Physical validation of simulators in Computer Graphics: A new framework dedicated to slender elastic structures and frictional contact


To cite this version:


HAL Id: hal-03217459
https://inria.hal.science/hal-03217459v2
Submitted on 7 May 2021

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.
Physical validation of simulators in Computer Graphics: A new framework dedicated to slender elastic structures and frictional contact

VICTOR ROMERO, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France
MICKAËL LY, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France
ABDULLAH HAROON RASHEED, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France
RAPHAËL CHARRONDIERE, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France
ARNAUD LAZARUS, Sorbonne Université, CNRS, Institut Jean Le Rond d’Alembert, UMR 7190, France
SÉBASTIEN NEUKIRCH, Sorbonne Université, CNRS, Institut Jean Le Rond d’Alembert, UMR 7190, France
FLORENCE BERTAILS-DESCOUBES, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, France

We introduce a selected set of protocols inspired from the Soft Matter Physics community in order to validate Computer Graphics simulators of slender elastic structures possibly subject to dry frictional contact. Although these simulators were primarily intended for feature film animation and visual effects, they are more and more used as virtual design tools for predicting the shape and deformation of real objects; hence the need for a careful, quantitative validation. Our tests, experimentally verified, are designed to evaluate carefully the predictability of these simulators on various aspects, such as bending elasticity, bend-twist coupling, and frictional contact. We have passed a number of popular codes of Computer Graphics through our benchmarks by defining a rigorous, consistent, and as fair as possible methodology. Our results show that while some popular simulators for plates/shells and frictional contact fail even on the simplest scenarios, more recent ones, as well as well-known codes for rods, generally perform well and sometimes even better than some reference commercial tools of Mechanical Engineering. To make our validation protocols easily applicable to any simulator, we provide an extensive description of our methodology, and we shall distribute all the necessary model data to be compared against.

CCS Concepts: • Computing methodologies → Animation; Physical simulation;

Additional Key Words and Phrases: Slender elastic structures, dry frictional contact, code benchmarking, experimental validation

ACM Reference Format:

1 INTRODUCTION

In the last decades, a number of numerical simulators have been provided by the Computer Graphics (CG) community to compute the statics and dynamics of slender elastic structures, possibly subject to frictional contact. Initially, these codes were primarily intended for the movie industry where the important evaluation criteria are visual realism and artistic authoring. Such simulators have been...
successfully leveraged to animate, for instance, hair, fur and cloth of virtual characters [Baraff and Witkin 1998; Baraff et al. 2003; Daviet 2020; Daviet et al. 2011; Kaufman et al. 2014; McAdams et al. 2009].

Yet nowadays, the growing demand of the movie industry for mixing virtual and real shots in a seamless fashion constantly pushes forward the limits of simulation in terms of realism, up to some quantitative matching with reality [Museth 2020]. Besides, with the advance of computational fabrication, CG simulators are now more and more used for virtual prototyping tasks, where the ultimate goal is not to make images, but to build actual objects with desired properties. Among recent developments in this area, we can cite cloth material identification and prototyping [Bartle et al. 2016; Wang 2018; Yang et al. 2017], architectural design [Gavrilii et al. 2020; laccone et al. 2019; Panetta et al. 2019], fabrication of soft robots [Coevoet et al. 2019; Vanneste et al. 2020], and the exploration of new meta-materials [Guseinov et al. 2020; Martínez et al. 2019; Schumacher et al. 2018]. Furthermore, in various communities ranging from Computer Vision to Biomedicine, a number of works now study how to learn physics from simulated data in order to predict reality [Liang et al. 2019; Yang and Lin 2016]; for training, CG simulators are often preferred to commercial, specialized FEM softwares, because in general the former offer a free, lighter, and more user-friendly environment compared to the latter.

Common to all these new applications is the concern to predict reality in a quantitative way – this is in contrast with traditional image making which instead focuses on capturing the real world qualitatively. Unfortunately, tools to measure the physical correctness and accuracy of simulators remain scarce in the Computer Graphics community. As a result, there is no guarantee that current CG simulators offer the predictability level required by the target application. While not critical in movie making, wrong or unreliable predictions may yield serious consequences in some more sensitive domains like architectural design or healthcare.

Our study is focused on static scenarios involving slender elastic structures (rods and plates) subject to clamped-free boundary conditions, gravity, and possibly natural curvature and frictional contact. The methodology we propose here is however general and could be extended to other situations. We hope our approach will be useful in the future to all the people who wish to design a new model or reimplement an existing one, by allowing them to check the physical validity of their model or/and of their code.

2 RELATED WORK

Unlike Mechanical Engineering which has a long tradition in physical and numerical validation, Computer Graphics is not used to evaluating its models quantitatively in a systematic manner. A first reason is that for movie and game making – the historical applications of Computer Graphics – many subjective criteria enter the loop, such as visual perception, authoring, or aesthetics; all of them remaining extremely hard to assess. A second reason is that phenomena of interest have long differed between Mechanical Engineering and Computer Graphics. While Computer Graphics was from its early stage interested in large displacements of structures, buckling, tearing, entanglement, and dynamics, all these phenomena were long considered as undesirable behaviors in Mechanical Engineering, thus restraining the search for corresponding numerical models and appropriate validation tools.

Since these early times, both fields have considerably evolved. While many researchers from Computer Graphics have turned from pure visual effect to material design applications – the latter requiring more predictability than the former – a sub-community of Mechanical Engineering now considers so-called extreme mechanics [Krieger 2012] as a new fruitful direction for research [Reis 2015]. However, validation methods applicable to the simulation of complex (or extreme) phenomena, such as post-buckled rods/plates and frictional contact, remain scarce. We review here the main practices used in both fields.

Overly simple scenarios. For planar rods, elementary tests can be performed under the small deflection assumption, comparing code results against solutions to the linear Euler-Bernoulli beam equations. Various scenarios can be considered, such as the linear cantilever beam experiment (clamped-free or clamped-clamped boundary conditions) or the multiple point bending test (horizontal beam subject to multiple vertical pressure points), which all generate some simple analytical solutions [Timoshenko 1953]. For instance, Martin et al. [2010] compare their unified model for rods and shells to the linear cantilever test, and Panetta et al. [2019] validate their rod model implementation using the four point bending test, characterized by a constant bending strain between the two downwards pressure points. However, such tests are not valid anymore for large displacements of the structure, which is the case we are interested in. In this paper we enrich the well-known linear cantilever scenario by considering nonlinear clamped rods and plates subject to possibly large gravitational forces (Cantilever test and Lateral Buckling test, respectively), as well as bending/twisting instabilities emerging for naturally curved structures (Bend-Twist test).

For frictional contact, a few authors have devised some simple tests like the falling drape over a plane [Li et al. 2018], which generates a constant velocity motion but only tests non-penetration as well as sliding friction. We go one step further by testing the sticking-sliding threshold (Stick-Slip test), independently of the material properties.

Qualitative only scenarios. To increase the complexity of validation scenarios, researchers in Computer Graphics soon departed from quantitative evaluation and turned instead to a qualitative assessment of their models. Progressively, a number of tests like cloth on rotating sphere [Bridson et al. 2002], the cloth funnel [Harmon et al. 2008], the reef knot [Harmon et al. 2009], or rod plec- tonemes [Sprillmann and Teschner 2007], became popular and were adopted for subsequent benchmarking studies [Bergou et al. 2008, Sec. 9.2] [Li et al. 2018, 2020], even though these remained merely
well-defined scenarios have been defined over time [Sze et al. 2004], while adding a new degree of freedom to the system (naturally curly (KES) [Kawabata and Niwa 1989] for cloth, or by creating new protocols suited for the model used [Wang et al. 2011], the aim is to fit model parameters to measurements so that the geometric output of the code matches the geometric output of the experiment. The underlying model does not need to be entirely physical but should possess enough parameters so that it can capture and reproduce the diverse phenomena at play in the experiments. This strategy has been widely used for the fast simulation of soft tissues [Bickel et al. 2009], the characterization of garments properties [Clyde et al. 2017; Miguel et al. 2012; Wang et al. 2011], or the design of inflatable structures [Skouras et al. 2012, 2014] and materials with desired properties [Bickel et al. 2010]. Of course, the next stage amounts to developing completely blind models through machine learning.

However, validation is much more than fitting: it requires setting physical parameters to their value measured in an independent fashion. Compared to validated physical models, fitted models are likely to possess a limited degree of predictability. Interestingly, we show in this paper that while our methodology is dedicated to the evaluation of physical numerical models, it can also be exploited to assess the potential of a fitted or partially fitted model.

**Particular dimensional studies.** In Mechanical Engineering, probably the most popular technique to validate a code quantitatively is to compare it against particular physical experiments spread across the literature. For instance, in the mechanics of plates and shells, well-defined scenarios have been defined over time [Sze et al. 2004], such as the sheared hemispherical shell or the pinched cylindrical shell. Such benchmark problems can be complex, but in general they rely on a particular, dimensional choice of parameters. As a result, although the test can be hard to set up in practice, it only assesses a particular setting of the code. Moreover, with these scenarios it can be tricky to compare various, disparate codes in a robust way.

**Towards a compact quantitative validation.** In contrast to particular dimensional benchmarks, each one of our tests includes a dimensionless scaling law which compactly represents several regimes of a complex physical experiment, hence describing a wide parameter range. Such a law can be verified against the output of any simulator able to reproduce the corresponding experimental setup, should the simulator be dimensionalized or dimensionless. This verification actually assesses many facets of the simulator at once.

Our main inspiration originates from the work of Rasheed et al. [2020] in Computer Vision. In their study, they train a neural network on a cloth simulator in order to predict friction coefficients in real cloth motion. To make sure their network learns from realistic data, they first validate their simulator against the stick-slip scaling law recently proposed in Soft Matter Physics by Sano et al. [2017].

We build on Rasheed et al.‘s validation idea and significantly extend it by proposing a set of four measurement protocols (among which is Sano et al.’s protocol) as new tests for code validation. We revisit in depth each one of these tests, reconstruct the corresponding scaling law accurately (adapting an existing one and devising a new one), and validate them experimentally.

