Innovative Strategies: Visualizations for Teaching and Learning Mathematics

Lila F. Roberts
LilaRoberts@clayton.edu
Department of Mathematics
Clayton State University
U.S.A.

Abstract: *The last 20 years have seen a dramatic evolution of software and hardware tools that can be used for creating demonstrations and models for teaching and learning mathematics. From flat 2D representations of three-dimensional objects, to robust 3D representations using sophisticated graphics software, to tactile objects manufactured using 3D printing, the important role of visualizations in teaching and learning mathematics cannot be overstated. Visualizations increase student understanding and learner motivation. 3D models of mathematically significant objects provide access to students who have low vision and have benefits for all students. In this paper, we investigate various free and proprietary software tools for generating 3D visualizations, both graphical representations and models created using 3D printers. Some practical aspects of 3D printing are also discussed.*

1. Introduction

Since ancient times, visualization has been important for understanding and communicating mathematics [1]. Although there is mixed evidence of the effectiveness to which visualizations support students' development and development of key concepts, there is a vast body of research that instructors and students enjoy interactive visualizations and animations and that these tools can bring about more student engagement through active learning ([2], [3]).

In the early days of computing technology and software for generating 3D visualizations for teaching, creating meaningful visualizations often required multiple software tools and many steps. In addition, at the time there were few tools available to make visualizations interactive. For example, the Demos with Positive Impact [4] repository of demos for teaching mathematics contains a variety of visualizations for solids of revolution. In Figure 1.1, the images are frames of an animated gif. The frames were created and exported frame-by-frame from a *Mathematica* [5] notebook. Afterward, the frames were compiled using another software package to create an animated gif. This visualization is the solid of revolution generated when the region bounded by the y-axis, x-axis, $x = 1$, and the graph of $y = x^2 + 1$, is revolved about the y-axis.

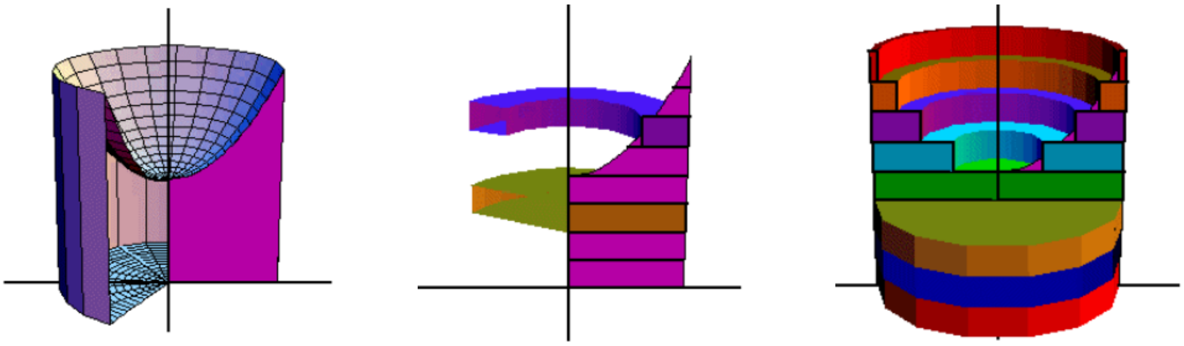


Figure 1.1 Static frames in three separate 3D animations [6]

This example provides a nice visual to illustrate that when one uses a disk approach to finding the volume of the solid of revolution, two integrals are required because while $0 < y < 1$, the cross section is a disk and when $1 < y < 2$, the cross sections are washers. This complexity illustrates that by using a different technique (Method of Shells, Figure 1.2) the volume calculation can be made using a single integral. As animated gifs, the visualizations are not as satisfying as they could be if there were slider tools available to engage the user in generating the images.

This paper will explore some significant improvements in technology tools for creating interactive 3D visualizations and how visualizations can, using 3D printing, be turned in to tactile objects from which all students can benefit, including those who have low vision.

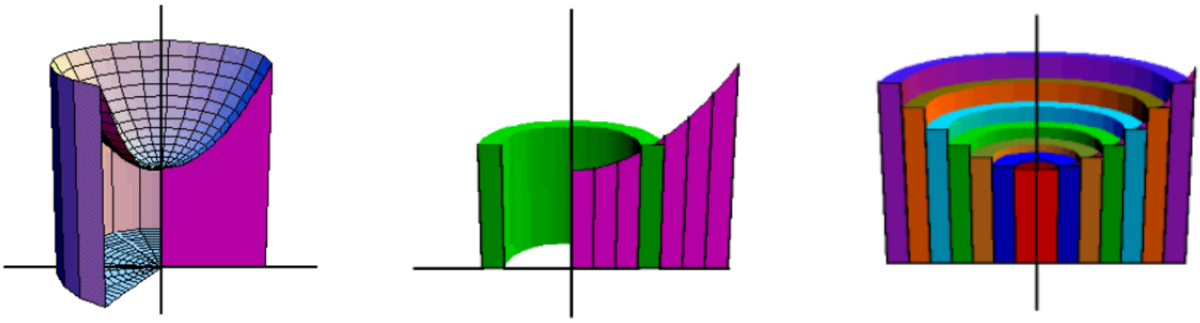


Figure 1.2 Static frames from 3D animations. [7]

2. Interactive 3D Visualizations

As graphical software has evolved, so has the ability to create interactive applications. In Figures 2.1 and 2.2(a,b), a visualization tool created using *Mathematica* Version 12.0 [8] shows the sequential steps in determining the region bounded by the graphs of $f(x) = x^2$and $g(x) = \sqrt{x}$ and using pushbutton controls and sliders, provides the ability to revolve the region about the *y*-axis and then about the *x*-axis.

