Recipe Manual for Validation

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 CHARRONDIÈRE, 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

ACM Reference Format:
Victor Romero, Mickaël Ly, Abdullah-Haroon Rasheed, Raphaël Charrondière, Arnaud Lazarus, Sébastien Neukirch, and Florence Bertails-Descoubes. 2021. Recipe Manual for Validation. *ACM Trans. Graph.* 40, 4, Article 66 (August 2021), 4 pages. https://doi.org/10.1145/3450626.3459931

This manual specifies the algorithms required to reproduce the protocols presented in our paper. Please refer to the paper for their full description and the notations used. Our goal is to make these algorithms as generic and universal as possible in order to help others reproduce our results or evaluate the physical realism of other simulators, albeit following a certain number of rules. The reader should be aware that some of our tests are very demanding. Hence, depending on the simulator used, false negative results can be obtained if one does not carefully setup the simulation for reproducing the experiment. In this manual we thus give the general rules to be followed for evaluating a model correctly (these are provided as algorithms), and at the same time we attempt to give all the tips we have applied to our own benchmarking study in order to avoid false negatives as much as possible. If you build a new evaluation study using one of our tests, please keep in mind to document precisely all the important parameters (spatial resolution, time step, tolerance, etc.), as well as the various tips used: this is part of the protocol.

1 GENERAL REMARKS

1.1 Dimensionless *vs* dimensional inputs

The protocols described in the main paper compare the results of the simulation to dimensionless scaling laws. This allows to check the physical accuracy of codes whether they take dimensionless or dimensional input parameters. In addition, for dimensional codes, the dimensionless tests also allow to check how sensitive these codes are to the scale of the inputs. In theory, regardless of the scale, input parameters producing the same dimensionless parameters (e.g. Γ, ϕ) should produce the same dimensionless output. However, different parameter scales might not have the same numerical conditioning and the accuracy may start to degrade outside a certain range.

Our tests can be used to quantify this phenomenon by sampling the input parameters over a large space that redundantly covers the testing range of the dimensionless parameters.

As an example, we provide such a redundant algorithm in Section 2.4 for the **Cantilever** test where the Γ space is sampled by varying several dimensional parameters. Otherwise, the default algorithms, described in Sections 2.4, 3.1, 4.1 and 5.1 do not test this dimensionless property and provide a protocol with specific input parameters. We recommend the user take the redundant version at least for the **Cantilever** test, to make sure that her tested code is not subject to scaling issues.

1.2 Notes for dynamical codes

The **Stick-Slip** experiment is likely to require small time steps to handle contact properly. For all other protocols, as only the static equilibrium matters, the calibration of dynamical parameters can be chosen to minimize the computing time without introducing any drift or divergence problems. For example, large time steps coupled with any kind of damping could be used. However, when using a dynamic code, be sure to check that equilibrium has been reached, i.e. that the speed is almost zero, before stopping the simulation. Otherwise, it could be a (bad) reason for a KO, especially for the **Bend-Twist** test, as explained in the paper.

1.3 Notes about discrete geometry

We do not give a precise protocol for generating input geometry and depending on models it might require different features, e.g. preferences between regular grid or asymmetrical meshes. However we use a common parameter for meshes, $\xi = L_{\text{mesh}}/L_{\text{plate}}$, where L_{mesh} is the typical size of a mesh element and L_{plate} the length of the plate.

2 THE **CANTILEVER** TEST

As described in the main paper, the configuration of this test is quite simple as it requires to compute/wait for the equilibrium with an horizontal clamp.

Authors' addresses: Victor Romero, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France, victor.romero@inria.fr; Mickaël Ly, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France, mickael.ly@inria.fr; Abdullah-Haroon Rasheed, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France, haroon.rasheed@inria.fr; Raphaël Charrondière, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France, raphael.charrondiere@inria.fr; Arnaud Lazarus, Sorbonne Université, CNRS, Institut Jean Le Rond d'Alembert, UMR 7190, 4 place Jussieu, case 162,, F-75005, Paris, France, arnaud.lazarus@upmc.fr; Sébastien Neukirch, Sorbonne Université, CNRS, Institut Jean Le Rond d'Alembert, UMR 7190, 4 place Jussieu, case 162,, F-75005, Paris, France, sebastien.neukirch@upmc.fr; Florence Bertails-Descoubes, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000, Grenoble, France, florence.descoubes@inria.fr.

