6.2 and the **Hanging cylinder** that similarly consists in a horizontal cylinder fixed at both ends and hanging under gravity as depicted in Figure 6.8. We will however see in Section 6.3.1 that boundary conditions are actually problematic. The material parameters are given in Table 6.1.

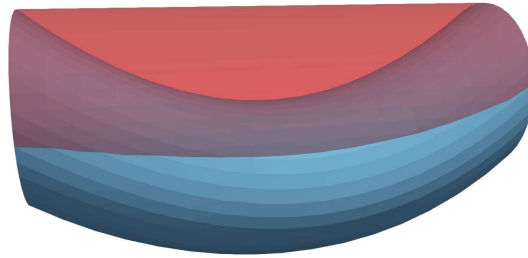


Figure 6.8: **Hanging cylinder** example generated with DISCRETE SHELL. The original rest shape is in red, and the deformed equilibrium shape in blue.

As this section is prospective, the results presented in this section are produced using different variants of the algorithm:

- the shell code: DISCRETE SHELL or LIBSHELL (MidEdgeAverage normals model), shortened respectively to DS and LIBSHELL;
- the coefficient of the DGC penalisation  $\alpha_{DGC}$  can either be fixed or adaptive using the Equation 6.7 (noted *adapt.*)
- the minimisation algorithm: BFGS, Gauss-Newton (GN) or a full Newton. The main differences between them are the same as observed in Section 5.4.1: BFGS has the most consistent behaviour, Newton is efficient, although very expensive and GN may be efficient provided that the gap  $\delta\rho$  between two problems is small enough.

These variants along with numerical parameters such as the convexification parameter  $\lambda$ , the mass gap  $\delta\rho$  or the additional regulariser  $\alpha_{smooth}$  (if present) will be specified at each result.

Regarding the minimisation criterion, the default behaviour is similar to that of Section 5.6.1 and to quit when the gradient or the displacement is small ( $\leq 1 \cdot 10^{-10}$ ) or when the iteration count is reached. However, unlike previously, the maximum number of iteration per minimisation is reduced as we solve subsequent problems that are assumed to be well warmstarted. It is often fixed between 50 and 200.



Example	$n_v$	$\sigma_{ref}$ ( $kg \cdot m^{-2}$ )	$k_L$ ( $N$ )	$k_A$ ( $N \cdot m^{-1}$ )	$k_B$ ( $N \cdot m$ )	$\rho_{ref}$ ( $kg \cdot m^{-3}$ )	$E$ ( $Pa$ )	$\nu$	$h$ ( $mm$ )
<b>Hanging sheet</b>	529	0.25	5.0	2.0	$1.0 \cdot 10^{-2}$	935	$1.5 \cdot 10^5$	0.4	2.0
<b>Hanging cylinder</b>	380	0.25	5.0	2.0	$1.0 \cdot 10^{-2}$	935	$1.5 \cdot 10^5$	0.4	2.0

Table 6.1: Material parameters for the two inversion examples for the two codes DISCRETE SHELL and LIBSHELL.

### 6.3.1 Penalisation strategy

In this section, we present the early tests using the developability penalisation described in Section 6.2.3 along with the weight continuation from Section 6.2.2 in order to identify the "best" mass value. For these two synthetic examples, the goal is to retrieve easily the original value of the surface/volumic density.

**Results with DISCRETE SHELL.** Let us begin with the results on DS, with a material more stretchy and resistant to bending than that of LIBSHELL.

For the **Hanging sheet**, visually, the result is quite similar to Figure 6.7d with the natural shape shrinking and unfolding as  $\sigma$  increases. However, the introduction of the DGC penalisation allows to retrieve a flat natural shape at  $\sigma = \sigma_{ref}$  (Figure 6.9c). The convergence plot in Figure 6.9e indeed shows that this flat shape is the original natural shape, satisfying exactly the physical and geometric objectives. Our procedure managed to identify the correct value of the mass.

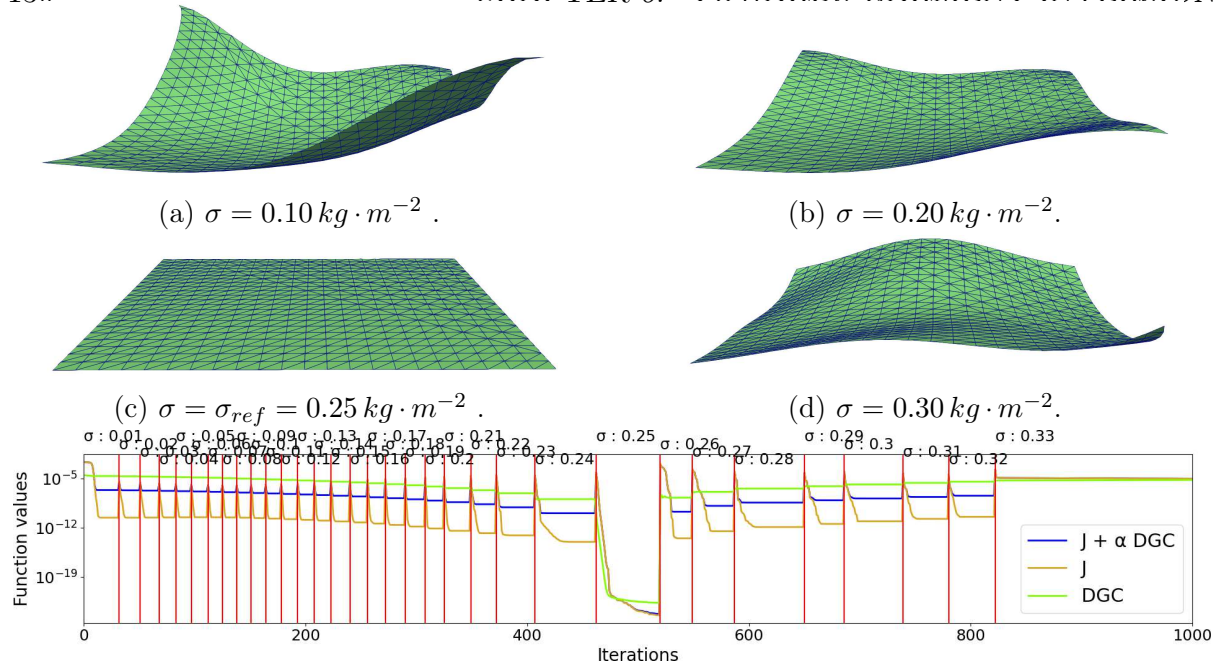
Note that this graph also illustrates the discussion on the "best" mass value of Section 6.2.4. Ideally, as it is depicted, both final objectives would start decrease as the mass grows until a point where the start to rise again, allowing the definition of a best (range of) values.

For the **Hanging cylinder**, we encounter a problem regarding the fixed points that has been discussed in Section 5.7.4. The positioning of such points is not regularised by their physics that is completely discarded by the objective of the first step  $J$  and as such, distorted meshes may arise.

Here, we observe that having the two extreme circles fixed causes convergence issues. With low values of the regularisers ( $\lambda, \alpha_{DGC}$ ) the evaluation of the implicit function would sometimes fail, stopping the whole process.

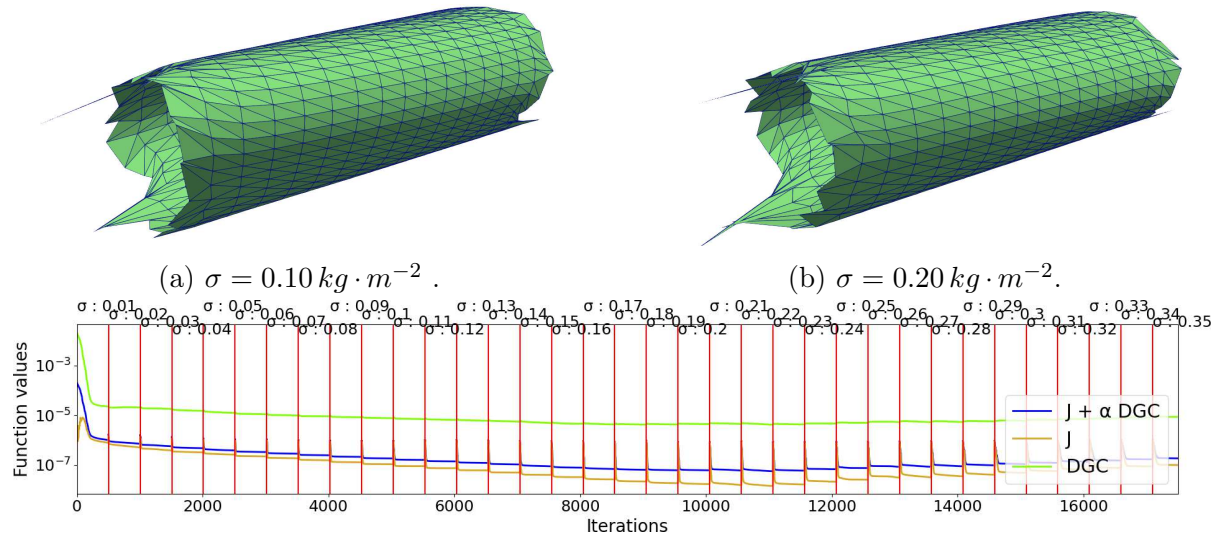
To be able to proceed safely with the continuation, we have to use relatively high values for the regularisations ( $\lambda = 50$  and  $\alpha_{DGC} = 10^{-2}$ ). Even with that, and using the Newton's method, the convergence rate is low and the graph shown in Figure 6.10c does not allow to recover the initial value  $\sigma_{ref} = 0.25 kg \cdot m^{-2}$ , but rather a best zone around  $\sigma_{ref} \approx 0.21 \sim 0.23 kg \cdot m^{-2}$ . Moreover, the shape produced have their boundary completely distorted (see Figures 6.10a) and 6.10b).

We highly suspect that the degenerated triangles trap the optimisation process and



(e) Convergence graph of the continuation process. Each vertical red line denotes a new value of  $\sigma$ . The inversion objective  $J$  is in orange, the developability DGC in green and the total objective  $J + \alpha_{DGC} DGC$  in blue.  $\sigma = 0.25 \text{ kg} \cdot \text{m}^{-2}$  appears clearly to be the best value.

Figure 6.9: Natural shapes and convergence graph of the inversion of the **Hanging sheet** with the mass continuation and fixed DGC penalisation. *Parameters:* DS,  $\alpha_{DGC} = 10^{-4}$ , GN,  $\lambda = 1$ ,  $\delta\sigma = 0.01 \text{ kg} \cdot \text{m}^{-2}$ .



(c) Convergence graph of the continuation process. Visually, the region  $\sigma_{ref} \approx 0.21 \sim 0.23 \text{ kg} \cdot \text{m}^{-2}$  seems to be the best solutions for this problem.

Figure 6.10: Inverting the **Hanging cylinder** with the mass continuation and fixed DGC penalisation. The natural shape correctly shrinks as the  $\sigma$  increases, but its boundaries are degenerated. *Parameters:* DS,  $\alpha_{DGC} = 10^{-2}$ , Newton,  $\lambda = 50$ ,  $\delta\sigma = 0.01 \text{ kg} \cdot \text{m}^{-2}$ .

are thus the cause of the very slow convergence rate.

The problems of the previous case all disappear if we change the boundary conditions of **Hanging cylinder** by fixing only 4 vertices, resulting in the case presented in Figure 6.11

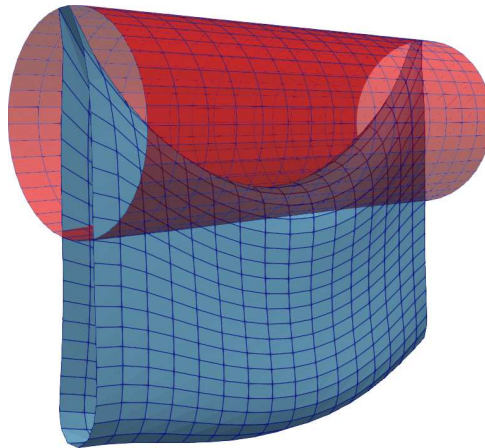


Figure 6.11: New **Hanging cylinder** example simulated with DISCRETE SHELL, with only 4 vertices fixed. The original rest shape is in red, and the deformed equilibrium shape in blue.

As for the **Hanging sheet**, our procedure applied to the new **Hanging cylinder** successfully recovers the initial value of the surface density. Numerically, the subsequent minimisations run smoothly, and we have a global minimum at  $\sigma = \sigma_{ref}$  on the graph of Figure 6.12e. Visually, the natural shape shrinks and deforms until it reaches at perfect cylinder again at  $\sigma = \sigma_{ref}$  (Figure 6.12c).

The problem of this boundary conditions is left aside and will be discussed again in Section 6.3.4. Starting now, we consider this case to be our new **Hanging cylinder** example.

**Results with LIBSHELL.** We reproduce the two previous cases with LIBSHELL. The deformed shape are quite different as the material is less stretchy and less resistant to bending. In order to reduced sharp edges on the **Hanging cylinder**, we separated the 4 pinned vertices but the deformed shape still exhibits some sharp creases.

However here, the inversion problem yields slightly less good results as we use an *absolute*  $\alpha_{DGC}$  of the same order of magnitude as in the previous code.

While the correct value  $\rho_{ref} = 935 \text{ kg} \cdot \text{m}^{-3}$  is successfully recovered for the **Hanging cylinder** thanks to the physical objective  $J$  and not the geometric terms as one can see in Figure 6.14, this is not the case for the **Hanging sheet**, whose results are in Figure 6.15. By looking at the convergence graph 6.15c, no value of  $\rho$  appears to emerge as better as the others. If we look at the geometry, it seems that the subsequent solutions are stuck into local minima that contained creases along the diagonals of the sheet.

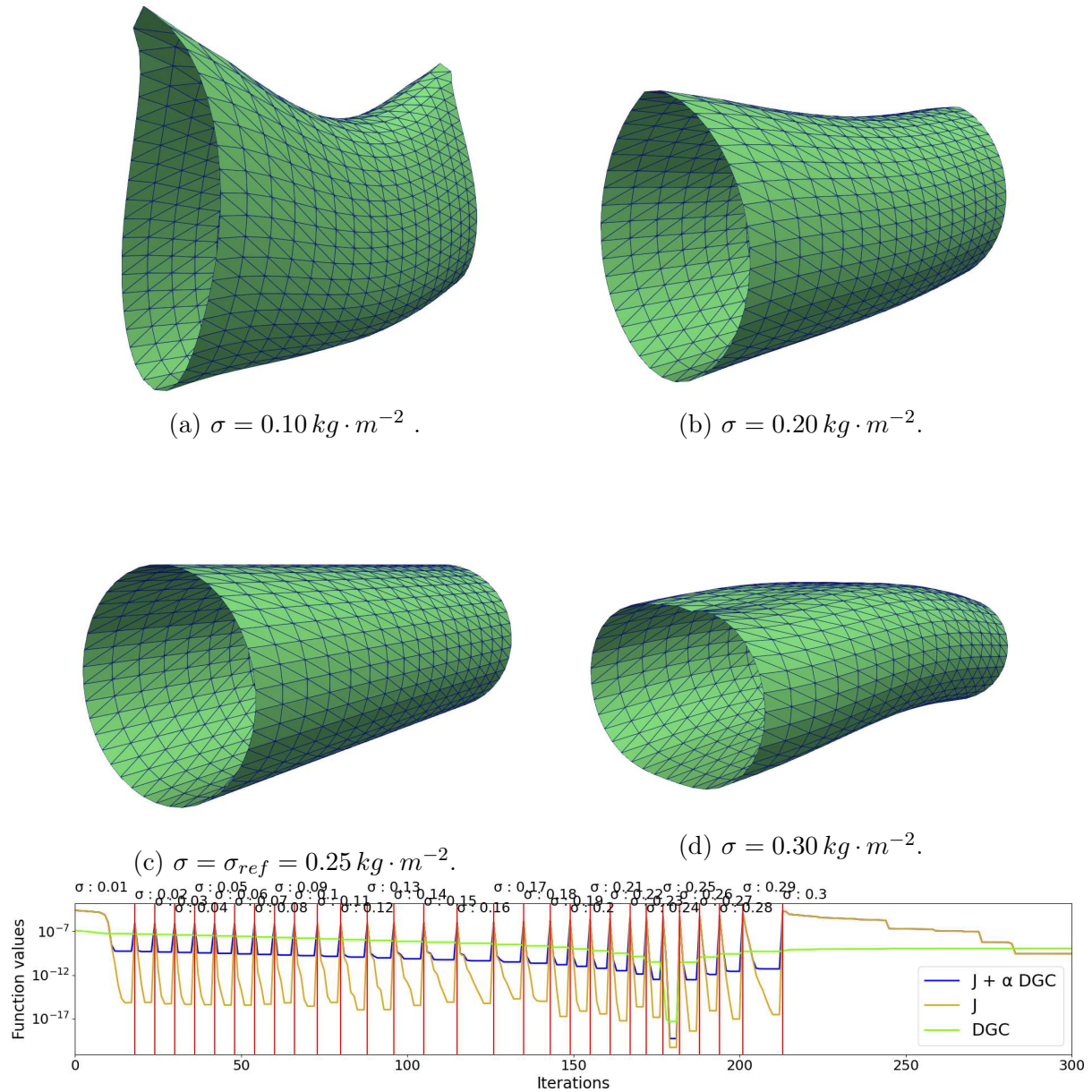


Figure 6.12: Natural shapes and convergence graph of the inversion of the new **Hanging cylinder** with the mass continuation and fixed DGC penalisation. *Parameters:* DS,  $\alpha_{DGC} = 10^{-8}$ , GN,  $\lambda = 10$ ,  $\delta\sigma = 0.01 \text{ kg} \cdot \text{m}^{-2}$ .

