Bowed string motion

Carl Olsson

May 8, 2014

Part I

Motivation

The violin has captivated physicists, notably Hermann von Helmholtz. As the instrument that sounds closest to the human voice, the violin is one of the most captivating and unique musical instruments. Apart from most string instruments, which are plucked, the violin is bowed; This means the musician draws a bow of horsehair over the string. The tone produced can vary greatly depending on the bowing technique used: spiccato, legato, détaché, sul ponticello, etc. As a violin player myself, it is very interesting to consider how the physical parameters that shape the tone produced by a bowed string.

Part II

Plucked strings

To begin to understand the physics of bowed string motion I first modeled the motion of a plucked string. For any string, the central equation of wave motion is given as

$$\frac{\delta^2 y}{\delta t^2} = c^2 \frac{\delta^2 y}{\delta x^2} \tag{0.1}$$

Where c is the wave velocity. Next, we consider the string as a series of discrete elements, such that each element has a length Δx and mass $\mu \Delta x$, where μ is the mass per unit length. If we rewrite Equation (0.1) in terms of a single string element and multiply by the mass of a string element, $\mu \Delta x$, we get

$$(\mu \Delta x) \frac{\delta^2 y(i)}{\delta t^2} = (\mu \Delta x) \, c^2 \frac{\delta^2 y(i)}{\delta x^2} \tag{0.2}$$

Incidentally, the left hand side of Equation (0.2) is just mass times the acceleration of the string element i, or simply the force on this string element. The right hand side of Equation (0.2) is the transverse component of the tension force on element i from its two adjacent elements. Writing this wave equation in finite difference form, we get

$$y(i, n+1) = 2\left[1 - r^2\right] y(i, n) - y(i, n-1) + r^2 \left[y(i+1, n) + y(i-1, n)\right] \quad (0.3)$$

Where i is the length along the string, n is the time step, and $r \equiv \frac{c\Delta t}{\Delta x}$. Using this equation, I can succesfully model the motion of a plucked string. Since we are dealing with a second-order differential equation, we require two initial conditions (i.e string profiles at times $n = 0, 1$). To do so, I modeled the initial pluck by giving the string a triangular profile, y_0, described by an amplitude and where along the string it is plucked and assumed that the string is held fixed with shape $y_0(x)$ prior to $t = 0$. The pseudocode for modeling the motion of a plucked string is below.

Pseudocode for plucked string motion

- Set the parameter combination $r = \frac{c\Delta t}{\Delta x}$
- For each time step n
  - Loop through the interior points along the string's length $i = 1$ through $i = M - 1$, excluding the ends of the string, which are fixed at a displacement of $y(0, n) = y(M, n) = 0$
  - Update the interior points' displacement according to Equation (0.3)
    * $y(i, n+1) = 2\left[1 - r^2\right] y(i, n) - y(i, n-1) + r^2 \left[y(i+1, n) + y(i-1, n)\right]$
  - The ends of the string at $i = 0$ and $i = M$ are fixed so $y(0, n) = y(M, n) = 0$ for all time steps n

Using this, I simulated the motion of a plucked violin string according to an initial triangular string profile. My results are shown below.

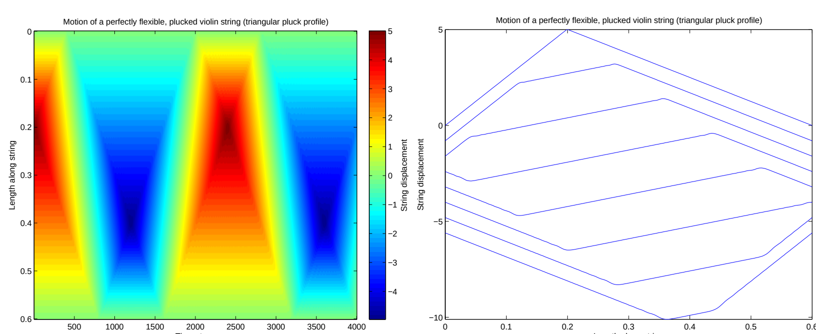

Figure 0.1: Motion of a perfectly flexible, plucked violin string

For the simulation, c or wave velocity is constant throughout the length of the string because violin strings are typically of uniform thickness and weight along their length. As is expected, after the string is released the initial kink in the string splits into two seperate kinks which propagate towards either end of the violin string. These kinks are reflected and inverted at the ends and after one half of a period they add together as an inverted version of the initial pluck.

