

Historical martial arts and weapons physics

[Home](#) [Documenting the dynamics of swords](#) [Posts](#) [Links](#) [About](#)

Simulating Sword Properties

Posted by **Vincent**

Over the years the topic of the simulation of the properties of swords has come up a number of times. I happen to have worked on that a few years ago, and this post describes the results I have obtained.

I have built a small [Scilab](#) program to simulate blade harmonics and dynamic properties (pivot points, center of gravity, mass), taking as inputs parameters from the materials (density, elasticity) and geometry (lengths and cross-sections). I chose a simple model for the sword, in fact one of the simplest models for slender, flexible beams (the name of the model is [Euler-Bernoulli](#)), that allows a varying cross-section along the length of the beam. The program solves the equation from the model using a 1D finite elements approach. In addition to rigid-body dynamic properties (mass, center of gravity, pivot points), it finds the modes of vibration, assuming that the sword is not constrained by anything. That is an appropriate model to simulate the very relaxed grip that is assumed when looking for vibration nodes.

Being satisfied with the behavior on basic test cases (uniform bar, simple taper, and so on), I decided to go for something a bit less obvious and tried to simulate a real sword of mine: my Angus Trim type XI. To be honest, I was expecting a more or less spectacular failure.

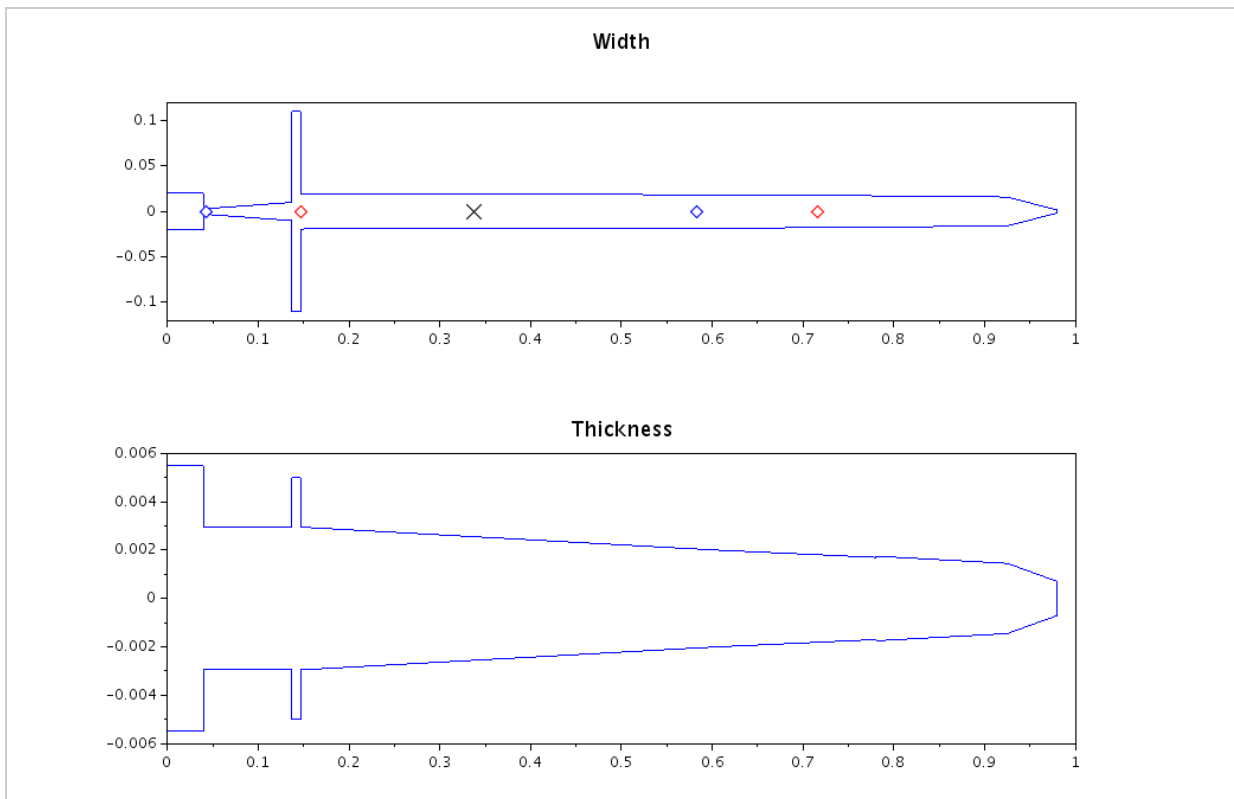
I measured as precisely as I could cross sections across the blade, as well as the pommel and cross-guard. I didn't want to go through the trouble of unmounting the sword so I made some reasonable guess for the tang based on what I remembered. I completely neglected the grip.

