

Nicholas Crane

PHY 307

Walter Freeman

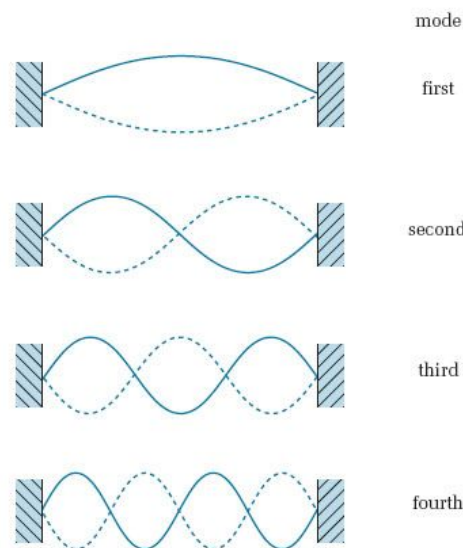
5 December 2019

### Project 7: Nonlinear Behavior in Vibrating Strings

Vibrating strings are very difficult to analyze in the real world. There are many factors that contribute to a vibrating string's frequency and the many interactions between all the particles in the string. Outside of very small amplitudes strings are essentially impossible to study analytically. The advent of computers allows very detailed approximations of vibrating strings, and this paper discusses one such method.

The premise of simulating a vibrating string in this project is to 'stretch' a series of nodes that are connected by nodes and holding the endpoints still while giving the other nodes some initial conditions (either some velocity or position). To ensure our simulation behaves accurately, we need to prescribe certain macroscopic parameters that we can measure in the real world and implement them into the simulation. For example, treating the string as one big spring, its spring constant is given by  $\alpha/L_0$ , where  $\alpha$  is the stiffness of the string (which can be measured) and  $L_0$  is the length of the unstretched (i.e. zero tension) string. The spring constant of each little spring connecting the nodes is given by  $N*(\alpha/L_0) = N*k_{\text{string}}$ , where  $N$  is the number of nodes in the string. The other parameters needed for the string are the linear mass density and the tension that are applied to the string. These can be used to solve for the mass of each node ( $m_{\text{node}} = \mu*L_0/N$ ) and the stretched length of the spring using Hooke's Law ( $L = k_{\text{string}}*T + L_0$ ).

With these tools, construction swiftly follows. Prescribing all the data, a string made of  $N$  nodes can be created with a spring connecting each node to the nodes adjacent to it. The force each node experiences can be determined using Hooke's Law. All that is left is to give the stretched system initial conditions and to analyze its behavior. Two initial conditions are used in this paper, the first of which is a sine wave that can simulate any number of 'modes'. A mode contains two nodes that do not move as the string vibrates. The number of the mode describes how many of these regions there are.



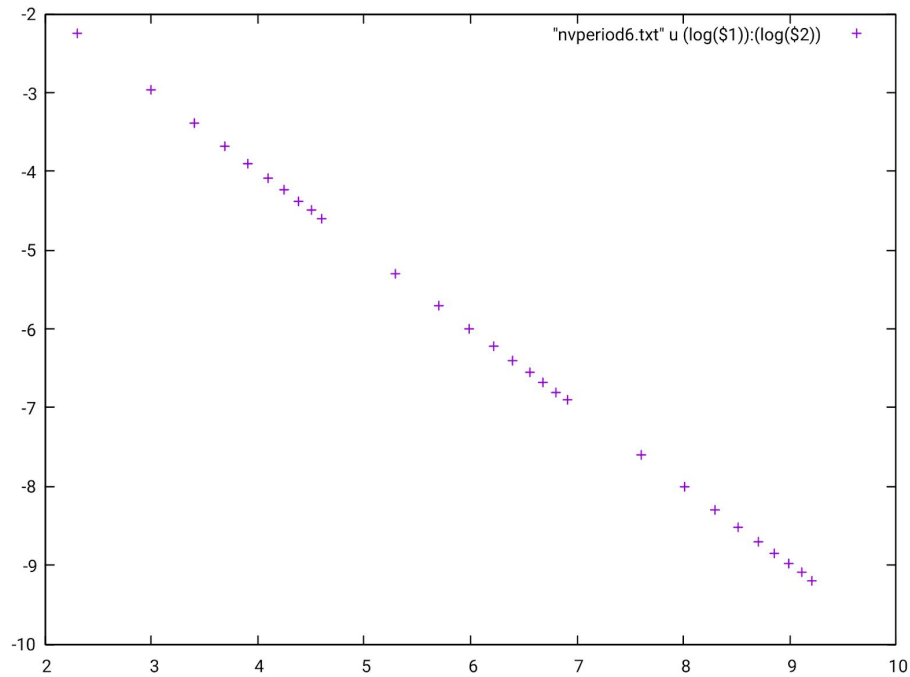
[https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=3514&extra=thumbnailfigure\\_idp4893104](https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=3514&extra=thumbnailfigure_idp4893104)