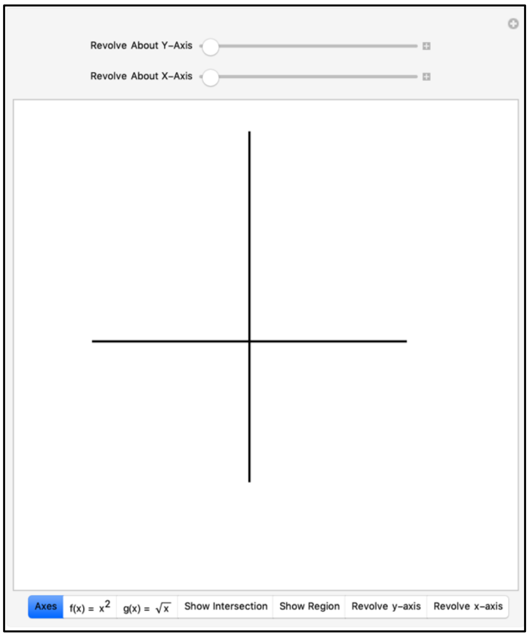


Figure 2.1 Interactive *Mathematica* visualization that uses buttons and sliders.

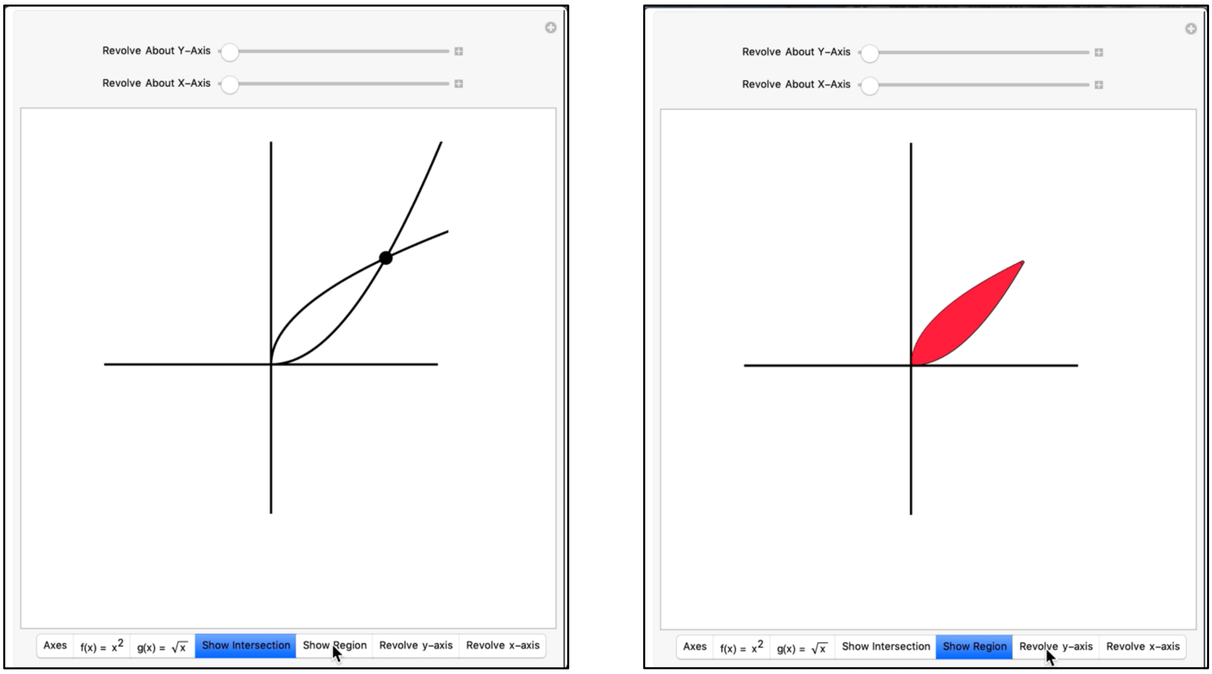


Figure 2.2a By sequentially pressing the controls, the graphs of *f* and *g* and the region between are generated.

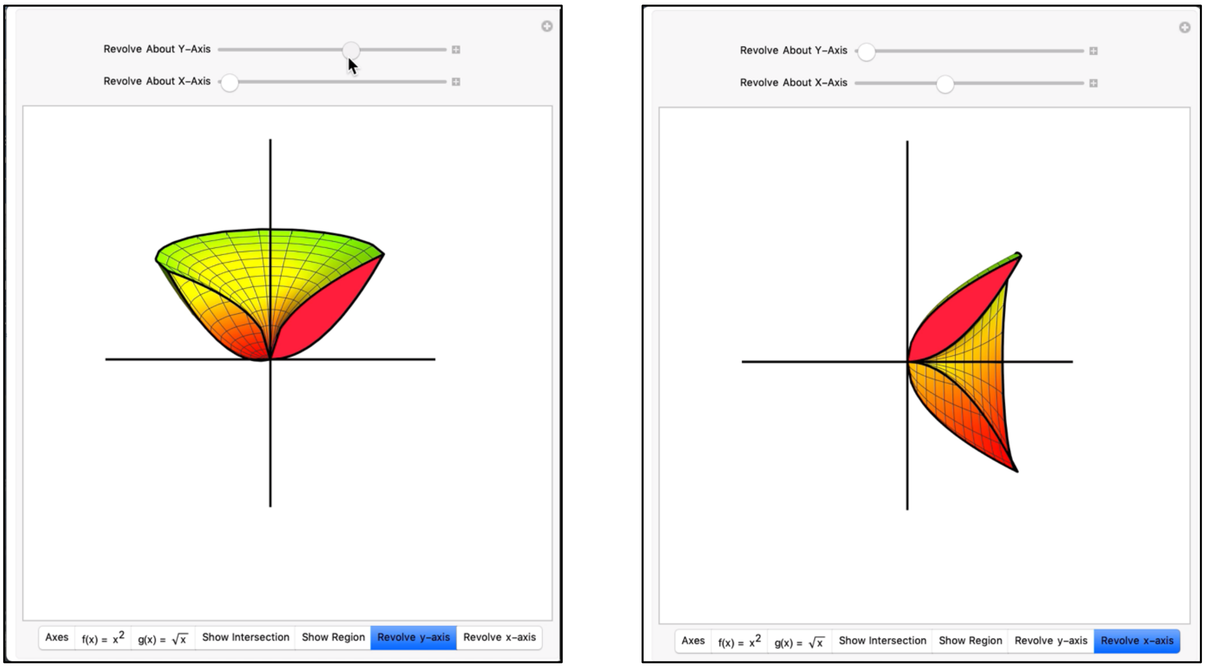


Figure 2.2b Using a slider, the region is revolved about the *y* or *x* axes to generate the solid of revolution.

