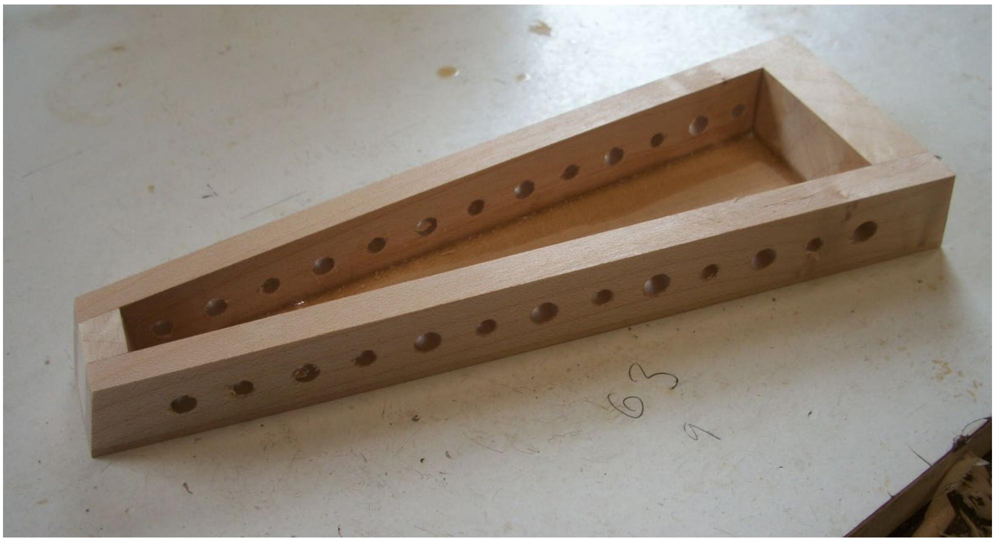
Luke Emmet
luke.emmet@orlando-lutes.com

## A Parametric Design Tool for Making a Family of Lute Pegboxes

Lute pegboxes come in a variety of sizes and angles to reflect the lutes they are attached to. There is some variation in style - and excluding extended neck lutes such as theorbos – a majority are a tapered box design, such as this 7 course example below which is quite typical:



Each pegbox is unique to that model of lute – the dimensions are more or less pre-determined by other dimensions and features of the instrument:

- Length of the pegbox – varies, determined by the number of pegs
- Width of pegbox cheeks – a fixed width, itself tapering from about 9mm to about 8mm
- Width at the top of the pegbox – a fixed width to allow adequate space to wind and adjust a string
- Width of pegbox – varies, determined by the specific pegbox slot cut into the neck

As a result of these constraints, each pegbox is unique and not simply scaled versions of each other. Traditionally, each pegbox is made as a one-off to fit each lute precisely, designed by hand to accommodate the above constraints, or to copy the dimensions of a specific historic model. You can do this on paper using a pencil and ruler to draw out the design and to empirically determine the pegbox taper angle which sets the overall geometry.

As a lute maker that also takes in a fair amount of repair and refurbishing work, I'm often making new pegboxes as part of a refurbishment. For example, many lutes still in circulation from the earlier days of the modern lute revival have a very chunky oversized open pegbox design and I'm often replacing these with new ones that are lighter, more elegant and historically appropriate.

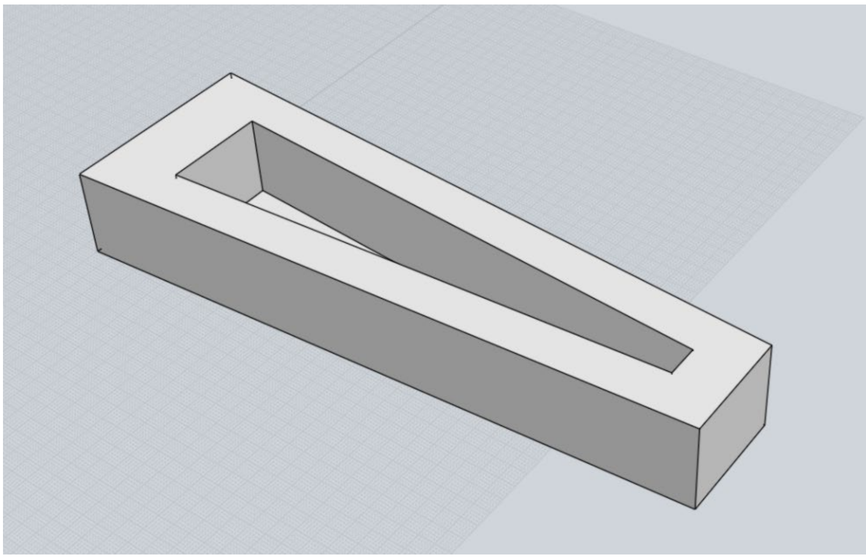
To assist this regular design work, I've been investigating whether it might be possible to use some CAD[1] tools to assist with the design of pegboxes. In particular, can we precisely determine the angle and dimensions of the pegbox blocks which defines the pegbox design?