I ended up with 7 parts for the sword, over which I assumed the variation were linear, that I further subdivided in elements roughly according to their length:

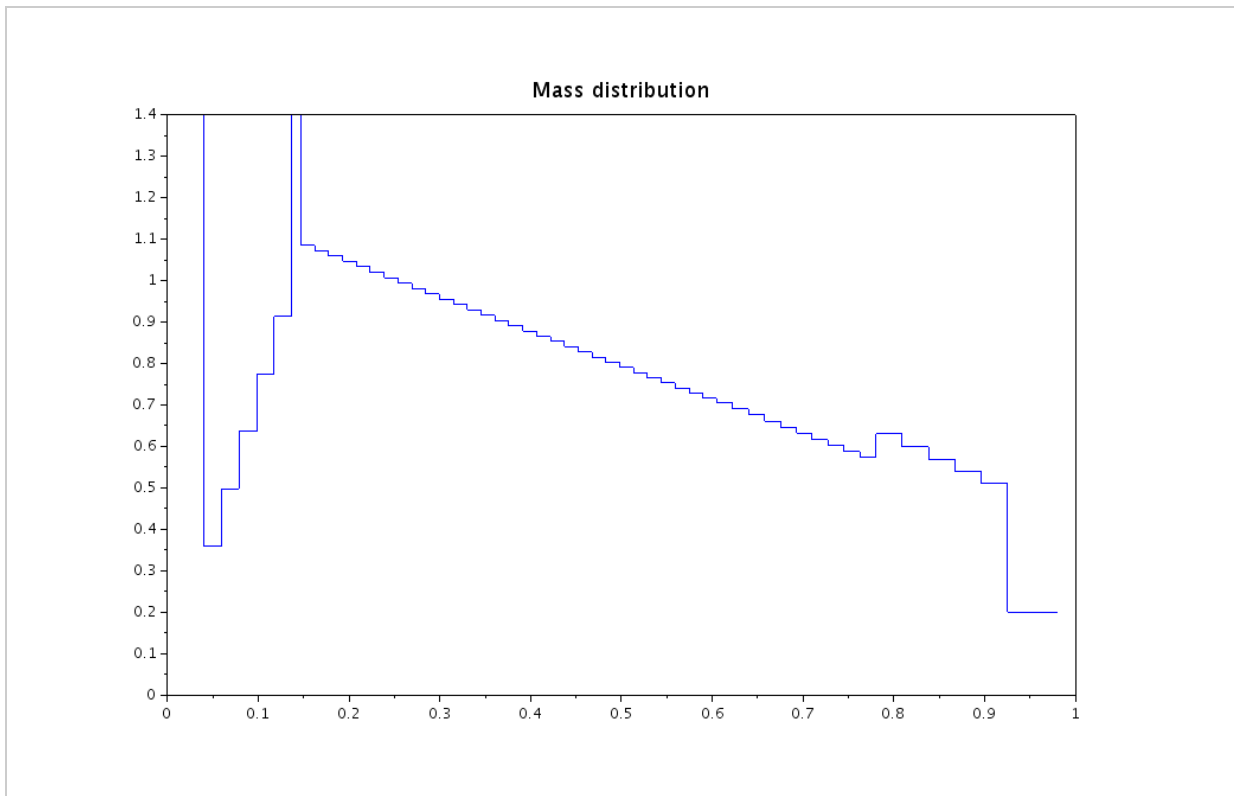
- pommel (1 element)
- tang (5 elements)
- cross (1 element)
- upper part of the blade (30 elements)
- middle part of the blade (10 elements), the fuller stops here
- upper part of the blade (5 elements), not fullered