With a bit more coding, a visualization that generates a partition, approximating rectangles, and a resulting approximating element (shells, washers), illustrates how the volume of the solid can be approximated using these elements. The hope is that the visualizations may help students grasp that volume calculations using integral calculus basically amount to knowing how to calculate the volume of a simpler geometric solid and then use summation and appropriate limits that give the volume of the solid of revolution (Figure 2.3).

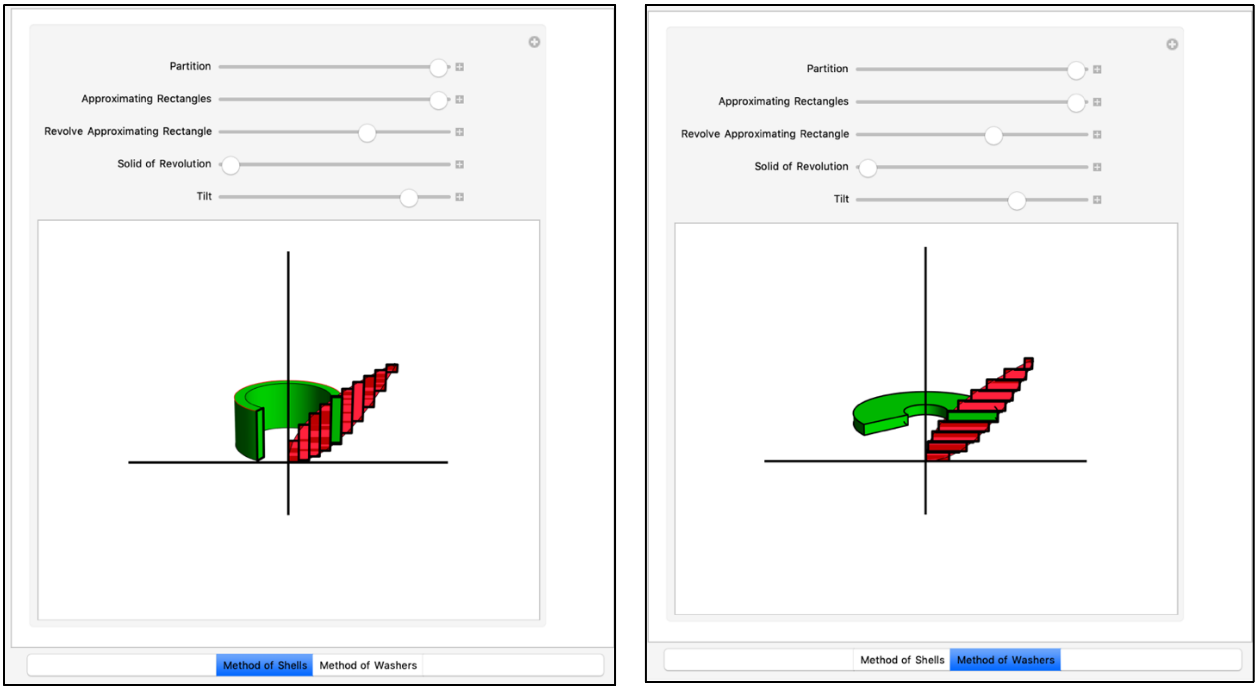


Figure 2.3 *Mathematica* interactive program to model solid approximations using cylindrical shells and washers.

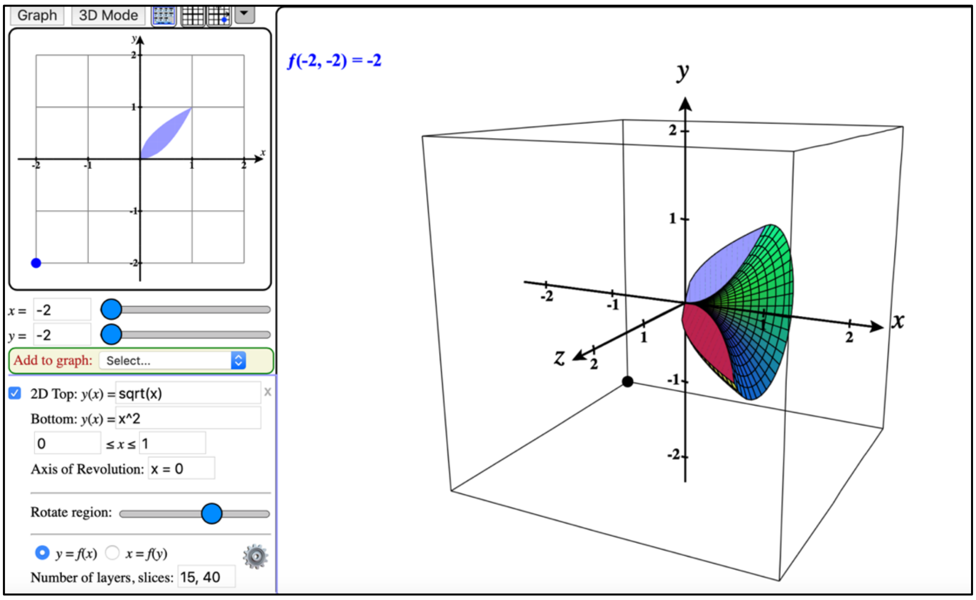
While *Mathematica* is a high end graphical and computer algebra system that has tremendous value beyond 3D visualization in the elementary calculus sequence, it is expensive and many students (and faculty) cannot afford to make that investment. Fortunately, there are some excellent free graphical tools that can provide interactive 3D explorations and more.