### 3 CONTRIBUTIONS

We evaluate the physical realism of popular simulators in Computer Graphics for simulating thin elastic rods and plates, possibly subject to contact and dry friction. To this end, we carefully select and adapt four measurement protocols from the Soft Matter Physics community, never introduced to Computer Graphics so far (Section 4). These protocols, namely Cantilever, Bend-Twist, Lateral Buckling, and Stick-Slip, are meant to assess various physical aspects of a simulator, such as bending, bend-twist coupling, or frictional contact (see Figure 1). Note that the Cantilever test we introduce in this paper is a dimensionless test, meaning that it is able to characterize full deformation regimes (depending on a combination of parameters), and not just particular deformations based on specific parameters (e.g. a particular material). In this respect, the Cantilever test is thus much richer than the dimensional cantilever test usually leveraged in Computer Graphics for validation or parameter fitting [Miguel et al. 2012; Wang et al. 2011].

We thoroughly derive scaling laws associated to each protocol (the one for Lateral Buckling being new), and perform careful experiments to validate theoretical predictions. These scaling laws are made to be sufficiently simple and rich for revealing the capacity of a given numerical model to reproduce and predict reality. We then select a number of popular codes from Computer Graphics as well as two reference codes from Mechanical Engineering (Section 5) and define a thorough methodology for benchmarking (Section 6). We give a (positive or negative) score to each code evaluated on all applicable protocols. Through a careful analysis of our results, we show that our tests help discriminate the codes sharply (Section 7).

We finally address limitations of our approach (Section 8), before concluding. Our protocols and data are made available to the community as supplementary materials, so that a thorough evaluation of additional codes can be performed easily by other researchers.

**Caveat: What this study is not about.** We do not evaluate performance nor user-friendliness of a code, even though some rough speed estimation of use and run are mentioned for merely indicative purposes. Neither do we evaluate the versatility of a code. Some codes are specific to one task (e.g. a static planar rod simulator), others might be more general (e.g. a dynamic cloth simulator). The range of phenomena which can be simulated by a code is not taken into account in the overall score attributed to a code.
4 SELECTION & DESIGN OF PHYSICAL PROTOCOLS

We present here the four protocols that will serve to construct our benchmarking study. All the details necessary to reproduce our experiments are available in our supplementary document (Section 4).

4.1 Scaling laws

Dimensional analysis [Buckingham 1914] and scaling laws are useful in Physics to understand the nature of phenomena. It is also a powerful tool for experimentalists who use it to measure unknown parameters. Yet we show here that scaling laws also have a high potential in terms of code validation. Consider for example the period

\[ T_{pd} = 2\pi \sqrt{\ell/g} \]

of the small-amplitude oscillations of a pendulum of length \( \ell \), where \( g \) is the gravity acceleration. If one were to test a simulator for the dynamics of such a pendulum, one could either choose several values of \( \ell \) and \( g \) and compare the numerical period to the formula \( T_{pd} = 2\pi \sqrt{\ell/g} \), or plot the numerical period as a function of \( \ell/g \) and verify that it follows a square-root law. The second method has several advantages: (i) it illustrates that two different numerical tests should yield the same period provided the ratio \( \ell/g \) is the same in both tests, (ii) it can detect some discrepancies in a simulator that would fail in a certain parameter range (small \( \ell/g \) for example), and (iii) it allows one to compare simulators working with dimensional parameters (i.e. having actual units, for example meters for \( \ell \)) to simulators working with dimensionless parameters (for example using the dimensionless time \( \Gamma = t/\sqrt{\ell/g} \) and hence finding the dimensionless period \( T_{pd} = 2\pi \)). In conclusion, using scaling laws allows us to test, on a unique set-up, a complete family of parameter values with all points collapsing on a single curve, which we shall refer to as the master curve in this paper.

Building a set-up with a simulator is a long process and it is more efficient if this set-up can serve to test multiple configurations of parameters at once. Once the code for reproducing a given experiment has been written, it is relatively straightforward to launch a large number of simulations in batches to sample the parameter range automatically. Having the results collapsing on a master curve is then very rich in terms of information. The readability of the results is considerably higher than in the case of a particular dimensional study where one would compare, for each run, 2 simulations or 1 simulation and 1 experiment.

4.2 The Cantilever test

We discuss a 2D test dating back to the 30’s [Bickley 1934], which is commonly used by the Soft Matter community for inferring mechanical parameters of rods and ribbons, see e.g. Duclaux [2006] or Fargette [2017]. The main advantage of this test is that, from purely geometrical considerations and measurements, one can estimate, with good accuracy, the elastic behaviour of the material. Hereafter we show that the same geometrical considerations can be harnessed either for intensive validation of physics-based simulators, or for calibration of non-physics-based simulators.

The soft cantilever consists of a slender, naturally straight rod or ribbon, clamped horizontally, and deformed in 2D by the action of gravity, see Figure 2, left. For a rod of radius \( r \), the parameters involved in the problem are the rod bending rigidity \( (EI) \), its material density \( (\rho) \), its length \( (L) \), the area \( A = \pi r^2 \) of its cross-section, and the gravity acceleration \( (g) \). Here \( E \) is the Young elastic modulus and \( I = \frac{\pi}{4} r^4 \) the second moment of area of the cross-section of the rod. It turns out that one can construct a unique length scale for this problem, namely the gravito-bending length \( L_{gb} = \sqrt{EI/(\rho Ag)} \), which compares the resistance to bending with the gravitational force. Thus, for an elastic object deformed by gravity with length \( L \), its equilibrium shape is effectively determined solely by the dimensionless gravito-bending parameter \( \Gamma = (L/L_{gb})^3 \), which reads

\[ \Gamma = \frac{DAgL^3}{EI} \]

(1)

Note that in the case of ribbons, when the cross-section is a rectangle of width \( w \) and thickness \( h \), with \( h \ll w \), the Young modulus \( E \) becomes \( E^* = E/(1-\nu^2) \), where \( \nu \) is the Poisson ratio of the material [Shield 1992]. As a result, \( \Gamma \) is simply replaced with \( \Gamma^* = (1 - \nu^2) \Gamma \),

\[ \Gamma^* = \frac{DA^*gL^3}{Dw} \]

(2)

where \( A^* \) is the area of the rectangular cross-section, \( A^* = wh \), and \( D \) is the bending rigidity of the plate, \( D = Er^3/(12(1 - \nu^2)) \).

The parameter \( \Gamma \) gives an immediate notion of the expected behaviour for the deformation of an elastic object. Large values of \( \Gamma \) lead to large deformations, and conversely low values yield small deformations. However, the main advantage of this test is that one can quantitatively compare simulations with experimental and analytical results for a broad range of mechanical and geometrical parameters. For such comparison, here we chose the aspect ratio of the final shape, \( H/W \) (see Figure 2), and show that this aspect ratio is an accurate indicator of the quality of the simulation: indeed the mapping between \( \Gamma \) and \( H/W \) is unique (strictly monotonic), so there is no need to compare global equilibrium shapes or curvature distributions.

Master curve: solution of the planar elastica. The final shape for rods and ribbons is governed by the planar elastica equations (i.e. the Kirchhoff equations for 2D rods, see [Landau and Lifshitz 1959]). The equilibrium for the internal force \( F \) and moment \( M \) for an infinitesimal variation of the arc length \( s \) read \( dF/ds + M_{ext} = 0 \) and \( dM/ds + t \times F = 0 \), with \( t \) the unit tangent to the rod. Free-end boundary conditions reads \( F(s = L) = 0 = M(s = L) \). The constitutive relation between the bending moment and the curvature \( d\theta/ds \) reads \( M = Eld\theta/ds \), with \( \theta \) the angle between the horizontal and the tangent, see Figure 2, left. When gravity is the only external force, \( M_{ext} = A\rho g \epsilon_y \), yielding \( F = A\rho g (L - s) \epsilon_y \) and \( dM/ds + A\rho g (L - s) \cos \theta \epsilon_z = 0 \). Introducing dimensionless variables \( \tilde{s} = s/L, \tilde{x} = x/L, \) and \( \tilde{y} = y/L \), the boundary value problem to be solved finally reads

\[ \frac{d^2\theta}{ds^2} + \Gamma (1 - \tilde{s}) \cos \theta = 0 \] \[ \text{with} \ \theta(0) = 0 \] \[ \frac{d\theta}{ds}(1) = 0 \] (3a)

\[ \frac{d\tilde{x}}{ds} = \cos \theta \text{ with } \tilde{x}(0) = 0 \] \[ \frac{d\tilde{y}}{ds} = \sin \theta \text{ with } \tilde{y}(0) = 0 \] (3b)
was clamped at different lengths and the aspect ratio gets larger the problem becomes stiff, and more powerful continuation methods are required. We use the collocation-based continuation package AUTO07p [Doedel et al. 1991] to compute the solution of (3) over 8 decades of \( \Gamma \) values, and we plot the aspect ratio \( H/W = y(L)/x(L) \) as a black solid line in Figure 2, right. Furthermore, one can find analytical expressions for the aspect ratio for small and large \( \Gamma \) values, these are \( H/W = \Gamma/8 \) for \( \Gamma \ll 1 \) [Gere 2004], and \( H/W = \sqrt{\Gamma}/2 \) for \( \Gamma \gg 1 \), plotted as red dashed lines in Figure 2.

Experimental validation. In addition to the analytical and numerical solutions, we include experimental results. We fabricated a long rod by filling a straight PVC tube with vinylpolysiloxane. The rod was clamped at different lengths and the aspect ratio \( H/W \) was measured, see circles in Figure 2. We measured the cross-section radius \( r \approx 2.0 \text{mm} \) and the rod density \( \rho \approx 1200 \text{kg/m}^3 \). The only uncertain parameter was the Young modulus, for which we only had an approximate range of values given by the manufacturer. Starting from an arbitrary value within this range, our measurements nicely collapsed onto a single curve having the exact same shape as the master curve, albeit slightly shifted horizontally (\( E \) is inversely proportional to \( \Gamma \)). We could adjust the value as \( E \approx 1.08 \text{MPa} \) through a simple alignment of our measurements on top of the master curve\(^1\).

We also present experiments on a ribbon cut from an acetate sheet of thickness of 0.17 mm. The ribbon has dimension 3×27 cm\(^2\). It is clamped horizontally at different lengths and the aspect ratio \( H/W \) is reported as squares in Figure 2. The best alignment in this case was obtained for \( E' = E/(1 - v^2) \approx 4.19 \text{GPa} \).