- tip (1 element)

I entered all this into my program, along with average values of density and Young modulus for steel, aft and forward positions on the grip (to compute aft and forward pivot points). I was then able to compute an approximate mass distribution, and rigid body dynamic properties:



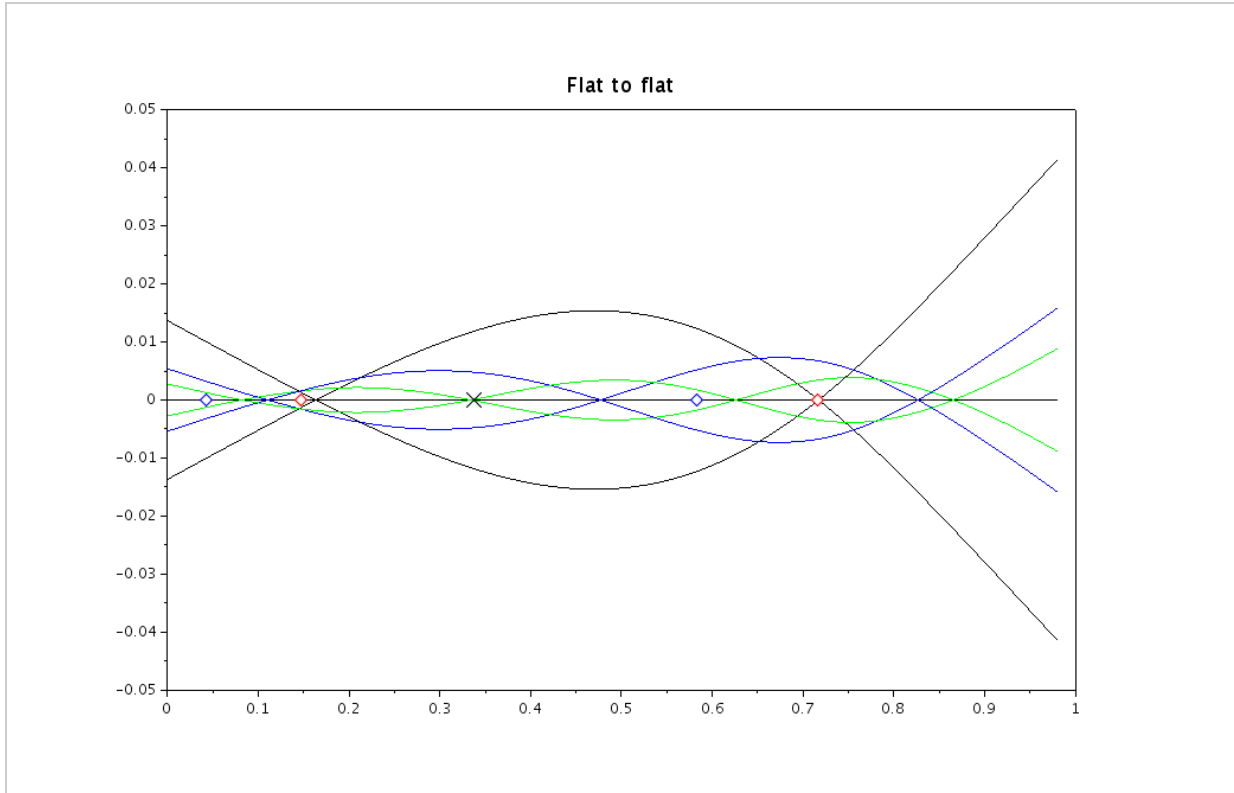
The variations of width and thickness used as inputs for the program.