In real life however, violin strings are not perfectly flexible and there will always be a stiffness force that opposes the bending of the string. Stiffness is added to the model by adding a term to the original wave equation and the corresponding finite difference form where ϵ is the stiffness and M is the number of spatial units (i.e $M = \frac{L}{\Delta x}$).

$$\frac{\delta^2 y}{\delta t^2} = c^2 \left(\frac{\delta^2 y}{\delta x^2} - \epsilon L^2 \frac{\delta^4 y}{\delta x^4} \right) \tag{0.4}$$

$$\begin{aligned} y(i, n+1) = & \left[2 - 2r^2 - 6\epsilon r^2 M^2\right] y(i,n) - y(i, n-1) \\ & + r^2 \left[1 + 4\epsilon M^2\right] \left[y(i+1, n) + y(i-1, n)\right] \\ & - \epsilon r^2 M^2 \left[y(i+2, n) + y(i-2, n)\right] \end{aligned} \tag{0.5}$$

Since Equation (0.5) involves the displacement at sites ± 2 units away from the site in question means we have to alter our boundary conditions. As used in the detailed modeling of piano strings, I assumed that the ends of the string are "hinged" [1]. This means we take the displacement at each end to be zero and assume there are phantom locations one unit beyond the ends which have displacements that are opposite the displacements at locations one unit inside the ends. In other words, the string ends have displacement $y(0, n) = y(M, n) = 0$ as usual, and the phantom locations have displacement corresponding to $y(-1, n) = -y(1, n)$ and $y(M+1, n) = -y(M-1, n)$.

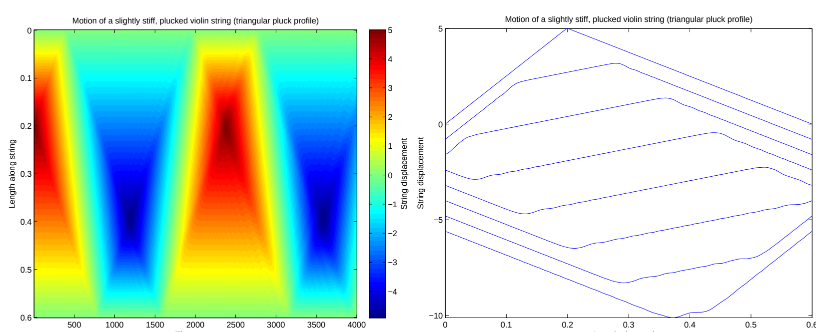

Figure 0.2: Motion of a slightly stiff, plucked violin string ($\epsilon = 1 * 10^{-5}$)

As is expected, the motion of a slighly stiff, plucked violin string is very similar to Figure 0.1. However, you can see the effect of string stiffness at the wave peak, which is more rounded than that of the perfectly flexible, plucked violin string.

Part III
Bowed string motion

As described by Helmholtz, bowed string motion consists of a "stick-slip" process as the bow's horsehair "sticks" to the string until the tension from the string overcomes the frictional forces between the horsehair and the string, at which point the bow's horsehair begins to "slip" across the string. Helmholtz discovered that the frequency of bowed string motion is the same as the as a plucked string.