4.3 The Bend-Twist test

While planar equilibria of naturally straight rods are a good starting point for the mechanical validation and calibration of a simulator, most of the complex phenomenology observed for elastic rods arises from the coupling between bending and twist. A physical situation where this coupling is at play was described by Miller et al. [2014]. We consider a rod of length \( L \) with a circular cross-section, and we note \( A \) the area, and \( I \) the second moment of area, of the cross-section. The rod is made from an isotropic, uniform, elastic material of density \( \rho \), but is cast in such a way that its natural shape is a twist-free circle of radius \( R \). The rod is then clamped vertically at one extremity and is left sagging under the action of gravity, while its other extremity is free. Depending on the material and physical parameters, the suspended equilibrium shape can be planar (with no twist), or deformed into a 3D curvy shape, see top panel in Figure 3.

Master curve: frontier between 2D and 3D configurations. The deformation of rods in 3D is governed by the Kirchhoff equations, see e.g. Audoly and Pomeau [2010]. We write the position of the rod as \( \mathbf{r} = \mathbf{r}(s) \), where the local frame \( \mathbf{t}(s) \), \( \mathbf{d}_1(s) \), \( \mathbf{d}_2(s) \) is used to follow the curvatures and twist deformations along the rod. We write \( \tau \) the twist, \( \kappa_{1,2} \) the two curvatures, and wrap these in a 3D vector \( \mathbf{\kappa}(s) = [\tau, \kappa_1, \kappa_2]^\top \). In its natural, uniform configuration \( \kappa_0 \), the rod has no twist and is curved in the plane orthogonal to \( \mathbf{d}_2 \) as \( \kappa(s) = \kappa_0 = [0, 0, 1/R]^\top \).

The local orthonormal frame follows the Darboux equation
\[
d\mathbf{R}/ds = \mathbf{R}[\mathbf{\kappa}]\times,
\]
where we use the matrix \( \mathbf{R} = [\mathbf{t} | \mathbf{d}_1 | \mathbf{d}_2] \) and the vector cross-product operator\(^2\).

The twist and curvatures are found by integrating the balance of torques \( \mathcal{K}_3 \cdot d\mathbf{R}/ds + [\kappa_3]_x \mathcal{K}_3 (\mathbf{\kappa} - \kappa_0) = [\mathbf{e}_x]_x \mathbf{R}^\top \mathbf{t}(s) \) where \( \mathbf{t}(s) = \rho A g(s - L) \mathbf{e}_z \) is the internal force and \( \mathcal{K}_3 = EI \text{ diag}(1/(1 + v), 1, 1) \) the diagonal stiffness matrix. Introducing dimensionless variables
\[
\bar{s} = s/R, \quad \bar{q} = L/R, \quad \bar{\kappa}(\bar{s}) = \kappa(s)/\kappa_0, \quad \bar{\kappa}_0 = \kappa_0 R, \quad \bar{\mathcal{K}}_3 = \mathcal{K}_3/EI,
\]
we finally obtain the dimensionless equation
\[
\bar{\mathcal{K}}_3 \frac{d\bar{\kappa}}{d\bar{s}} + [\kappa_3]_x \bar{\mathcal{K}}_3 (\bar{\kappa} - \bar{\kappa}_0) = \frac{\Gamma}{\bar{q}} (\bar{s} - \bar{q}) [\mathbf{e}_x]_x \mathbf{R}^\top \mathbf{e}_z,
\]
\(^1\)Actually, the quantity that we wish to measure accurately is the product \( EI \). An adjustment on the Cantilever master curve proves more robust than using for instance a traction test, which would measure only the Young modulus \( E \).
\(^2\)For a 3D vector \( \mathbf{u} = [u_0, u_1, u_2]^\top \) the cross-product operator is represented by the skew symmetric matrix
\[
[u]_x = \begin{bmatrix} 0 & -u_2 & u_1 \\ u_2 & 0 & -u_1 \\ -u_1 & u_2 & 0 \end{bmatrix} \quad \text{such that } \mathbf{u} \times \mathbf{v} = [u]_x \mathbf{v}.
\]
with $\Gamma$ defined in Section 4.2. Equations (4) and (6) are subject to the boundary conditions $x(0) = y(0) = z(0) = 0$ (clamping point located at the origin), $\mathcal{R}(0) = [-e_x] - e_y e_{y_0}$ (vertical downwards clamping), and $\mathbf{d}(\phi) = (0, 0, 1)$ (circular natural shape). Both are integrated in the interval $\mathcal{S} \in [0, \phi^*]$. From Equation (6), we see that this test involves two dimensionless parameters: the gravito-bending parameter $\Gamma$, and the curvature parameter $\varphi$. Note that $\varphi/(2\pi)$ corresponds to the number of turns of the twist makes in its natural shape. The full analysis of this equation is out of reach for this paper. However the key observation is that despite the fact it is always possible to find a planar solution (twist $\tau(s) = 0$, $V(s)$), depending on the values of the parameters $\Gamma$ and $\varphi$, it might not be stable. There is indeed a specific region in the phase diagram where the planar solution is unstable and is then replaced by a 3D solution. The parameter range, see Figure 3, then has two well-defined regions. Planar solutions (in orange), are observed for low and high values of $\sqrt{\Gamma}/\varphi$ and, in contrast, 3D solutions (in turquoise) prevail for rod with large $\varphi/(2\pi)$, but are restricted to $\sqrt{\Gamma}/\varphi \lesssim 0.62$. Above this limit, the rod remains planar whatever its length and natural curvature, and for long length cases, the deformed shape is mainly straight with a hook near the free-end to comply with boundary conditions. The frontier between 2D and 3D solutions is characterized by a pitchfork bifurcation and we use AUTO07p (with $\nu = 0.5$) to numerically follow this bifurcation point and plot the transition curve in black in Figure 3.

**Experimental validation.** Curly rods were fabricated by injecting vinylpolysiloxane in a PVC tube coiled around a rigid cylinder and letting the elastomer cure for 24 hours [Miller et al. 2014]. We obtained samples with a circular cross-section of radius $r = 1.8$ mm, a natural curvature $1/R = 57.02$ mm$^{-1}$, and a small natural twist $\tau = 2.65$ mm$^{-1}$. This small intrinsic twist is due to the manufacturing process as soon as rods have $L > 2\pi R$, but it only slightly affects the results. We used the tabulated value of vinylpolysiloxane Poisson’s ratio, $\nu = 0.5$, and we reused the Young’s modulus value, $E \approx 1.08$ MPa, measured in Section 4.2. The density $\rho \approx 1267$ kg/m$^3$ was obtained by simply weighing our samples.

We took snapshots of the rod clamped at different lengths, see top panel in Figure 3, and reported experimental parameters with large circles in the phase diagram, see bottom panel in Figure 3. Discrimination between 2D and 3D configurations was done visually. We found a good agreement between our experimental measurements and the numerical 2D/3D transition. We complemented the diagram with experimental points (small circles) and numerically-computed frontier points (triangles) from Miller et al. [2014].

### 4.4 The Lateral Buckling test

We now turn to a test involving an elastic structure with a width $w$ commensurated to its length $L$, namely an elastic plate. The thickness $h$ of the plate is small compared to both $L$ and $w$. As in the **Bend-Twist** test, we let the plate hang and sag under its own weight and wait for a (pitchfork) bifurcation to occur. In its reference configuration, the plate lies in the $(x, z)$ plane with $0 \leq X \leq L$ and $-w \leq Z \leq 0$. Gravity is oriented along the $-e_y$ direction. The plate is clamped along its $X = 0$ edge while the other three edges are free. The clamped orientation is vertical, along the $z$ axis. In the deformed configuration, the plate can either stay in the $(x, z)$ plane or deflect laterally in the $e_x e_y$ direction and adopt a 3D shape. This lateral buckling typically occurs when gravity overcomes the elastic rigidity of the structure.

Lateral buckling instabilities are an important mode of failure in civil engineering where long steel beams carry important shear loads, and the history of lateral buckling analysis goes back to the

---

In the **Bend-Twist** protocol, rod self-contact is not accounted for.

---

4The sharp reader may notice that some experimental rods (Figure 3 top right) are labeled as 2D whereas they slightly look 3D. This 3D shift is due to our residual natural twist stemming from fabrication constraints, and not to any buckling of the rod. We accounted for this residual when labeling the experimental rods as 2D or 3D (see also our supplementary document, Section 4).
19th century [Michell 1899]. Still, we are not aware of any study carefully analyzing and building up a collapsing curve (with only two dimensionless parameters) for this test. This is what we propose to achieve here.

As in previous sections we nondimensionalize, i.e. we rescale, lengths with $L$ and forces with $Dw/L^2$, where $D$ is the bending rigidity of the plate, $D = E h^3/(12(1 − ν^2))$ (see Section 4.2). Once these two parameters are set away, we are left with the following free parameters: the Poisson ratio $ν$, the width $w/L$, the thickness $h/L$, and the total weight $Γ^v = Mg/(Dw/L^2)$, where the total mass of the plate is noted $M = phwL$. Recall from Section 4.2 that $Γ^v$ differs from $Γ = Mg/(EI/L^2)$ by a scaling factor, $Γ^v = (1 − ν^2)Γ$, with $Dw = EI$.

**Master curve:** Our test experiment consists in holding the plate clamped as explained above, and increasing the parameter $Γ^v$ while keeping the other parameters fixed. For $Γ^v = 0$ the plate is in its reference configuration. As $Γ^v$ increases, the plate only suffers in-plane shear and extension, but stays in the $(x, z)$ plane. Only when $Γ^v > Γ_c^v$ does the plate buckle in the third direction and normal curvature sets in. The critical $Γ_c^v$ value depends on the other free parameters $Γ_c^v = Γ_c^v(w/L, h/L, ν)$. To generate an interesting master curve, our idea was to plot $Γ_c^v$ as a function of $w/L$, as shown in Figure 4, bottom panel. We work with thin plates $h/L ≪ 1$, $h/w ≪ 1$, that is in the regime where only little transverse (i.e. through the thickness) shear deformation is present. Consequently $h$ is not a parameter anymore and we assume we have reached the limit where $Γ_c^v = Γ_c^v(w/L, ν)$. To be on the safe side numerically, we nevertheless use the S8R thick shell finite-elements setup of ΩAibaqs to determine $Γ_c^v$. We set $ν = 0.35$ (corresponding to polyester, which we use in experiments) and compute the threshold curve with the buckling eigenvalue analysis module of Abaqus using a high resolution mesh, i.e. for each $w/L$, we compute the critical $Γ_c^v$ for which the total stiffness matrix, composed of the matrix of the base state and the tangent matrix, becomes singular. The results are plotted in Figure 4, using $h/L = 6 × 10^{-4}$. We see in Figure 4 that $Γ_c^v(w/L = 0)$ corresponds to the value evaluated with the Kirchhoff elastic rod theory $Γ_c^v(kirchhoff) = 18.2(1 − ν)\sqrt{1 + ν}$ [Michell 1899], and that, remarkably, $Γ_c^v$ increases approximately linearly (with slope $≈ 14.5$) with $w/L$ in the range $0 < w/L$ that we considered. Interestingly, we note that this simple linear dependence, extremely practical to evaluate numerical codes, actually matches the observations previously made in a lateral buckling problem with a different loading [Reissner 1995].