Note that a small smoothness regulariser is also present on these examples, but proves insufficient to get the **Hanging sheet** to flatten.

To briefly conclude this subsection, we saw that the penalisation strategy may prove successful to recover the mass value. Yet, the last results show us the sensitivity of the

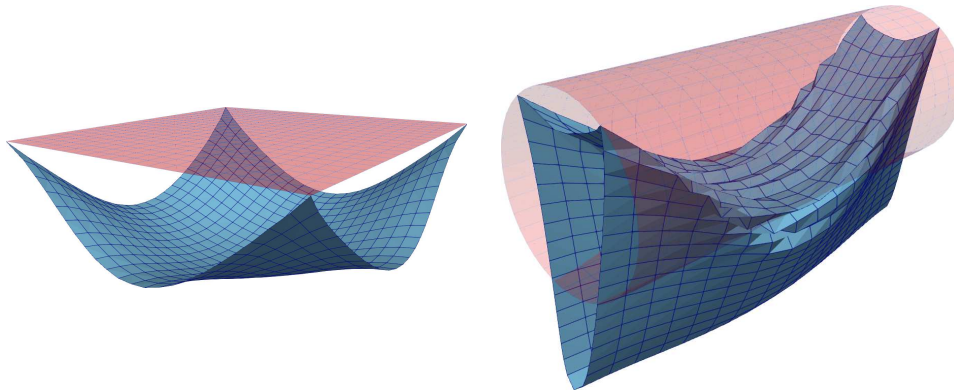


Figure 6.13: Examples generated with LIBSHELL. The original rest shape is in red, and the deformed equilibrium shape in blue.

problem *w.r.t.* the different coefficients and the need for a rule to adjust them. The results using the adaptive coefficient rule 6.7 are presented in the next section.

### 6.3.2 Adaptive coefficient

In this section, we show the results of the adaptive rule for  $\alpha_{DGC}$  described in Section 6.2.3 in order to obtain proper results for the LIBSHELL code.

With the new heuristic, the **Hanging sheet** now works fine, allowing to identify the  $\rho_{ref} = 935 \text{ kg} \cdot \text{m}^{-3}$  as the best value on the convergence graph in Figure 6.16c Regarding the geometry, we obtain a shape that is almost isometric to the flat square, with a small difference induced by the natural curvature.

Note that the coefficients produced by our update rule are of order of magnitude  $\alpha_{DGC} \approx 0.1 \sim 1$ , which is much higher than that the coefficients used for *DiscreteShell* and thus may explain the failure observed in Figure 6.15.

For the **Hanging cylinder**, the adaptive coefficient strategy also proves successful to identify the good mass value (Figure 6.17c) although the resulting natural shape is crumpled (Figure 6.17a). To counterbalance this crumpling caused by the low dependence on the natural curvature due to the low bending coefficient, it may be tempting to add the smoothness regulariser as done for Figure 6.14. Even though it actually seems to slightly unfold the natural shape, it remains crumpled and moreover this additional term seems to affect the convergence, and the best value is less identifiable.

Ergo, our adaptive coefficient for the developability seems relatively efficient to allow the identification of the mass value on these two synthetic cases. However, further work to tune the smoothness regulariser, or in a subsequent final step is needed if we wish to enforce a smooth natural shape.

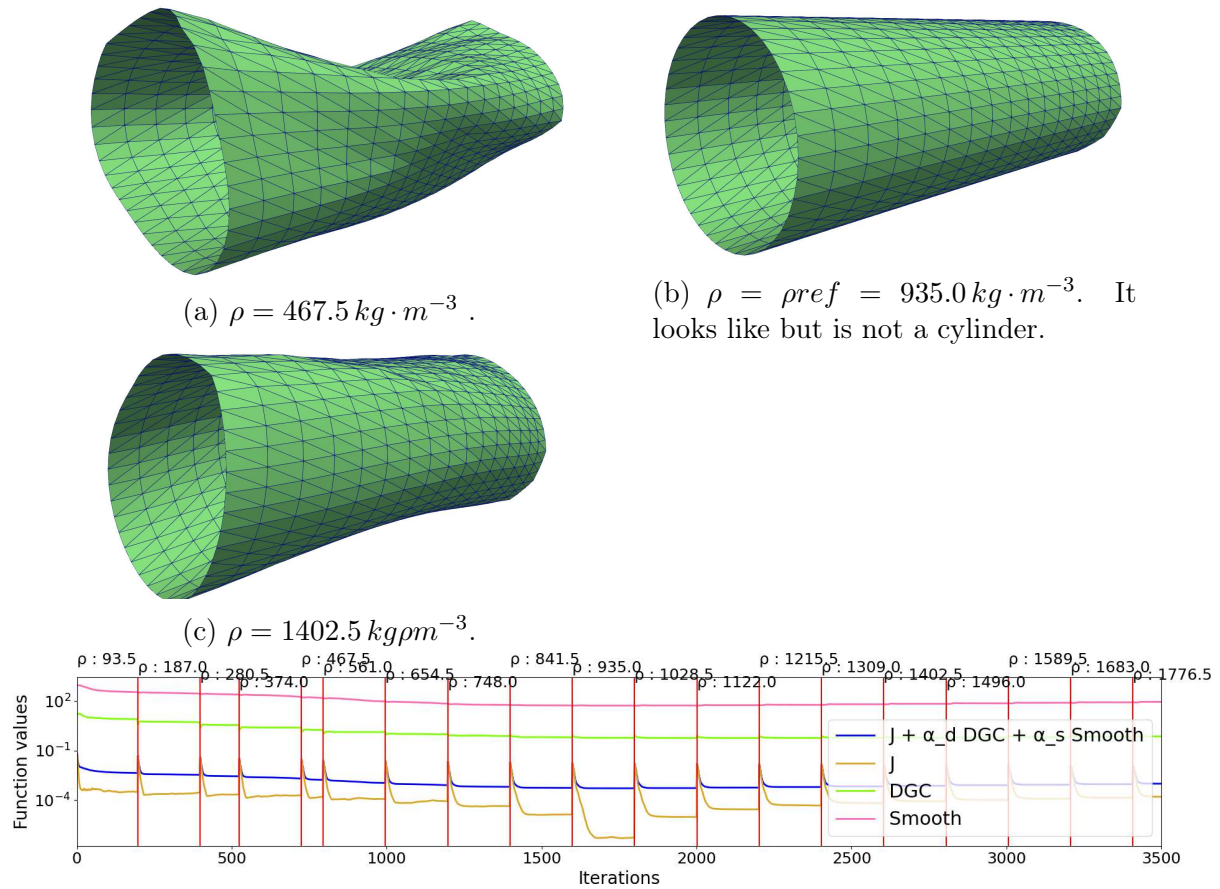


Figure 6.14: Inverting the **Hanging cylinder** with the mass continuation and fixed DGC penalisation. *Parameters:* LIBSHELL,  $\alpha_{DGC} = 2 \cdot 10^{-4}$ ,  $\alpha_{smooth} = 1 \cdot 10^{-5}$ , BFGS,  $\lambda = 1$ ,  $\delta\rho = 93.5 \text{ kg} \cdot \text{m}^{-3}$ .

### 6.3.3 Remeshing and hierarchical strategy

Since Section 5.7.1 and Table 5.2, we have seen that the computation time is a huge limitation of our algorithm. To alleviate the cost of treating highly refined meshes, we would like to use the results of problems using coarser meshes in order to accelerate the computation. More precisely, in our continuation strategy, we can first estimate roughly the mass value and then refine its value as we solve with increasing meshes resolution.

In order to devise such a strategy, we first observe the behaviour of the simulator as we refine the meshes. To do so, we compute several equilibria for the **Hanging sheet** using the parameters of Table 6.1 with different meshes resolutions for the natural shape: 529, 1013, 3921 and 7921 vertices. The resulting shapes are depicted in Figure 6.18. We can observe that low-resolution meshes exhibit a stiffer material behaviour than expected<sup>1</sup>. Ergo, solving the continuation problem at low resolution should yield an upper bound on

<sup>1</sup>This phenomenon is sometimes misnamed *locking* in the litterature.

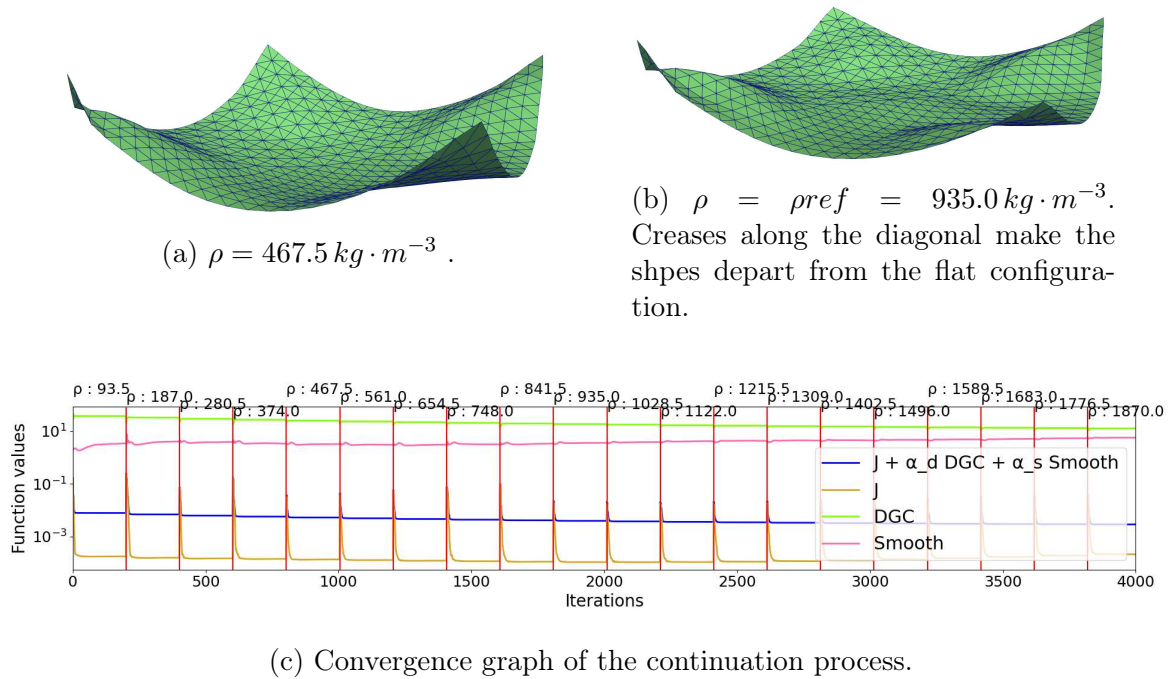


Figure 6.15: Natural shapes and convergence graph of the inversion of the new **Hanging sheet** with the mass continuation and fixed DGC penalisation. *Parameters:* LIBSHELL,  $\alpha_{DGC} = 2 \cdot 10^{-4}$ ,  $\alpha_{smooth} = 1 \cdot 10^{-5}$ , BFGS,  $\lambda = 1$ ,  $\delta\rho = 93.5 \text{ kg} \cdot \text{m}^{-3}$ .

the mass value since a stiffer material should be able to withstand more weight.

To test this "hierarchical" strategy, we build again upon the **Hanging sheet**. Starting from the equilibrium shape of 7921 vertices that we wish to invert while recovering the mass value, we remesh it to lower resolutions using [the Mmg remeshing software](#) (Dobrzynski and Frey, 2008; Dapogny et al., 2014). We generate several meshes of different resolutions: 749, 1761, 3055 and 7921 (depicted in Figure 6.19) that we are going to subsequently invert. To use the invert shape of a coarser problem to warmstart the next one, we simply compute a linear mapping between the corresponding target shapes that upsamples the coarser mesh to the finer one and report this mapping on the computed natural shape.

Let us begin by inverting the coarser version (749 vertices) using our algorithm with the adaptive developability penalisation. Surprisingly and contrary to what we expected, the best value in the sense "the most developable with a reasonably good value for  $J$ " that we can read from Figure 6.20 is  $\rho = 748 \text{ kg} \cdot \text{m}^{-3}$ , that is below the expected final value  $\rho_{ref} = 935 \text{ kg} \cdot \text{m}^{-3}$ . For  $\rho > \rho_{ref}$ , both values of  $J$  and  $C_{DGC}$  are larger.

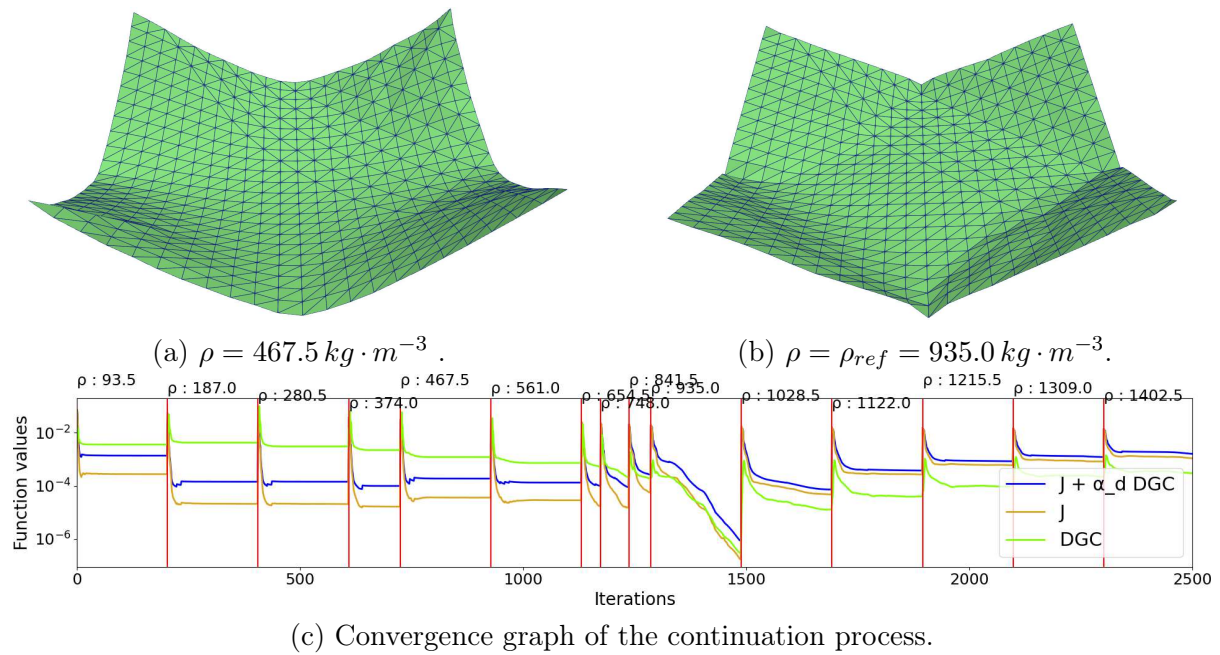


Figure 6.16: Natural shapes and convergence graph of the inversion of the **Hanging sheet** with the mass continuation and adaptive DGC penalisation. The adaptive coefficient allows the identification of the best volumic mass value. *Parameters:* LIBSHELL,  $\alpha_{DGC} = adapt.$ , BFGS,  $\lambda = 1$ ,  $\delta\rho = 93.5 \text{ kg} \cdot \text{m}^{-3}$ .

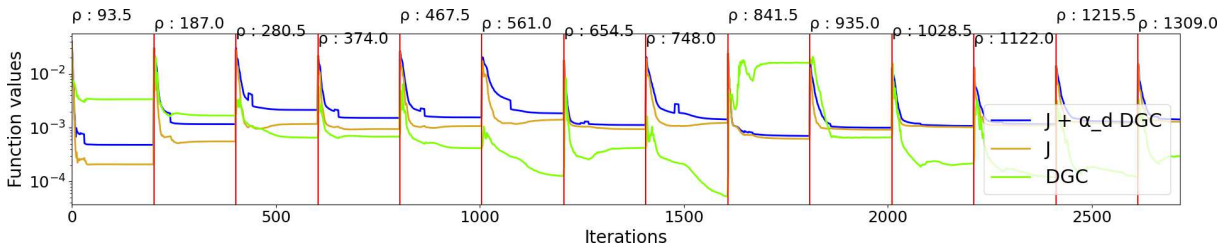
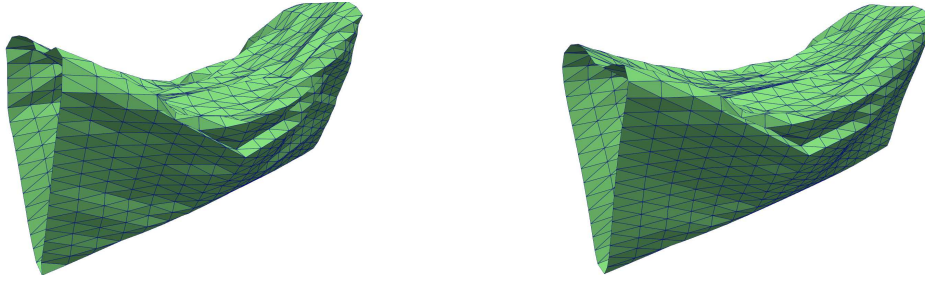


Figure 6.20: Convergence graph of the inversion of the remeshed version at 749 vertices of the high-resolution **Hanging cylinder**. Surprisingly, the best value underestimates the correct value  $\rho_{ref} = 935 \text{ kg} \cdot \text{m}^{-3}$ . *Parameters:* LIBSHELL,  $\alpha_{DGC} = adapt.$ , BFGS,  $\lambda = 4$ ,  $\delta\rho = 93.5 \text{ kg} \cdot \text{m}^{-3}$ .