One example is *CalcPlot3D*, a robust 3D interactive graphing tool developed by Dr. Paul Seeburger at Monroe Community College (United States) [9]. Originally written in Java, which had inherent usability problems as mobile technology evolved, the *CalcPlot3D* has been converted to JavaScript. The goal of the project is to “enhance the geometric intuition of calculus students so that they are able to visualize the concepts and actually ‘see’ the rich visual relationships described by calculus concepts.” [9] In addition to providing a robust tool for graphing functions of two variables, an additional level of visualization sophistication is provided—3D mode for use with 3D glasses. There is also capability of loading scripts so that additional functionality may be included.

There are several built-in visualizations, or the user may add his or her own. One such feature is “Surface of Revolution.” A pre-made example that is provided is one we saw earlier and is illustrated in Figures 2.4 and 2.5.

In addition to Surface of Revolution, the user may choose graphing tools such as space curves, vector fields, points in space, vectors in three dimensions, implicit surfaces, parametric surfaces and cylindrical coordinates.

All of these features and the fact that *CalcPlot3D* was designed with teaching and learning in mind, make *CalcPlot3D* a very robust and versatile 3D graphing utility. It is a superb web app, runs in any browser, can be used with 3D glasses, is compatible with mobile devices and best of all, it is free.



$x =$ -2

$y =$ -2

☑ 2D Top: $y(x) =$ sqrt(x)

Bottom: $y(x) =$ x^2

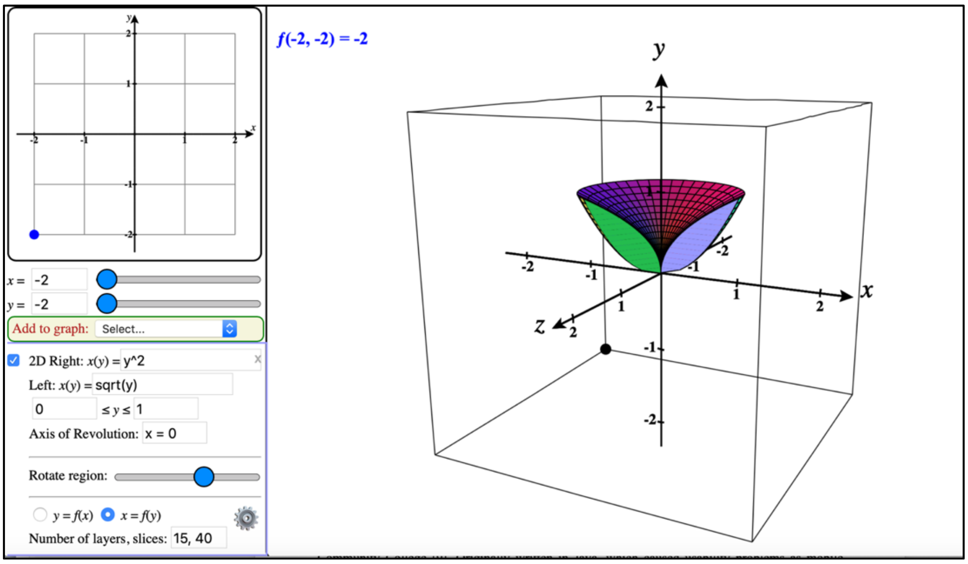
0 $\leq x \leq$ 1

Axis of Revolution: x = 0

● $y = f(x)$ ○ $x = f(y)$

Number of layers, slices: 15, 40

Figure 2.4 Surface of revolution formed by revolving region bounded between $y = x^2$and $y = \sqrt{x}$ about the *x*-axis. *CalcPlot3D*



$x =$ -2

$y =$ -2

☑ 2D Right: $x(y) =$ y^2

Left: $x(y) =$ sqrt(y)

0 $\leq y \leq$ 1

Axis of Revolution: x = 0

○ $y = f(x)$ ● $x = f(y)$

Number of layers, slices: 15, 40

Figure 2.5 Surface of revolution formed by revolving region bounded between $x = y^2$and $x = \sqrt{y}$ about the *y*-axis. *CalcPlot3D*

Another interesting demo in the *Demos with Positive Impact* collection is one entitled “Applying Calculus to Approximating Volumes of Physical Objects.”[10] Inspired by another contribution in the collection [11], this demo illustrated how to use the method of cross sections to approximate the volume of an object that is not a solid of revolution. From an instructional standpoint, such examples are useful so that students understand that the approximation techniques we learn are widely applicable. The demo involves approximating the wine bottle (Figure 2.6) using cylindrical cross sections. Note that the wine bottle cannot be modeled using a solid of revolution.

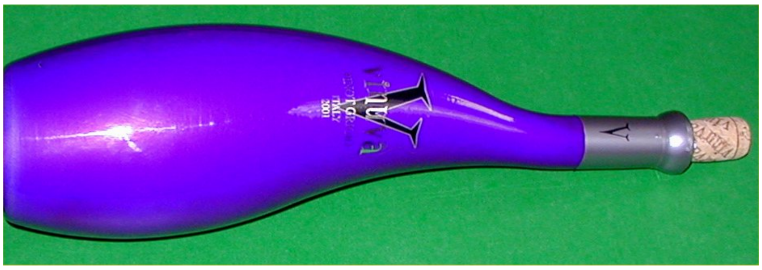


Figure 2.6 Interesting wine bottle for 3D modeling.

Using *Mathematica*, we can import photos, collect data by choosing points along the perimeter of the bottle and then interpolate the functions along the "top" and "bottom" of the image. Figure 2.7 shows the interpolating functions for the sets of points that were collected.

