Virtually Reconstructing an Ancient Musical Instrument

Helena Welch

Introduction

This project aims to help restore ancient musical tradition by reconstructing the chelys, a Greek chordophone over 2500 years old [1]. I will first construct a computational model of the instrument, where size and shape can be changed to test the spectrum of frequencies the instrument can play (musical spectrum).

Plan

I hypothesize that there are optimal dimensions of the chelys which best fit the ancient Greek musical system. A method of testing different dimensions and their respective acoustical outputs will be used to discover these parameters. How will altering the dimensions of the chelys affect its musical spectrum? Which dimensions best fit the musical system that its ancient players would have used? To answer these questions, a computational model of the chelys must be derived.

Model

The musical spectrum of the chelys is produced when string vibrations resonate inside the instrument's body, in this case a tortoise shell [1, 2, 3] . Therefore, my model will need to accurately represent the shape and acoustical properties of a tortoise shell. To determine such a model, I assume that the base of the tortoise shell is elliptical. Thus, the first equation of my model is the standard formula of an ellipse, set in the x-z plane:

$$1 = \left(\frac{x-h}{j}\right)^2 + \left(\frac{z-k}{m}\right)^2 \tag{1}$$

where h and j are the center and radius of the major axis, respectively, and k and m are the center and radius of the minor axis, respectively. Figure 1 shows an example of such an elliptical base with the input parameters labelled.

Next, the curve of the tortoise shell can be fitted to a set of parabolas. The first will be in the x-y plane, and the second will be a series of parabolas in the y-z plane, each perpendicular to the x-y parabola and with vertices intersecting the x-y parabola.

The x-y parabola is given by:

$$y(x) = a(x-h)^2 + p \tag{2}$$

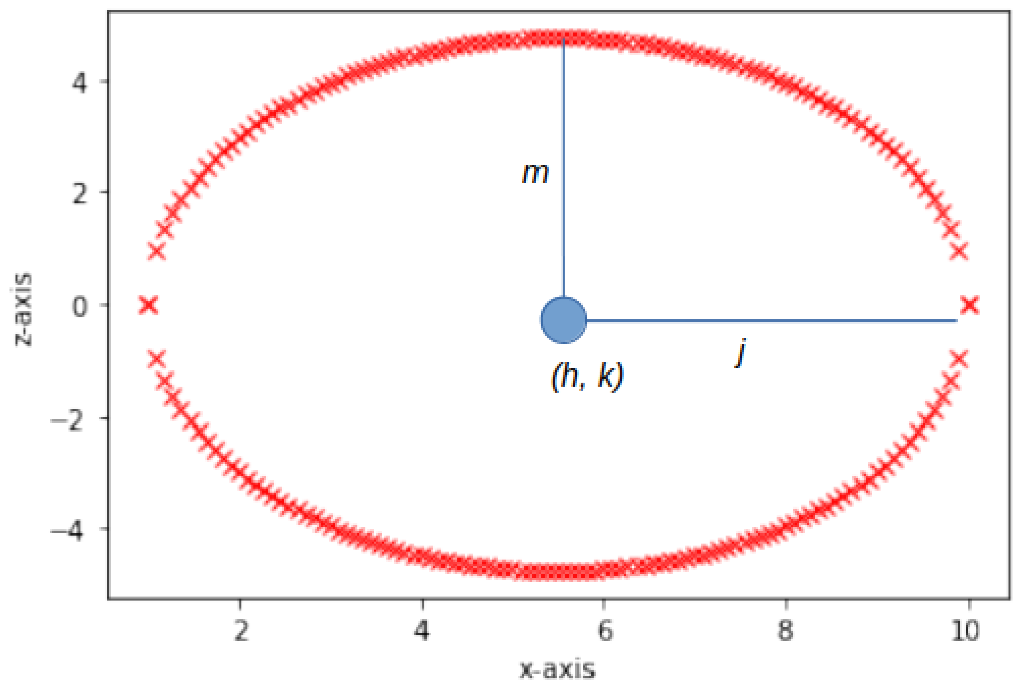

Figure 1: An example of an elliptical base of the tortoise shell with parameters m=5, j=4.5, h=5.5, and k=0.