To evaluate the dependence of $Γ_c^v(w/L, ν)$ w.r.t. the Poisson ratio $ν$, we computed the curve for several $ν$ values and found them to be approximately parallel, each one emerging from the Kirchhoff $w = 0$ limit. We conclude that an approximated formula for $Γ_c^v$ may be written as $Γ_c^v(w/L, ν) \approx 18.2(1 − ν)\sqrt{1 + ν} + 14.5 \frac{w}{L}$.

**Experimental validation.** We perform experiments with naturally flat Poly-Styrene plates with thickness $h = 0.10$ or $0.14$ mm. We measure densities to be $ρ_{0.10} = 1410$ and $ρ_{0.14} = 1260$ kg/m$^3$. Using the Cantilever test, we measure the Young modulus to be $E_{0.10}(1 − ν^2) = 7.75$ GPa and $E_{0.14}(1 − ν^2) = 4.19$ GPa. For the Poisson ratio we use the value $ν = 0.35$, tabulated at polymerdatabase.com. We then have $L_{gb0.10} = 7.8$ cm and $L_{gb0.14} = 8.2$ cm.

We use five samples of widths cut in the range $3 < w < 12$ cm, and having either $h = 0.10$ or $0.14$ mm. For each sample, the plate is clamped vertically, with a suspended length $L$ emerging from the clamp. Each experiment consists in increasing $L$ from $L = 14$ to $L = 27$ cm. For each length value, we place a square on the phase diagram of Fig. 4 whose color indicates the 2D/3D nature of the equilibrium shape. Writing $y = Γ^v$ and $x = w/L$, each experiment follow a hyperbola of equation $y = (w/L_{gb})^3/\sqrt{x^3}$.

Experimentally this is a challenging system as the lateral buckling instability is very sensitive to both the verticality of the clamp and the flatness of the plate. Despite these difficulties, we retrieve a fairly good agreement with the numerical prediction, as shown in the phase diagram in Figure 4.
In this so-called Stick-Slip test, a clamped elastic strip, of length $L$ and rigid enough to neglect gravity effects ($\Gamma \ll 1$), is pushed quasi-statically against a solid substrate, see Figure 5. The strip and the substrate interact through normal ($P$) and tangential ($Q$) forces at the contact. As the vertical displacement ($\Delta y$) increases forces do as well.

Three well-defined regimes are observed in this test depending on the friction coefficient ($\mu$) and the vertical strain ($\epsilon_y = \Delta y / L$): stick, slip, and extended contact (orange, turquoise, and gray regions respectively in Figure 5). In the stick phase, contact is localized at the lower end of the elastic strip and the contact point does not move. In the slip phase, friction is no longer able to hold the end of the strip which slips along the horizontal plane. Finally, in the extended contact regime the contact interface spreads to a larger portion of the elastic strip. These three regions are separated by (i) a Master curve (computed below), (ii) the line $\epsilon_y = \epsilon_{y,c} \approx 0.33$, and (iii) the line $\mu = \mu_c \approx 0.36$. These three border curves meet at the triple point $(\epsilon_{y,c}, \mu_c)$. The phase diagram of Figure 5 does not depend on any mechanical or geometrical property of the system. Therefore, one can compare frictional contact solvers over a large range of mechanical scenarios and test their ability to reproduce the phase diagram.

**Master curve: transition between stick and slip regimes.** To compute the Stick-Slip master curve, which partitions the stick and slip regions, we compute the equilibrium of a clamped planar elastica as the parameter $\epsilon_y$ is varied. The equilibrium solution gives the forces $P = -P \epsilon_y$ and $Q = Q \epsilon_x$ and friction is accounted for with the Amontons-Coulomb law. More precisely, once contact is made, the strip buckles as soon as the vertical force $P > 20.2EI/L^2$. At buckling the horizontal force is $Q = 0$ and no friction is required for the equilibrium to hold. Nevertheless as $\epsilon_y$ is increased the ratio $Q/P$ grows, and slip occurs as soon as $Q/P$ reaches the Coulomb coefficient $\mu$. The transition (master) curve is then reached when the clamped-pinned equilibrium is such that $Q/P = \mu$. We therefore compute the ratio $Q/P$ at equilibrium, as function of $\epsilon_y$, and plot it in the phase diagram, see the black curve in Figure 5. Using the dimensionless arc length $\bar{s} = s/L$, equilibrium is found as the solution of the following boundary value problem,

$$
\frac{d^2\theta}{d\bar{s}^2} = \frac{P \bar{L}^2}{EI} \sin \theta + \frac{Q \bar{L}^2}{EI} \cos \theta \tag{7a}
$$

$$
\frac{d\bar{x}}{d\bar{s}} = \sin \theta \quad \text{with } \bar{x}(0) = 0, \ \bar{x}(1) = 0 \tag{7b}
$$

$$
\frac{d\bar{y}}{d\bar{s}} = \cos \theta \quad \text{with } \bar{y}(0) = 0, \ \bar{y}(1) = 1 - \epsilon_y^* \tag{7c}
$$

where we note $\epsilon_y^*$ the value of $\epsilon_y$ on the transition curve.

Note that in Eq. (7a), $P$ and $Q$ only appear divided by the scaling force $EI/L^2$ and will therefore go to be proportional to it. Hence, the ratio $\mu = Q/P$ will be independent of the mechanical ($EI$) and geometrical ($L$) properties of the elastic strip.

One can analytically solve Eq. (7) up to a second order in $\theta$ by writing $\sin \theta \approx \theta$ and $\cos \theta \approx 1 - \theta^2$. With this weakly nonlinear approximation, the solution for the master curve reads

$$
\frac{Q}{P} = 0.445 \sqrt{\epsilon_y}. \tag{8}
$$

**4.5 The Stick-Slip test**

Up to now we have dealt with suspended elastic objects deformed by the action of gravity. However, in many applications in Computer Graphics and Computer Vision, elastic slender objects are supposed to interact with their surroundings. Thus, testing contact and frictional solvers is of foremost importance for assessing realistic scenarios.

Rasheed et al. [2020] recently introduced the work of Sano et al. [2017] as a way to evaluate the ARGUS code [Li et al. 2018] that they used for generating cloth training data. Inspired by their approach, we similarly leverage the Sano’s stick-slip criterion for precisely evaluating the realism of frictional contact solvers. Unlike them however, we re-build the mathematical formulation of the master curve from scratch and carefully verify it experimentally, with a more elaborated experimental set-up compared to [Sano et al. 2017]. Furthermore, we test multiple codes on this master curve, including plate and 2D rods simulators coupled with various frictional contact solvers.

![Fig. 5. Master curve and experimental validation for the Stick-Slip test. Left: Experimental snapshot of a Stick-Slip test with a high friction coefficient, for $\epsilon_y = 0.2$. The superposition of the numerical planar elastica solution (blue line), for the same value $\epsilon_y = 0.2$, illustrates the very good agreement between experiments and numerics. Top-Right: our experimental ribbons under frictional contact with the ground, with vertical deflection $\epsilon_y$. The background (originally black) has been colored to indicate whether the plate and 2D rods simulators coupled with various frictional contact solvers interact through normal ($P$) and tangential ($Q$) forces at the contact. As the vertical displacement ($\Delta y$) increases forces do as well.](image-url)
This solution (dashed white line in Figure 5) is not accurate enough to serve in the present test. A detailed, semi-analytical, solution of this problem can be found, see e.g. [Mikata 2006], but involves elliptic functions and a root-finding step. Here, without any lost of accuracy, we simply solve system (7) numerically and plot $Q/P$ as a function of $\epsilon_y$, see the black curve in Figure 5. The triple point $(\epsilon_y, c, \mu_c) \simeq (0.33, 0.36)$ is reached when the equilibrium solution is such that $\theta(s = L) = \pi$.

**Experimental validation.** We quasi-statically compress a 260 μm thick, naturally flat, elastic strip of Poly-Styrene against a rigid substrate. The strip is clamped at the top end to a Futek load cell which measures the load $P$. The load cell itself is attached to a Thorlabs LTM300 motorised stage used to control the vertical displacement $\Delta y = \epsilon_y L$. The substrate is mounted on two horizontal free-to-roll cylinders, to minimise resistance to horizontal displacement. A second Futek load cell holds the substrate horizontally and measures the load $Q$. An experiment starts at $\epsilon_y = 0$ and consists in increasing the vertical displacement with a series of 1mm steps, each followed by a wait-in-position step of 2 seconds where 2000 force measurements are performed and averaged, for both $P$ and $Q$. At each displacement step we take a snapshot of the strip to compare its shape with numerical computations (see e.g. the image on Figure 5). We use different sample lengths ($L = 20.5$ to 28.5cm) and widths ($w = 3$ to $w = 4.5$cm) to explore different $L^3/EI$ values. We stop experiments at vertical strain $\epsilon_y = 0.33$, and make sure no material plasticity was encountered during the test.

Depending on the frictional characteristics of the substrate we obtain two qualitatively different responses of the system.

On the one hand, for smooth substrates (copper, plastic, and aluminium), we observe the stick to slip transition for $\epsilon_y = \epsilon_y^c$. At this transition point, the force measurements yield a ratio $Q/P$ which is in good agreement with the numerical Stick-Slip master curve, see square markers in Figure 5. Note that the value of $Q/P$ at this point is a way to evaluate the Coulomb friction coefficient $\mu$ of the material. On the other hand, for rough substrates (e.g. sandpaper with grit size 180) the elastic strip first sticks then transitions into the extended contact regime. In this case, plotting the experimentally measured ratio $Q/P$ as a function of $\epsilon_y$ yields the purple curve in Figure 5, which is in good agreement with the numerical master curve. For such substrates we cannot infer the friction coefficient $\mu$ from the experimental data, we only know that $\mu > \mu_c \simeq 0.36$.