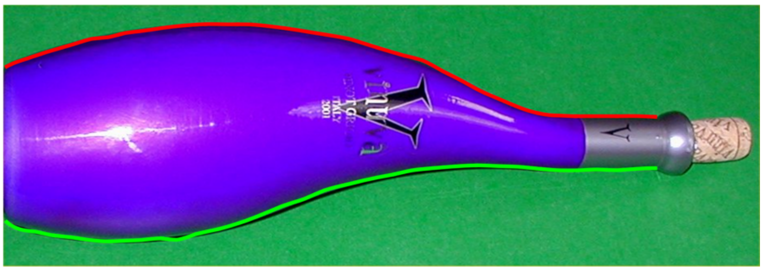


Figure 2.7 Interpolating functions to model the wine bottle.

The wine bottle has circular cross sections, so we can build a model by approximating the centers of the cross sections and then build a model using circular disks with constant thicknesses. Figure 2.8 illustrates the model of the bottle and the approximation using circular disks.

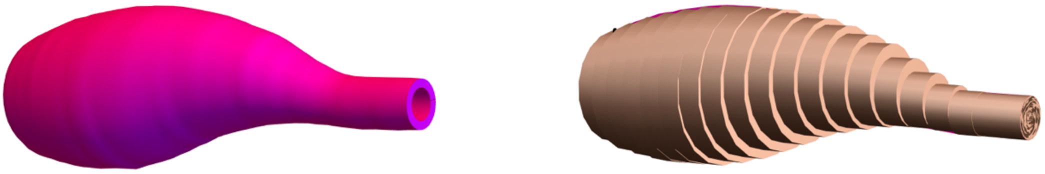
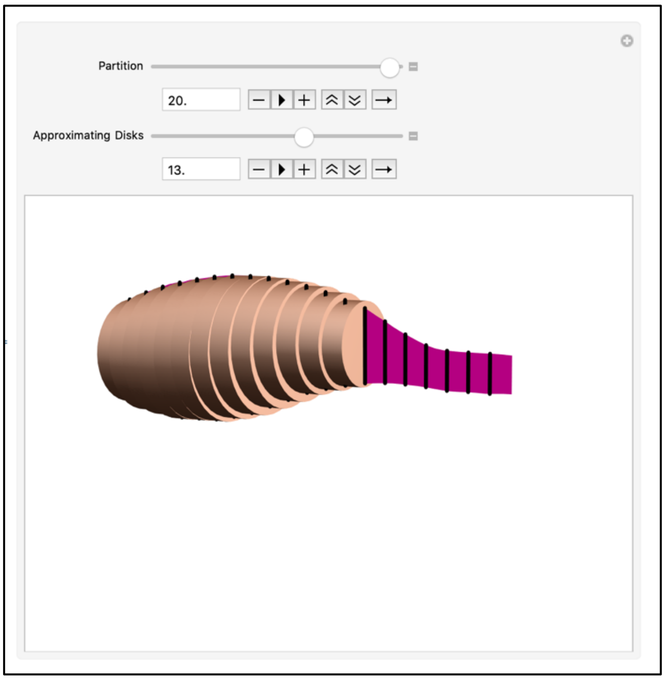


Figure 2.8 Model of wine bottle and approximation using circular disks.

Figure 2.9 shows a *Mathematica* interactive tool to "build the bottle."

No discussion of 3D visualization tools would be complete without inclusion of GeoGebra [12]. A powerful computer algebra system, geometric construction, 2D graphing tool and utility for data analysis, GeoGebra combines 3D graphing capabilities with full-featured user interface and scripting. GeoGebra is a robust utility for interactivity and dynamic exploration. GeoGebra runs on multiple platforms, including mobile devices and provides cloud-based hosting for notebooks and GeoGebra Books [13]. It is open source software that is freely available for non-commercial use and is used by millions around the world. GeoGebra is a powerful authoring tool for development of dynamic applications for teaching and learning.

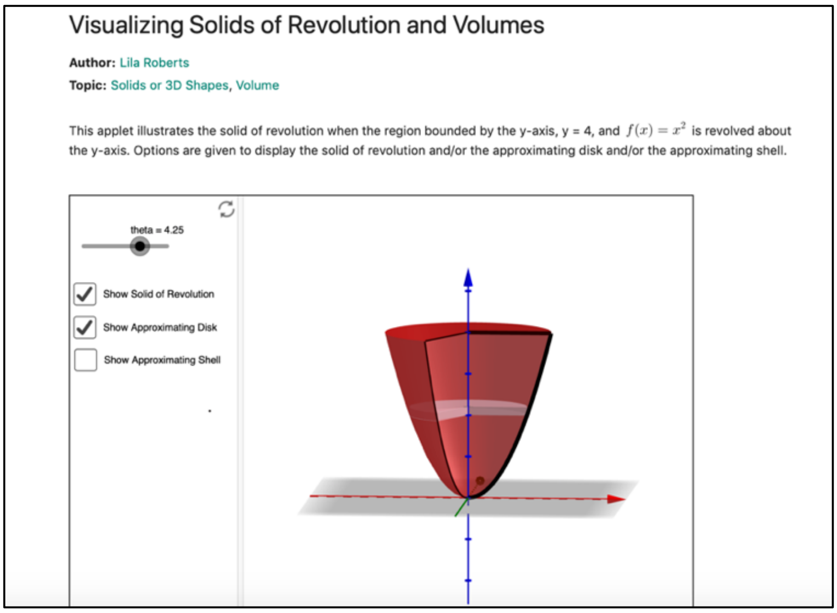


Partition 20.

Approximating Disks 13.

Figure 2.9 *Mathematica* interactive tool for the wine bottle approximation.

Figures 2.10a and 2.10b show some of the examples in a GeoGebra Book illustrating dynamic visualization of solids of revolution [14].