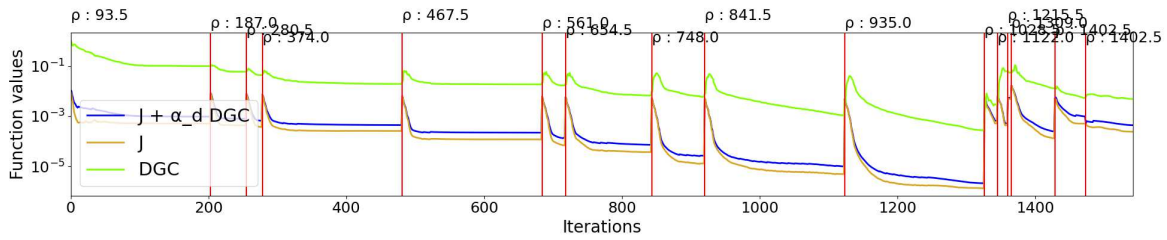
Nonetheless, if we stick to the original plan, the value of  $\rho_{best}$  found is an overestimation and so we use the natural shape computed for  $\rho = 654.5 \text{ kg} \cdot \text{m}^{-3}$  to warmstart the next problem at 1761 vertices. On Figure 6.21, we can see that a "best zone" arises around the expected value  $\rho \in [841.5, 1028.5]$  although it is again the lowest value that solves best the problem.



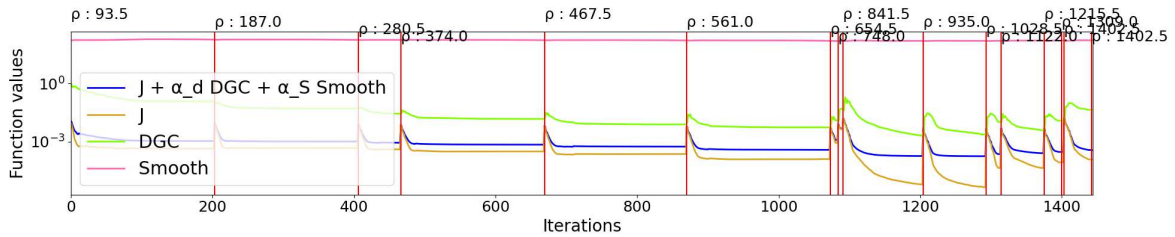


(a)  $\rho = \rho_{ref} = 935.0 \text{ kg} \cdot \text{m}^{-3}$ , without the smoothness regulariser.

(b)  $\rho = \rho_{ref} = 935.0 \text{ kg} \cdot \text{m}^{-3}$ , with the smoothness regulariser  $\alpha_{smooth} = 10^{-6}$ .



(c) Convergence graph of the continuation process without the smoothness regulariser.



(d) Convergence graph of the continuation process without the smoothness regulariser with the smoothness regulariser  $\alpha_{smooth} = 10^{-6}$ .

Figure 6.17: Inverting the **Hanging cylinder** with the mass continuation and adaptive DGC penalisation. The correct mass value can be identified with or without the smoothness regulariser, although further tuning is required to retrieve the initial cylinder. *Parameters:* LIBSHELL,  $\alpha_{DGC} = adapt.$ , BFGS,  $\lambda = 8$ ,  $\delta\rho = 46.75 \text{ kg} \cdot \text{m}^{-3}$ .

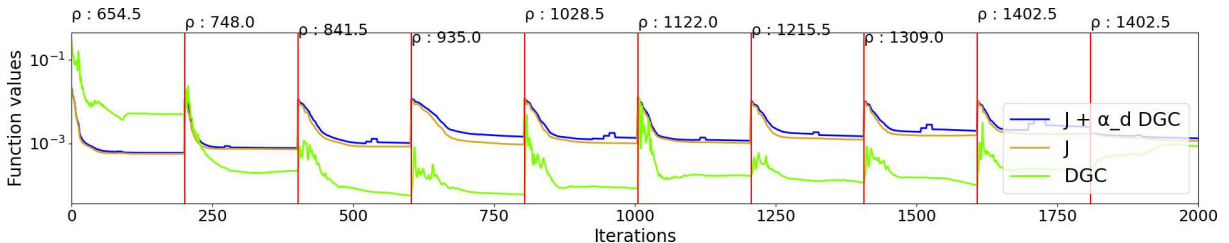


Figure 6.21: Convergence graph of the inversion of the remeshed version at 1761 vertices of the high-resolution **Hanging cylinder** warmstarted with a solution of the problem at 749 vertices. *Parameters:* LIBSHELL,  $\alpha_{DGC} = adapt.$ , BFGS,  $\lambda = 4$ ,  $\delta\rho = 93.5 \text{ kg} \cdot \text{m}^{-3}$ .

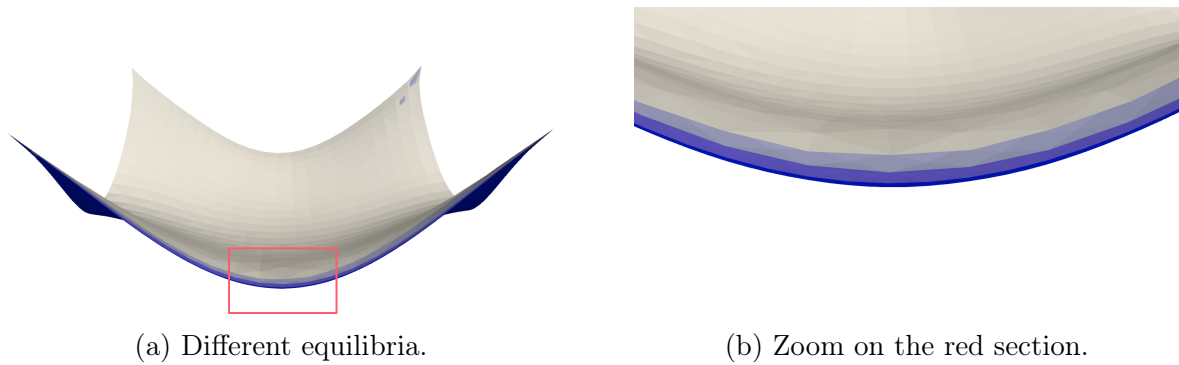


Figure 6.18: Equilibria of the **Hanging cylinder** with different mesh resolutions (529, 1013, 3921 and 7921 vertices) computed with LIBSHELL. The darker the colour is, the finer the mesh is. Low resolution meshes exhibit a stiffer material behaviour.

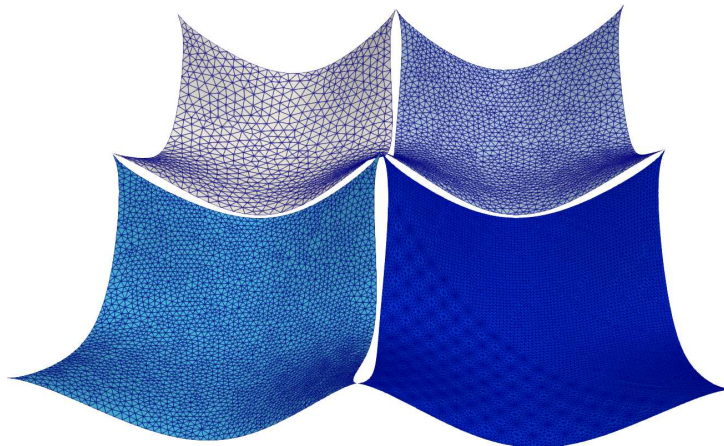


Figure 6.19: At the bottom right, in the darkest blue, equilibrium of the **Hanging cylinder** (7921 vertices). In other shades of blue, remeshed versions using Mmg, a darker colour indicating a higher mesh resolution (749, 1761 and 3055 vertices).

We pursue again the same strategy, taking the inferior value  $\rho = 748 \text{ kg} \cdot \text{m}^{-3}$  to warmstart the next problem at 3055 vertices. In the convergence graph displayed in Figure 6.22, we find again a best zone centred around the reference value albeit no specific value seems to emerge.

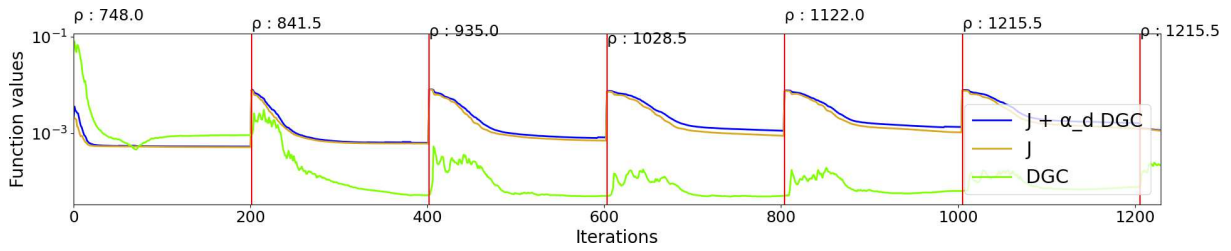


Figure 6.22: Convergence graph of the inversion of the remeshed version at 3055 vertices of the high-resolution **Hanging cylinder** warmstarted with a solution of the problem at 1761 vertices. *Parameters:* LIBSHELL,  $\alpha_{DGC} = \text{adapt.}$ , BFGS,  $\lambda = 4$ ,  $\delta\rho = 93.5 \text{ kg} \cdot \text{m}^{-3}$ .

Finally, we apply the same method to solve the initial high-resolution problem. On our computers, minimising with the BFGS algorithm was impossible as we were unable to allocate the inverse hessian approximation. To lower the memory cost, we use the L-BFGS method with a memory of size 15.

Once again, in the convergence graph of Figure 6.23 there is a best zone that includes the reference value  $\rho_{ref}$  but no specific value seems to emerge. We suspect that the whole process, combining the resolution of approximate problems and a geometrical upsampling, drives the natural shape towards a local minimum that is relatively close a the global minimum albeit unattainable as the dimensionality of the problem grows when the meshes are refined.

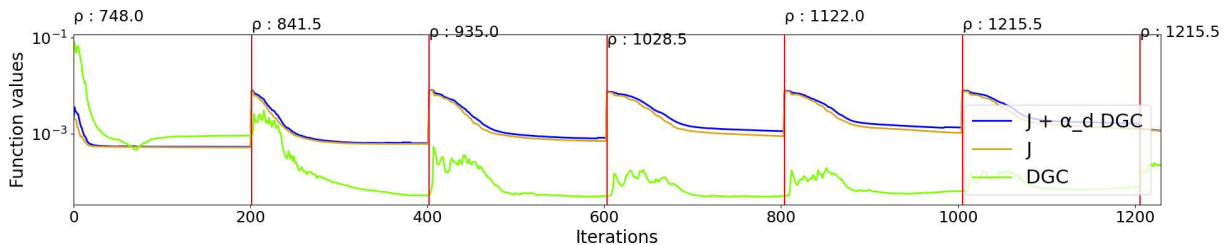


Figure 6.23: Convergence graph of the inversion of the high-resolution **Hanging cylinder** warmstarted with a solution of the problem at 3055 vertices. We obtain a range of best parameters, but no specific value. *Parameters:* LIBSHELL,  $\alpha_{DGC} = \text{adapt.}$ , L-BFGS-15,  $\lambda = 4$ ,  $\delta\rho = 46.75 \text{ kg} \cdot \text{m}^{-3}$ .

To conclude this experiment on a hierarchical solve, we can say that this is an interesting avenue although much work remains to make it efficient. There, the second and third steps did very little to refine the value of  $\rho$  and could have been skipped. Moreover, one could think of a better upsampling method that does not only rely on geometrical considerations. And finally, the question on why low-resolution solutions underestimate instead of overestimating the final value remain to assess.

### 6.3.4 Friction penalisation

In this section, we present a very naive way to deal with frictional contact within this continuation framework. As discussed in Sections 5.7.2 and 6.3.1, fixed regions are totally ignored by the first step objective  $J$  which may result in distorted meshes. As such regions often stems from contact with an external object, not decoupling the contact and the free vertices would prevent the degeneracy of the meshes in that kind of cases.

Very naively here, we add to the optimisation problem the "Coulomb error" function of the second step defined in Section 5.5.2 that we restrict to the forces at the contact points  $G_{\text{contact}}(\Phi(\bar{x}), \bar{x})$ , which yield the problem:

$$\min_{\bar{x}} J(\bar{x}) + \alpha_{DGC} C_{DGC}(\bar{x}) + \alpha_{\mu} G_{\text{contact}}(\Phi(\bar{x}), \bar{x}).. \quad (6.8)$$

$\alpha_{DGC}$  is adaptive *w.r.t.*  $J$  and  $DGC$  only, and  $\alpha_{\mu}$  is set to a fixed value.

We test the continuation procedure with this modified objective on the following **Beams** examples, where a square sheet composed of 1089 vertices is dropped onto two cylinders. We simulate it using LIBSHELL using the same material parameters as Table 6.1 and the friction coefficient  $\mu = 0.6$ . The resulting equilibrium is depicted in Figure 6.24

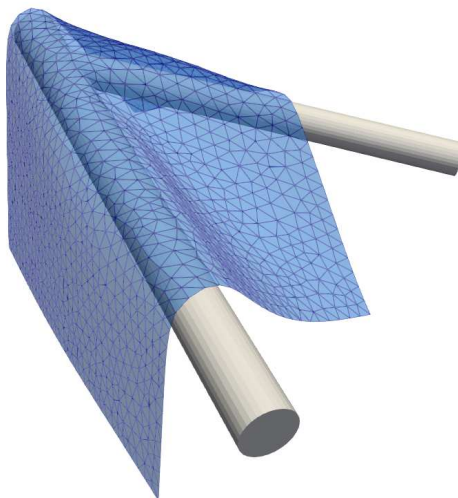
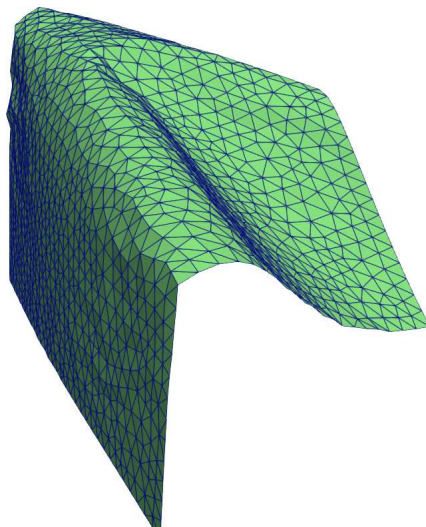
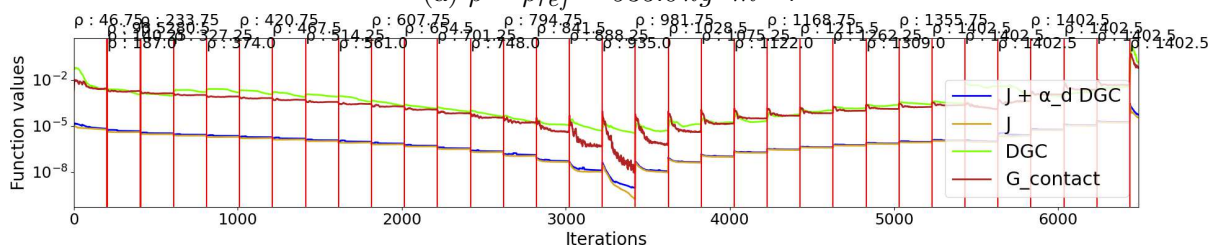


Figure 6.24: **Beams** example generated with LIBSHELL.

The results of the inversion process are in Figure 6.25c. Numerically, the minimisation runs smoothly and all the terms agree to correctly designate  $\rho = 935.0 \text{ kg} \cdot \text{m}^{-3} = \rho_{ref}$  as the best value. Visually however, we have a similar problem as previously that is that the shape, although developable, does not unfold.

Moreover, as we said in Section 5.7.2, combining both steps with  $J$  that is totally oblivious to the physics of the contacting points yield a slow convergence rate and expensive evaluations due to the implicit function.

(a)  $\rho = \rho_{ref} = 935.0 \text{ kg} \cdot \text{m}^{-3}$ .

(b) Convergence graph of the continuation process.

(c) Inverting the **Beams** with the mass continuation, adaptive DGC penalisation and the Coulomb error. The correct mass value can be identified with or without the smoothness regulariser, although further tuning is required to retrieve the initial cylinder. *Parameters:* LIBSHELL,  $\alpha_{DGC} = \text{adapt.}$ ,  $\alpha_{\mu} = 10^{-1}$ , BFGS,  $\lambda = 4$ ,  $\delta\rho = 46.75 \text{ kg} \cdot \text{m}^{-3}$ .

## 6.4 Conclusion

To conclude this section, we can say that we have explored an interesting avenue to introduce the developability and to identify a material parameter within the inverse problem.

Yet, there is much work left to obtain a robust algorithm. The difficulty of this problem comes from the fact that we have several possibly conflicting objectives:

- The term  $J$  that relies on the implicit function only deals with the contact free zones, for which it seeks the closest natural shape that allows the associated equilibrium to be *as close as possible* to the target. It is totally oblivious to the physics of the contact points and may disrupt them;
- The developability term  $C_{DGC}$  that we aim at treating as a *constraint* for fabrication purposes. However, it may conflict with the physics. In addition, the current formulation assumes a single developable patch;
- The Coulomb error  $G_{\text{contact}}$  that also needs to be treated as a constraint;

- And finally geometrical regulariser such as  $C_{smooth}$  that may be used and should be strong enough to have an impact without interfering with the other terms.