## 5 SELECTION OF EXISTING CODES

We have selected a number of numerical models from Computer Graphics, which are well-known and widely used in the community. As a baseline for comparison, we also picked two reference codes from Mechanical Engineering, among which the widely used commercial software ©ABAQUS. The full list of the 14 codes used (together with some variants) is provided in Table 1, left column.

Of course, our selection is not exhaustive: it is rather meant to gather some representative examples. But we hope that the methodology we describe in this paper will encourage others to validate their own numerical model, which will help the community enrich these results.

### 5.1 Implementation

As far as possible, to limit implementation bias, we have striven to use the original code from the authors. This was possible for all but four of the academic numerical models we have selected, either by using the free implementation distributed online by the authors (SUPER-CLOTHOID, LibShell, ArcSim, So-Bogus, Argus, FenicsShell), or upon direct request to the authors (Super-Helix, Discrete Elastic Rod, Super-Ribbon, Projective Friction). The four exceptions comprise Projective Dynamics, Discrete Shell, Bridson-Harmon, and Viscous Friction. However, these four algorithms are simple enough so that we could reimplement them quite easily with a high level of confidence. Finally, for the commercial software ©ABAQUS, we had to use a license paid by our lab, and did not have access to the source code.

It is noteworthy that the codes we have tested are very disparate in terms of implementation, parameter input, and general behavior. In the supplementary document (Section 7) we summarize the main characteristics of all the codes we have tested.

### 5.2 Thin elastic rods

A staple of rod models in Computer Graphics is the Discrete Elastic Rod model [Bergou et al. 2010, 2008], which has been used for various applications ranging from hair simulation [Daviet et al. 2020; Kaufman et al. 2014] to the design of meta-materials [Schumacher et al. 2018] and grid shells [Pianetta et al. 2019]. The Discrete Elastic Rod (dynamic) model relies on a nodal formulation of the rod centerline tightly coupled to a material frame via some material twist around the discrete Bishop frame. While the first paper rigidly fixes the material frame as the one minimizing the potential elastic energy of the rod (assuming a vanishing cross-section), the second one considers in contrast some twist inertia and updates the material frame in time through parallel transport. The latter model has the advantage of yielding a sparse-block stiffness matrix at first order, allowing for fast implicit integration. The implementation that we have used was provided by the authors: it is based on this second method, which offers more stability than the first one.

We have also considered the family of curvature-based models, namely the Super-Helix [Bertails et al. 2006] and the Super-Clothoid [Casati and Bertails-Descoubes 2013] models. In contrast to position-based formulations, these models rely on a curvature-based parameterization, allowing the formulation of the kinematics of the rod (i.e. bending and twisting modes) in a fully reduced way, without the need for inextensibility constraints. In the Super-Helix model, each element is characterized by uniform material curvatures and twist and takes the form of a circular helix (or circular arc in 2D). In our test, we have used the 2D and 3D implementations provided by the authors (dynamic codes). Super-helices were successfully used in Computer Graphics for simulating the forward and inverse dynamics of human hair [Daviet et al. 2011; Derouet-Jourdan et al. 2013; Hu et al. 2017], and their reduced parameterization has been recently harnessed in the context of soft robotics [Boyer et al. 2020]. Besides, the approach was extended to the Super-Clothoid model, which features material curvatures and twist linearly varying along the elements. For our tests we took a static version (provided by the authors) of the reference (dynamic) implementation available online [Casati and Bertails-Descoubes 2013].
We initially intended to test further popular rod models from Computer Graphics [Hadj 2006; Pai 2002; Spillmann and Teschner 2007], but unlike for plates/shells, we had trouble finding reference implementations. As rod models are often tricky to implement, we refrained from using a non-reference code which may not fairly reflect the original published method. However, our result table is meant to be dynamic and filled in by others with their own models and implementations.

5.3 Thin elastic plates, shells and ribbons

Plates and shells. For plate and shell computations, Computer Graphics codes compete in attempting to formulate simple yet accurate membrane and bending formulations, as well as their coupling. For in-plane deformations, linear elasticity has been proven to be computationally cheap and yet relies on strong theoretical bases ensuring its accuracy and showing its limits. However, for out-of-plane deformations, to the best of our knowledge, no model has been validated yet.

A widespread bending energy for plates or shells in Computer Graphics has been simultaneously proposed by Grinspun et al. [2003] and Bridson et al. [2003]. Although the former derived it from a discretization of the square of the difference of the mean curvatures and the latter from heuristics, both yield a similar (up to a scaling coefficient) hinge energy based on the dihedral angle between faces at their shared edges. The simplicity of this energy made it a popular way to measure the bending in Computer Graphics applications [Coros et al. 2012; Skouras et al. 2012; Wang et al. 2011]. Bridson et al. [2003]’s model is part of the Arcsim implementation[1] [Narain et al. 2013, 2012] and, introducing a slight modification of the bending coefficient (from $\frac{1}{2}$ to 3), we could readily switch to Grinspun et al. [2003]’s bending model (see supplementary document, Section 3). This modified code is denoted in the following as Discrete Shell + Arcsim. Note that for Arcsim, we used the more recent Argus implementation [Li et al. 2018] for the sake of simplicity (without frictional contact, Argus boils down to Arcsim).

Another, more recent approach, comes from the discrete geometry community [Chen et al. 2018; Weischedel 2012]. By replacing metrics such as the fundamental forms used in the continuous description of shells by discrete equivalents, new numerical shell models were derived. The open-source library L preprocess SHELL [Chen et al. 2018] implements a discrete shell model based on Koiter’s thin shell energy following this principle. The code possesses a few variants, characterized by the choice of elastic behavior – Saint Venant-Kirchhoff (StV-K) or Neo-Hookean (NH) laws – and the choice of discretization scheme employed for the computation of the normals – one “first order” M ID EGE A VERAGE and two “second order” M ID EGE S IN and M ID EGE T AN schemes [Weischedel 2012]. Note that this overall approach departs from traditional finite-elements formulations which discretize quantities over functional spaces. By incorporating the [Grinspun et al. 2003]’s bending model into L preprocess SHELL, we were able to test yet another variant of this approach, referred to as Discrete Shell (+ LinShell). We expect this variant to be close to Discrete Shell + Arcsim as, in principle, both models use the same in-plane and out-of-plane energies. Anticipating our results (Section 7), we however noticed some discrepancies between them.

Ribbons. Lying in between rods and plates, ribbons have recently regained some interest in the Soft Matter Physics community [Fosdyck and Fried 2015]. Based on the Sadowsky-Wunderlich theory [Sadowsky 1929; Wunderlich 1962], and obtained through dimension reduction, equations for ribbons have been reformulated as enriched Kirchhoff rod equations [Dias and Audoly 2015]. Inspired by the Super-Clothoid model, an implementation of the Sadowsky-Wunderlich ribbon model has recently been proposed, based on elements with a linear normal curvature and a quadratic twist [Charondière et al. 2020]. This work has been published in Mechanical Engineering, but since it is deeply related to the super-model formulation introduced in Computer Graphics, we have included it in our study. Moreover, since the model lies in-between rods and plates, we found it particularly interesting to assess its relevance w.r.t. our rod and plate tests. We have used the implementation borrowed from the authors, that will be referred to as Super-Ribbon in the following.

5.4 Frictional contact

Perhaps one of the most popular model for frictional contact in Computer Graphics is the Bridson-Harmon algorithm [Bridson et al. 2002; Harmon et al. 2008], which explicitly resolves non-penetration and Coulomb constraints by successive filters applied to the object velocity. Such a model is known for yielding satisfying qualitative results (e.g. avoiding any interpenetration artifact) at a cheap cost.

In contrast, implicit constraint-based methods that were later introduced in Computer Graphics focus on physical accuracy and on the accurate capture of the Coulomb threshold. We have selected the So-Bogus solver [Daviet et al. 2011], where the contact forces are treated as unknowns and are tied to the velocities through complementarity constraints that are resolved implicitly. This solver has been successfully used to simulate large assemblies of hair [Kaufman et al. 2014] and cloth [Li et al. 2018]. The So-Bogus code is freely available as a standalone library, and for our tests we coupled it with the Super-Helix 2D model following the work of Daviet et al. [2011]. We also use its coupling with the cloth simulator Arcsim, also known as the Argus code, which is freely distributed by their authors [Li et al. 2018]. Additionally, we consider the much faster Projective Friction algorithm, which yields results qualitatively similar to Argus [Ly et al. 2020]. We used the original code provided by the authors.

Finally, although the Viscous Friction model – which simply sets friction forces proportional to the object velocity – is not adapted to solid friction, it is quite often used for its simplicity. We thus decided to include it in our study to emphasize on its differences with dry frictional contact models.

5.5 Reference codes of Mechanical Engineering

Originally released in 1978, ©ABAQUS [Dassault-Systems 2005] is a widely used commercial software designed for the numerical modelling of solid and fluid mechanics as well as electromagnetic

---

5Bridson et al. [2003] rely on the sine of the dihedral angle whereas Grinspun et al. [2003] use the angle itself, which is equivalent in the limit of small angles.

6We used version 0.2.1 of Arcsim, recommended by their authors over version 0.3.1 when shells and fracture are not needed.
problems. It is based on finite-elements analysis and supports various types of elements (including high-order elements for rods, plates and shells), both in static and dynamic settings. The user may choose between a graphical user interface and Python scripting for using the software. ABAQUS’s robustness and capability to treat nonlinear problems involving contact, plasticity, etc., as well as explicit and implicit dynamics, have made it one of the reference tools in Mechanical Engineering, both in industry and academia. We therefore use it as a reference in our validation work, taking advantage of its many ready-to-use models, and despite the fact that most of its routines are closed.

In contrast, FEniCS is an open-source computing platform which comprises a C++ core engine interfaced with Python [Habera et al. 2018]. FEniCS codes are easily parallelized and run on laptops, desktops, or clusters. The FEniCS environment allows users to quickly enter various types of partial differential equations through the Unified Form Language (UFL) with which users convert mathematical models into finite-elements formulation in a straightforward way. FEniCSShell is a set of libraries for solving linear and nonlinear plate and shell models [Hale et al. 2018]. Sheep-locking is dealt with using several techniques, among which MITC (Mixed Interpolation of Tensorial Components) or PSRI (Partial Selective Reduced Integration, which we have used). As with UFL, the user only has to write down the elastic energy, the physical model is easily changed and moreover several Newton and linear algebra solvers are accessible (including for example PETSC). The actual implementations for the Cantilever and Lateral Buckling tests are based on an example provided online [Brunetti et al. 2018].