Visualizing Solids of Revolution and Volumes

Author: Lila Roberts

Topic: Solids or 3D Shapes, Volume

This applet illustrates the solid of revolution when the region bounded by the y-axis, y = 4, and $f(x) = x^2$ is revolved about the y-axis. Options are given to display the solid of revolution and/or the approximating disk and/or the approximating shell.

theta = 4.25

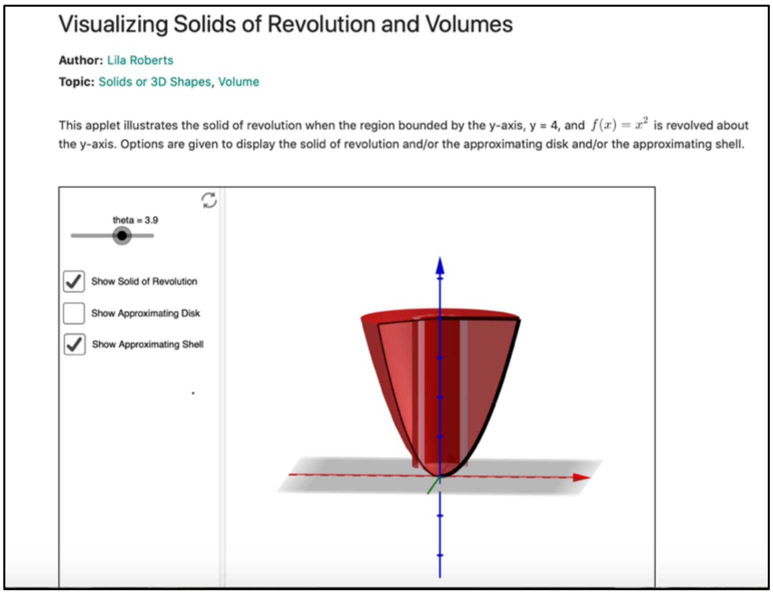
☑ Show Solid of Revolution

☑ Show Approximating Disk

□ Show Approximating Shell

Figure 2.10a Visualization of Solid of Revolution using GeoGebra.

GeoGebra has many capabilities that other high-end 3D visualization tools provide. For example, using the strategy demonstrated in [15], it is straightforward to model an object as a solid of revolution using a photograph, shown in Figure 2.11.



Visualizing Solids of Revolution and Volumes

Author: Lila Roberts

Topic: Solids or 3D Shapes, Volume

This applet illustrates the solid of revolution when the region bounded by the y-axis, y = 4, and $f(x) = x^2$ is revolved about the y-axis. Options are given to display the solid of revolution and/or the approximating disk and/or the approximating shell.

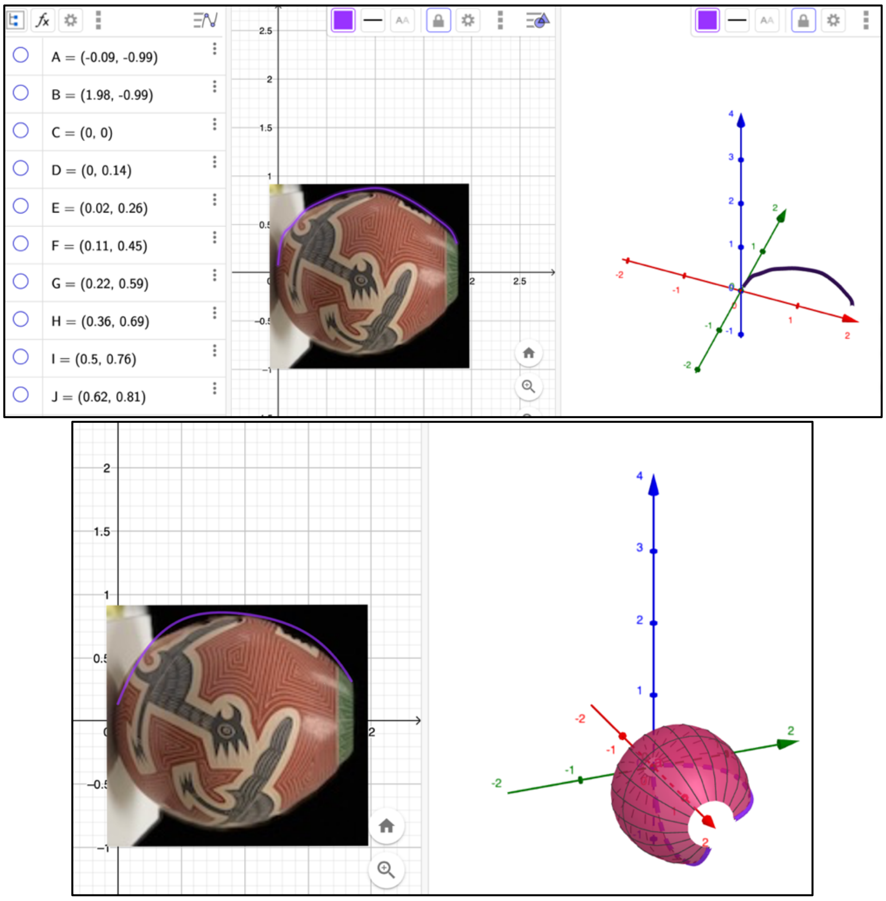
theta = 3.9

☑ Show Solid of Revolution

□ Show Approximating Disk

☑ Show Approximating Shell

Figure 2.10b Visualization of solid of revolution using GeoGebra.



○ A = (-0.09, -0.99)

○ B = (1.98, -0.99)

○ C = (0, 0)

○ D = (0, 0.14)

○ E = (0.02, 0.26)

○ F = (0.11, 0.45)

○ G = (0.22, 0.59)

○ H = (0.36, 0.69)

○ I = (0.5, 0.76)

○ J = (0.62, 0.81)

Figure 2.11 Modeling a vase as a solid of revolution.

It is easier than ever to create interactive and innovative 3D visualizations for teaching and learning and that these visualizations can help students to "see" the mathematics in a way that was previously hard or just not possible. However, there is an audience for whom these visualizations are not particularly useful. This audience includes students who have visual impairments.