The other initial condition will be discussed later. In this paper,  $n$  will represent the mode.

There are a few tests that can be run to ensure the simulation is behaving realistically. Analytically, in the limit of a small amplitude the period of each of these modes is given by  $\tau = \lambda_n/v$ , where  $\lambda_n$  is the wavelength of the  $n$ th normal mode and  $v$  is the velocity of the wave, which is also determined analytically and will not be covered in this paper. This can be rewritten as  $(2L/n)\sqrt{\mu/T}$ . Using a  $n$  of 1, an  $\alpha$  of 10 N/m, a tension of 5 N, a  $L_0$  of 1 m, and a  $\mu$  of 1 kg/m

the analytic period is 1.095445s. In our simulation, we can track one antinode and determine when it's velocity changes sign. This time gives one half the period. Experimenting with a range of  $N$  varying from 10 to 10,000 and constructing a log-log graph of the fractional difference in period ( $\Delta\tau = |\tau_{\text{an}} - \tau_{\text{num}}|/\tau_{\text{num}}$ ) vs  $N$ , we get a linear graph with a slope of -1:

Plot of  $\ln(\Delta\tau)$  vs  $\ln(N)$



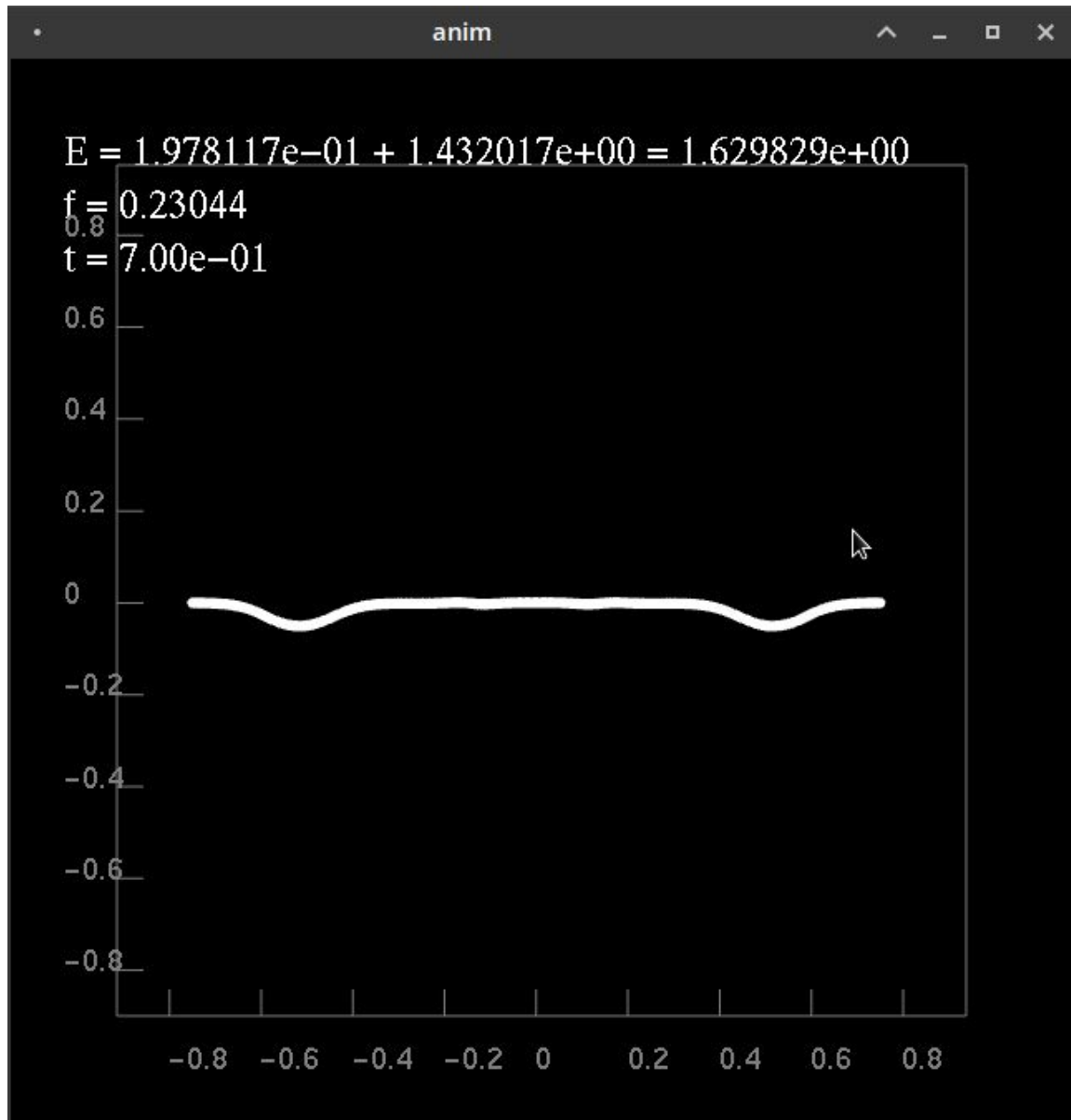
This tells two things; one is that the string behaves more realistically as the number of nodes increases (which makes conceptual sense) and that our method for simulating a string is accurate only to the first-order, i.e. the error in our simulation is proportional to  $1/N$ , rather than  $1/N^2$  or something greater. While this does mean we have to be conscious of computational error in our studies, with 10,000 nodes our fractional difference was negligible (0.000101 with a  $dt$  of  $10^{-7}$ s). This simulation uses the leapfrog algorithm which is accurate to the second order however, allowing us to use a lower time step with similar results. The fractional difference with a  $dt$  of

$10^{-6}$ s was 0.000114, close enough to that with a  $\Delta t$  of  $10^{-7}$ s. For the rest of the paper a timestep of  $10^{-6}$ s and an  $N$  of 10,000 nodes is used.