Moreover, in non-synthetic examples *i.e.* when we are not sure of a solution satisfying a global minimum (*e.g.* the remeshed deformed shapes), the current technique yield a Pareto optimum. A more systematic way to determine the best value of the continuation process would be required to treat more complex examples.

Finally, related to the conflicting objective, the convergence rate and the subsequent computation time are problematic to deal with more detailed examples. As one may have noticed in the convergence graphs, the objective often descends rather quickly at the beginning of a  $\rho$  iteration, then makes a plateau. The convergence rate there is slow, but not null, and we have found out that cutting the minimisation too early may perturb the continuation process and prevent the identification of the correct value. As such, a minimisation scheme that would speed up the convergence rate would be of great interest in our approach.

# General conclusion

## Summary of the contributions

Although the focus of my thesis has been on the topic of inverse garment design, I had the opportunity to work on and contribute to related projects on the topic of shells and frictional contact. I summarise below my contributions to the whole of these research projects.

To begin with, I presented our addition to the *Projective Dynamics* algorithm (Bouaziz et al., 2014) in order to embed frictional contact in that framework (Ly et al., 2020). Despite the success of that method to produce fast and stable simulations of deformable objects, one could regret that dry friction was not included and only dealt with in a post-processing step. To address this limitation, I have followed an idea similar to that of the original method, which consists in the design of a simple step where forces can be estimated and further refined during an iterative process. That idea resulted in an additional contact step where the equation of motion is approximated in a splitting-scheme like manner so as to build an explicit relation between the velocities and the contact forces, both submitted to the Signorini-Coulomb law. Thanks its the explicit form, the constraint can be easily integrated into the dynamics. The resulting modified algorithm is able to reproduce qualitatively (and rather quantitatively according to the results presented in Chapter 3) the threshold effect of the dry friction.

Convinced that physical simulators developed Computer Graphics have the potential to address problems with rather strong requirements in terms of physical accuracy, our research group introduced several protocols inspired by some measurement protocols of the Soft Matter Physics community in order to evaluate the simulators (Romero et al., 2021). By providing dimensionless laws to be compared with, our tests are designed to test directly the correctness of the geometry produced by the numerical simulator, regardless of the scale of the objects. Within this project, I had the opportunity to contribute mainly to early tests for the **Cantilever** protocol and to the design of the **Lateral Buckling**, to test some codes on these two protocols and analyse their results. Our data have shown that the methods developed in the Computer Graphics community have nothing to be ashamed of in comparison to codes from Mechanical Engineering in terms of physical accuracy.

Finally, for the shell inverse design, I proposed to add a correction step to the algorithm initially derived by Casati (2015). This new step takes the form of an optimisation problem that aims at bringing the forces at the contact points into the Coulomb cones in order to enforce a static equilibrium with dry friction forces (Ly et al., 2018). We showed that the resulting algorithm can robustly produce natural shapes that solve the problem on a reasonable parameter range. We then further explored avenues to modify this algorithm in order to apply our method to real garments aiming at, in the long term, recovering their patterns. To do so, we proposed on the one hand a continuation method that helps explore the material parameters space, alleviating the need for an exact knowledge of the material parameter, and on the other hand to introduce a discrete developability constraint so as to try to enforce the recovery of natural shapes that can relatively easily be flattened. In the prospective work presented, both of these ideas lead to promising results albeit further work is naturally required.

## Perspectives

For our *Projective Friction* algorithm, we strongly believe that the simplicity of our addition to an already popular method can raise some interest. Regarding the algorithm itself, an exciting direction of research would be to increase the convergence speed of our estimation procedure. Indeed, extensive work has been proposed to enhance the convergence speed of *Projective Dynamics* (Wang, 2015; Macklin et al., 2020), so it would be interesting to see if similar treatments can be applied to our modified framework to allow the friction forces to converge accurately and quickly.

With the work of our group on the physical validation of numerical simulators, we hope that it will encourage the Computer Graphics community to assess more systematically the accuracy of the simulators and also to develop similar simple yet rich dimensionless laws to compare with. Moreover, the range of behaviours covered by our protocols are quite narrow (static slender structures clamped under gravity) and do not reflect the richness of the situations that happen in the Computer Graphics applications. Ergo, designing tests that are able to evaluate more complex situations is an interesting avenue albeit more difficult as deriving theoretical results becomes harder.

Lastly, the static shell inversion algorithm with contact, developability and mass continuation has exhibited promising results. However, it is in the need of a proper numerical framework that would allow to consistently reach the objective of a material parameter and a developable natural shape that best solve the inverse design problem under Coulomb constraints, with the shape as smooth or unfolded as possible. The computation time is also an issue and prevents the treatment of highly refined meshes which are needed for complex shapes in order to accurately model the folds for instance, or to ensure the convergence of the physical model. A classical avenue to solve this issue lies in the search of numerical methods, such as preconditioning or other optimisation schemes in order to



improve the convergence rate. Another possibility would be to look into the direction of differentiable physics such as in (Liang et al., 2019). Such methods may allow to build a new physical shell inversion process, built on dynamic sequences instead of static poses, that would offer new insights and could be lighter or better conditioned.



# Appendix A

## Notions of linear elasticity

In this appendix, we propose a short practical introduction to the 3D linear elasticity to give requirements for the discussions and the derivations from the main body.

Consider an object described by a subset of  $\Omega \subset \mathbb{R}^3$ . We are interested in describing the displacement of this object defined by the function  $f : \Omega \rightarrow \mathbb{R}^3$  submitted to external forces and considered to be elastic (without any external forces, at rest,  $u$  is a rigid transformation).

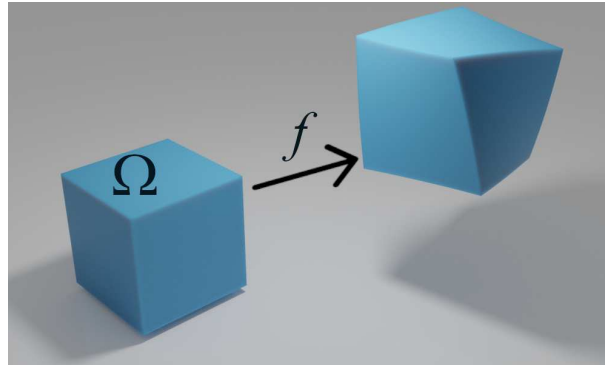


Figure A.1: Displacement field.

**General relations.** The elastic law aims at linking the displacement field  $u$ , the *strain*  $\epsilon : \Omega \rightarrow \mathbb{R}^{3 \times 3}$  i.e. the deformation and the *stress*  $\sigma : \Omega \rightarrow \mathbb{R}^{3 \times 3}$  i.e. the force density within the body.

With the hypothesis of the *small deformation* assumption, the strain-displacement equation is given by the Green strain:

$$\epsilon = \frac{1}{2} (\nabla f + (\nabla f)^\top + (\nabla f)^\top \nabla f). \quad (\text{A.1})$$

Another possible strain-displacement relation is the Cauchy strain where the quadratic part is removed

$$\epsilon_{cauchy} = \frac{1}{2} (\nabla f + (\nabla f)^\top) \quad (\text{A.2})$$

However, this strain formulation is not suitable for *large displacements* (rotations) where it exhibits artefacts.

The stress and the strain are linked by a linear relation, the *Hooke's law*:

$$\sigma = \mathbf{C} : \epsilon \quad (\text{A.3})$$

with  $\mathbf{C} \in \mathbb{R}^{3 \times 3 \times 3 \times 3}$  the fourth-order stiffness tensor. Detailed with the components, the previous relation can be written using the Einstein summation convention:

$$\sigma_{ij} = \mathbf{C}_{ijkl} \epsilon_{kl}. \quad (\text{A.4})$$

For symmetry reasons, there are actually 6 independent coefficients in  $\epsilon$  and  $\sigma$ , and 21 distinct coefficients in  $\mathbf{C}$ . This leads  $\sigma$  and  $\epsilon$  to be in  $\mathbb{R}^6$  and  $\mathbf{C} \in \mathcal{S}_6$ . In 2D, a similar mapping can be made to have  $\sigma, \epsilon \in \mathbb{R}^3$  and  $\mathbf{C} \in \mathcal{S}_3$ .

Further simplifications can be made if the material has symmetries. We give below the formulas for the case of an orthotropic material and of an isotropic material.

**Orthotropic material.** A material is orthotropic when it has 3 orthogonal planes of symmetry (2 in 2D). The behaviour can then be described by Young modulus  $E_i$ , Poisson ratio  $\nu_{ij}$  and shear modulus  $G_i$  in the different directions.

Assuming that these planes are aligned with the directions 1, 2, 3 (*i.e.* the symmetry planes are 1 – 2, 1 – 3 and 2 – 3), the strain-stress relation is given in 3D by

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{bmatrix} = \begin{bmatrix} \frac{E_1(1-\nu_{23}\nu_{32})}{\delta} & \frac{E_2(\nu_{12}-\nu_{13}\nu_{32})}{\delta} & \frac{E_3(\nu_{13}-\nu_{12}\nu_{23})}{\delta} & 0 & 0 & 0 \\ & \frac{E_2(1-\nu_{31}\nu_{13})}{\delta} & \frac{E_2(\nu_{23}-\nu_{13}\nu_{21})}{\delta} & 0 & 0 & 0 \\ & & \frac{E_3(1-\nu_{12}\nu_{21})}{\delta} & 0 & 0 & 0 \\ & & & 2G_{12} & 0 & 0 \\ & (sym) & & & 2G_{13} & 0 \\ & & & & & 2G_{23} \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{23} \end{bmatrix} \quad (\text{A.5})$$

with  $\delta = 1 - \nu_{12}\nu_{21} - \nu_{13}\nu_{31} - \nu_{23}\nu_{32} - 2\nu_{12}\nu_{23}\nu_{31}$ , and in 2D

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \frac{E_1}{1-\nu_{12}\nu_{21}} & \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & 0 \\ & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ & & 2G \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{12} \end{bmatrix}. \quad (\text{A.6})$$

Note the 0 that appeared in  $\mathbf{C}$  due to the symmetries.

**Isotropic material.** In an isotropic material, the behaviour is the same regardless of the direction. That is, every plane is a plane of symmetry. This reduces the model to be parametrized by 2 coefficients. In practice, for instance, we can take the orthotropic case with the same  $E$  and  $\nu$  in both directions.

After a short computation, the 2D and the 3D case can be reduced in a single equation of the form

$$\sigma = 2G\epsilon + \lambda \operatorname{tr}(\epsilon)\mathbf{I} \quad (\text{A.7})$$

with  $\lambda$  the first coefficient of Lamé,  $\mu$  the shearing modulus, also called second coefficient of Lamé, and  $\mathbf{I}$  the identity. Note that  $\lambda$  do not have the same expression in 2D and in 3D, and that in the isotropic case,  $G$  is linked to  $E$  and  $\nu$ .

$$\lambda_{3D} = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \lambda_{2D} = \frac{E\nu}{1-\nu^2} \quad G = \frac{E}{2(1+\nu)}. \quad (\text{A.8})$$



# Appendix B

## Fundamental forms for the mechanics of shells

In this appendix, we briefly introduce the first and the second fundamental forms of surfaces in the continuous and discrete settings.

### B.1 Continuous setting

In this section, we follow closely the clear exposition from (Casati, 2015) of (do Carmo, 1976)

Let  $\Omega \subset \mathbb{R}^2$  be the parametrization domain of a surface  $r : \Omega \rightarrow \mathbb{R}^3$ . We assume that  $r$  is  $\mathcal{C}^2$ , and that  $\forall s \in \Omega$ ,  $Dr(s)$  is full-rank, that is there is always a tangent plane.

Consider  $s \in \Omega$  and  $ds \in \mathbb{R}^2$  such that  $s + ds \in \Omega$ . The distance variation on the surface induced by this small displacement in the parametrization is given by

$$\|r(s + ds) - r(s)\|^2 = \|Dr(s)ds + o(ds)\|^2 = \langle ds, Dr(s)^\top Dr(s)ds \rangle + o(\|ds\|^2). \quad (\text{B.1})$$

That is, the norm induced on the surface is the norm defined by the matrix  $Dr^\top Dr$ . This matrix is symmetric definite positive due to our assumption on the existence of the tangent plane, and so we can define the following application - the *first fundamental form* :

$$\begin{aligned} \mathbf{A} : \Omega &\rightarrow \mathcal{S}_2^{++} \\ s &\mapsto Dr(s)^\top Dr(s). \end{aligned} \quad (\text{B.2})$$

This application also links the surface element of the surface to the surface element of the parametrization domain

$$dS(s) = \|\partial_1 r(s) \times \partial_2 r(s)\| ds = \sqrt{\det \mathbf{A}(s)} ds \quad (\text{B.3})$$

which explains the term in  $\sqrt{\det \mathbf{A}}$  that we can find in the Koiter's energy.

The previous development gave the *in-plane* surface variation on the tangent space. Similarly, we can repeat the process to observe the out-of-plane change, that is in the normal direction. The normal is defined by the application

$$\begin{aligned} n : \Omega &\rightarrow \mathcal{S}(0,1) \\ s &\mapsto \frac{\partial_1 r(s) \times \partial_2 r(s)}{\|\partial_1 r(s) \times \partial_2 r(s)\|}. \end{aligned} \quad (\text{B.4})$$

To compute the normal variation, we have to push the expansion of  $r$  to the second order, and so it yields

$$\begin{aligned} n(s)^\top (r(s+ds) - r(s)) &= \underbrace{n(s)^\top Dr(s)}_{=0} ds + \frac{1}{2} n(s)^\top D^2 r(s)(ds, ds) + o(\|ds\|^2) \\ &= \frac{1}{2} ds^\top \left[ \sum_{k=1}^3 n_k(s) \partial_i \partial_j r_k(s) \right]_{i,j} ds + o(\|ds\|^2). \end{aligned} \quad (\text{B.5})$$

So the out-of-plane displacement is obtained by a quadratic form defined by the matrix called the *second fundamental form*, that reads as

$$\begin{aligned} \mathbf{B} : \Omega &\rightarrow \mathcal{S}_2 \\ s &\mapsto [n_k(s) \partial_i \partial_j r_k(s)]_{i,j} \end{aligned} \quad (\text{B.6})$$

using Einstein's summation convention. Another expression can be computed by deriving the relation  $n^\top Dr = 0$  which gives

$$\begin{aligned} \mathbf{B} : \Omega &\rightarrow \mathcal{S}_2 \\ s &\mapsto -Dn(s)^\top Dr(s). \end{aligned} \quad (\text{B.7})$$

## B.2 Discrete setting

Let  $\mathcal{M} = \{\mathcal{E}, \mathcal{F}\}$  be a triangular mesh of  $n_v$  vertices with  $\mathcal{E} \in \llbracket 1, n_v \rrbracket^{2n_e}$  the set of edges and  $\mathcal{F} \in \llbracket 1, n_v \rrbracket^{3n_f}$  the set of triangular faces.

We do not consider here a global parametrization  $r$  as in the previous section, but only local parametrization per face that will allow the definition of the fundamental forms over the faces. Let  $\{ijk\} \in \mathcal{F}$ . Then a possible local parametrization of this face is

$$\begin{aligned} r_{ijk} : \mathcal{T} &\rightarrow \mathbb{R}^3 \\ (u, v) &\mapsto x_i + u(x_j - x_i) + v(x_k - x_i) \end{aligned} \quad (\text{B.8})$$

with  $x \in R^{3n_v}$  the vector of the nodes positions, and  $x_i \in \mathbb{R}^3$  the position of the  $i^{\text{th}}$  vertex.

If we redo the previous reasoning on the in-plane deformation by computing



$\|r(u + du, v + dv) - r(u, v)\|^2$ , we immediately get the first fundamental form

$$\mathbf{A}_{ijk} = \begin{bmatrix} \|x_j - x_i\|^2 & (x_j - x_i)^\top(x_k - x_i) \\ \cdot & \|x_k - x_i\|^2 \end{bmatrix} \quad (\text{B.9})$$

that is constant over the triangle and symmetric.

For the second fundamental form, it is less trivial. A simple way to compute a discrete equivalent is to do a finite difference of the formulation B.7.  $Dr$  is the easiest to compute:  $\partial_u r = (x_j - x_i)$  and  $\partial_v r = (x_k - x_i)$ .

For  $Dn$ , in the mid-edge model, the normals are attached to the mid-points of the edges and not to the vertices. An approximation here is to use the Thales's theorem to say that an edge is parallel and twice the length of the segment linking the two mid-points of the two other edges. As such, the mid-edge normals are reported to the vertices, and divided differences gives  $\partial_u n = \frac{n_{jk} - n_{ik}}{2}$  and  $\partial_v n = \frac{n_{jk} - n_{ij}}{2}$ , with  $n_{ij}$  the mid-edge normal of the edge  $ij$ . Finally, we obtain

$$\mathbf{B}_{ijk} = -\frac{1}{2} \begin{bmatrix} (n_{jk} - n_{ik})^\top(x_j - x_i) & n_{jk}^\top(x_j - x_i) \\ \cdot & (n_{jk} - n_{ij})^\top(x_k - x_i) \end{bmatrix}. \quad (\text{B.10})$$

The symmetry is obtained by recalling that a normal is orthogonal to the edge it is attached to.