3. “Visualizations for All” Using 3D Printing—Making Mathematics Accessible

In the United States, the Americans with Disabilities Act (ADA) requires that universities must provide “meaningful access to the educational experience” [16] for all students. Mathematics, such as geometry and calculus, rely heavily upon visualization. Meeting the requirement of providing “meaningful access” to visual models may not always be an easy task, depending on the university’s resources.

3D physical models provide a tactile approach to teaching and learning for students who have visual impairments. Increasing affordability of 3D printers and ready availability of resources to design 3D models make 3D printing a useful option for addressing accessibility. Some studies have shown that increasing accessibility for students with visual impairments often enhances learning for all students [17] because physical models are important for hands-on active learning [1]. This section will demonstrate that many of the tools for visualization for sighted students can also produce the necessary format to create tactile models using 3D printing to generate tactile models for those visually impaired. The examples provided here are appropriate for novices who have the necessary hardware and software.

Getting started with 3D printing requires a method to produce a file that can be converted to a format a 3D printer can use to manufacture the solid. A common file format is STL, which is an abbreviation for STereo Lithography (also referred to as “Standard Tessellation Language” or “Standard Triangle Language”). STL encodes the surface geometry of a 3D object using tessellations or tiling [18]. It can be straightforward to generate an STL file in *Mathematica*. Consider the following example. The commands

```
x[u_, v_] := (1 + (v/2) Cos[u/2]) Cos[u]
y[u_, v_] := (1 + (v/2) Cos[u/2]) Sin[u]
z[u_, v_] := (v/2) Sin[u/2]
plot = ParametricPlot3D[{x[u, v], y[u, v], z[u, v]}, {u, 0,
   2 Pi}, {v, -.5, .5}, Boxed -> False, Axes -> False]
```

generate the Moebius [19] strip shown in Figure 3.1.

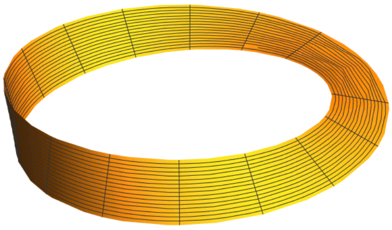


Figure 3.1 A Moebius strip [19] generated using *Mathematica*.

While this is a nice graphic, it is not suitable for 3D printing because it has no thickness. However, changing the ParametricPlot3D instruction to

```
moebiusplot = ParametricPlot3D[{x[u, v], y[u, v], z[u, v]}, {u, 0, 2
Pi}, {v, -.5, .5}, Boxed -> False,  Axes -> False,
PlotTheme->”ThickSurface”]
```

generates a Moebius strip that has non-zero thickness (Figure 3.2). The strip can now be printed.

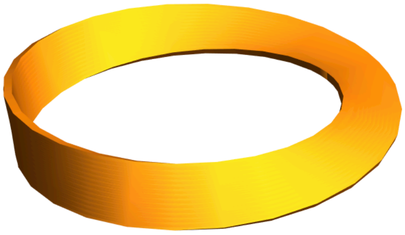


Figure 3.2 Modified Moebius band for compatibility with 3D printing.

Using *Mathematica*'s Export function, a simple command exports the desired format:

```
Export["mobius.stl", moebiusplot, STL]
```

Next, one needs to have a "slicer" utility that converts the STL file into a language the printer can read. Typically, a given printer will have a proprietary slicer but there are some 3rd party slicers that usually have more options (e.g. [20]). For the Ultimaker 3 [21] printer, the slicer software is called Ultimaker Cura (Cura). When the STL file is loaded into Cura, it is ready for slicing (Figure 3.3).

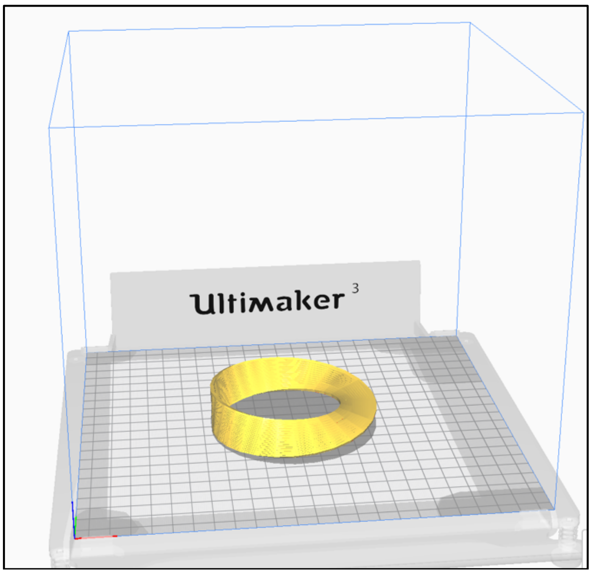


Figure 3.3 Moebius STL file loaded into Ultimaker Cura.

Before sending the object to the printer, we need to select print settings, such as controlling extrusion, material and resolution. We also need to ensure that the print will adhere to the build plate and that overhangs are supported. Notice that the Moebius strip is not flat on the build plate, so some support will be required for a successful print. After adjusting settings, and adding a raft and supports, the object is ready to be printed (Figure 3.4). Once the print is complete, the raft and support materials will be broken off, leaving the Moebius strip.

While there is much more to generating files suitable for 3D printing than this simple example, the example illustrates that one does not need to be an expert to generate a 3D print that could be useful as a model for teaching and learning. It is also the case that you need not have a high-end graphics package like *Mathematica* or CAD software to export STL files. Both *CalcPlot3D* and GeoGebra can export STL files for 3D printing (Figure 3.5).