## Traditional CAD

If we wanted to build a standard CAD model of a pegbox blank, it would start from something like the following model. It is squarer than the final pegbox, which is achieved by planning the height and pegbox sides, as well as cutting the angle on the end where it meets the pegbox slot at around 80-85 degrees.

Most CAD processes involve the design of a specific element, based on dimensions that are known in advance. So once we know the final dimensions of your pegbox, we can build a CAD model (2D or 3D) to capture these.

But this design process it is quite laborious to build such a model since again it has to be designed as a one-off for the lute in front of you – you have to know all the dimensions in advance.



So this individual approach doesn't really save much time or energy compared to designing by hand. And when you come to the next pegbox, the work you did before cannot be reused.

What is really needed is some kind of pegbox designing system, where you can enter the primary design constraints (inner length of pegbox, outer width to match the neck slot, inner width at the top of the pegbox), and then the other dimensions are automatically

[1] Computer Aided Design

calculated. This would be a useful flexible design tool for a whole family of individual pegboxes.

## Parametric CAD

Parametric CAD is based around the principle that some elements of the design are not fixed but may vary within a set of overall constraints and parameters. In this way it is closer to something like a spreadsheet, in which we can enter a formula in a cell that is calculated based on other numbers in the worksheet, which in turn may themselves have been calculated from other numbers. For example we might calculate a parts order as follows:

| B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | **quantity** | **unit price** | **cost** | | |
| | **nuts** | 10 | £0.15 | £1.50 | | |
| | **bolts** | 10 | £0.10 | £1.00 | | |
| | **screws** | 55 | £0.03 | £1.65 | | |
| | | | | | | |
| | | | Total | =SUM(F3:F5) | | |
| | | | | SUM(**number1**, [number2], ...) | | |
| | | | | | | |
| | | | | | | |

Such a worksheet is reusable for the next order, where the number of items required may be different, and also the unit price may vary over time. So a spreadsheet software package lets us design a reusable order calculation system for a wide variety of actual orders. We can also use a spreadsheet to "Goal seek" whereby we set the final price, then calculate how many items it would permit us to purchase – going backwards from an imposed constraint on the total cost

Parametric CAD is an extension of CAD that lets us take a similar approach for design. It is a feature of a number of high end industrial CAD tools such as *AutoDesk*[2]. However such industrial grade tools are priced accordingly[3] and not so easily justified for craft-based instrument designers and makers.

*SolveSpace*[4] is a parametric 2D/3D CAD software application that works in a similar way for CAD designs. It is free and open source[5] which makes it more accessible to designers who wouldn't normally have access to higher end industrial design tools.