where a relates to the parabola's curvature and p is the height of the parabola. An example of an x-y parabola with input parameters labelled is shown in Figure 2.

a can be derived by setting x and y equal to the x and y coordinates of the parabola's extremum and solving algebraically for a. The resulting formula is as follows:

$$a = -\frac{p}{j^2} \tag{3}$$

The y-z parabolas are described by the following equation:

$$y(x) = b(x)(z - k)^2 + r(x) \tag{4}$$

where $r(x)$ is the height of each parabola and $b(x)$ relates to the curvature of the parabola. $y(x)$, $b(x)$, and $r(x)$ depend on x since the curvature and

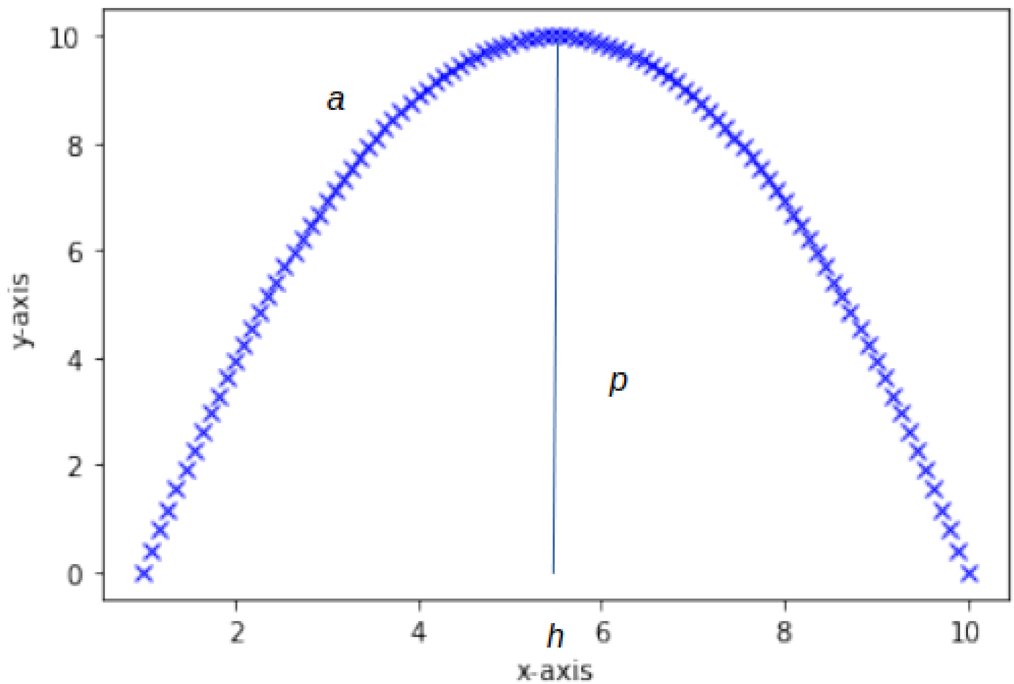

Figure 2: An example of an x-y parabola with input parameters p=10, h=5.5, and a=-2.23.

height of the parabolas change in accordance with the change in height of the x-y parabola. Figure 3 shows an example of a y-z parabola with input parameters labelled.