We assume that the string is bowed at a point that is a distance βL from the end of the string, where β is a ratio of the string length, L. As the bow begins to move across the string at time $t = 0$, the string "sticks" and moves with a velocity equivalent to that of the bow, namely v_{bow}. This abrupt transition to sticking excites waves that travel away from the point of contact with the bow towards the two ends of the string. These waves are reflected and result in an additional force between the bow and the string that can cause string to "slip" from the bow (if the force is large enough). We assume that there are two coefficients of friction, one when the string is either "sticking" μ_s, and one when the string is slipping, μ_k. For stick-slip motion to occur $\mu_s > \mu_k$, which makes sense since the coefficient of kinetic friction is almost always greater than the coefficient of static friction.

To create our equation of motion, we reuse the single element wave equation, Equation (0.2). If we include an externally applied force, F_h, (such as the friction force produced by a bow), we can simply add it to the right hand side of Equation (0.2).

$$(\mu \Delta x) \frac{\delta^2 y(i)}{\delta t^2} = (\mu \Delta x) c^2 \frac{\delta^2 y(i)}{\delta x^2} + F_h \tag{0.6}$$

If we include string stiffness and put the resulting equation into finite difference form, we have

$$(\mu \Delta x) \frac{\delta^2 y}{\delta t^2} = (\mu \Delta x) \quad c^2 \left(\frac{\delta^2 y}{\delta x^2} - \epsilon L^2 \frac{\delta^4 y}{\delta x^4} \right) + F_h \tag{0.7}$$

$$\begin{aligned} y(i, n+1) = & \left[2 - 2r^2 - 6\epsilon r^2 M^2\right] y(i,n) - y(i, n-1) \\ & + r^2 \left[1 + 4\epsilon M^2\right] \left[y(i+1, n) + y(i-1, n)\right] \\ & - \epsilon r^2 M^2 \left[y(i+2, n) + y(i-2, n)\right] \\ & + \frac{(\Delta t)^2}{\mu \Delta x} F_h \end{aligned} \tag{0.8}$$

Depending on whether the string is slipping, the external force F_h will change in Equation (0.8). When the string is slipping the force has magnitude

$$F_{slip} = \mu_k N \tag{0.9}$$

and when the string is sticking we know that the force is no greater than $\mu_S N$, thus

$$F_{stick} \leq \mu_s N \tag{0.10}$$

where N is the normal force. The pseudocode for modeling the motion of a bowed string is below.

Pseudocode for bowed string motion

- Let the velocity of the bow be v_{bow}, and the string displacement at the bowing point be $y(i_{bow}, n)$ where n is the time step
  - Start with the string in the "sticking" state, with the string velocity at the bowing point equal to v_{bow}
  - Advance one time step and assume that the string continues to stick: $y(i_{bow}, n+1) = y(i_{bow}, n) + v_{bow}\Delta t$
  - Use Equation (0.8) to calculate the value of the force required to obtain $y(i_{bow}, n+1)$. If the required force is less than the limit in Equation (0.10), then the string continues in the "sticking" mode
  - If the required force is larger than the limit in Equation (0.10), then the string goes into the "slipping" mode and the force is given by Equation (0.9). One must then recalculate $y(i_{bow}, n+1)$ using Equation (0.8) with the force F_{slip}
  - The motion of the rest of the string is calculated as before with Equation (0.8) using an external force of zero, $F_h = 0$
  - Repeat this procedure for the desired number of time steps