0730-0301/2021/8-ART66 $15.00
https://doi.org/10.1145/3450626.3459931

2.1 Algorithm for reproducibility

```
1  r ← 5.10^−4;
2  InitializeRodGeometry(L = 1, CrossSectionRadius= r);
3  InitializePhysicalParameters(E = 4/(πr^4), density ρ = 1/(πr^2));
4  for i ← −3 to 4 by 0.025 do
5  |   Γ ← 10^i;
6  |   SetGravity(0, Γ);
7  |   ComputeEquilibrium();
8  |   x, y ←coordinatesOfEndPoint();
9  |   Record(Γ, y/x);
10 end
```

Algorithm 1: Cantilever

This algorithm can be adapted to ribbons by setting the width to $w = 0.1$, the thickness to $h = 10^{-3}$, the plate bending coefficient to $D = 1$, and the density to $\rho = 1/h$. Note that this test is insensitive to the Poisson ratio ν.

2.2 Remarks about the algorithm

L.1-3 If the rod has a non-circular cross section, just set $E = \frac{1}{I}$ and $\rho = 1/A$.

L.6,8 Although models are 3D, the **Cantilever** protocol is purely 2D (see Figure 2 of the paper).

L.7 Previous solution can be used as a warm-start for the computation of the next equilibrium.

2.3 Discrete geometry characteristics

We used two variants for meshes: $\xi = \frac{1}{120}$ in Res 0 or $\xi = \frac{1}{240}$ in Res +. For our evaluated rod models, the number of elements was less than 50 for curvature-based models, and less than 300 for position-based ones. These number can serve as reference for further studies.

2.4 Redundant test

As mentioned before, the general idea is to check that the dimensionless law is recovered, regardless of the scale of the inputs (in a reasonable range).

As an example, we describe below the protocol used to test LibShell, Arcsim and Discrete Shell.

A similar warm-start strategy as described above was used, we proceeded by continuation by descending values of E for given values of L and h.

Note that this way of sampling does not guarantee that each Γ value will be tested twice or more times with different entries, but it provides a dense sampling and assumes a "continuous" behaviour of the simulator.

To test the exact same values, one could put Γ in the entries along other parameters but one (e.g. L), and compute this parameter so as to recover the wanted Γ value $\left(\text{e.g. for a rod } L = \sqrt[3]{\frac{\Gamma EI}{\rho A g}}\right)$.

```
1  lengths ← {0.5m, 0.8m};
2  thicknesses ← {1mm, 2.5mm};
3  young_moduli ← 25 log-samples in [10kPa, 10GPa];
4  ν ← 0.3;
5  ρ ← 1287kg/m^3;
6  g ← 9.81m/s^2;
7  SetGravity(0, −g);
8  for (L, h, E) in (lengths, thickness, young_moduli) do
9  |   Γ ← ComputeGamma(L, h, E, ν, ρ, g);
10 |   ComputeEquilibrium();
11 |   x, y ←coordinatesOfEndPoint();
12 |   Record(Γ, y/x);
13 end
```

Algorithm 2: Redundant **Cantilever**, relying on the sampling of dimensional parameters, combining differently to generate a dense sampling of the Γ range: all data should collapse on the same curve.