# Appendix C

## Reformulation of the Koiter energy

This section is meant to be a short guide to pass from the Koiter's energy as described by [Ciarlet \(2005, Section 4.1\)](#) to the formulation used in this manuscript (see e.g. [Section 1.1](#)).

Note that ([Ciarlet, 2005](#)), the quantities related to the deformed configuration are noted as depending of the displacement field  $\eta$  while the rest configuration quantities do not have any argument, and the thickness  $h$  is  $2\epsilon$ .

The main properties used to rewrite the energy are that

- The usual fundamental forms have covariant components, noted by subscript indices:  $\mathbf{A} = [\mathbf{A}_{ij}]$ ;
- The contravariant components are noted by superscript indices and are defined by  $\mathbf{A}_i \cdot \mathbf{A}^j = \delta_i^j$ ,  $\delta$  the Kronecker symbol. This implies that  $\mathbf{A}^{ij} = (\mathbf{A}^{-1})_{ij}$ .
- $[\mathbf{A}_{ij}]$  and  $[\mathbf{A}^{ij}]$  are symmetric.

We derive the reformulation for the stretching term, similar computation can be done for the bending term.

Let's start from [Ciarlet \(2005\)](#)'s expression of the Koiter's shell energy, where the terms  $G_{\alpha\beta} = \frac{1}{2}(\mathbf{A}_{\alpha\beta} - \bar{\mathbf{A}}_{\alpha\beta})$  that denote the change of metric tensor have been replaced

$$E_{\text{el}} = \frac{1}{2} \iint_{\Omega} \frac{h}{2} \left( \frac{4\lambda G}{\lambda + 2G} \bar{\mathbf{A}}^{\alpha\beta} \bar{\mathbf{A}}^{\sigma\tau} + 2G(\bar{\mathbf{A}}^{\alpha\sigma} \bar{\mathbf{A}}^{\beta\tau} + \bar{\mathbf{A}}^{\alpha\tau} \bar{\mathbf{A}}^{\beta\sigma}) \right) \underbrace{\frac{1}{2}(\mathbf{A}_{\alpha\beta} - \bar{\mathbf{A}}_{\alpha\beta})}_{M_{\alpha\beta}} \frac{1}{2}(\mathbf{A}_{\alpha\tau} - \bar{\mathbf{A}}_{\alpha\tau}) \sqrt{\det \bar{\mathbf{A}}} ds.$$

Using the symmetries, we can regroup the terms in that fashion

$$E_{\text{el}} = \frac{h}{16} \iint_{\Omega} \left( \frac{4\lambda G}{\lambda + 2G} (\bar{\mathbf{A}}^{\alpha\beta} M_{\alpha\beta}) (\bar{\mathbf{A}}^{\sigma\tau} M_{\sigma\tau}) + 2G (\bar{\mathbf{A}}^{\alpha\sigma} M_{\sigma\tau} \bar{\mathbf{A}}^{\tau\beta} M_{\beta\alpha}) + 2G (\bar{\mathbf{A}}^{\alpha\tau} M_{\tau\sigma} \bar{\mathbf{A}}^{\sigma\beta} M_{\beta\alpha}) \right) \sqrt{\det \bar{\mathbf{A}}} ds.$$

Finally, we can replace the Lamé parameters and the contravariant components and we can identify in the parenthesis the matrix norms, which finally yields

$$E_{\text{el}} = \frac{h}{16} \iint_{\Omega} \left( \frac{2E\nu}{1-\nu^2} \left( \text{tr}(\bar{\mathbf{A}}^{-1}M) \right)^2 + \frac{2E}{1-\nu} \text{tr} \left( \left( \bar{\mathbf{A}}^{-1}M \right)^2 \right) \right) \sqrt{\det \bar{\mathbf{A}}} ds. \quad (\text{C.1})$$

# Appendix D

## Second derivatives in the implicit function theorem

In the main body of this manuscript, in Section 5.4.1, we used the implicit function theorem to compute the first derivative of an implicit function. However, it may also possess higher order derivatives depending on the regularity of the equation that links the two variables (Berger, 1997).

For the case of the second derivative, we follow (Papadimitriou and Giannakoglou, 2008; Caplan, 2011) to explain how to compute it in practice.

### D.1 Framework

We are interested in solving the following optimisation problem

$$\min_{t \in \mathbb{R}^n} f(t) = F(t, u(t)) \tag{D.1}$$

with  $F : \mathbb{R}^N \times \mathbb{R}^M \rightarrow \mathbb{R}$ ,  $u : \mathbb{R}^N \rightarrow \mathbb{R}^M$  the implicit function satisfying the relation  $\forall t \in \mathbb{R}^n : \mathcal{R}(t, u(t)) = 0$  with  $\mathcal{R} : \mathbb{R}^N \times \mathbb{R}^M \rightarrow \mathbb{R}^M$ .

$F$  is supposed to be differentiable twice, so as to consider its hessian, and  $\mathcal{R}$  regular and injective "enough".

### D.2 Computation methods

We wish to compute the hessian of  $f$ , and Papadimitriou and Giannakoglou (2008); Caplan (2011) distinguish three different ways<sup>1</sup>.

Note that in the following, we will make use of the Einstein convention for the summations when the tensors products are ambiguous. We track the dimensions of the different

---

<sup>1</sup>Actually four, but the "Hyper-Dual" method is applicable only to complex functions.

tensors in the computations.

**Direct-direct method.** This method is the "brute-force" one, computing the derivatives of  $f$  by computing explicitly the derivatives  $\frac{du}{dt}$  and  $\frac{d^2u}{dt^2}$ .

$$\nabla f = \underbrace{\frac{\partial F}{\partial t}}_N + \left( \underbrace{\frac{du}{dt}}_{M \times N} \right)^\top \underbrace{\frac{\partial F}{\partial u}}_M \quad (\text{D.2a})$$

$$\nabla^2 f = \underbrace{\frac{\partial^2 F}{\partial^2 t}}_{N \times N} + \left( \underbrace{\frac{du}{dt}}_{M \times N} \right)^\top \left( \underbrace{\frac{\partial^2 F}{\partial t \partial u}}_{N \times M} + \left( \underbrace{\frac{du}{dt}}_{M \times N} \right)^\top \underbrace{\frac{\partial^2 F}{\partial^2 u}}_{M \times M} \right)^\top + \left( \underbrace{\frac{d^2 u}{d^2 t}}_{M \times N \times N} \right)^\top \underbrace{\frac{\partial F}{\partial u}}_M + \left[ \left[ \underbrace{\frac{d^2 u}{d^2 t}}_{M \times N \times N} \right]_{kij} \left[ \underbrace{\frac{\partial F}{\partial u}}_M \right]_k \right]_{ij} \quad (\text{D.2b})$$

However, we would like to avoid using these terms as their computation and their storage are heavy. The formula for the first derivative is well-known and involves the inversion of a matrix, and the formula for the second derivative is even more complex to evaluate (Berger, 1997).

**Adjoint-adjoint method.** The idea in the adjoint framework is to add extra variables, the adjoint states, that satisfy some well chosen relations that provide alternate ways of computing fragments of the hessian.

For the gradient computation, one adjoint vector  $\phi \in \mathbb{R}^M$  is used, and for the hessian, two adjoint matrices  $\nu, \mu \in \mathbb{R}^{N \times M}$  are used. We introduce the notations  $\nu_{i,:}$  and  $\nu_{:,j}$  which represents respectively the  $i$ -th row and  $j$ -th column of the matrix  $\nu$ .

The gradient and the hessian are computed using the following systems

$$\left\{ \begin{array}{l} \left( \underbrace{\frac{\partial \mathcal{R}}{\partial u}}_{M \times M} \right)^\top \phi = - \underbrace{\frac{\partial F}{\partial u}}_M \\ \nabla f = \left( \underbrace{\frac{\partial \mathcal{R}}{\partial t}}_{M \times N} \right)^\top \phi + \underbrace{\frac{\partial F}{\partial t}}_N \end{array} \right. \quad (\text{D.3a})$$

$$\left\{ \begin{array}{l}
\forall i \in \llbracket 1, N \rrbracket \quad \underbrace{\left( \frac{\partial \mathcal{R}}{\partial u} \right)}_{M \times M} \underbrace{\nu_{i,:}}_M = - \underbrace{\frac{\partial \mathcal{R}}{\partial t_i}}_M \\
\forall i \in \llbracket 1, N \rrbracket \quad \left( \frac{\partial \mathcal{R}}{\partial u} \right)^\top \underbrace{\mu_{i,:}}_M = - \underbrace{\frac{\partial^2 F}{\partial u \partial t_i}}_M - \left( \frac{\partial^2 \mathcal{R}}{\partial u \partial t_i} \right)^\top \phi \\
\qquad \qquad \qquad - \underbrace{\frac{\partial^2 F}{\partial^2 u}}_{N \times N} \underbrace{\nu_{i,:}}_M - \underbrace{\partial_u \left( \nu_{i,:}^\top \frac{\partial \mathcal{R}^\top}{\partial u} \phi \right)}_{\substack{= \left[ \sum_{m,n=1}^M \nu_{i,n} \frac{\partial^2 \mathcal{R}_m}{\partial u_k \partial u_n} \phi_m \right]_{k \in \llbracket 1, M \rrbracket}}} \\
\forall i, j \in \llbracket 1, N \rrbracket \quad \nabla^2 f_{i,j} = \frac{\partial^2 F}{\partial t_i \partial t_j} + \sum_{m=1}^M \left( \frac{\partial^2 \mathcal{R}_m}{\partial t_i \partial t_j} + \frac{\partial \mathcal{R}_m}{\partial t_k} \mu_{i,j} + \frac{\partial^2 F}{\partial u_m \partial t_j} \nu_{i,m} \right) \\
\qquad \qquad \qquad + \sum_{m,n=1}^M \frac{\partial^2 \mathcal{R}_M}{\partial u_n \partial t_j} \nu_{i,n} \phi_m
\end{array} \right. \tag{D.3b}$$

So this method requires to solve  $2N + 1$  linear systems in order to compute the adjoint states. However, the good point here is that the left hand side of these equations is constant :  $\frac{\partial \mathcal{R}}{\partial u}$ .

However, we may notice that  $\nu$  is actually  $\left(\frac{du}{dt}\right)^\top$ , *i.e.* the gradient of the implicit function. This defeats the purpose of the adjoint method to compute the gradient.

Therefore, this methods is morally equivalent to the following one concerning the gradient. We will see that the later is more effective, as it requires less linear systems.

**Direct-adjoint method.** This methods assumes that  $\frac{du}{dt}$  has been computed for instance by solving the  $N$  first linear systems of Equation D.3b. Surprisingly, this methods also make use of the adjoint state  $\phi$  obtained by solving the first linear system of Equation D.3a.

With that, the hessian is computed using the following formula

The hessian is then given by

$$\forall i, j \in \llbracket 1, N \rrbracket \quad \nabla^2 f_{i,j} = \frac{\partial^2 F}{\partial t_i \partial t_j} + \phi^\top \frac{\partial^2 \mathcal{R}}{\partial t_i \partial t_j} + \frac{du}{dt}^\top \left( \frac{\partial^2 F}{\partial^2 u} + \phi^\top \frac{\partial^2 \mathcal{R}}{\partial^2 u} \right) \frac{du}{dt_j} \\
+ \frac{\partial^2 F}{\partial t_i \partial u} \frac{du}{dt_j} + \frac{\partial^2 F}{\partial t_j \partial u} \frac{du}{dt_i} \\
+ \left( \phi^\top \frac{\partial^2 \mathcal{R}}{\partial t_i \partial u} \right) \frac{du}{dt_j} + \left( \phi^\top \frac{\partial^2 \mathcal{R}}{\partial t_j \partial u} \right) \frac{du}{dt_i} \tag{D.4}$$

This approach is much more efficient as it requires to solve  $N + 1$  linear systems before assembling the matrix.

### D.3 Application to our inverse problem

With the previous section, we can now compute the hessian of our inverse problem. We have to replace using the following conversion array from the notations of the main body

Formulas	Our case
$N$	$3n$
$M$	$3n$
$t$	$\bar{x}$
$u$	$\Phi(\bar{x}) \equiv x$
$\phi$	$p$
$F$	$(\bar{x}, x) \rightarrow \ x - x_t\ ^2$
$R$	$\nabla E_p$

**Adjoint-adjoint method.** The system of the adjoint-adjoint method becomes in our case

$$\begin{cases} \nabla_{xx}^2 E_p p = -(u(\bar{x}) - x_t) \\ \nabla f = (\nabla_{x\bar{x}}^2 E_p)^\top p + 0 \end{cases} \quad (\text{D.5a})$$

$$\left\{ \begin{array}{l} \forall i \in \llbracket 1, N \rrbracket \quad \nabla_{xx}^2 E_p \nu_{i,:} = -\nabla_{x\bar{x}_i}^2 E_p \\ \forall i \in \llbracket 1, N \rrbracket \quad \nabla_{xx}^2 E_p \mu_{i,:} = 0 - \nabla_{\bar{x}_i x x}^3 E_p p \\ \quad \quad \quad - \nu_{i,:} - \left[ \sum_{m,n=1}^M \nabla_{x_i x_k x_m}^3 E_p \nu_{i,n} E_p p_m \right]_{k \in \llbracket 1, N \rrbracket} \\ \forall i, j \in \llbracket 1, N \rrbracket \quad \nabla^2 f_{i,j} = 0 + \sum_{m=1}^M \left( \nabla_{x_m \bar{x}_i \bar{x}_j}^3 E_p + \nabla_{x_m \bar{x}_j}^2 E_p \mu_{i,j} + 0 \right) \\ \quad \quad \quad + \sum_{m,n=1}^M D_{x_m x_n \bar{x}_j}^3 E_p \nu_{i,n} \phi_m \end{array} \right. \quad (\text{D.5b})$$

As mentioned in the previous section, we only use this method to compute the gradient, the adjoint state  $p$  and the derivative of the implicit function  $\nu = \left(\frac{du}{dt}\right)^\top$ .

**Direct-adjoint method.** When we replace our problem in this method, we obtain the following equation for the hessian

$$\begin{aligned} i, j \in \llbracket 1, N \rrbracket \quad D^2 f_{i,j} &= 0 + p^\top D_{x\bar{x}_i \bar{x}_j}^3 E_p + \frac{du}{d\bar{x}_i}^\top \left( \mathbf{I} + p^\top D_{xxx}^3 E_p \right) \frac{du}{d\bar{x}_j} \\ &0 + 0 \\ &+ \left( p^\top D_{xx\bar{x}_i}^3 E_p \right) \frac{du}{d\bar{x}_j} + \left( p^\top D_{xx\bar{x}_j}^3 E_p \right) \frac{du}{d\bar{x}_i} \end{aligned} \quad (\text{D.6})$$



This is the formula we use in practice. Note that there are some symmetries in the derivatives of  $E_p$  that can be used to lighten the evaluation of the third derivatives.



# Bibliography

- Vincent Acary and Bernard Brogliato. 2008. *Numerical methods for nonsmooth dynamical systems*. Lecture Notes in Computational and Applied Mechanics, Vol. 35. Springer.
- Mohammed A. Agwa and Antonio Pinto da Costa. 2021. Existence and multiplicity of solutions in frictional contact mechanics. Part I: A simplified criterion. *European Journal of Mechanics - A/Solids* 85 (2021), 104062. <https://doi.org/10.1016/j.euromechsol.2020.104062>
- Pierre Alart and Alain Curnier. 1991. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput. Methods Appl. Mech. Eng.* 92, 3 (1991), 353–375.
- Guillaume Amontons. 1699. De la résistance causée dans les Machines, tant par les frottemens des parties qui les composent, que par roideur des cordes qu'on y employe, & la manière de calculer l'un & l'autre. *Histoire de l'Académie royale des sciences* (12 1699), 206–227.
- Lars-Erik Andersson, Antonio Pinto da Costa, and Mohammed A. Agwa. 2016. Existence and uniqueness for frictional incremental and rate problems – sharp critical bounds. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 96, 1 (2016), 78–105. <https://doi.org/10.1002/zamm.201400143> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/zamm.201400143>
- Mihai Anitescu and Gary D. Hart. 2004. A constraint-stabilized time-stepping approach for rigid multibody dynamics with joints, contact and friction. *Internat. J. Numer. Methods Engrg.* 60, 14 (2004), 2335–2371. <https://doi.org/10.1002/nme.1047> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1047>
- Douglas Arnold and Franco Brezzi. 1997. Locking free finite element for shells. *Math. Comput.* 66 (Jan. 1997), 1–14. <https://doi.org/10.1090/S0025-5718-97-00785-0>
- Uri M. Ascher, Robert M.M. Mattheij, and Robert D. Russell. 1995. *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. SIAM.