Using this, I simulated the motion of a bowed string

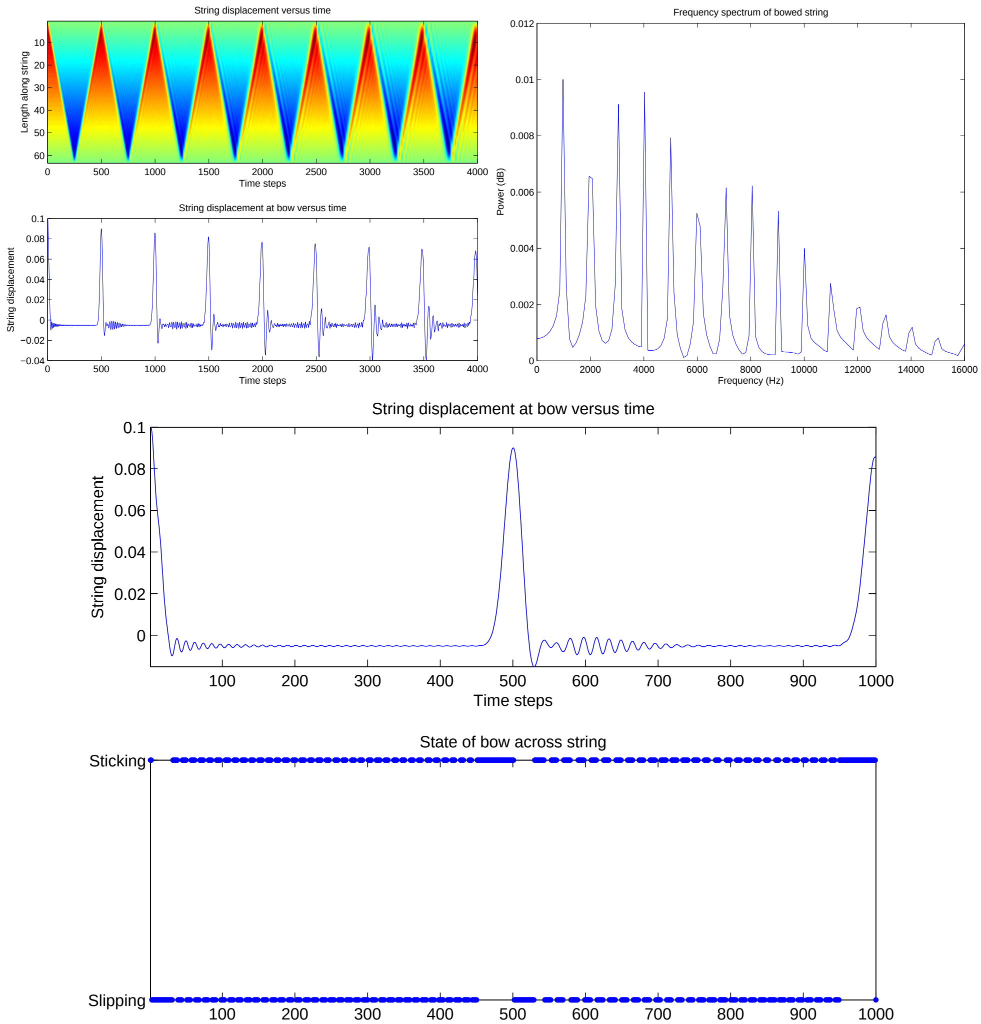

Figure 0.3: Motion of a slightly stiff, bowed violin string. The parameters used here are $L = 0.15$ m, $\beta = 0.06$, $\Delta x = 0.06$ mm, $c = 300\frac{m}{s}$, $r = 0.25$, $\mu_s = 0.9$, $\mu_1 = 0.55$, $\mu_2 = 0.35$, $v_{bow} = 0.2\frac{m}{s}$, $\epsilon = 1 * 10^{-5}$, and $N = 0.65$ N. I calculated μ_k using $\mu_k = \mu_1 exp\left(\frac{-v_{bow}}{v_1}\right) + \mu_2$ where $v_1 = 0.1\frac{m}{s}$