3 THE **BEND-TWIST** TEST

3.1 Algorithm for reproducibility

As for the previous test, configuring this test is relatively easy as it evaluates equilibriums under a simple boundary condition. However, as mentioned below in the remarks, because we are interested here in an instability, a perturbation might be required to reveal it.

```
1  r ← 5.10^−4;
2  InitializeRodGeometry(L = 1, CrossSectionRadius= r);
3  InitializePhysicalParameters(ν = 0.5, E = 4/(πr^4), density
   |  ρ = 1/(πr^2));
4  X ← logspace(0.105, 0.95, 30);
5  Y ← logspace(0.3, 10, 23);
6  foreach y ∈ Y do
7  |   φ = 2πy;
8  |   ResetRodShape();
9  |   Rod.SetNaturalCurvatures(τ = 0, κ1 = φ, κ2 = 0);
10 |   foreach x ∈ X do
11 |   |   RodShapeEpsilonPerturbation();
12 |   |   Γ ← (xφ)^3;
13 |   |   ComputeEquilibrium();
14 |   |   d ← Rod.3DMeasure();
15 |   |   Record(x, y, d);
16 |   end
17 end
```

Algorithm 3: Bend-Twist

3.2 Remarks about the algorithm

L.9 Only one bending curvature of the natural shape is nonzero.

L.11 Using the previous solution as a warm-start of the algorithm is a good way to shorten the convergence time to the next

equilibrium. However, when starting from a 2D configuration, algorithms have sometimes difficulties to reach 3D configurations. To resolve such a situation one could introduce some perturbation, e.g. add 0.001 to the curvatures. If the different equilibria however always remain 2D, one can try to increase this perturbation.

L.14 For the 3D measure, we suggest the use the norm of the twist of the rod $\int_{s=0}^{L} |\tau(s)| ds$. Note the second curvature is also a measure of the 3D configuration. However if a model does not handle the curvatures as a degree of freedom, an alternative measure is the lateral deviation of the center-line $\int_{s=0}^{L} |y(s)| ds$.
To transform any of these measures to a Boolean, a tolerance of $\epsilon = 10^{-3}$ can be used.

3.3 Discrete geometry characteristics

In our tests, 30 elements proved sufficient for curvature-based models. Position-based models however require a much larger number of elements, we found that at least 700 elements were needed for Discrete Elastic Rod to properly account for strongly curled configurations.

4 THE **LATERAL BUCKLING** TEST

4.1 Algorithm for reproducibility

Whether it be stable or unstable, the 2D configuration is always an equilibrium in this test. In order to avoid some of the codes to be artificially stuck in a 2D configuration, we introduce an initial rotation to prepare the system. The rotation consists in introducing an angle θ between the gravity direction, $\mathbf{e}_z$, and the clamping direction. Once gravity has been fully turned on, the initial rotation is discarded. The lateral displacement is the distance d_y as shown in Figure 4 of the paper.

4.2 Remarks about the algorithm

L.4-11 The beginning of the algorithm searches a good warm start for the procedure. Depending on the solver, the loop lines 7–10 can be adapted, e.g. for FenicsShell there are 30 steps, for Super-Ribbon only 1.

L.11 Changing the θ angle implies rotating the plate adequately. If needed, it can be done in several steps, similarly to 7–10.

L.16 The ϵ value should indicate whether the plate is flat or not, we suggest $\epsilon = 10^{-4}$. To be sure which value to choose, plotting the curve $d_y(\Gamma)$ will help. Please note that the critical Γ_c value is the locus of a pitchfork bifurcation, but that the incipient post-buckled part, $\Gamma \gtrsim \Gamma_c$, of the curve might look somehow flat.

L.17-19 The value of L being 1, the present width value indeed yields w/L.

4.3 Discrete geometry characteristics

We set two variants for meshes: $\xi = \frac{1}{50}$ in Res 0 or $\xi = \frac{1}{75}$ in Res +. For curvature based models, the number of elements should be 30.

```
1  InitializePhysicalParameters(ν = 0.35, D = 1);
2  for width from 0.1 to 1 by 0.1 do
3  |  InitializePlateGeometry(L = 1, w = width, h = 10^-3);
4  |  InitializeClampingTheta(0.2);
5  |  SetGravity(0 , 0, 0);
6  |  ComputeEquilibrium();
7  |  for Γ from 0 to 40 do
8  |  |  SetGravity(0 , 0, −Γ);
9  |  |  ComputeEquilibrium();
10 |  end
11 |  ChangeClampingTheta(0);
12 |  for Γ from 40 down to 10 by 0.5 do
13 |  |  SetGravity(0 , 0, −Γ);
14 |  |  ComputeEquilibrium();
15 |  |  d_y ←PlateLateralDisplacement();
16 |  |  if |d_y| < ε then
17 |  |  |  Record(width, Γ, 2D);
18 |  |  else
19 |  |  |  Record(width,Γ, 3D);
20 |  |  end
21 |  end
22 end
```