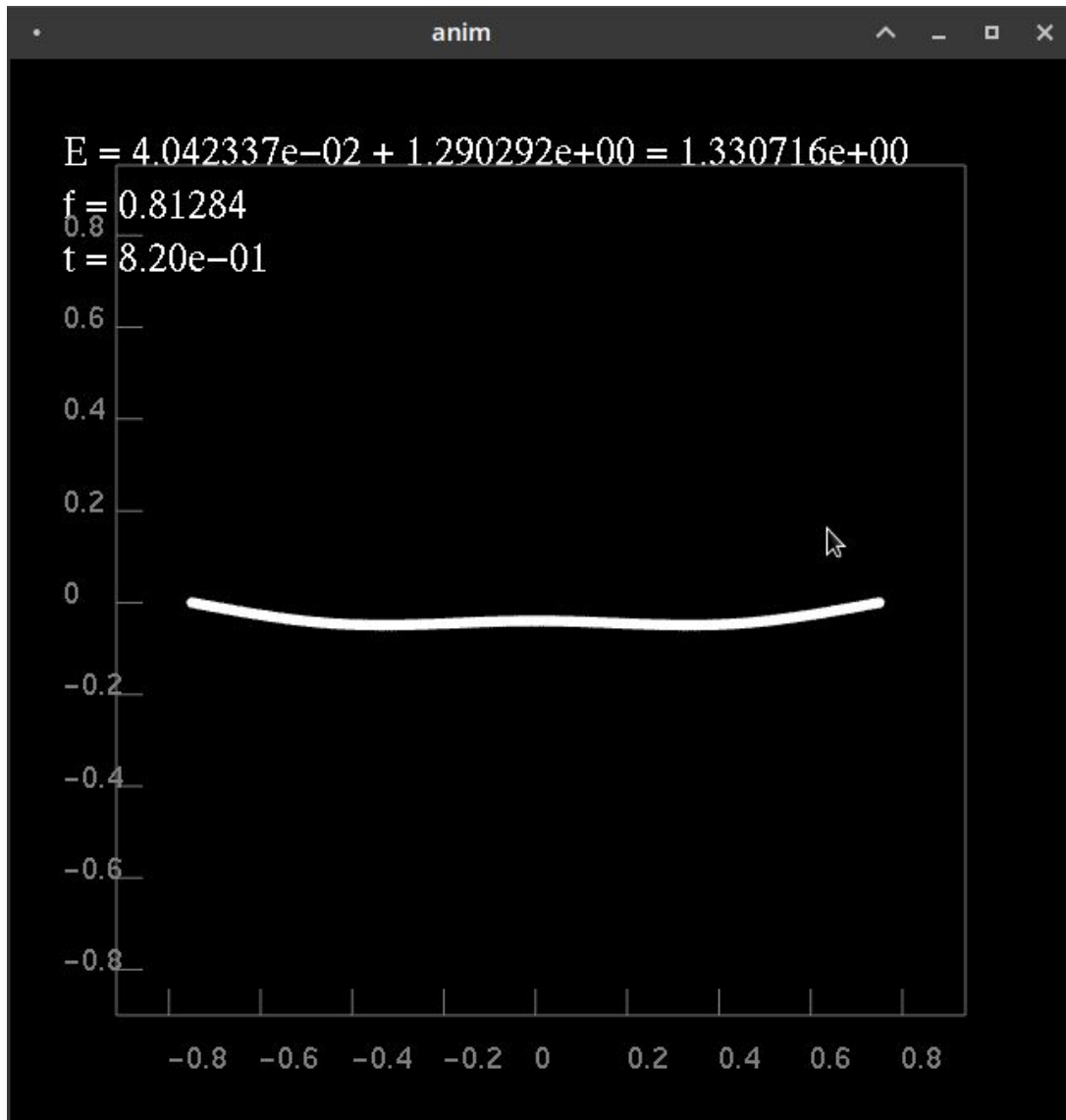
Another check is to change the parameters and ensure the program behaves appropriately. Following the process described above and adjusting the unstretched length, tension, and linear mass density individually, the fractional difference in period was never more than 0.000120, indicating our system is behaving as intended.

The other initial condition is known as the ‘Gaussian bump’ and is given by  $A \exp(-(x_i - x_c)^2 / \sigma^2)$  where  $A$  is the amplitude,  $x_i$  is the  $x$  position of the node,  $x_c$  is the center of the bump, and  $\sigma$  describes the width of the bump. This bump can be used to simulate a wide variety of real-world interactions with a string, like a finger or guitar pick plucking a string. Animating the string allows us to visualize what goes on in a string under these different bumps.

These bumps can be classified into two different categories; small bumps and large bumps. This is not in reference to the amplitude, but instead describes the width of the bump. Large bumps can be seen affecting the entire string all at once while smaller bumps appear to affect only small portions at a time. Acoustically, large bumps create darker tones while small bumps create brighter tones. This can be tested yourself if you have a guitar or other string instrument lying around. Pluck the guitar string with your finger, and then pluck it with a guitar pick. Your finger, which applies the force over a large area creating a larger bump, sounds darker and fuller than when the string is plucked with the pick, which applies the force to a very small area. This can also be visualized through animation of the string. Exciting the string with a small bump(s) creates many small wiggles throughout the string:



While larger bumps tend to only create distinct bumps throughout the string:



This follows what was said earlier; small bumps create little wiggles which have more energy in higher modes leading to a brighter sound than the large bumps which have more energy in the lower modes and as such have a deeper tone, which we observed with the guitar string.

In summation, simulation of real world objects can lead to better understanding that is not possible by direct observation in the real world. Taking quantities that we can measure and implementing them into our simulation ensures realistic results while allowing for easy analysis.