---

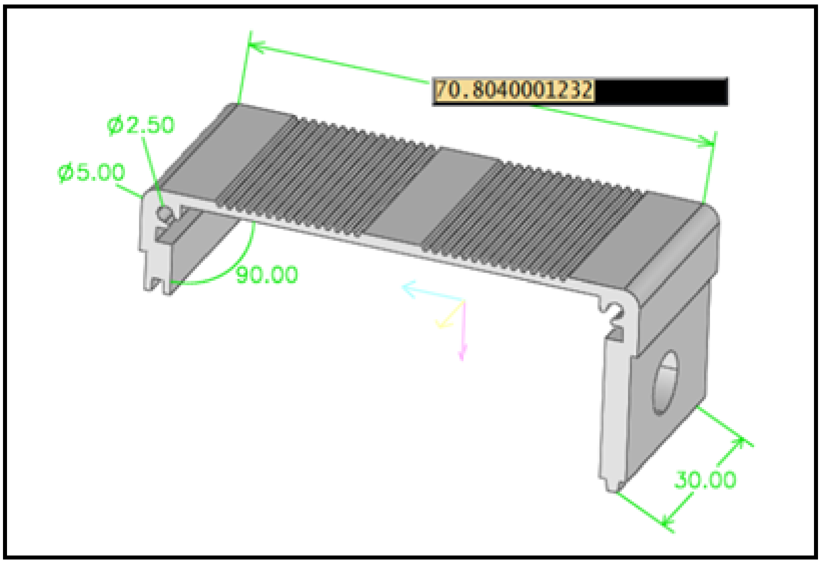
[2] https://www.autodesk.co.uk

[3] At the time of writing, priced at £1,986 /year

[4] https://solvespace.com

[5] Gnu Public License version 3

In the example widget design screenshot, taken from the SolveSpace website, we can see a widget model in which the user can vary some parameters of the design – such as its length without that change affecting other parameters. So it is not simply scaling up the design.

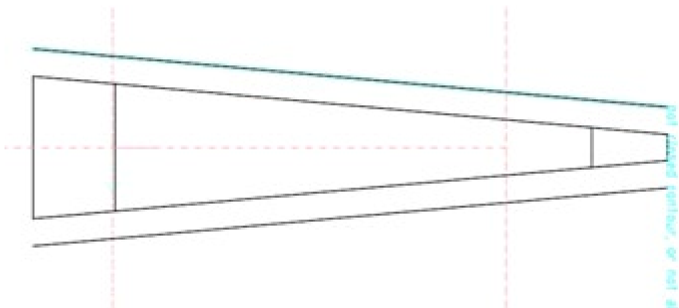



In building a parametric CAD model we can define a number of constraints between key dimensions, including:

- Two lengths or angles being equal
- A line or a point being a mirror reflection of another about some axis
- One line being at a fixed angle or length from another
- Fixed or varying hole or corner diameters

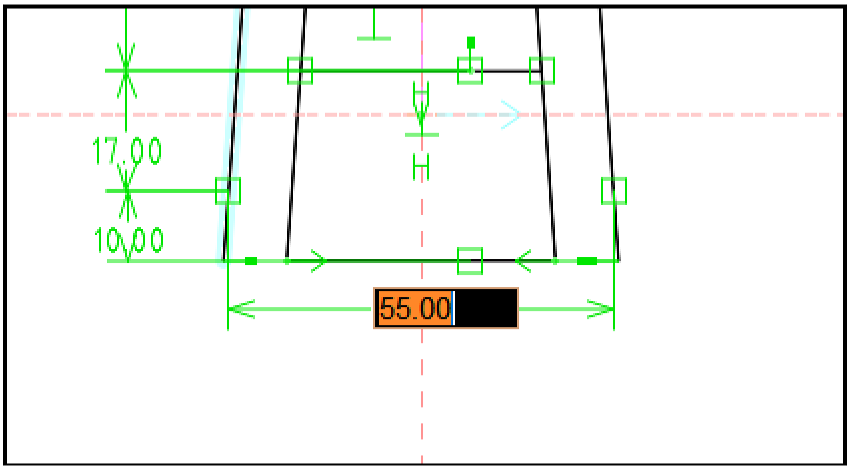
These constraints are the equivalent of the formulae in our spreadsheet example. The design process is somewhat different to traditional CAD, in that we have to conceptualise a design as a set of dependent parameters, rather than as absolute dimensions. The software uses a mathematical constraint solver to determine other dimensions, keeping the design within the overall constraints.

We can therefore use a parametric CAD approach within SolveSpace to build a parametric model of lute pegboxes, based on the following design elements of two end blocks and two pegbox sides.



In the following images we can see the constraints set on the model. For example, the total width of the Pegbox 1 at the lower block is 63mm and the inner height is 157mm. These constraints we shall vary per pegbox, whereas other constraints are fixed, such as the pegbox cheeks being set to 9mm, and the inner width at the top of the pegbox being 13mm.