6 METHODOLOGY FOR BENCHMARKING

Comparing a number of codes thoroughly is a particularly challenging task. Common and unconscious biases include treating codes heterogeneously, favoring one’s code in place of others’, or failing to give sufficient details for reliable replication by others. To avoid or at least limit these pitfalls, we have established a rigorous methodology which we strive to make reproducible.

6.1 As consistent as possible

Benchmarking codes requires a common methodology even though the set of tested codes may be fairly diverse. In our case for instance, some codes may be static while others dynamic, some may use dimensionless parameters while others dimensional data. To compare codes consistently, we should be able to free ourselves from peculiarities and offer clear and non-ambiguous guidelines that can be followed equally for each code to be tested.

To this aim, we advocate a (possibly iterative) two-step strategy:

1. First, prepare the simulation scenario, that is find a combination of physical parameters of the code $C$ able to reproduce the test $T$ in the right physical regime;
2. Second, find, by continuation, the optimal numerical parameters of the code $C$ able to “agree with the test”.

After defining the meaning of “agreeing with the test” in terms of a test error, we formulate our final rule for assessing a code on a given test.

Test error. Quantifying the proximity of the results produced by a code $C$ to the expected ones on a given test $T$ can be achieved by defining an error $E(C, T)$, which measures a distance between the two. In practice, we compare the reference master curve of the test $T$, over its full range, to the one generated by the code $C$. The code will be said to be in good agreement with the test result if the output of the code is indeed a continuous curve at our scale of observation, and if it matches the reference curve visually (i.e. $E(C, T) < \epsilon_T$ with a tolerance $\epsilon_T \approx 5\%$ of the plot size, in the sense of the Hausdorff distance).

Consistent solver calibration. For each test, we perform a large number of simulations by sampling a wide parameter range. From one parameter bound to another one (e.g. from a small value of $T$ to a large one), the complexity of the simulations (e.g. small deformation vs. large deformation) may vary a lot, hence calling for different tunings of the discretization and numerical solver (number of elements, timestep, tolerance, etc.). However, to keep benchmarks as simple as possible, instead of building complex adaptive numerical strategies, we preferred to find out a common set of numerical parameters allowing for proper convergence of the solver over the entire parameter range. In other words, for a given code, the results reported stem from the same numerical configuration, and only the physical parameters vary.

Success versus failure. To simplify code evaluation, we have chosen a binary measure of success. A code $C$ is assessed as successful (OK) on test $T$ if there exists a set of numerical parameters such that, for any refinement of these parameters and any combination of physical parameters $p$ which lead to the same regime (or dimensionless data) $R_T$ of the test, the code output agrees with the master curve. Otherwise, it is considered as failing (KO) on test $T$.

Mathematically, this reads

$$T(C) = \text{OK} \iff \exists n \text{Elts}^*, \bar{d}t^*, \text{tol}^* \text{ such that }$$

$$\forall n \geq n \text{Elts}^*, \bar{d}t \leq \bar{d}t^*, \text{tol} \leq \text{tol}^*, \forall p \in R_T, \quad E(C, T) < \epsilon_T. \quad (9)$$

As a result, a code $C$ will be marked as KO on test $T$ if we fail to find a set of physical and numerical parameters passing the test $T$, or, if we just find an arbitrary and isolated set of parameters which works, without any possible refinement.

Our success rule is thus highly demanding, since it requires, at the same time, a correct physical (dimensionless) behavior of the code (\(\forall p \in R_T\)), the numerical convergence in space and time of the numerical model (\(\forall n \geq n \text{Elts}^*, \bar{d}t \leq \bar{d}t^*, \text{tol} \leq \text{tol}^*\)), and the convergence towards the physical target ($E(C, T) < \epsilon_T$).

Note however that our rule does not discriminate between “fast” and “slow” codes, in terms of convergence. A code may be qualified of OK even though it may require a huge mesh resolution or and an extremely small time step to reach the target, making such a configuration impractical. To compare code convergence speed (and thus their practicability), we hence adjoin the information of the numerical values ($n \text{Elts}^*$, $\bar{d}t^*$, $\text{tol}^*$) to the test status (OK) in case of success.

Anticipating on the result section, the reader may take a look at Table 1, which summarizes all our test results.
6.2 As fair as possible

Dealing equally with some codes that are yours and some that have been written by others is difficult, for two main reasons. First, having a perfect knowledge of the model, its various options, and the underlying implementation details, of course helps understand how to tune things and quickly get the most out of the code. This process might be much slower and may eventually not converge to the optimal behavior when using a code that one does not master. Second, it is unconsciously always tempting to spend less efforts on others’ code to make it work than on one’s code, because of a lack of neutrality (even though one’s honesty is unquestionable).

In the two cases mentioned above, the risk is clearly to underestimate the potential of others’ code or method. To mitigate this risk, we have adopted the principles listed below.

A set of well-experimented programmers. The authors of this paper who took in charge the benchmarking of Computer Graphics codes are only senior PhD students, postdocs or permanent researchers with good programming skills and a minimum experience of three years with complex programming in Computer Graphics.

Understanding of the underlying numerical model. Fortunately, most of the numerical models we have been using are well documented in open publications. Moreover, except for ©ABAQUS, we have access to all source codes, which has allowed us to check the conformity between the written model and its implementation. From the publications, we could also verify the limitations of the proposed algorithm, which generally matched our own observations. Overall, we were able, in most cases, to make sure that our observations were consistent (or at least, not contradictory) with the claims made in the original papers. The ©ABAQUS case was the most problematic one, because it is a closed code and its optimal usage may require specific training. To limit bias regarding our usage of this software, two authors of this paper were trained on ©ABAQUS for several weeks. Additionally, we have consulted several colleagues of Mechanical Engineering conducting their research with ©ABAQUS, who could share their expertise with us.

Overall, we estimate we did our best efforts to get the most of each software in our hands.

Systematic investigation in case of failure. After trying in vain parameter refinement or various tricks, when it comes that a code fails to a given test, we strive to analyze the origin of such a failure. And, if possible, we attempt to provide a (documented) fix to the code so that it may eventually pass the test. In our study, some of our fixes helped in succeeding a test, while others did improve results but not to the point of a success.

Future evaluations. Finally, despite all these precautions, there is theoretically always a small probability that we may have misleadingly categorized a code as a failure on a given test, because we may have failed to test all possible alternatives. In this case, we may have unwillingly produced false negatives. In contrast, given the high requirement level of each test, we believe that false positives are extremely unlikely to arise. Hence, successful codes can reasonably be considered as certified by our study, under their precise condition of usage.

To mitigate the possible limitation on false negatives, we count on future benchmarking made by the community to refine and enrich our results. This is why reproducibility of our methodology is of foremost importance.

6.3 As reproducible as possible

Our goal is twofold. First, allow others to reproduce our benchmarking results (at least for publicly available codes). Second, allow others to validate rigorously their own models or codes, following our exact procedure. Our supplementary material “Recipe Manual for Validation” should give all the necessary algorithms and information to meet these two objectives. We also plan to deliver data regarding the reference master curves to compare against.

7 BENCHMARKING RESULTS

The general results of our benchmarking study are summarized in Table 1. As mentioned sooner, OK is the success score, while KO denotes a failure. A dash (–) indicates that the test is not applicable to the corresponding code, while a cross (×) shows that we have not tried to test the code because of expected failure. From these results one can already make a few important observations.

First, the most well-known rod models of Computer Graphics, namely Discrete Elastic Rod and Super-Helix (and its higher-order counterpart Super-Clothoid), perfectly pass the two rod tests Cantilever and Bend-Twist. This is not surprising, as these models have benefited from physical insights from their construction stage. Furthermore, these codes have already been cross-validated [Casati and Bertails-Descoubes 2013] and used in other communities like Soft Matter Physics [Audoly and Pomeau 2010; Jawed et al. 2014].

In contrast, the popular adaptive model Arcsim for plates fails on Cantilever, even though it is a mere 2D test. We carefully analyze such failure and attempt to provide a fix to the code.

Then, the only Computer Graphics code able to cope with the demanding Lateral Buckling test is LibShell, although its success depends on the exact choice of the bending energy formulation. Note that FenicsShell successfully passes the test as well.

Regarding frictional contact, most Computer Graphics solvers fail on the Stick-Slip test, mainly due to a lack of accuracy at convergence. The only successful Computer Graphics code is So-Bogus, provided a low tolerance of the solver is used.

Finally, the commercial package ©ABAQUS, widely used in mechanics, performs moderately well. While it easily passes Cantilever and Stick-Slip, it surprisingly has a hard time on Bend-Twist, which is in contrast perfectly passed by the three Computer Graphics codes for rods that we have tested. It similarly fails on Lateral Buckling, where both LibShell and FenicsShell succeed.

Our results and observations are further detailed and commented on in the remainder of this section. Please also watch our accompanying video for an illustration of our benchmarking study. Many further information such as convergence plots or detailed experiments and results are provided in supplementary material. Moreover, our second supplementary material, called “Recipe Manual for Validation”, explains all the rules, criteria and tips that we have been using to conduct the study thoroughly.
Table 1. General result summary of all the benchmarks we have performed.

<table>
<thead>
<tr>
<th>Tested Code</th>
<th>Cantilever</th>
<th>Bend-Twist</th>
<th>Lateral Buckling</th>
<th>Stick-Slip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Rod</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISCRETE ELASTIC ROD [Bergou et al. 2010]</td>
<td>OK (300 elts)</td>
<td>OK (200 elts)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SUPER-Helix [Bertails et al. 2006]</td>
<td>OK (50 elts)</td>
<td>OK (30 elts)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>SUPER-CLOTHOIDS [Casati and Bertails-Descoubes 2013]</td>
<td>OK (20 elts)</td>
<td>OK (25 elts)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Ribbon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUPER-Ribbon [Charrondiere et al. 2020]</td>
<td>OK (20 elts)</td>
<td>–</td>
<td>KO</td>
<td>–</td>
</tr>
</tbody>
</table>

**Plate**

|            |            |            |                  |            |
| FENICSHELL [Hale et al. 2018] | OK (Res 0, P_{2,4} elts) | –             | OK (Res 0, P_{2,4} elts) | –          |
| ©ABAQUS    | OK (200 P_2 elts) | KO | KO | OK (dt=9 µs) |

7.1 Results on the Cantilever test

The **Cantilever** test stands for an interesting 2D protocol to validate the bending formulation of all kinds of models – rods, ribbons, and plates – whether they are inextensible or not. Indeed, even though the master curve has been derived in the mere inextensible case, it remains valid in the presence of stretching within the range of $\Gamma$ used, $\Gamma < 10^4$. When simulating a simple finite-difference rod model allowing for both bending and stretching, we were able to check that the in-plane elongation remains indeed negligible ($\approx 1\%$ at $\Gamma = 10^4$). In the codes allowing for stretching and shearing, we made sure to take a negligible thickness $h \ll w$ so that bending is the preferred deformation mode.