$r(x)$ is the y-coordinate of a given point on the x-y parabola as a function of x:

$$r(x) = \left(-\frac{p}{j^2}\right)(x-h)^2 + p \tag{5}$$

$b(x)$ is derived in the same manner as a for the x-y parabola:

$$b(x) = -\frac{r(x)}{m^2\left(1 - \left(\frac{x-h}{j}\right)^2\right)} \tag{6}$$

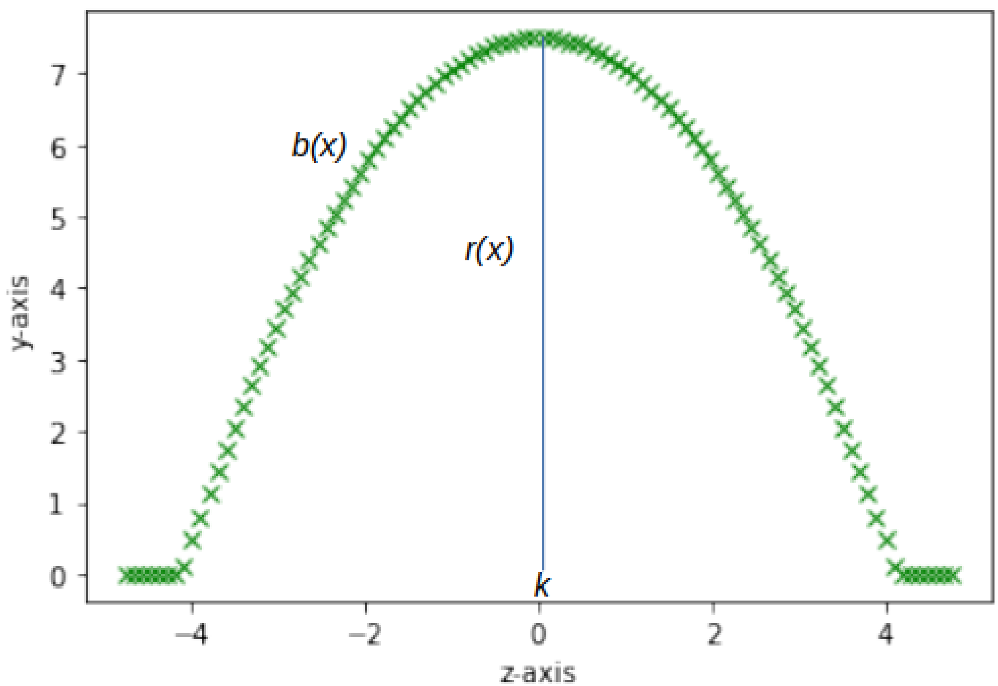

Figure 3: An example y-z parabola where k=0, $r(x)$=7.5, and $b(x)$=-0.25.

Implementation and Expected Results

Values of the input parameters (h, j, k, m, and p) will be changed to produce different dimensions of the tortoise shell. How the chelys' dimensions affect its acoustical properties will be tested through the following procedure:

1) Using the model described above, different variants of tortoise shell will be programmed in the mesh generator Gmsh [4, 5].

2) The Gmsh meshes will then be fed into a solver that uses finite element analysis to output certain acoustical properties of the shells, following the example given by Jeremy Bleyer [7] . Examining these acoustical properties, I hope to discover the optimal combination of timbre and pitch that fit the ancient Greek musical system [6, 8].

