Bow Simulator version 0.1.0
User Manual

http://bow-simulator.sourceforge.io/

Contents

1 Introduction

This is the User Manual for Bow Simulator, a program for simulating the statics and dynamics of bow and arrow. It contains information about the usage of the program, explanations for input and output data as well as some background information.

For the latest version of the software and this manual visit
`http://bow-simulator.sourceforge.io/`.

2 Input Data

2.1 Bow parameters

2.1.1 Material

Two material constants are needed here: The density ρ and Young's modulus E. If you don't know Young's modulus of your material have a look at section 6 for a simple bending test.

2.1.2 String

- **Strand density:** Linear density of the string material (mass per unit length). See the examples below for two ways to get this value.

  Example 1: String material with diameter $d = 0.5\text{mm}$ and density $\rho = 1.38\frac{\text{g}}{\text{cm}^3}$. Calculation of the linear density:

  $$\begin{aligned}\text{Linear density} = \rho A &= \rho\,\frac{\pi}{4}\,d^2 \\ &= 1.38 \cdot 10^{-3}\frac{\text{kg}}{\text{m}^3} \cdot \frac{\pi}{4}(0.5 \cdot 10^{-3}\text{m})^2 = 0.00027\frac{\text{kg}}{\text{m}}\end{aligned}$$

  Example 2: A piece of string material with length $L = 150\text{cm}$ weighs $m = 0.405\text{g}$. Calculation of the linear density:

  $$m = \rho A\,L \quad \Leftrightarrow \quad \rho A = \frac{m}{L} = \frac{0.405 \cdot 10^{-3}\text{kg}}{150 \cdot 10^{-2}\text{m}} = 0.00027\frac{\text{kg}}{\text{m}}$$

- **Strand stiffness:** Stiffness of the strands against elongation (force per unit strain). In static analysis the stiffness does not have much effect as long as it is large enough to prevent significant elongation of the string. In dynamic analysis however it can be an important parameter.

2.1.3 Masses

- **String center:** Additional mass at the string center to account for serving and nocking point.
- **String tip:** Additional mass at the ends of the string to account for the serving of the ears.
- **Limb tip:** Additional mass at the limb tips to accouont for overlays.

2.1.4 Operation

- **Brace height:** Distance between string center and the coordinate origin ($x = 0$, $y = 0$) in braced condition.
- **Draw length:** Distance between string center and coordinate origin in fully drawn state.
- **Arrow mass:** Total mass of the arrow.

2.1.5 Profile

The profile describes the shape of the bow in unbraced state. The curve you can edit here is the back of the bow. Use the table on the left to edit the profile curve. Each row specifies an arc-segment with a certain length and curvature. (The curvature is the inverse of the segment's radius. For curvature = 0 you get a straight line.)

If your bow has a stiff middle section/riser you can use the x-, y- and angular offsets to change the starting point and orientation of the limb.

2.1.6 Width and Height

Here you can specify the width and height of the limb's cross sections. Use the table on the left to specify a number of values at certain positions along the limb. A smooth curve (cubic spline) passing through the supplied values is constructed. The position parameters can be anything you want, they just have to be ascending. They are mapped to the limb profile curve in such a way that the first position corresponds to the start of the limb and the last one to the limb tip. You could use for example percentages or lengths.

2.2 Numerical settings

The numerical settings can be accessed via the menu or the toolbar. You can use them to fine-tune the simulation and to balance accuracy versus computing time. For most use cases the default values should be fine though.

2.2.1 Limb and String elements

The number of finite elements into which limbs and string are divided. With more elements the approximation gets better but also the computing time increases. When in doubt, increase the number of elements gradually until the simulation results do not change significantly anymore.

2.2.2 Draw steps

Number of steps that are performed by the static simulation from brace height to full draw. This determines the resolution of the static results. If you're only interested in the fully drawn state or the dynamics you could set this to something very low.

2.2.3 Time span factor

This controls how long the dynamics of the bow are simulated. A value of 1 corresponds to the time in which the arrow travels from full draw to brace height. The default value is larger than 1 in order to capture some of the things that occur after the arrow left the bow, like for example the maximum dynamic loads on limb and string.

2.2.4 Time step factor

To understand this parameter, some details about the dynamic solution method are necessary. When carrying out the simulation the program will use the current state of the bow at time t to calculate the next state at time $t + \Delta t$ where Δt is some small timestep. This is repeated over and over until the desired time span has been simulated. The timestep Δt has to be chosen small enough to get an accurate and stable solution (i.e. that doesn't "explode" numerically) but also as large as possible to keep the number of steps that have to be performed low. The program can calculate a crude estimation for the optimal timestep, but to be on the safe side this estimation is reduced by a safety factor between 0 and 1. This is what you can manipulate here. The default value has been chosen relatively low, favouring robustness over perfromance. Usually the simulation can be safely sped up by increasing this value.