- Marco Attene, Marco Livesu, Sylvain Lefebvre, Thomas Funkhouser, Stefano Ellero, Szymon Rusinkiewicz, Jonàs Martínez, and Amit Haim Bermano. 2018. *Design, Representations, and Processing for Additive Manufacturing*. Synthesis Lectures on Visual Computing: Computer Graphics, Animation, Computational Photography, and Imaging, Vol. 10. 146 pages. <https://hal.inria.fr/hal-01836525>
- Basile Audoly and Yves Pomeau. 2010. *Elasticity and Geometry: from hair curls to the nonlinear response of shells*. Oxford University Press.
- David Baraff. 1991. Coping with friction for non-penetrating rigid body simulation. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'91)*. ACM, 31–40.
- David Baraff. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'94)*. ACM, New York, NY, USA, 23–34.
- David Baraff and Andrew Witkin. 1998. Large Steps in Cloth Simulation. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'98)*. 43–54.
- Aric Bartle, Alla Sheffer, Vladimir G. Kim, Danny Kaufman, Nicholas Vining, and Flo-raine Berthouzoz. 2016. Physics-driven Pattern Adjustment for Direct 3D Garment Editing. *ACM Transactions on Graphics* 35, 4, Article 50 (July 2016), 11 pages. <https://doi.org/10.1145/2897824.2925896>
- Stéphanie Basseville and Alain Léger. 2006. Stability of equilibrium states in a simple system with unilateral contact and Coulomb friction. *Archive of Applied Mechanics* 76, 7 (Dec. 2006), 403–428. <https://doi.org/10.1007/s00419-006-0040-x>
- James V. Beck and Keith A. Woodbury. 1998. Inverse problems and parameter estimation: integration of measurements and analysis. *Measurement Science and Technology* 9, 6 (1998), 839.
- Erich Berger. 1997. Remarks on the Analytic Implicit Function Theorem. *J. Math. Anal. Appl.* 209, 2 (1997), 435–459. <https://doi.org/10.1006/jmaa.1997.5285>
- Miklós Bergou, Basile Audoly, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2010. Discrete Viscous Threads. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'10)* 29, 4 (2010).
- Miklós Bergou, Max Wardetzky, David Harmon, Denis Zorin, and Eitan Grinspun. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (Cagliari, Sardinia, Italy) (SGP '06)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 227–230.

- Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. 2008. Discrete elastic rods. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'08)* 27, 3 (2008), 1–12. <https://doi.org/10.1145/1360612.1360662>
- Amit H. Bermano, Thomas Funkhouser, and Szymon Rusinkiewicz. 2017. State of the Art in Methods and Representations for Fabrication-Aware Design. *Comput. Graph. Forum* 36, 2 (May 2017), 509–535. <https://doi.org/10.1111/cgf.13146>
- Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. 2006. Super-Helices for Predicting the Dynamics of Natural Hair. *ACM Transactions on Graphics* (2006). <https://hal.inria.fr/inria-00384718>
- Florence Bertails-Descoubes. 2017. *Numerical modeling of elastic slender structures subject to contact and friction: from dynamic simulation to inverse static design*. Habilitation à diriger des recherches. Université Grenoble Alpes. <https://hal.archives-ouvertes.fr/tel-01925260>
- Florence Bertails-Descoubes and Basile Audoly. 2019. Graphyz, The first Graphics-Physics Workshop. Grenoble, France. <https://project.inria.fr/graphyz/>
- Florence Bertails-Descoubes, Florent Cadoux, Gilles Daviet, and Vincent Acary. 2011. A nonsmooth Newton solver for capturing exact Coulomb friction in fiber assemblies. *ACM Transactions on Graphics* 30, Article 6 (February 2011), 14 pages. Issue 1. <https://doi.org/10.1145/1899404.1899410>
- Florence Bertails-Descoubes, Alexandre Derouet-Jourdan, Victor Romero, and Arnaud Lazarus. 2018. Inverse design of an isotropic suspended Kirchhoff rod: theoretical and numerical results on the uniqueness of the natural shape. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474, 2212 (April 2018), 1–26. <https://doi.org/10.1098/rspa.2017.0837>
- Floraine Berthouzoz, Akash Garg, Danny M. Kaufman, Eitan Grinspun, and Maneesh Agrawala. 2013. Parsing Sewing Patterns into 3D Garments. *ACM Trans. Graph.* 32, 4, Article 85 (jul 2013), 12 pages. <https://doi.org/10.1145/2461912.2461975>
- Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Hyunho Richard Lee, Hanspeter Pfister, Markus Gross, and Wojciech Matusik. 2010. Design and Fabrication of Materials with Desired Deformation Behavior. *ACM Trans. Graph.* 29, 4, Article 63 (July 2010), 10 pages. <https://doi.org/10.1145/1778765.1778800>
- Bernd Bickel, Moritz Bächer, Miguel A Otaduy, Wojciech Matusik, Hanspeter Pfister, and Markus Gross. 2009. Capture and Modeling of Non-Linear Heterogeneous Soft Tissue. *ACM Trans. Graph.* 28, 3, Article 89 (July 2009), 9 pages. <https://doi.org/10.1145/1531326.1531395>

- Bernd Bickel, Paolo Cignoni, Luigi Malomo, and Nico Pietroni. 2018. State of the Art on Stylized Fabrication. *Computer Graphics Forum* 37 (2018). <http://vcg.isti.cnr.it/Publications/2018/BCMP18>
- William G. Bickley. 1934. The heavy elastica. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 17, 113 (1934), 603–622. <https://doi.org/10.1080/14786443409462419>
- ]Blender Blender. [n.d.]. Blender - a 3D modelling and rendering package. <http://www.blender.org>
- Alejandro Blumentals, Bernard Brogliato, and Florence Bertails-Descoubes. 2016. The contact problem in Lagrangian systems subject to bilateral and unilateral constraints, with or without sliding Coulomb’s friction: A tutorial. *Multibody System Dynamics* 38, 1 (Sept. 2016), 43–76. <https://doi.org/10.1007/s11044-016-9527-6>
- Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. 2010. *Polygon Mesh Processing*. AK Peters / CRC Press. 250 pages. <https://hal.inria.fr/inria-00538098>
- Sofien Bouaziz, Sebastian Martin, Tiantian Liu, Ladislav Kavan, and Mark Pauly. 2014. Projective Dynamics: Fusing Constraint Projections for Fast Simulation. *ACM Trans. Graph.* 33, 4, Article 154 (July 2014), 11 pages. <https://doi.org/10.1145/2601097.2601116>
- Derek Bradley, Tamy Boubekeur, and Wolfgang Heidrich. 2008. Accurate multiview reconstruction using robust binocular stereo and surface meshing. In *Computer Vision and Pattern Recognition (CVPR’08)*.
- Robert Bridson, Ronald Fedkiw, and John Anderson. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH’02)* 21, 3 (2002), 594–603. <http://www.cs.ubc.ca/~rbridson/docs/cloth2002.pdf>
- Robert Bridson, Sebastian Marino, and Ronald Fedkiw. 2003. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA’03)*. ACM-EG SCA, 28–36. <http://www.cs.ubc.ca/~rbridson/docs/cloth2003.pdf>
- Rémi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design Preserving Garment Transfer. *ACM Transactions on Graphics* 31, 4 (2012), 36:1–36:11.

- George E. Brown, Matthew Overby, Zahra Forootaninia, and Rahul Narain. 2018. Accurate Dissipative Forces in Optimization Integrators. *ACM Trans. Graph.* 37, 6, Article 282 (Dec. 2018), 14 pages. <https://doi.org/10.1145/3272127.3275011>
- Pierre-Thomas Brun, Neil M Ribe, and Basile Audoly. 2012. A numerical investigation of the fluid mechanical sewing machine. *Physics of Fluids* 24, 4 (2012), 043102. <https://doi.org/10.1063/1.3703316> arXiv:<https://doi.org/10.1063/1.3703316>
- Edgar Buckingham. 1914. On physically similar systems; illustrations of the use of dimensional equations. *Physical Review* 4, 4 (1914), 345–376.
- Thomas Buffet, Damien Rohmer, Loïc Barthe, Laurence Boissieux, and Marie-Paule Cani. 2019. Implicit Untangling: A Robust Solution for Modeling Layered Clothing. *ACM Trans. Graph.* 38, 4, Article 120 (July 2019), 12 pages. <https://doi.org/10.1145/3306346.3323010>
- Andrew Butts, Ben Porter, Dirk Van Gelder, Mark Hessler, Venkateswaran Krishna, and Gary Monheit. 2018. Engineering Full-Fidelity Hair for Incredibles 2. In *ACM SIGGRAPH 2018 Talks* (Vancouver, British Columbia, Canada) (*SIGGRAPH '18*). Association for Computing Machinery, New York, NY, USA, Article 14, 2 pages. <https://doi.org/10.1145/3214745.3214798>
- Florent Cadoux. 2009. *Méthodes d'optimisation pour la dynamique non-régulière*. Ph.D. Dissertation. Université Joseph Fourier.
- Mauricio F. Caliri, Antonio J.M. Ferreira, and Volnei Tita. 2016. A review on plate and shell theories for laminated and sandwich structures highlighting the Finite Element Method. *Composite Structures* 156 (2016), 63–77. <https://doi.org/10.1016/j.compstruct.2016.02.036> 70th Anniversary of Professor J. N. Reddy.
- Anca Capatina. 2014. *Variational Inequalities and Frictional Contact Problems*. Vol. 31. <https://doi.org/10.1007/978-3-319-10163-7>
- Philip Caplan. 2011. Numerical computation of second derivatives with applications to optimization problems. *Unpublished academic report, MIT* (2011).
- Michel Carignan, Ying Yang, Nadia Magnenat-Thalmann, and Daniel Thalmann. 1992. Dressing Animated Synthetic Actors with Complex Deformable Clothes. *SIGGRAPH Comput. Graph.* 26, 2 (July 1992), 99–104. <https://doi.org/10.1145/142920.134017>
- Romain Casati. 2015. *Some contributions to the numerical modeling of slender structures for computer graphics*. Theses. Université Grenoble Alpes. <https://tel.archives-ouvertes.fr/tel-01502808>

- Romain Casati, Gilles Daviet, and Florence Bertails-Descoubes. 2016. *Inverse elastic cloth design with contact and friction*. Research Report. Inria Grenoble, Université de Grenoble. <https://hal.archives-ouvertes.fr/hal-01309617>
- Julie S. Chalfant and Takashi Maekawa. 1998. Design for Manufacturing Using B-Spline Developable Surfaces. *Journal of Ship Research* 42, 03 (09 1998), 207–215. <https://doi.org/10.5957/jsr.1998.42.3.207> arXiv:<https://onepetro.org/JSR/article-pdf/42/03/207/2231554/sname-jsr-1998-42-3-207.pdf>
- Raphaël Charrondière, Florence Bertails-Descoubes, Sébastien Neukirch, and Victor Romero. 2020. Numerical modeling of inextensible elastic ribbons with curvature-based elements. *Computer Methods in Applied Mechanics and Engineering* 364 (June 2020), 1–32. <https://doi.org/10.1016/j.cma.2020.112922>
- Hsiao-Yu Chen, Arnav Sastry, Wim M. van Rees, and Etienne Vouga. 2018. Physical Simulation of Environmentally Induced Thin Shell Deformation. *ACM Trans. Graph.* 37, 4, Article 146 (July 2018), 13 pages. <https://doi.org/10.1145/3197517.3201395>
- Tian Chen, Julian Panetta, Max Schnaubelt, and Mark Pauly. 2021. Bistable Auxetic Surface Structures. *ACM Trans. Graph.* (2021).
- Xiang Chen, Changxi Zheng, Weiwei Xu, and Kun Zhou. 2014. An Asymptotic Numerical Method for Inverse Elastic Shape Design. *ACM Transactions on Graphics* 33, 4, Article 95 (July 2014), 11 pages. <https://doi.org/10.1145/2601097.2601189>
- Sunny Chiu-Webster and John R. Lister. 2006. The fall of a viscous thread onto a moving surface: a ‘fluid-mechanical sewing machine’. *Journal of Fluid Mechanics* 569 (2006), 89–111. <https://doi.org/10.1017/S0022112006002503>
- Kwang-Jin Choi and Hyeong-Seok Ko. 2002. Stable but Responsive Cloth. *ACM Trans. Graph.* 21, 3 (July 2002), 604–611. <https://doi.org/10.1145/566654.566624>
- Philippe G. Ciarlet. 2000. *Mathematical Elasticity, Volume 3: Theory of Shells*.
- Philippe G. Ciarlet. 2005. An Introduction to Differential Geometry with Applications to Elasticity. *Journal of Elasticity* 78-79 (01 2005), 1–215. <https://doi.org/10.1007/s10659-005-4738-8>
- David Clyde, Joseph Teran, and Rasmus Tamstorf. 2017. Modeling and Data-Driven Parameter Estimation for Woven Fabrics. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation (Los Angeles, California) (SCA '17)*. Association for Computing Machinery, New York, NY, USA, Article 17, 11 pages. <https://doi.org/10.1145/3099564.3099577>



- Eulalie Coevoet, Adrien Escande, and Christian Duriez. 2019. Soft robots locomotion and manipulation control using FEM simulation and quadratic programming. In *RoboSoft 2019 - IEEE International Conference on Soft Robotics*. Seoul, South Korea. <https://hal.inria.fr/hal-02079151>
- David Cohen-Steiner and Jean-Marie Morvan. 2003. Restricted Delaunay Triangulations and Normal Cycle. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry* (San Diego, California, USA) (*SCG '03*). Association for Computing Machinery, New York, NY, USA, 312–321. <https://doi.org/10.1145/777792.777839>
- Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. 2009. *The Linear Complementarity Problem*. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9780898719000>  
arXiv:<https://epubs.siam.org/doi/pdf/10.1137/1.9780898719000>
- Charles Augustin Coulomb. 1781. *Théorie des machines simples en ayant égard au frottement de leurs parties et à la roideur des cordages*.
- Michael A Crisfield and Gordan Jelenić. 1998. Objectivity of strain measures in the geometrically exact three-dimensional beam theory and its finite-element implementation. *Proc. Royal Society of London, Series A* 455, 1983 (1998), 1125–1147.
- Peter A. Cundall. 1971. A computer model for simulating progressive, large-scale movements in blocky rock systems.
- Charles Dapogny, Cécile Dobrzynski, and Pascal Frey. 2014. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics* (April 2014). <https://doi.org/10.1016/j.jcp.2014.01.005>
- Gilles Daviet. 2016. *Modèles et algorithmes pour la simulation du contact frottant dans les matériaux complexes : application aux milieux fibreux et granulaires*. Theses. Université Grenoble Alpes. <https://tel.archives-ouvertes.fr/tel-01684673>
- Gilles Daviet. 2020. Simple and Scalable Frictional Contacts for Thin Nodal Objects. *ACM Trans. Graph.* 39, 4, Article 61 (July 2020), 16 pages. <https://doi.org/10.1145/3386569.3392439>
- Gilles Daviet, Florence Bertails-Descoubes, and Laurence Boissieux. 2011. A hybrid iterative solver for robustly capturing Coulomb friction in hair dynamics. *ACM Transactions on Graphics* 30 (2011), 139:1–139:12. Issue 6.

- Fernando de Goes, Donald Fong, and Meredith O'Malley. 2020. Garment Refitting for Digital Characters. In *ACM SIGGRAPH 2020 Talks* (Virtual Event, USA) (*SIGGRAPH '20*). Association for Computing Machinery, New York, NY, USA, Article 74, 2 pages. <https://doi.org/10.1145/3388767.3407348>
- Erik Demaine and Tomohiro Tachi. 2017. Origamizer: A Practical Algorithm for Folding Any Polyhedron. <https://doi.org/10.4230/LIPIcs.SoCG.2017.34>
- Alexandre Derouet-Jourdan, Florence Bertails-Descoubes, Gilles Daviet, and Joëlle Thollot. 2013. Inverse Dynamic Hair Modeling with Frictional Contact. *ACM Transactions on Graphics* 32, 6, Article 159 (Nov. 2013), 10 pages. <https://doi.org/10.1145/2508363.2508398>
- Alexandre Derouet-Jourdan and Joëlle Bertails-Descoubes, Florence and Thollot. 2010. Stable Inverse Dynamic Curves. *ACM Transactions on Graphics* 29, Article 137 (December 2010), 10 pages. Issue 6. <https://doi.org/10.1145/1882261.1866159>
- Manfredo P. do Carmo. 1976. *Differential geometry of curves and surfaces*. Prentice Hall. I–VIII, 1–503 pages.
- Cécile Dobrzynski and Pascal Frey. 2008. Anisotropic Delaunay mesh adaptation for unsteady simulations. In *17th international Meshing Roundtable*. United States, 177–194. <https://hal.archives-ouvertes.fr/hal-00353786>
- Eusebius Doedel, Herbert B Keller, and Jean Pierre Kernevez. 1991. Numerical Analysis and Control of Bifurcation Problems (I) Bifurcation in Finite Dimensions. *International Journal of Bifurcation and Chaos* 1, 3 (1991), 493–520.
- Virginie Duclaux. 2006. *Pulmonary occlusions, eyelid entropion and aneurysm : a physical insight in physiology*. Ph.D. Dissertation. Université de Provence - Aix-Marseille I. <https://tel.archives-ouvertes.fr/tel-00130610>
- Aurélié Fargette. 2017. *Soft Interfaces: from elastocapillary snap-through to droplet dynamics on elastomers*. Ph.D. Dissertation. Université Pierre et Marie Curie - Paris VI.
- Roy Featherstone. 1987. *Robot Dynamics Algorithm*. Kluwer Academic Publishers, USA.
- Laura W. Ferris. 1968. A Standard Series of Developable Surfaces. *Marine Technology and SNAME News* 5, 01 (01 1968), 52–62. <https://doi.org/10.5957/mt1.1968.5.1.52> arXiv:<https://onepetro.org/MTSN/article-pdf/5/01/52/2200329/sname-mtsn-1968-5-1-52.pdf>