The computed mass distribution (I have cropped the peaks at the cross and pommel)

The center of gravity is marked here with a black cross. The aft pivot point pair is represented by blue diamonds, the forward pivot point pair by red ones. The dynamic properties are found with a precision better than half a centimeter (far better than I can measure anyway). I just tweaked slightly the pommel shape to adjust even better (when I say slightly, it's 1mm in thickness...). And that is with all the approximation made, and a very rough model of the fuller.

My program can compute modes of vibration for the sword, which of course gives the nodes in particular.



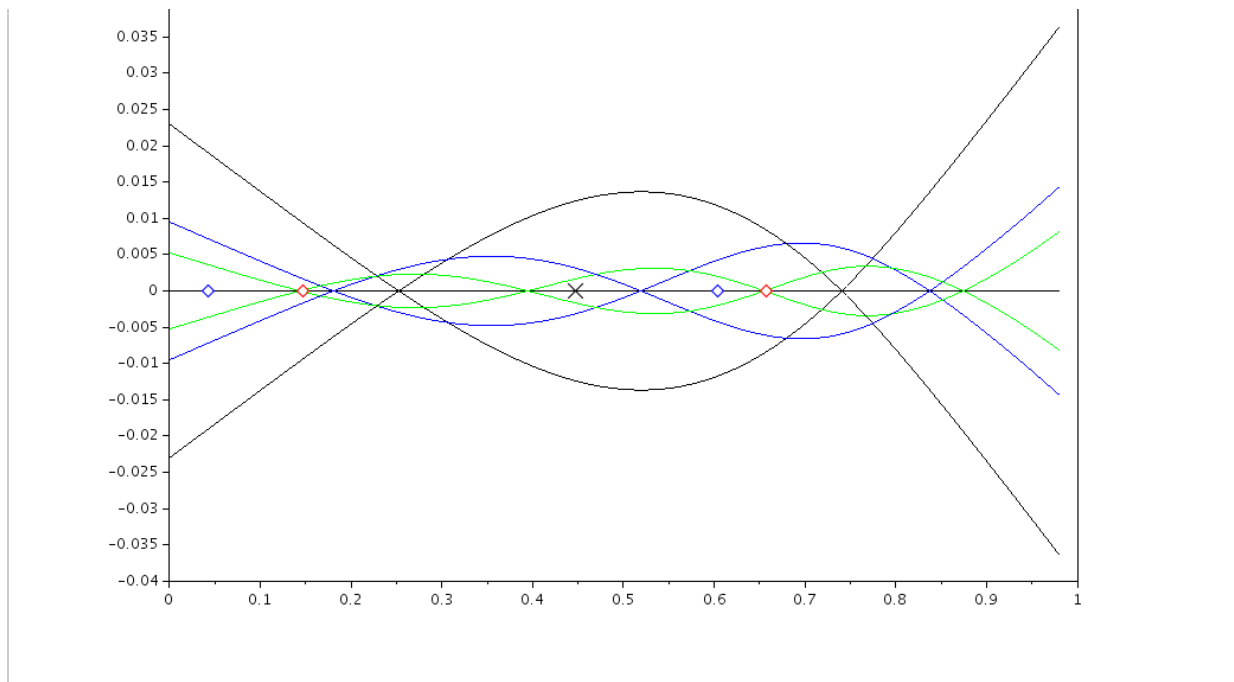
The modes of vibration, computed here in the flat to flat plane, which is the one we are able to observe.