2.2.5 Sampling time

This value limits the time resolution of the dynamic output data. This is done because the simulation time steps can be quite small. Not including all ot them in the output saves memory and computing time.

2.3 Comments

The comments box can be accessed via the menu or the toolbar. It is meant for documentation. Any notes or remarks about the bow and the simulation can be added here.

3 Output

3.1 Static analysis

- **String length:** Length that the string must have so that the bow meets the required brace height
- **Final draw force:** Draw force in fully drawn state
- **Drawing work:** The work done by drawing the bow. This is equal to the area under the draw curve.
- **Storage ratio:** This is an indicator of the bow's capability to store energy and is defined as

  $$\text{storage\_ratio} = \frac{\text{drawing\_work}}{1/2 \cdot \text{draw\_force} \cdot (\text{draw\_length} - \text{brace\_height})}.$$

  It's the amount of energy stored by the bow in relation to the energy that would have been stored if the draw curve was linear. Values > 1 are good.

3.2 Dynamic analysis

- **Arrow velocity:** Final velocity of the arrow
- **Arrow energy:** Final kinetic energy of the arrow
- **Efficiency:** Energy conversion efficiency of the bow, ratio between energy input (static drawing work) and useful energy output (kinetic energy of the arrow)

4 Theory: The Bow Model

This section is intended to give interested users an idea about how the program works internally, what the physical model for the bow is and what the assumptions and limitations are. This section will soon be replaced by a separate and complete technical documentation of the simulation model.

Limb:

The limb is regarded as an Euler-Bernoulli beam, this means that all cross-sections of the beam are assumed to stay flat and perpendicular to the beam axis during deformation. Euler-Bernoulli beam theory only accounts for bending deformation and neglects the influence of shear, which is a valid thing to do for long, slender beams. The material of the limb is considered linear-elastic, the relation between material stress σ and strain ϵ at any point in the limb given as $\sigma = E\,\epsilon$ with Young's modulus E. The overall behaviour of the limb however is nonlinear for geometrical reasons, as arbitrarily large deformations are allowed. For the dynamic simulation an additional point mass at the limb tip is used to account for overlays and the like.

String:

Contrary to the limb, the string only transfers longitudinal forces and has no flexural rigidity. The material is considered linear-elastic as well. The string has a constant cross section and is internally implemented as a chain of point masses connected by springs. Additional point mass at the center and the tips represent servings and nocking point.

Arrow:

The arrow is modeled as another point mass. This means that the characteristic bending motion of the arrow known as "archer's paradox" is neglected. However, as the scope of this program is not the finetuning of the arrow but to evaluate the bow's overall performance (i.e. final arrow velocity, degree of efficiency, etc.), this is sufficient.

Symmetry:

The bow is assumed to be symmetric. This is often an approximation, as most bows besides crossbow prods are actually slightly asymmetric. The assumption of symmetry simplifies the definition of the parameters by the user (no need to define the limb twice). It also allows the program to simulate only one half of the bow, which reduces the computing time. As a user you don't have to take this into account, all input and output data of the program corresponds to the complete bow.

5 Validation

It is quite important to make sure that the results obtained by simulation agree reasonably well with real world examples. This section shows the validation efforts made so far. It is still very sparse, so if anyone has used this program for a real world application, let me know about your results and they will be added here.

5.1 Statics of a straight steel bow

In this experiment the draw curve and limb shapes of a small steel bow have been measured and compared to version 2014.4 of Bow Simulation Tool (now Bow Simulator). The bow is shown in figure 1 and has been made from an old saw blade. Spring steel is a good material for this kind of test, because its mechanical properties are homogenous and well-known. Even though the exact steel type is unknown, a Young's modulus of $E = 210$ GPa will most likely be a good estimate.

Figure 1: Steel bow. Cross section: 16.85×0.75mm. Length: 269mm. Brace heighth: 49.8mm

The experiment was done by hanging the bow up on a thread and hanging a plastic bag at the string center. The draw force was then gradually applied by counting steel balls with a known mass into the bag. After every load step the draw length was measured a photo of the bow was taken.

Figure 2 shows a comparison between the measured and the simulated draw curve and 3 compares the pictures of the limb against the simulated limb shapes.