- Marco Fratarcangeli, Valentina Tibaldo, and Fabio Pellacini. 2016. Vivace: A Practical Gauss-Seidel Method for Stable Soft Body Dynamics. *ACM Trans. Graph.* 35, 6, Article 214 (Nov. 2016), 9 pages.
- William Frey. 2002. Boundary Triangulations Approximating Developable Surfaces that Interpolate a Close Space Curve. *Int. J. Found. Comput. Sci.* 13 (04 2002), 285–302. <https://doi.org/10.1142/S0129054102001096>
- Johan Gaume, Alec van Herwijnen, Ted Gast, Joseph Teran, and Chenfanfu Jiang. 2019. Investigating the release and flow of snow avalanches at the slope-scale using a unified model based on the material point method. *Cold Regions Science And Technology* 168 (2019), 102847. <https://doi.org/10.1016/j.coldregions.2019.102847> International Snow Science Workshop (ISSW), Oct 07-12, 2018, Innsbruck, AUSTRIA. This is an open access article under the terms of the Creative Commons Attribution License.
- Konstantinos Gavriil, Ruslan Guseinov, Jesús Pérez, Davide Pellis, Paul Henderson, Florian Rist, Helmut Pottmann, and Bernd Bickel. 2020. Computational Design of Cold Bent Glass Façades. *ACM Trans. Graph. (SIGGRAPH Asia 2020)* 39, 6, Article 208 (Dec 2020), 16 pages.
- Moritz Geilinger, David Hahn, Jonas Zehnder, Moritz Bächer, Bernhard Thomaszewski, and Stelian Coros. 2020. ADD: Analytically Differentiable Dynamics for Multi-Body Systems with Frictional Contact. *CoRR* abs/2007.00987 (2020). arXiv:2007.00987 <https://arxiv.org/abs/2007.00987>
- James M. Gere. 2004. *Mechanics of Materials* (6th ed.). Thomson-Brooks/Cole.
- Rony Goldenthal, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. 2007. Efficient simulation of inextensible cloth. In *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'07 (San Diego, California) (SIGGRAPH '07))*. ACM, New York, NY, USA, Article 49. <https://doi.org/10.1145/1275808.1276438>
- Gene H. Golub and Charles F. Van Loan. 1996. *Matrix Computations* (third ed.). The Johns Hopkins University Press.
- Alvaro A. Grandi, Suzie Protière, and Arnaud Lazarus. 2021. Enhancing and controlling parametric instabilities in mechanical systems. *Extreme Mechanics Letters* 43 (2021), 101195. <https://doi.org/10.1016/j.eml.2021.101195>
- Eitan Grinspun, Yotam Gingold, Jason Reisman, and Denis Zorin. 2006. Computing discrete shape operators on general meshes. *Computer Graphics Forum* 25, 3 (2006), 547–556. <https://doi.org/10.1111/j.1467-8659.2006.00974.x>

- Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. 2003. Discrete Shells. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'03)*. ACM-EG SCA, 62–67.
- Ruslan Guseinov, Connor McMahan, Jesús Pérez, Chiara Daraio, and Bernd Bickel. 2020. Programming temporal morphing of self-actuated shells. *Nature Communications* 11 (Jan. 2020). <https://doi.org/10.1038/s41467-019-14015-2>
- Sunil Hadap. 2006. Oriented strands - dynamics of stiff multi-body system. In *ACM SIGGRAPH - EG Symp. on Comp. Anim. (SCA'06)*. ACM-EG SCA, 91–100.
- Christian Hafner and Bernd Bickel. 2021. The Design Space of Plane Elastic Curves. *ACM Trans. Graph.* (2021).
- Jack S. Hale, Matteo Brunetti, Stéphane PA Bordas, and Corrado Maurini. 2018. Simple and extensible plate and shell finite element models through automatic code generation tools. *Computers & Structures* 209 (2018), 163–181.
- David Harmon, Etienne Vouga, Breannan Smith, Rasmuth Tamstorf, and Eitan Grinspun. 2009. Asynchronous contact mechanics. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'09)*. <http://www.cs.columbia.edu/cg/ACM/>
- David Harmon, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2008. Robust Treatment of Simultaneous Collisions. *SIGGRAPH (ACM Transactions on Graphics)* 27, 3 (2008), 1–4.
- David Hilbert and Stephan Cohn-Vossen. 1952. *Geometry and the Imagination*. AMS Chelsea Pub.
- Philipp Holl, Vladlen Koltun, and Nils Thuerey. 2020. Learning to Control PDEs with Differentiable Physics. arXiv:2001.07457 [cs.LG]
- Nan Hu and Rigoberto Burgueño. 2015. Tailoring the elastic postbuckling response of cylindrical shells: A route for exploiting instabilities in materials and mechanical systems. *Extreme Mechanics Letters* 4 (2015), 103–110. <https://doi.org/10.1016/j.eml.2015.05.003>
- Yuanming Hu, Jiancheng Liu, Andrew Spielberg, Joshua B. Tenenbaum, William T. Freeman, Jiajun Wu, Daniela Rus, and Wojciech Matusik. 2019. ChainQueen: A Real-Time Differentiable Physical Simulator for Soft Robotics. In *2019 International Conference on Robotics and Automation (ICRA)*. 6265–6271. <https://doi.org/10.1109/ICRA.2019.8794333>

- Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. 2013. Artistic Simulation of Curly Hair. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Anaheim, California) (*SCA '13*). Association for Computing Machinery, New York, NY, USA, 63–71. <https://doi.org/10.1145/2485895.2485913>
- Takeo Igarashi and John F. Hughes. 2003. Clothing Manipulation. *ACM Transactions on Graphics* 22, 3 (July 2003), 697–697. <https://doi.org/10.1145/882262.882328>
- Florin Isvoranu, Julian Panetta, Tian Chen, Etienne Bouleau, and Mark Pauly. 2019. X-Shell Pavilion: A Deployable Elastic Rod Structure. *Proceedings of the IASS Annual Symposium - Structural Membranes 2019* (2019).
- Caigui Jiang, Klara Mundilova, Florian Rist, Johannes Wallner, and Helmut Pottmann. 2019. Curve-Pleated Structures. *ACM Trans. Graph.* 38, 6, Article 169 (Nov. 2019), 13 pages. <https://doi.org/10.1145/3355089.3356540>
- Fritz John. 1965. *Estimates for the derivatives of the stresses in a thin shell and interior shell equations.*
- Fritz John. 1971. Refined interior equations for thin elastic shells. *Comm. Pure Appl. Math.* 24 (1971), 583–615.
- Danny Kaufman, Timothy Edmunds, and Dinesh K. Pai. 2005. Fast frictional dynamics for rigid bodies. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH'05 )* 24, 3 (2005), 946–956.
- Danny Kaufman, Shinjiri Sueda, Doug L. James, and Dinesh K. Pai. 2008. Staggered Projections for Frictional Contact in Multibody Systems. *ACM Transactions on Graphics (Proc. ACM SIGGRAPH Asia'08 )* 27, 5 (2008), 164:1–164:11.
- Danny Kaufman, Rasmus Tamstorf, Breannan Smith, Jean-Marie Aubry, and Eitan Grinspun. 2014. Adaptive Nonlinearity for Collisions in Complex Rod Assemblies. *ACM Transactions on Graphics* (2014). <http://www.cs.columbia.edu/cg/adonis>
- Sueo Kawabata and Masako Niwa. 1989. Fabric Performance in Clothing and Clothing Manufacture. *The Journal of The Textile Institute* 80, 1 (1989), 19–50. <https://doi.org/10.1080/00405008908659184>
- Anders Klarbring and Jong-Shi Pang. 1998. Existence of Solutions to Discrete Semicoercive Frictional Contact Problems. *SIAM J. on Optimization* 8, 2 (Feb. 1998), 414–442. <https://doi.org/10.1137/S105262349629784X>
- Warner T. Koiter. 1966. On the Nonlinear Theory of Thin Elastic Shells. *Proc. Koninkl. Ned. Akad. van Wetenschappen, Series B* 69 (1966), 1–54.

- Martin Komaritzan and Mario Botsch. 2018. Projective Skinning. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 1, Article 12 (July 2018), 19 pages.
- Mina Konaković-Luković, Julian Panetta, Keenan Crane, and Mark Pauly. 2018. Rapid Deployment of Curved Surfaces via Programmable Auxetics. *ACM Trans. Graph.* 37, 4, Article 106 (July 2018), 13 pages. <https://doi.org/10.1145/3197517.3201373>
- Kim Krieger. 2012. Extreme mechanics: buckling down. *Nature* 488, 7410 (Aug. 2012), 146–147.
- Francesco Laccone, Luigi Malomo, Nico Pérez, Jesús and Pietroni, Federico Ponchio, Bernd Bickel, and Paolo Cignoni. 2019. FlexMaps Pavilion: a twisted arc made of mesostructured flat flexible panels. In *FORM and FORCE, IASS Symposium 2019, Structural Membranes 2019*. CIMNE, 498–504. <http://vcg.isti.cnr.it/Publications/2019/LMPPPBC19>
- Lev D. Landau and Evgenii M. Lipshitz. 1959. *Theory of elasticity*. Pergamon London.
- Marie Le Merrer, Jacopo Seiwert, David Quéré, and Christophe Clanet. 2008. Shapes of hanging viscous filaments. *EPL - Europhysics Letters* 84, 5 (2008), 56004. <https://doi.org/10.1209/0295-5075/84/56004>
- Remco Leine and Nathan van de Wouw. 2008. Stability properties of equilibrium sets of non-linear mechanical systems with dry friction and impact. *Nonlinear Dynamics* 51 (03 2008), 551–583. <https://doi.org/10.1007/s11071-007-9244-z>
- Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. 2018. Shape Reconstruction Using Volume Sweeping and Learned Photoconsistency. In *European Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 11213)*. Springer, Munich, Germany, 796–811. [https://doi.org/10.1007/978-3-030-01240-3\\_48](https://doi.org/10.1007/978-3-030-01240-3_48)
- Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. 2017. BendSketch: Modeling Freeform Surfaces through 2D Sketching. *ACM Trans. Graph.* 36, 4, Article 125 (July 2017), 14 pages. <https://doi.org/10.1145/3072959.3073632>
- Cheng Li, Min Tang, Ruofeng Tong, Ming Cai, Jieyi Zhao, and Dinesh Manocha. 2020b. P-Cloth: Interactive Complex Cloth Simulation on Multi-GPU Systems Using Dynamic Matrix Assembly and Pipelined Implicit Integrators. *ACM Trans. Graph.* 39, 6, Article 180 (Nov. 2020), 15 pages. <https://doi.org/10.1145/3414685.3417763>
- Jie Li, Gilles Daviet, Rahul Narain, Florence Bertails-Descoubes, Matthew Overby, George Brown, and Laurence Boissieux. 2018a. An Implicit Frictional Contact Solver for Adaptive Cloth Simulation. *ACM Transactions on Graphics* 37, 4, Article 52 (Aug. 2018), 15 pages.

- Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny Kaufman. 2020a. Incremental Potential Contact: Intersection- and Inversion-free, Large-Deformation Dynamics. *ACM Transactions on Graphics* 39 (07 2020), 20. <https://doi.org/10.1145/3386569.3392425>
- Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018b. FoldSketch: Enriching garments with physically reproducible folds. *ACM Transactions on Graphics* 37, 4, Article 133 (Aug. 2018), 13 pages.
- Junbang Liang, Ming C. Lin, and Vladlen Koltun. 2019. Differentiable Cloth Simulation for Inverse Problems. In *Conference on Neural Information Processing Systems*.
- Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J. Gortler. 2008. A Local/Global Approach to Mesh Parameterization. In *Proceedings of the Symposium on Geometry Processing (Copenhagen, Denmark) (SGP '08)*. Eurographics Association, Goslar, DEU, 1495–1504.
- Yong-Jin Liu, Kai Tang, and Ajay Joneja. 2007. Modeling dynamic developable meshes by the Hamilton principle. *Computer-Aided Design* 39, 9 (2007), 719–731. <https://doi.org/10.1016/j.cad.2007.02.013>
- Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. 2019. Reparameterizing Discontinuous Integrands for Differentiable Rendering. *ACM Transactions on Graphics* 38, 6 (Nov. 2019), 228:1 – 228:14. <https://doi.org/10.1145/3355089.3356510>
- Mickaël Ly, Romain Casati, Florence Bertails-Descoubes, Mélina Skouras, and Laurence Boissieux. 2018. Inverse Elastic Shell Design with Contact and Friction. *ACM Transactions on Graphics* 37, 6 (Nov. 2018), 1–16. <https://doi.org/10.1145/3272127.3275036>
- Mickaël Ly, Jean Jouve, Laurence Boissieux, and Florence Bertails-Descoubes. 2020. Projective Dynamics with Dry Frictional Contact. *ACM Transactions on Graphics* 39, 4 (2020), Article 57:1–8. <https://doi.org/10.1145/3386569.3392396>
- Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Tae-Yong Kim. 2020. *Primal/Dual Descent Methods for Dynamics*. Eurographics Association, Goslar, DEU. <https://doi.org/10.1111/cgf.14104>
- Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Viktor Makoviychuk. 2019. Non-Smooth Newton Methods for Deformable Multi-Body Dynamics. *ACM Trans. Graph.* 38, 5, Article 140 (Oct. 2019), 20 pages. <https://doi.org/10.1145/3338695>

- Miles Macklin, Matthias Müller, and Nuttapong Chentanez. 2016. XPBD: Position-Based Simulation of Compliant Constrained Dynamics. In *Proceedings of the 9th International Conference on Motion in Games (Burlingame, California) (MIG '16)*. Association for Computing Machinery, New York, NY, USA, 49–54. <https://doi.org/10.1145/2994258.2994272>
- Sebastian Martin, Peter Kaufmann, Mario Botsch, Eitan Grinspun, and Markus Gross. 2010. Unified Simulation of Elastic Rods, Shells, and Solids. 29, 4, Article 39 (July 2010), 10 pages. <https://doi.org/10.1145/1778765.1778776>
- Jonàs Martínez, Mélina Skouras, Christian Schumacher, Samuel Hornus, Sylvain Lefebvre, and Bernhard Thomaszewski. 2019. Star-Shaped Metrics for Mechanical Metamaterial Design. *ACM Trans. Graph.* 38, 4 (July 2019), Article No. 82 :1–13. <https://doi.org/10.1145/3306346.3322989>
- Marvelous Designer. 2010. <http://www.marvelousdesigner.com>.
- Hammad Mazhar, Toby Heyn, Dan Negrut, and Alessandro Tasora. 2015. Using Nesterov’s Method to Accelerate Multibody Dynamics with Friction and Contact. *ACM Trans. Graph.* 34, 3, Article 32 (May 2015), 14 pages. <https://doi.org/10.1145/2735627>
- Aleka McAdams, Andrew Selle, Kelly Ward, Eftychios Sifakis, and Joseph Teran. 2009. Detail preserving continuum simulation of straight hair. *ACM Trans. Graph. (Proc. ACM SIGGRAPH'09 )* 28, 3 (2009), 1–6. <https://doi.org/10.1145/1531326.1531368>
- Dereck S. Meek and Desmond J. Walton. 2000. On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. *Computer Aided Geometric Design* 17, 6 (2000), 521–543. [https://doi.org/10.1016/S0167-8396\(00\)00006-6](https://doi.org/10.1016/S0167-8396(00)00006-6)
- Anthony G. M. Michell. 1899. Elastic stability of long beams under transverse forces. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 48, 292 (1899), 298–309.
- Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A. Otaduy, and Steve Marschner. 2012. Data-Driven Estimation of Cloth Simulation Models. *Comp. Graph. Forum* (May 2012), 519–528. <https://doi.org/10.1111/j.1467-8659.2012.03031.x>
- Jay T. Miller, Arnaud Lazarus, Basime Audoly, and Pedro Reis. 2014. Shapes of a Suspended Curly Hair. *Physical Review Letters* 112, 6 (2014).



- Matthew Moore and Jane Wilhelms. 1988. Collision detection and response for computer animation. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'88)*. 289–298.
- Jean-Jacques Moreau. 1988. Unilateral contact and dry friction in finite freedom dynamics. *Nonsmooth mechanics and applications*, CISM Courses Lect. 302, 1-82 (1988)..
- Jean-Jacques Moreau. 2000. Contact and friction in the dynamics of rigid body system. *Revue Européenne des éléments finis*, 9, 9-28 (2000).
- Jean-Jacques Moreau. 2006. Facing the plurality of solutions in nonsmooth mechanics. In *Nonsmooth/Nonconvex Mechanics with Applications in Engineering*, C.C. Baniotopoulos (Ed.). Thessalonique, Greece, 3–12. <https://hal.archives-ouvertes.fr/hal-01824588>
- Matthias Müller, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. 2007. Position based dynamics. *Journal of Visual Communication and Image Representation* 18, 2 (2007), 109–118. <https://doi.org/10.1016/j.jvcir.2007.01.005>
- Paul M. Naghdi. 1972. The Theory of Shells and Plates In *Handbuch der Physik*, Vol. VIa/2 C. Truesdell.
- Rahul Narain, Armin Samii, and James F. O'Brien. 2012. Adaptive Anisotropic Remeshing for Cloth Simulation. *ACM Trans. Graph.* 31, 6, Article 152 (Nov. 2012), 10 pages. <https://doi.org/10.1145/2366145.2366171>
- Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization*. Springer.
- Tony J. Nolan. 1971. Computer-Aided Design of Developable Hull Surfaces. *Marine Technology and SNAME News* 8, 02 (04 1971), 233–242. <https://doi.org/10.5957/mt1.1971.8.2.233> arXiv:<https://onepetro.org/MTSN/article-pdf/8/02/233/2202947/sname-mtsn-1971-8-2-233.pdf>
- Miguel A. Otaduy, Rasmus Tamstorf, Denis Steinemann, and Markus H. Gross. 2009. Implicit Contact Handling for Deformable Objects. *Computer Graphics Forum (Proc. Eurographics'09)* 28, 2 (apr 2009). <http://www.gmr.v.es/Publications/2009/OTSG09>
- Matthew Overby, George E. Brown, Jie Li, and Rahul Narain. 2017. ADMM  $\supseteq$  Projective Dynamics: Fast Simulation of Hyperelastic Models with Dynamic Constraints. *IEEE Trans. Vis. and Comp. Graph.* 23, 10 (Oct 2017), 2222–2234.
- Julian Panetta, Florin Isvoranu, Tian Chen, Emmanuel Siefert, Benoit Roman, and Mark Pauly. 2021. Computational Inverse Design of Surface-based Inflatables. *ACM Trans. Graph.* (2021).