Appendix

All plots were made with MatPlotLib[9]. The following code produces a model of the turtle shell's dimensions. I will use this code to mesh my object in Gmsh.

```
#This is the equation for r, the max height of each y-z parabola

def rx(p,h,j,x):
    r = -p/j**2 * (x-h)**2 + p
    return r


#This is the equation for b, the prefactor for
#curvature of each y-z parabola

def bx(r,h,j,m,x):
    if ((x-h)/j)**2 != 1.0:
        b = -r/((m**2)*(1-((x-h)/j)**2))
    else:
        b = 0.0
    return b

#This finds the y coordinates corresponding to the
#x and z coordinates rastered along the y-z parabolas
#If the z coordinate is not on a given parabola,
#the corresponding y coordinate defaults to -1
#The function then returns an array of y coordinates
#defined as the Yraster

def Parabs(p,h,k,j,m,Xraster,Zraster, Nx, Nz):
    Parabs_list = np.zeros(Nx*Nz,dtype=float)
    Parabs_array = Parabs_list.reshape([Nx,Nz])
    i = 0
    for x in Xraster:
        d = 0
        r = rx(p,h,j,x)
        b = bx(r,h,j,m,x)
        for z in Zraster:
            if z <= (1-((x-h)/j)**2)**0.5*m+k and \
            z >= -(1-((x-h)/j)**2)**0.5*m+k:
                Parabs_array[i][d] = b*(z-k)**2 + r
```

```
            else:
                Parabs_array[i][d] = -1
            d = d+1
        i = i+1
    return Parabs_array


#This shows how many points along each y-z parabola were rastered
#It also shows the first and last indices of each line of the array
#to have a y-coordinate on the parabola
#This will be used to generate the mesh in Gmsh

def FindBoundaries(Yraster, Nx, Nz):
    numbers = np.zeros(Nx, dtype=int)
    zindex0 = np.zeros(Nx, dtype=int)
    zindex1 = np.zeros(Nx, dtype=int)
    for i in range(0, Nx):
        counter1 = 0
        firstindex = -1
        lastindex = -1
        for d in range(0, Nz):
            if Yraster[i][d] > -1:
                counter1 += 1
                if firstindex < 0:
                    firstindex = d
            if lastindex < 0 and firstindex > 0 \
            and Yraster[i][d] < 0:
                lastindex = d-1
        numbers[i] = counter1
        zindex0[i] = firstindex
        if lastindex < 0:
            lastindex = Nz - 1
        zindex1[i] = lastindex

    return numbers, zindex0, zindex1
```

References

[1] Bakarezos, Efthimios; Vathis, Vasilios; Brezas, Spyros; Orphanos, Yannis; Papadogiannis, Nektarios A. "Acoustics of the Chelys–An ancient Greek tortoise-shell lyre," *Applied Acoustics* **2012**, *73*, 478-483.

[2] Butler, Paul. "Greek Lyre," `https://crab.rutgers.edu/users/pbutler/greeklyre.html`, Accessed October 12, 2023.

[3] Killmer, J. "Chelys-Lyra, Greece 400 BCE," Smith College History of Science, Museum of Ancient Inventions, `https://www.smith.edu/hsc/museum/ancient_inventions/hsc12b.htm`, Accessed October 12, 2023.

[4] Geuzaine, C.; Remacle, J.-F. "Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering* **2009**, *79*, 1309-1331.

[5] Baratta, Igor A.; Dean, Joseph P.; Dokken, Jørgen S.; Habera, Michal; Hale, Jack S.; Richardson, Chris N.; Rognes, Marie E.; Scroggs, Matthew W.; Sime, Nathan; Wells, Garth N. "*DOLFINx*: the next generation *FEniCS* problem solving environment,", Zenodo, `https://doi.org/10.5281/zenodo.10447666` (2023).

[6] Yokoyama, Masao; Takei, Amane; Shioya, Ryuji; Yagawa, Genki. "Coupled simulation of vibration and sound field of Stradivari's violin," *Proceedings of Meetings on Acoustics* **2023**, *51*, 035001.

[7] Bleyer, Jeremy. "Numerical Tours of Computational Mechanics with FEniCS,", Zenodo, `https://doi.org/10.5281/zenodo.1287832` (2018).

[8] Kaselourix, E.; Bakarezos, M.; Tatarakis, M.; Papadogiannis, N.A.; Dimitriou, V. "A Review of Finite Element Studies in String Musical Instruments," *acoustics* **2022**, *4*, 183-202.

[9] Hunter, J. D. "Matplotlib: A 2D graphics environment", *Computing in Science & Engineering 9*, *3*, `https://doi.org/10.1109/MCSE.2007.55` (2007), 90-95.