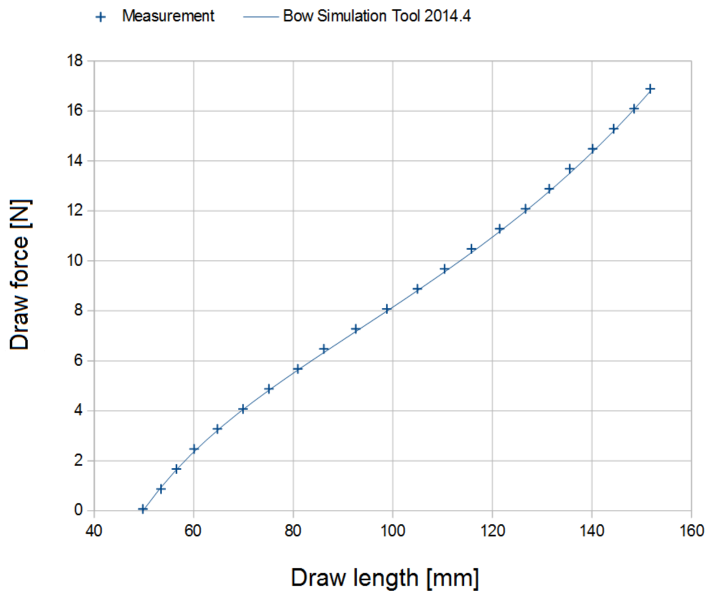


Figure 2: Experimental and simulated draw curves

The agreement between experiment and simulation is surprisingly good here. It really shows the potential of such kinds of simulations, provided that the material properties are well known and a the bow can be built exactly as simulated, with low tolerances. But this is still a very simplistic example. The next step would be to repeat this experiment with bows that have varying cross sections and non-straight profiles. Another open question are the dynamic simulation results. It's unclear if they can match experiments as good as the static results do, because the dynamic simulation has much more uncertainties.

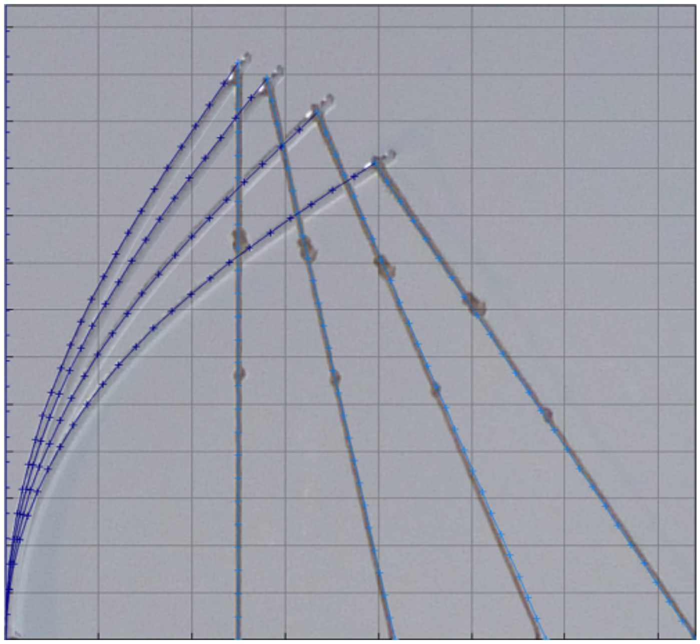


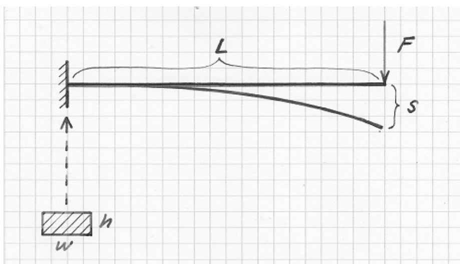
Figure 3: Experimental and simulated limb shapes

6 A Simple Bending Test

A bending test is an easy way to determine Young's modulus of a material without any special equipment. In the test considered here a ledge with length L is clamped on one side and subjected to a vertical force F at the free end. The deflection s due to this load is measured. Young's modulus can then be calculated using the equations below.

<table>
<tr>
<td>

Constant cross sections

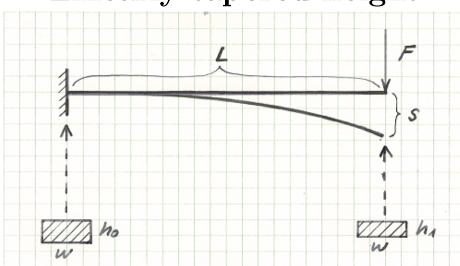


</td>
<td>

$$E = \frac{4}{wh^3}\,\frac{FL^3}{s}$$

</td>
</tr>
<tr>
<td>

Linearly tapered height



</td>
<td>

$$E = \frac{12\,\ln(h_1 L)+6}{w\,(h_1-h_0)^3}\,\frac{FL^3}{s}$$

</td>
</tr>
</table>

A few practical considerations:

- A simple way to apply a defined load is to hang a mass m onto the beam tip and use $F = m\,g$, with $g = 9.81\,{}^{\mathrm{m}}/_{\mathrm{s}^2}$.
- The equations above hold for long, slender beams and small deflections. The test setup should be chosen accordingly.
- If there is some small initial deflection due to gravity, then s is the difference in deflection after application of the tip force.