Even though it remains difficult to compare finely two models discretized with different approaches (e.g., finite-elements against models based on discrete differential geometry), we made sure we were using similar mesh resolutions for all the plate models that we have tested. For the **Cantilever** test we have typically used two kinds of meshes: a Res 0 mesh, made of 120 discretization nodes along the length, and a Res + mesh with a double-size resolution.

Figure 6 presents the results for all the tested codes. Except **Arcsim** (whose results are shifted) and **Projective Dynamics** (whose results are too scarce), all the codes are in excellent agreement with the master curve. However, for some bending plate models like **Discrete Shell**, an increased resolution (Res +) was needed to cover the range $10^3 < \Gamma < 10^4$ properly. Computation time ranges between $\approx 1$ h (for most models) and a few hours (for nodal models requiring a high number of elements or a high-resolution mesh).

**Bridson et al. vs Grinspun et al.’s bending energy.** Naturally, since **Arcsim**’s bending energy [Bridson et al. 2003] only differs from that of **Discrete Shell** [Grinspun et al. 2003] by a multiplicative factor, both cannot pass the test. Our test thus allows to discriminate between the two factors, leading to the conclusion that Grinspun et al.’s factor is the correct one (see Figure 7, top).

However, we found the Grinspun et al. [2003]’s discrete energy to converge much more slowly compared to other mesh-based approaches like **LibShell** or **FENICSHELL**. In practice, we had to use a twice as more resolute mesh (Res +) for **Discrete Shell (+ LibShell)** to observe a good agreement with the master curve.

Interestingly, Grinspun et al. [2006] report that **Discrete Shell** does not converge in the general case, albeit in the case of triangular meshes with equilateral triangles. Our meshes were close to this ideal case, which probably explains our observation of convergence.

**Investigations on Arcsim.** After fixing **Arcsim** with the correct factor (**Discrete Shell + Arcsim**), we observe a reduction of the
with Grinspun’s factor nicely removes the shift (Figure 7, top). In-when using the LibShell ACM Transactions on Graphics, Vol. 40, No. 4, Article 66. Publication date: August 2021.

spent in performing the test. However, for the sake of simplicity, unable to capture the smoothness and the results start to scatter. For plates, without a sufficient resolution, the resulting mesh is the clamp (the latter required meshes twice as resolute as the former). Shell + Arcsim

Discrete Elastic Rod nodal models like Arcsim

Curvature-based models such as Super-Ribbon can easily capture this phenomenon with few elements (50 and 20 elements, respectively). However, nodal models like Discrete Elastic Rod, LibShell or Discrete Shell + Arcsim require meshes with a fairly good resolution around the clamp (the latter required meshes twice as resolute as the former). For plates, without a sufficient resolution, the resulting mesh is unable to capture the smoothness and the results start to scatter (see supplementary document, for example Figure 28b).

A way to circumvent this issue would be to use adapted meshes, by subdividing more around the clamp. Additionally, for all models including rods, this would have the advantage to reduce the time spent in performing the test. However, for the sake of simplicity, unfortunately does not help improve results.

High $\Gamma$ values. The region $\Gamma > 10^3$ is numerically challenging for the codes as the expected shapes are rods/ribbons heavily bent at the clamp and then almost vertical. Curve-based models such as Super-Helix or Super-Ribbon can easily capture this phenomenon with few elements (50 and 20 elements, respectively). However, nodal models like Discrete Elastic Rod, LibShell or Discrete Shell + Arcsim require meshes with a fairly good resolution around the clamp (the latter required meshes twice as resolute as the former).

A way to circumvent this issue would be to use adapted meshes, by subdividing more around the clamp. Additionally, for all models including rods, this would have the advantage to reduce the time spent in performing the test. However, for the sake of simplicity, we decided in the present study to use uniform meshes for all the tested codes (except Arcsim in its original version).

Successful ®Abaqus. The ©Abaqus software easily passes the test. Note that 200 2D Timoshenko beam elements with quadratic interpolation functions have been used. Based on a Newton solver with automatic stabilization, 200 static equilibria where found within approximately 1 hour (no warm-start, only changing the length of the rod every independent computation).

Tentative fitting of Projective Dynamics. Finally, the Projective Dynamics method, which relies on arbitrary (non-physical) weights in front of its three energy terms (gravity $G$, stretch $S$, bend $B$), provides a good example of calibration offered by our method. We can indeed fit at least its bending weight against the Cantilever master curve. We take $h \ll w$, for each vertex $G = 1/3Ap$ where $A$ is the area adjacent to the vertex, $S = Eh$, and $B = aEh^3/(12 – (1 – \nu^2))$. We found that for our implementation of Projective Dynamics, $a = 0.25$ gives the best agreement to the master curve. However, the output values yielded by the Projective Dynamics code are scarce as the regime of this experiment is not favorable to the convergence of the method. In practice, we did not find any appropriate solver parameters allowing for a sampling as dense as for the other codes. Indeed, in this experiment the stretching coefficient $S$ is much greater than the bending coefficient $B$, leading to a poor convergence of the local/global scheme, which is even worse at low $\Gamma$’s where $G$ becomes much smaller than $B$. Nonetheless, the code produces fairly good results in the range $0.1 < \Gamma < 500$ beyond which it crosses the master curve and departs from it.

7.2 Results on the Bend-Twist test

The Bend-Twist test allows to challenge twist formulation of 3D rod models and displays a known numerical limit between 2D and 3D deformed configurations as explained in Section 4.3. Our three codes for rods, namely Discrete Elastic Rod, Super-Helix, and Super-Clothoid, as well as ©Abaqus, were tested on this protocol.

Figure 8 shows that all the codes except ©Abaqus perform well. However, as the protocol is much more demanding than the Cantilever test, the simulation setup needs to be carefully designed.

Successful graphics rod models. The three rod models from Computer Graphics yield perfect results, provided enough elements are used for highly curved rods (200 elements for the nodal model Discrete Elastic Rod, 30 and 20 elements for the curvature-based models Super-Helix and Super-Clothoid, respectively). The interested reader can see convergence plots in the supplementary material (Section 1).

Yet, to yield successful results, important details in the setup deserve to be emphasized. First, to generate a 3D configuration, it is generally needed to start from an initially 3D rod (otherwise there is a risk that the rod remains in an unstable 2D state). If one starts from an initially 2D curved rod, one idea is to apply a small transverse perturbation at the beginning of the simulation. Second, for this test it is crucial to make sure that dynamic models (Discrete Elastic Rod and Super-Helix) have perfectly stabilized after each simulation. Otherwise there is a high risk to label wrongly a 2D configuration as a 3D one. For large $\phi$ and $\Gamma$ (long and curvy...
7.3 Results on the Lateral Buckling test

The Lateral Buckling test can be seen as an extension of the Cantilever test to elastic plates. Although both tests are meant to evaluate the bending formulation, the former is much more demanding than the latter. Indeed, **Cantilever** is a 2D test, whereas Lateral Buckling incorporates 3D effects through the width and is subject to a pitchfork instability (transverse buckling). A code not working to a pitchfork instability (transverse buckling). A code not working to a pitchfork instability (transverse buckling).

**Difficulties for Abaqus.** Similarly to **Super-Clothoid**, we have used **Abaqus** with a static version relying on the Newton method. But as **Super-Clothoid** converges to the correct equilibria on the whole set of parameters with 20 elements, **Abaqus** gives wrong results close to the 2D/3D frontier and at large natural curvature \( \phi/2\pi \). Increasing the number of elements (up to 785 for the largest \( \phi/2\pi \) value), adding some stabilization parameters to the Newton algorithm, and using previously computed deformed configurations as warm starts, all greatly improve a phase diagram that was initially much worse than the one presented in Figure 8d. But after days of trying, we have not been able to obtain entirely satisfactory results.

**Results close to the 2D/3D frontier and at large natural curvature.**

**Failure of Abaqus.** Apart from two values of \( w/L = 0.1 \) and 0.2 for which the transition between 2D and 3D equilibrium shapes are correctly predicted by **Abaqus**, the latter returns no result for \( w/L > 0.2 \). The reason of this failure lies in the incapacity of the Newton solver to converge to the starting point of the demanding Lateral Buckling protocol. Although an equilibrium is always found when the plate is slightly tilted at \( \Gamma^* = 40 \), when one strives to remove this rotation angle, the Newton solver keeps diverging whatever the stabilization options and the mesh resolution for \( w/L > 0.2 \), so that the Lateral Buckling experiment cannot even start.

**Limitation of the Wunderlich ribbon model.** Based on a one dimensional formulation, the Super-Ribbon code nevertheless exhibits a correct linear dependence of \( F^* \) with respect to \( w/L \), although it displays a clear offset from the theoretical curve. We found out that this offset was caused by the developability constraint of the Wunderlich ribbon model, which turns out to be unrealistic in the Lateral Buckling test. Indeed, due to the orientation of the clamp, in-plane extension and shear are bound to happen. To overcome this limitation while still benefiting from a reduced formulation, it would be interesting to develop a stretchable ribbon model.

**Discriminating LibShell variants.** As in the Cantilever test, we note that the elastic law has little impact here, hence we evaluate LibShell using only the Saint-Venant Kirchhoff (SrV-K) material. All the variants of LibShell pass the test, however it appears that the discretization schemes for the discrete normals computation are discriminated by this test. Indeed, the two “second-order” models MidEdgeSin and MidEdgeTan pass at the resolution Res + while the “first-order” model MidEdgeAverage requires an even increased resolution Res ++ (see supplementary document, Figure 13).