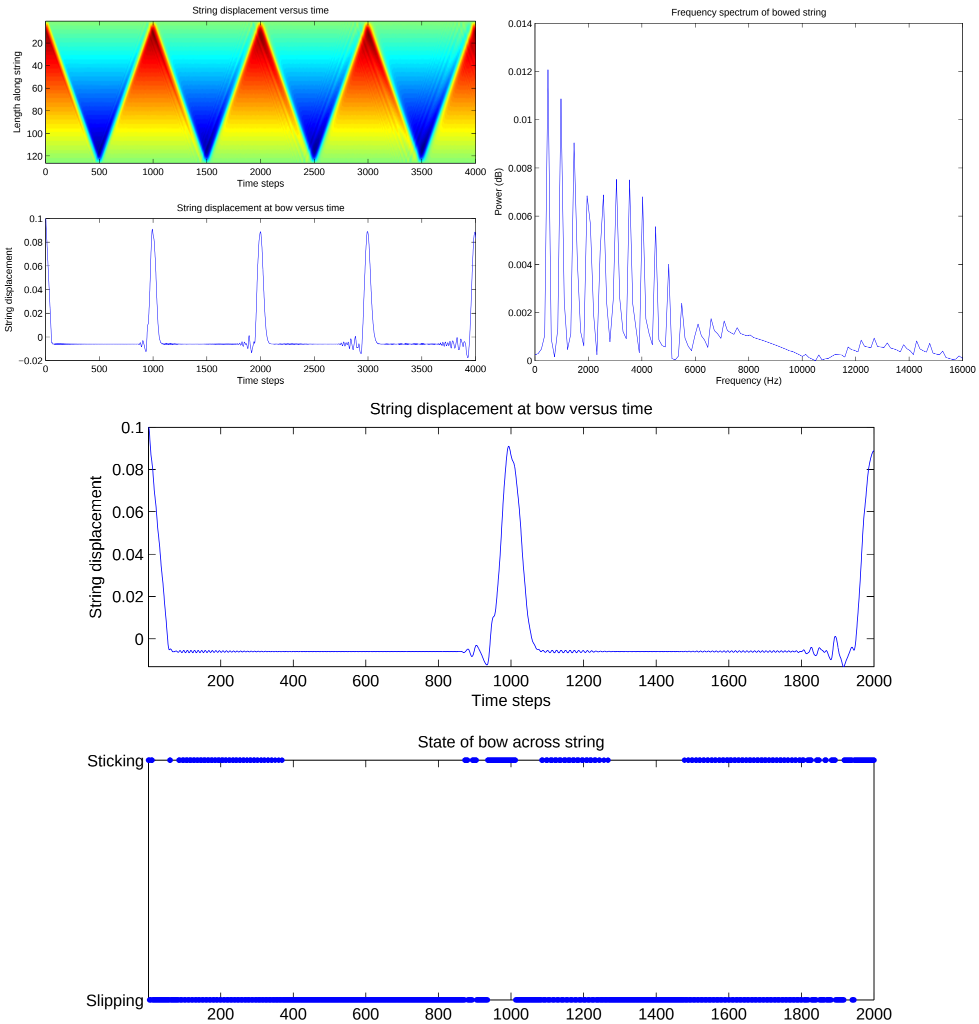

Figure 0.4: Motion of a slightly stiff, bowed violin string. The parameters used here are the same as before but with $L = 0.3$ m