Algorithm 4: Lateral Buckling

5 THE **STICK-SLIP** TEST

5.1 Algorithm for reproducibility

The **Stick-Slip** test is a 2D test which is only valid for negligible gravity, so one should take zero gravity, yielding $\Gamma = 0$. Remaining degrees of freedom are the friction coefficient μ and the bending force EI/L^2. The test should be insensitive to the latter parameter, so it can be taken equal to 1.

5.2 Remarks about the algorithm

L.3 With So-Bogus and Argus we needed to take $dt = 0.5$ ms to obtain good results (see convergence plot for So-Bogus in the supplemental). Note that it is possible to perform the test for static simulators which only output equilibria, in that case there is not time step to set. Note that in our study, we have not evaluated any static simulator on **Stick-Slip**.

L.8 The initial setup is the vertical (straight) rod, with $\Delta_y = 0$

L.9 In the simulation, ϵ_y is increased with the speed $\dot{\epsilon}_y = 0.01$/sec. Consequently, there are 3333 time-steps for each iteration of the loop L.9-L.17.

L.11 For dynamic simulators: when decreasing Δ_y, the dynamical simulation should be quasi-static to avoid any effect of the speed of the rod. Damping should be used to dissipate energy and stabilize the simulation.

L.13 ϵ can be taken equal to 10^{-3}. For dynamic simulators: it is possible that the initial impact (i.e. when the rod touches the substrate for the first time) creates a small horizontal displacement ϵ_x of the rod tip at the very beginning. Such an impact can be removed by carefully positioning the rod initially so

```
1  InitializePhysicalParameters(L = 1, EI = 1, ρA = 1);
2  SetInitialRodStraight();
3  SetInitialRodClamping(0 , −L, 0);
4  InitializeTimeStep(dt);
5  SetGravity(0 , 0, 0);
6  for μ from 0 to 0.35 by 0.05 do
7  |  SetFrictionCoefficient(μ);
8  |  ResetRodSetup();
9  |  for ϵ_y from 0 to 0.6 by 1/60 do
10 |  |  ChangeClamp(Δ_y = ϵ_y/L, mode=quasiStatic);
11 |  |  r_x ←ComputeHorizontalPositionOfTheRodTip();
12 |  |  if |r_x| > ϵ then
13 |  |  |  Record(ϵ_y, μ, SLIP);
14 |  |  else
15 |  |  |  Record(ϵ_y, μ, STICK);
16 |  |  end
17 |  end
18 end
```

Algorithm 5: Stick-Slip

that it just brushes the substrate. If one does not manage to get rid of this initial impact, one should then compare $|r_x - \epsilon_x|$ against ϵ in L.13.

L.12-17 A possible variant is to decide whether the state is stick or slip by computing the contact force. This is actually the test we perform experimentally, because we easily have access to forces (but not to the friction coefficient μ). Yet when testing the codes we have not retained this variant, because it may not be applicable to all of them: some codes do not output forces. However, it can be a nice complementary test to perform for codes which do provide forces. For So-Bogus and Argus for example, we did this additional test and obtained good results as well (see our convergence plot for So-Bogus in the supplemental, computed using this technique). To use this variant, one should change L.12–17 with block algorithm 6

```
1  Q, P ←ComputeForces();
2  if Q/P > μ then
3  |  Record(ϵ_y, μ, SLIP);
4  else
5  |  Record(ϵ_y, μ, STICK);
6  end
```

Algorithm 6: Block for the **Stick-Slip** variant with force test

5.3 Discrete geometry characteristics

In the codes that use meshes, we choose $\xi = \frac{1}{50}$.