The straight black line is the axis of the undisturbed sword. The tip of the sword is to the right. The curves represent the modes of vibration. The first mode, that is used to determine the blade node and hilt node, is in black. The curves are scaled according to the energy of the sword when it vibrates in that mode: the higher the mode, the less ample the vibration becomes. I plotted only the three first modes to keep the figure clear. For each mode, I made two symmetric curves, that indicate the deformation of the sword bending one way, then the other. The harmonic nodes are precisely computed, from what I can see. The blade node is exactly right, the hilt node is a bit more difficult to measure, but seems to be spot-on as well.

I had to adjust the density slightly to account for the mass of the sword. My first simulation was giving a mass of 950g, while my sword weights 1009g. This uniform change does not moves the nodes or pivot points at all (this is a mathematical property of the model I use). It might indicate that my measurements were not perfectly accurate, which is likely given the approximation made especially on the cross-section shape.

Interesting virtual experiments can be done with this program. For example, the pommel and cross can be virtually removed. Here is how the properties change:





Modes of vibration of the bare blade, virtually removing cross and pommel.

I remain convinced that the interest of such simulation is limited, because the perceptible results (dynamics, nodes of the first mode of vibration) are easily measured on swords. However, it shows what is sufficient to obtain the observed effects. It also disproves a relatively widespread idea on this matter, that it should take an army of scientists and a cluster of supercomputers to accurately model and simulate the behavior of a sword because they are somehow “infinitely complex”. I must stress that I did all that using well-known mathematical methods and the simplest models, implemented in the most direct fashion, and that the core of the computation runs in about 0.1 second on my laptop (far less than what it spends plotting the results, in fact. The total is about 1s...). It proves me wrong as well, because before doing this I used to think that the number of cross sections that should be measured would have to be far greater...

Most modern 3D CAD programs would make these computations easier, and probably more accurate and stable as well. I have mainly done this code to gain some experience in finite element modeling on a topic I enjoy. I'm not really expecting this to be robust or easy to use. [Find the code in this zip file!](#) Just unzip it, launch Scilab, set the working directory to the folder containing swordSimulFunc.sce, edit swordSimul.sce, or even create a new file as needed and execute it. I do not have time to support or document this much more, so you will be pretty much on your own if you want to dig deeper. If you do end up with something valuable, please share!

2017/06/16 in Physics and biomechanics, Weapon design. Tags: pivot points, simulation, sword, vibration nodes

Related posts



Modelling impacts and damage



Measuring swords as pendulums



The Sword – Form & Thought

[← Measuring swords as pendulums](#)

[A Plan of Thibault's Académie de l'Espée →](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website



Save my name, email, and website in this browser for the next time I comment.

Confirm you are NOT a spammer

Recent Comments

Matthieu Dutheil on [Etude de rapières historiques](#)

Vincent on [On Silver's times](#)

Hamza on [On Silver's times](#)

James vandsandt on [Capoferro weapon length](#)

Dylan Thomas on [My approach of HEMA](#)

Subscriptions

[RSS for Entries](#)

[RSS for Comments](#)

[Facebook page](#)

Tags

[Alfieri](#) [approach](#) [blunt](#) [Capoferro](#) [centre of oscillation](#) [centre of percussion](#) [comparison](#) [convention](#) [cutting](#)
[definition](#) [Di Grassi](#) [distance](#) [dynamic](#) [elbow](#) [entering](#) [Fabris](#) [geometry](#) [Giganti](#) [goals](#) [graphical](#) [hema](#) [history](#) [i33](#)
[inertia](#) [instance](#) [interpretation](#) [mass distribution](#) [mechanics](#) [motion](#) [personal](#) [physics](#) [rapier](#) [science](#)
[sharp](#) [shoulder](#) [Silver](#) [speed](#) [sword](#) [Thibault](#) [time](#) [viggiani](#) [visual](#) [waggle test](#) [weighing](#)
[wrist](#)

© 2013-2019 [Vincent Le Chevalier](#), All Rights Reserved | Theme based on Expound by [Konstantin Kovshenin](#)