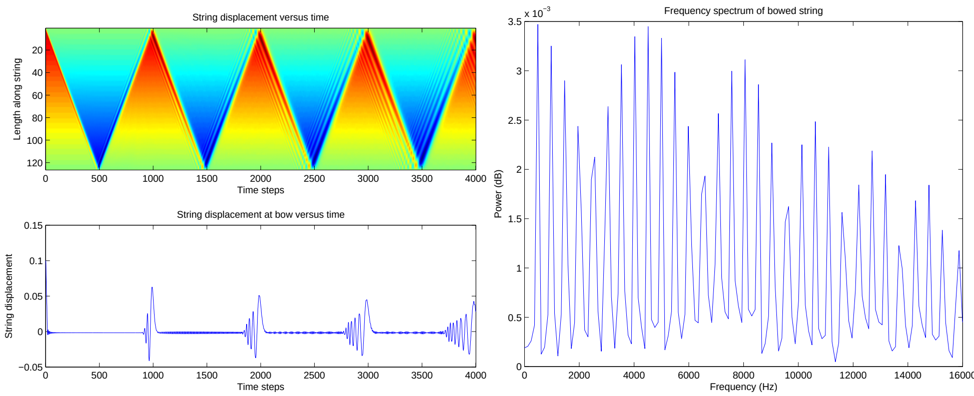

Figure 0.5: Motion of a slightly stiff, bowed violin string. The parameters used here are the same as before but with the bow position moved towards the bridge, $\beta = 0.02$. This is to simulate playing using the "sul ponticello" technique, which means "on the bridge" in Italian. Bowing the string over the bridge makes it virtually impossible to set up stable, regular Helmholtz motion, and rather easy to excite, at least briefly, some harmonic Helmholtz motion. Notice how the frequency spectrum changes, gaining many high frequency harmonics and having a weak fundamental frequency. This illustrates why playing "sul ponticello" results in an eerie sound that sounds weak and thin and is full of high frequency harmonics.

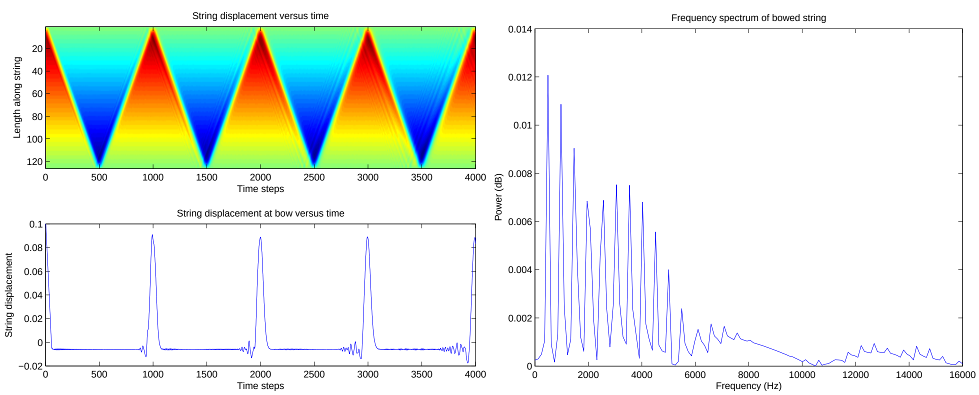

Figure 0.6: Motion of a slightly stiff, bowed violin string. The parameters used here are the same as before but with the the friction coefficients decreased drastically: $\mu_1 = 0.25$ and $\mu_2 = 0.15$. This is to simulate playing without rosin, a sticky substance used to increase the horsehair's stickiness. It is very similar to playing with rosin and the frequency spectrum validates this.

References

[1] Antoine Chaigne. Numerical simulations of piano strings. I. A physical model for a struck string using finite difference methods, 1994.

[2] Antoine Chaigne. Numerical simulations of piano strings. II. Comparisons with measurements and systematic exploration of some hammer-string parameters, 1994.

[3] Giordano N.; Nakanishi H. *Computational Physics.* Prentice Hall, Upper Saddle River, NJ, 2 edition, 2006.

[4] M.E. McIntyre and J. Woodhouse. Friction and the bowed string, 1986.

[5] Popp K.; Stelter P. Stick-Slip Vibrations and Chaos. *Philosophical Transactions: Physical Sciences and Engineering*, 3(1624):89–105, 1990.

[6] John C. Schelleng. The bowed string and the player, 1973.

[7] Stefania Serafin, Federico Avanzini, Dip Ing, and Davide Rocchesso. Bowed String Simulation Using an Elasto-Plastic Friction Model. *Stockholm Music Acoustics Conference*, 2003:1–4, 2003.

[8] M. Sterling and M. Bocko. Empirical physical modeling for bowed string instruments. *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010.

[9] J Woodhouse. Physical modeling of bowed strings. *Computer Music Journal*, 16:43–56, 1992.