**Successful FenicsShell.** It is noteworthy that FenicsShell successfully passes the test using default resolution meshes, in comparison with LibShell which requires higher resolution meshes. This can probably be explained by the fact that FenicsShell relies on...
we record the continuous value of the lateral displacement $d$. ACM Transactions on Graphics, Vol. 40, No. 4, Article 66. Publication date: August 2021.

such as the solver tolerance and the timestep. We set the timestep to 0.5 ms and the solver tolerance to $10^{-13}$ N. Furthermore, it is necessary to ensure the quasi-static behavior of the dynamic simulators when replicating the experiment. A detailed study on convergence of So-Bogus utilizing the Super-Helix 2D model is provided in the supplementary material, Figure 16.

No adaptivity for Argus. While the bending formulation of Super-Helix 2D was validated on Cantilever, it is not the case for Arcsim: we have seen in Section 7.1 that Arcsim’s bending energy (relying on Bridson et al. [2003]’s model) had to be fixed by a constant factor. Moreover, despite this fix, the agreement to the master curve remained unsatisfactory. Interestingly, succeeding in the Cantilever test is not necessary for Argus to pass the Stick-Slip test, which demonstrates the independence of the latter with respect to the relative scaling of the bending and gravitational energies. This way we retrieve the results previously obtained by Rasheed et al. [2020]. However, although it generates fair results, the original adaptive version of Argus does not properly match the master curve, presumably due to small impacts during adaptive refinement: we had to disable adaptivity in the Argus code to pass the Stick-Slip test (see Figures 11c and 11d), as was also done by Rasheed et al. [2020].

Insufficient accuracy for explicit solvers. Explicit frictional contact solvers from Computer Graphics, such as BRIDSON-HARMON and PROJECTIVE FRICTION, suffer from a lack of accuracy and are not able to pass the Stick-Slip test, even when using a small timestep and/or a large number of iterations. For both solvers we test a timestep down to 0.1 ms, and observe no improvement over refinement. For PROJECTIVE FRICTION we test the number of local/global iterations up to 500. Note that unlike VISCOS FRICTION, both methods do capture a stick-slip threshold, but they yield excessive sticking due to their projective nature. This is especially prominent in the BRIDSON-HARMON method, which systematically overestimates the slipping threshold. Our study thus confirms the observations previously made by Li et al. [2018] when comparing Argus to BRIDSON-HARMON. In contrast, PROJECTIVE FRICTION manages to generate a correct threshold value for moderate friction coefficients (up to $\mu = 0.25$). Yet, in accordance with Ly et al. [2020]’s observations, the model reaches a convergence plateau which binds it to a mere sticking behavior for higher $\mu$ values.

Successful results for ©Abaqus. Finally, ©Abaqus performs very well on the Stick-Slip test. We work in a 2D-space and use 40 Timoshenko isoparametric beam elements with quadratic interpolation (B22) to model Sano’s rod. The rod motion is simulated thanks to

Complementary rotation experiment. The interested reader can refer to our supplementary document (Section 5) for an additional test based on this experiment. In this complementary test, the clamped orientation is tilted by an angle $\alpha$ from the vertical direction, and we record the continuous value of the lateral displacement $d_\alpha$ as $\alpha$ is varied. This rotation test has the advantage of yielding a smooth signal in place of the present binary output of the Lateral Buckling test, thus being less demanding compared to our retained test. In particular, it can be leveraged to analyze failures on the Lateral Buckling test.

7.4 Results on the Stick-Slip test

Finally, the Stick-Slip test assesses the ability of a code to properly capture the sticking-slipping threshold of Coulomb’s friction. Figure 11 gathers our results obtained on our six different codes for a planar rod (or a plate) subject to frictional contact.

Successful So-Bogus. We first notice that the Stick-Slip test is particularly demanding, as among our tested Computer Graphics solvers, only the nonsmooth implicit So-Bogus solver is able to pass it, whether being coupled to a planar rod model like Super-Helix 2D or to a plate model like Arcsim – incorporated within Argus.

Yet, it must be noted that for successful agreement with the master curve, one has to carefully set the parameters of the simulation, such as the solver tolerance and the timestep. We set the timestep to

Fig. 9. Results of the Lateral Buckling test.

Fig. 10. Detailed results of the Lateral Buckling test for Discrete Shell and our two different mesh resolutions up to $\Gamma = 80$.

high-order elements (P$_{2,3}$), with faster convergence compared to LibShell.

Insufficient accuracy for explicit solvers. Explicit frictional contact solvers from Computer Graphics, such as BRIDSON-HARMON and PROJECTIVE FRICTION, suffer from a lack of accuracy and are not able to pass the Stick-Slip test, even when using a small timestep and/or a large number of iterations. For both solvers we test a timestep down to 0.1 ms, and observe no improvement over refinement. For PROJECTIVE FRICTION we test the number of local/global iterations up to 500. Note that unlike VISCOS FRICTION, both methods do capture a stick-slip threshold, but they yield excessive sticking due to their projective nature. This is especially prominent in the BRIDSON-HARMON method, which systematically overestimates the slipping threshold. Our study thus confirms the observations previously made by Li et al. [2018] when comparing Argus to BRIDSON-HARMON. In contrast, PROJECTIVE FRICTION manages to generate a correct threshold value for moderate friction coefficients (up to $\mu = 0.25$). Yet, in accordance with Ly et al. [2020]’s observations, the model reaches a convergence plateau which binds it to a mere sticking behavior for higher $\mu$ values.

Successful results for ©Abaqus. Finally, ©Abaqus performs very well on the Stick-Slip test. We work in a 2D-space and use 40 Timoshenko isoparametric beam elements with quadratic interpolation (B22) to model Sano’s rod. The rod motion is simulated thanks to
an explicit dynamic analysis with large Rayleigh internal damping ($\alpha = 10$) to damp the lower frequencies that could appear. Contacts between the bottom surface and the beam nodes are enforced with a kinematic contact algorithm (predictor-corrector algorithm that modifies accelerations to obtain a corrected configuration in which the contact constrains are enforced). The contact tangential behavior is a basic Coulomb friction model with static friction $\mu$. Sticking between the rod and the bottom surface is ensured by a penalty method. The minimal increment being $9.223 \times 10^{-6}$ s to ensure stability of the explicit scheme for a total time of 60 s for one Stick-Slip test, it takes over $6.5 \times 10^6$ increments for a CPU time of approximately 1 hour in total.

8 LIMITATIONS AND FUTURE WORK

In our study we have provided a rich set of protocols to validate physically any numerical simulator for rods, plates, or frictional contact. Still, our framework does not cover all possible validation facets and could be further developed to handle more complex cases.

Validation versus verification. As recalled by Museth [2020], validation queries the ability of a simulator to solve the correct physics, whereas verification deals with the correct numerical solving of a given model [Pham 1999]. In that sense, our study focuses on the physical validation of numerical codes, not on their numerical verification. Indeed, numerical verification is only meaningful once one knows exactly the (continuous) physical model from which the code derives, as well as the physical validity of this model. While this is often the case in Mechanical Engineering, which clearly decouples the use of well-established physical models from their subsequent finite-element discretization, this is not necessarily the case in Computer Graphics, where simulators may be directly written as new numerical models integrating physics and numerics altogether. Our validation methodology then allows us to treat equally simulators deriving from a continuous physical model from that directly written on a discrete form, making it applicable to any numerical model.

However, we have seen that numerical verification, such as the proof of numerical convergence, often turns out to be a prerequisite to physical validation. For instance, a suitable choice of numerical parameters (time step, number of elements) can only be done if one has the guarantee that results will improve over refinement. Convergence studies are not always conducted in Computer Graphics papers, and we encourage the authors of new models to provide them systematically in the future.

While convergence is indirectly assessed by our protocols, other relevant numerical properties are however not evaluated. Typically, important features that we miss are the objectivity of the model – i.e. its invariance w.r.t. rigid motion [Crisfield and Jelenić 1998] – and non-locking – i.e. the ability of the model to continue converging properly upon some parameter refinement, typically when the thickness of the structure tends towards zero [Arnold and Brezzi 1997]. We plan to investigate these issues in the future.

Towards more complex physical protocols. Although richer than the one commonly used, our protocols remain too elementary to evaluate the full richness of Computer Graphics simulators. To augment their validation potential, one first step would be to consider dynamical scenarios, such as vibrating structures [Virgin 2007]. More complex contacting scenarios, such as 2D and 3D confinement of buckled slender structures [Roman and Pocheau 2002], could also be considered. An extension of our methodology to the validation of thin shells would also be valuable, but finding relevant master curves, which exploit characteristic instabilities depending on a limited set of dimensionless parameters, remains a challenging task.

Finally, to be able to deal with fully integrated phenomena that are common to Computer Graphics scenarios (large fiber assemblies, cloth with frictional contact, etc.), one should develop some innovative experimental setups. We believe such a large step forward will only be possible through a strong cooperation between experts.
from the three communities Computer Graphics, Mechanical Engineering, and Soft Matter Physics. We hope our paper will foster more research in this fascinating pluridisciplinary area.

9 CONCLUSION
We have proposed a new framework to validate quantitatively the physical realism of numerical simulators for rods, plates and frictional contact. Our methodology relies on the introduction of four physical protocols inspired from experimental measurement setups of Soft Matter Physics. As these protocols exhibit macroscopic instabilities which depend on very few dimensionless physical parameters, they are characterized by a clear master curve against which any code can be robustly compared. This way, we can easily determine whether a code successfully passes each validation test, and thus evaluate its level of physical realism. We have tested a number of representative codes from Computer Graphics, as well as two reference codes from Mechanical Engineering. Our results demonstrate that even though some popular codes fail, some may be improved upon slight modification. Furthermore, remarkably a few Computer Graphics codes do succeed on all tests and even perform better than reference models of Mechanical Engineering. We hope that our study will help the Computer Graphics community better assess and diffuse their models in the future, and that the set of proposed protocols will be enriched by a joined work across the communities.

ACKNOWLEDGMENTS
We would like to thank Benoit Roman and Jose Bico for sharing insight on the Cantilever experiment, Ken Museth for fruitful discussions on validation, Rahul Narain, Jie Li, Amâncio Fernandes, and Corrado Maurini, for their help with ARCSIM, @ABAQUIS, and FENICS-SHELL respectively, Mélina Skouras and Thibaut Métivet for their careful proofreading of our paper submission, Francesco Dal Corso for providing us with useful bibliographic references, as well as the anonymous reviewers for their constructive comments. This work was supported in part by the ERC grant GEM (StG-2014-639139).

REFERENCES