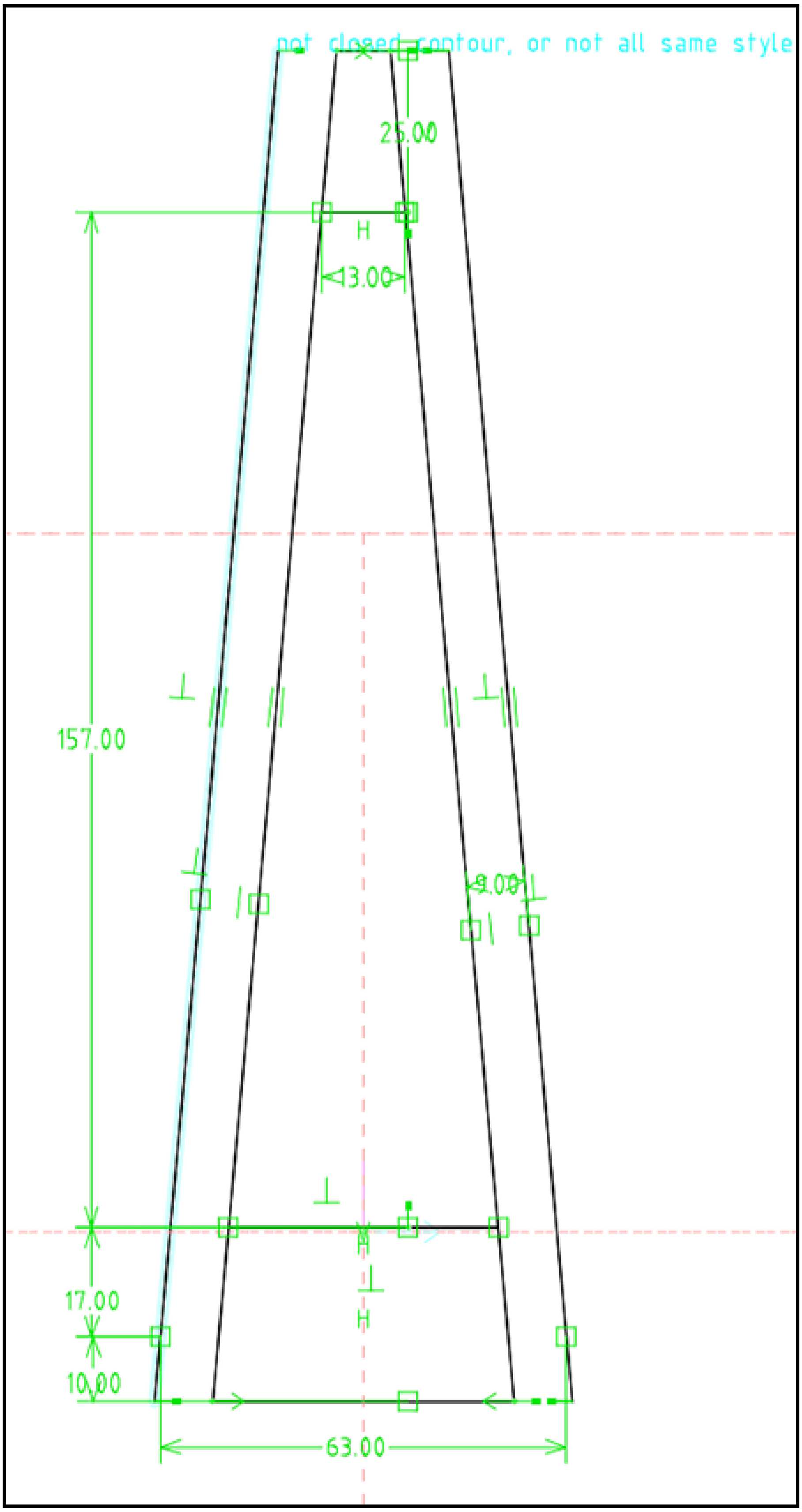
As a result the angle and dimensions of the blocks is determined by the constraint solver.




In Pegbox 2 we use the same model to derive a design for a pegbox for a 55mm slot and a 145mm inner length. Other dimensions are unaffected such as the length of the legbox blocks or the width of the pegbox cheeks.

In conclusion, I have found this to be a rewarding process to assist in design – it has simplified the steps needed to design any future lute pegbox of this style I may need to design – I can simply enter the parameters, the model adjusts itself and I can go straight to printing out the resultant design.

Pegbox 1: 63mm wide, inner length 157



Pegbox 2: 55mm wide, inner length 145mm