- J. Panetta, M. Konaković-Luković, F. Isvoranu, E. Bouleau, and M. Pauly. 2019. X-Shells: A New Class of Deployable Beam Structures. *ACM Trans. Graph.* 38, 4, Article 83 (July 2019), 15 pages. <https://doi.org/10.1145/3306346.3323040>
- Dimitrios I. Papadimitriou and Kyriakos C. Giannakoglou. 2008. Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems. *International Journal for Numerical Methods in Fluids* 56, 10 (2008), 1929–1943. <https://doi.org/10.1002/fld.1584> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1584>
- Davide Pellis, Martin Kilian, Felix Dellinger, Johannes Wallner, and Helmut Pottmann. 2019. Visual Smoothness of Polyhedral Surfaces. *ACM Trans. Graph.* 38, 4, Article 31 (July 2019), 11 pages. <https://doi.org/10.1145/3306346.3322975>
- Jesús Pérez, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, José A. Canabal, Robert Sumner, and Miguel A. Otaduy. 2015. Design and Fabrication of Flexible Rod Meshes. *ACM Trans. Graph.* 34, 4, Article 138 (July 2015), 12 pages. <https://doi.org/10.1145/2766998>
- Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. 2017. ClothCap: Seamless 4D Clothing Capture and Retargeting. *ACM Transactions on Graphics* 36, 4, Article 73 (July 2017), 15 pages. <https://doi.org/10.1145/3072959.3073711>
- Serban D. Porumbescu, Brian Budge, Louis Feng, and Kenneth I. Joy. 2005. Shell Maps. *ACM Transactions on Graphics* 24, 3 (July 2005), 626–633. <https://doi.org/10.1145/1073204.1073239>
- Xavier Provot. 1997. Collision and self-collision handling in cloth model dedicated to design garments. In *Computer Animation and Simulation '97*, Daniel Thalmann and Michiel van de Panne (Eds.). Springer Vienna, Vienna, 177–189.
- Michael Rabinovich, Tim Hoffmann, and Olga Sorkine-Hornung. 2018a. Discrete Geodesic Nets for Modeling Developable Surfaces. *ACM Transactions on Graphics* 37, 2 (2018).
- Michael Rabinovich, Tim Hoffmann, and Olga Sorkine-Hornung. 2018b. The Shape Space of Discrete Orthogonal Geodesic Nets. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)* 37, 6 (2018).
- Laks Raghupathi and François Faure. 2006. QP-Collide: A New Approach to Collision Treatment. In *Journées du groupe de travail Animation et Simulation, GTAS 06, June, 2006 (Annual French Working group on Animation and Simulation)*. Institut de Recherche en Informatique de Toulouse, Toulouse, France, 91–101.

- Vivek Ramachandran, Michael D. Bartlett, James Wissman, and Carmel Majidi. 2016. Elastic instabilities of a ferroelastomer beam for soft reconfigurable electronics. *Extreme Mechanics Letters* 9 (2016), 282–290. <https://doi.org/10.1016/j.eml.2016.08.007>
- Abdullah-Haroon Rasheed, Victor Romero, Florence Bertails-Descoubes, Stefanie Wuhrer, Jean-Sébastien Franco, and Arnaud Lazarus. 2020. Learning to Measure the Static Friction Coefficient in Cloth Contact. In *CVPR 2020 - IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle, United States, 9909–9918. <https://doi.org/10.1109/CVPR42600.2020.00993>
- Stéphane Redon, Abderrahmane Kheddar, and Sabine Coquillart. 2002. Fast Continuous Collision Detection between Rigid Bodies. In *Computer Graphics Forum (Proc. Eurographics )*, Vol. 21. 279–288. Issue 3.
- Pedro M. Reis. 2015. A Perspective on the Revival of Structural (In)Stability With Novel Opportunities for Function: From Buckliphobia to Buckliphilia. *Journal of Applied Mechanics* 82, 11 (Sept. 2015). <https://doi.org/10.1115/1.4031456> 111001.
- Eric Reissner. 1995. The problem of lateral buckling of cantilever plates. *ZAMM - Journal of Applied Mathematics and Mechanics* 8 (1995), 615–621.
- Yingying Ren, Julian Panetta, Tian Chen, Florin Isvoranu, Samuel Poincloux, Christopher Brandt, Alison Martin, and Mark Pauly. 2021. 3D Weaving with Curved Ribbons. *ACM Trans. Graph.* (2021).
- Neil M. Ribe. 2004. Coiling of Viscous Jets. *Proceedings: Mathematical, Physical and Engineering Sciences* 460, 2051 (2004), 3223–3239. <http://www.jstor.org/stable/4143202>
- Victor Romero, Florence Bertails-Descoubes, Alexandre Derouet-Jourdan, and Arnaud Lazarus. 2018. Inverse design of a suspended Kirchhoff rod: From theory to practice. In *ESMC10 2018 - 10th European Solid Mechanics Conference*, Vol. 25. Bologna, Italy, 184302 – 184302. <https://hal.inria.fr/hal-01778474>
- Victor Romero, Mickaël Ly, Abdullah Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. 2021. Physical validation of simulators in Computer Graphics: A new framework dedicated to slender elastic structures and frictional contact. *ACM Trans. Graph.* (2021).
- Kenneth Rose, Alla Sheffer, Jamie Wither, Marie-Paule Cani, and Boris Thibert. 2007. Developable Surfaces from Arbitrary Sketched Boundaries. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (Barcelona, Spain) (SGP '07)*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 163–172. <http://dl.acm.org/citation.cfm?id=1281991.1282014>

- Tomohiko G. Sano, Tetsuo Yamaguchi, and Hirofumi Wada. 2017. Slip Morphology of Elastic Strips on Frictional Rigid Substrates. *Physical Review Letters* 118, 17 (2017), 178001–5.
- Nikolas Schmitt, Martin Knuth, Jan Bender, and Arjan Kuijper. 2013. Multilevel Cloth Simulation using GPU Surface Sampling. <https://doi.org/10.2312/PE.vriphys.vriphys13.001-010>
- Christian Schumacher, Steve Marschner, Markus Gross, and Bernhard Thomaszewski. 2018. Mechanical Characterization of Structured Sheet Materials. *ACM Trans. Graph.* 37, 4, Article 148 (July 2018), 15 pages. <https://doi.org/10.1145/3197517.3201278>
- R. T. Shield. 1992. Bending of a beam or wide strip. *Quarterly Journal of Mechanics and Applied Mathematics* 45, 4 (1992), 567–573.
- Antonio Signorini. 1959. Questioni di elasticità non linearizzata e semilinearizzata. *Rend. Mat. Appl* 18, 5 (1959), 95–139.
- Morten Silcowitz, Sarah Niebe, and Kenny Erleben. 2009. Nonsmooth Newton Method for Fischer function reformulation of contact force problems for interactive rigid body simulation. 105–114. <https://doi.org/10.2312/PE/vriphys/vriphys09/105-114>
- Mélina Skouras, Bernhard Thomaszewski, Bernd Bickel, and Markus H. Gross. 2012. Computational Design of Rubber Balloons. *Computer Graphics Forum (Proc. Eurographics)* (2012).
- Mélina Skouras, Bernhard Thomaszewski, Stelian Coros, Bernd Bickel, and Markus Gross. 2013. Computational design of actuated deformable characters. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 32, 4 (July 2013), 82:1–82:10.
- Mélina Skouras, Bernhard Thomaszewski, Peter Kaufmann, Akash Garg, Bernd Bickel, Eitan Grinspun, and Markus H. Gross. 2014. Designing Inflatable Structures. *ACM Transactions on Graphics* 33, 4, Article 63 (July 2014), 10 pages. <https://doi.org/10.1145/2601097.2601166>
- Breannan Smith, Danny M Kaufman, Etienne Vouga, Rasmus Tamstorf, and Eitan Grinspun. 2012. Reflections on Simultaneous Impact. *ACM Trans. Graph.* 31, 4, Article 106 (July 2012), 12 pages. <https://doi.org/10.1145/2185520.2185602>
- Yousuf Soliman, Dejan Slepčev, and Keenan Crane. 2018. Optimal Cone Singularities for Conformal Flattening. *ACM Trans. Graph.* 37, 4 (2018).

- Justin Solomon, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. 2012. Flexible Developable Surfaces. *Comput. Graph. Forum* 31, 5 (Aug. 2012), 1567–1576. <https://doi.org/10.1111/j.1467-8659.2012.03162.x>
- Oded Stein, Eitan Grinspun, and Keenan Crane. 2018. Developability of Triangle Meshes. *ACM Trans. Graph.* 37, 4, Article 77 (July 2018), 14 pages. <https://doi.org/10.1145/3197517.3201303>
- David E. Stewart and Jeffrey C. Trinkle. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction. *ijnme* 39, 15 (1996).
- K.Y. Sze, X.H. Liu, and S.H. Lo. 2004. Popular benchmark problems for geometric nonlinear analysis of shells. *Finite Elements in Analysis and Design* 40, 11 (2004), 1551 – 1569. <https://doi.org/10.1016/j.finel.2003.11.001>
- Rasmus Tamstorf and Eitan Grinspun. 2013. Discrete Bending Forces and Their Jacobians. 75, 6 (Nov. 2013), 362–370. <https://doi.org/10.1016/j.gmod.2013.07.001>
- Chengcheng Tang, Pengbo Bo, Johannes Wallner, and Helmut Pottmann. 2016. Interactive Design of Developable Surfaces. *ACM Trans. Graph.* 35, 2, Article 12 (Jan. 2016), 12 pages. <https://doi.org/10.1145/2832906>
- Kai Tang and Charlie C. L. Wang. 2005. Modeling Developable Folds on a Strip. *Journal of Computing and Information Science in Engineering* 5, 1 (03 2005), 35–47. <https://doi.org/10.1115/1.1804206>
- Min Tang, Tongtong Wang, Zhongyuan Liu, Ruofeng Tong, and Dinesh Manocha. 2018. I-Cloth: Incremental Collision Handling for GPU-Based Interactive Cloth Simulation. *ACM Trans. Graph.* 37, 6, Article 204 (Dec. 2018), 10 pages. <https://doi.org/10.1145/3272127.3275005>
- Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. 1987. Elastically Deformable Models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. Association for Computing Machinery, New York, NY, USA, 205–214. <https://doi.org/10.1145/37401.37427>
- Stephen Timoshenko. 1953. *History of strength of materials*. The McGraw-Hill Book Company.
- Emmanuel Turquin, Jamie Wither, Laurence Boissieux, Marie-Paule Cani, and John F. Hughes. 2007. A Sketch-Based Interface for Clothing Virtual Characters. *IEEE Comput. Graph. Appl.* 27, 1 (Jan. 2007), 72–81. <https://doi.org/10.1109/MCG.2007.1>

- Christopher D. Twigg and Zoran Kačić-Alesić. 2011. Optimization for sag-free simulations. In *ACM SIGGRAPH - EG Symposium on Computer Animation (SCA'11)* (Vancouver, British Columbia, Canada). ACM-EG SCA, 225–236. <https://doi.org/10.1145/2019406.2019437>
- Erva Ulu, James McCann, and Levent B. Kara. 2019. Structural Design Using Laplacian Shells. *Computer Graphics Forum* 38, 5 (2019), 85–98. <https://doi.org/10.1111/cgf.13791>
- Nobuyuki Umetani, Danny Kaufman, Takeo Igarashi, and Eitan Grinspun. 2011. Sensitive Couture for Interactive Garment Editing and Modeling. *ACM Transactions on Graphics* 30, 4 (2011). <http://www.cs.columbia.edu/cg/SC/>
- Félix Vanneste, Olivier Goury, Jonas Martinez, Sylvain Lefebvre, Herve Delingette, and Christian Duriez. 2020. Anisotropic soft robots based on 3D printed meso-structured materials: design, modeling by homogenization and simulation. *IEEE Robotics and Automation Letters* 5, 2 (Jan. 2020), 2380–2386. <https://doi.org/10.1109/LRA.2020.2969926>
- Pascal Volino, Martin Courshesnes, and Nadia Magnenat-Thalmann. 1995. Versatile and Efficient Techniques for Simulating Cloth and Other Deformable Objects. In *Computer Graphics Proceedings (Proc. ACM SIGGRAPH'95)*, Robert Cook (Ed.). Addison Wesley, 137–144.
- Huamin Wang. 2015. A Chebyshev Semi-Iterative Approach for Accelerating Projective and Position-Based Dynamics. *ACM Trans. Graph.* 34, 6, Article 246 (Oct. 2015), 9 pages.
- Huamin Wang. 2018. Rule-free sewing pattern adjustment with precision and efficiency. *ACM Transactions on Graphics* 37, 4, Article 53 (Aug. 2018), 14 pages.
- Huamin Wang, Ravi Ramamoorthi, and James F. O'Brien. 2011. Data-driven elastic models for cloth: modeling and measurement. *ACM Transactions on Graphics (SIGGRAPH 2011)* 30, 4, Article 71 (Aug. 2011), 12 pages. <https://doi.org/10.1145/2010324.1964966>
- Ziqi Wang, Peng Song, and Mark Pauly. 2021. State of the Art on Computational Design of Assemblies with Rigid Parts. *Computer Graphics Forum* (2021). <https://doi.org/10.1111/cgf.142660>
- Zhendong Wang, Longhua Wu, Marco Fratarcangeli, Min Tang, and Huamin Wang. 2018. Parallel Multigrid for Nonlinear Cloth Simulation. *Computer Graphics Forum* 37 (10 2018), 131–. <https://doi.org/10.1111/cgf.13554>

- Max Wardetzky. 2007. *Discrete Differential Operators on Polyhedral Surfaces - Convergence and Approximation*. Ph.D. Dissertation. Freie Universität Berlin.
- Max Wardetzky, Miklós Bergou, David Harmon, Denis Zorin, and Eitan Grinspun. 2007. Discrete Quadratic Curvature Energies. *Comput. Aided Geom. Des.* 24, 8–9 (Nov. 2007), 499–518. <https://doi.org/10.1016/j.cagd.2007.07.006>
- Nicholas J. Weidner, Kyle Piddington, David I. W. Levin, and Shinjiro Sueda. 2018. Eulerian-on-Lagrangian Cloth Simulation. *ACM Trans. Graph.* 37, 4, Article 50 (July 2018), 11 pages. <https://doi.org/10.1145/3197517.3201281>
- Clarisse Weischedel. 2012. A discrete geometric view on shear-deformable shell models.
- Ryan White, Keenan Crane, and David A. Forsyth. 2007. Capturing and animating occluded cloth. *ACM Transactions on Graphics* 26, 3, Article 34 (July 2007). <https://doi.org/10.1145/1276377.1276420>
- Thomas Wolf, Victor Cornillère, and Olga Sorkine-Hornung. 2021. Physically-based Book Simulation with Freeform Developable Surfaces. *Computer Graphics Forum (proceedings of EUROGRAPHICS 2021)* 40, 2 (2021).
- Zangyueyang Xian, Xin Tong, and Tiantian Liu. 2019. A Scalable Galerkin Multigrid Method for Real-Time Simulation of Deformable Objects. *ACM Trans. Graph.* 38, 6, Article 162 (Nov. 2019), 13 pages. <https://doi.org/10.1145/3355089.3356486>
- Jinlong Yang, Jean-Sébastien Franco, Franck Hetroy-Wheeler, and Stefanie Wuhler. 2018a. Analyzing Clothing Layer Deformation Statistics of 3D Human Motions. In *ECCV 2018 - European Conference on Computer Vision (Lecture Notes in Computer Science, Vol. 11211)*. Springer, Munich, Germany, 245–261. [https://doi.org/10.1007/978-3-030-01234-2\\_15](https://doi.org/10.1007/978-3-030-01234-2_15)
- Shan Yang, Junbang Liang, and Ming C. Lin. 2017. Learning-based cloth material recovery from video. In *Proceedings of the IEEE International Conference on Computer Vision*. 4383–4393.
- Shan Yang and Ming C. Lin. 2016. MaterialCloning: Acquiring elasticity parameters from images for medical applications. *IEEE Trans. on Visualization and Computer Graphics* 22, 9 (2016), 2122–2135.
- Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C. Lin. 2018b. Physics-Inspired Garment Recovery from a Single-View Image. *ACM Trans. Graph.* 37, 5, Article 170 (Nov. 2018), 14 pages. <https://doi.org/10.1145/3026479>

- Bo Zhu, Mélina Skouras, Desai Chen, and Wojciech Matusik. 2017. Two-Scale Topology Optimization with Microstructures. *ACM Trans. Graph.* 36, 4, Article 120b (July 2017), 16 pages. <https://doi.org/10.1145/3072959.3095815>
- Simon Zimmermann, Roi Poranne, James M. Bern, and Stelian Coros. 2019. PuppetMaster: Robotic Animation of Marionettes. *ACM Trans. Graph.* 38, 4, Article 103 (July 2019), 11 pages. <https://doi.org/10.1145/3306346.3323003>