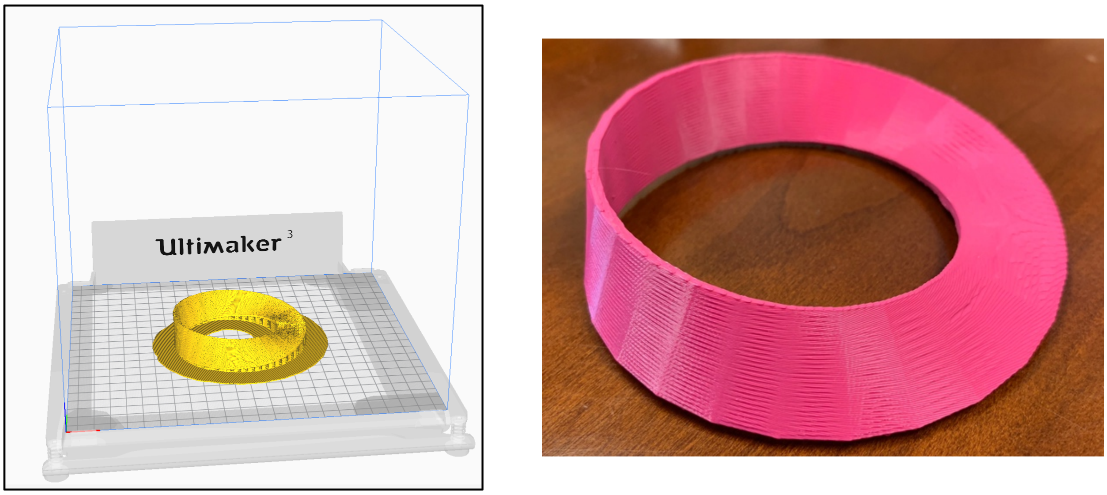


Figure 3.4 Moebius strip with supports and raft, and the finished product, raft and supports removed.

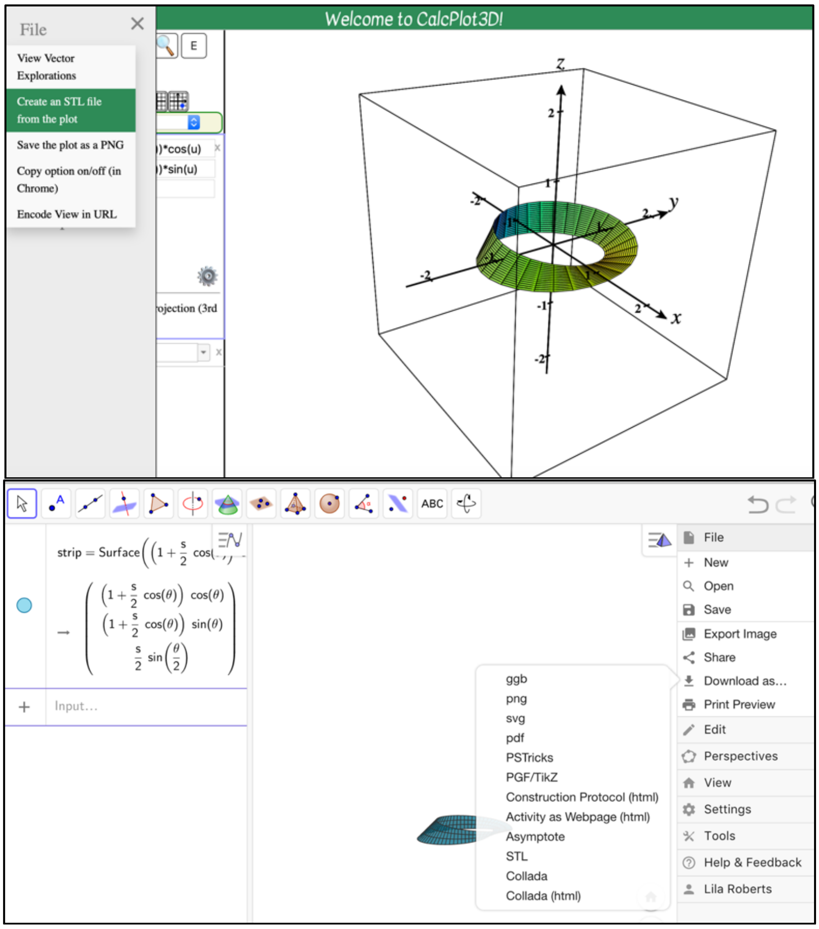


Figure 3.5 STL export using *CalcPlot3D* and GeoGebra.

The ability to export STL files from common 3D graphics utilities provides a means to create physical models of the visualizations that were described in Part 2 of this paper. For example, exporting the models for the wine bottle approximation described in Figure 2.7, yields delightful 3D prints of the bottle and the approximation using circular disks (Figure 3.6). Because of size limitations dictated by the build volume of the printer, the models shown here are about 15% of the size the models would be if they were full-sized.

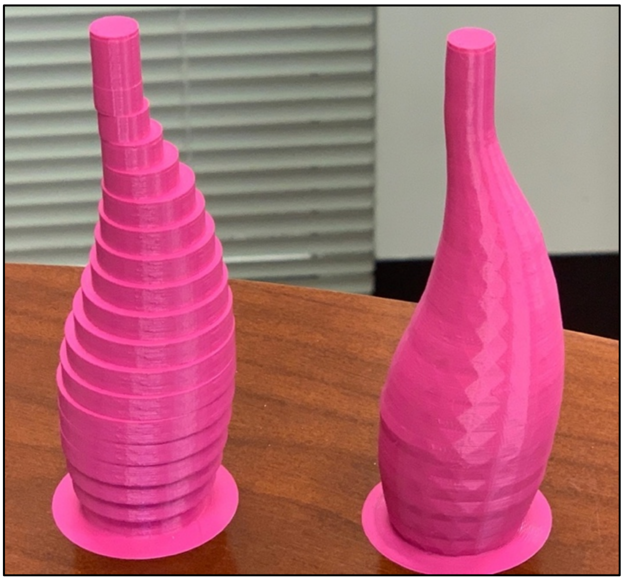


Figure 3.6 Wine bottle approximations using STL files exported from *Mathematica*.

Exporting the STL files from 3D applications allows us to bring models to life with tactile versions of the visualizations. Figure 3.7 shows photos of the 3D printed models of the visualizations described in Figures 1.1 and 1.2.

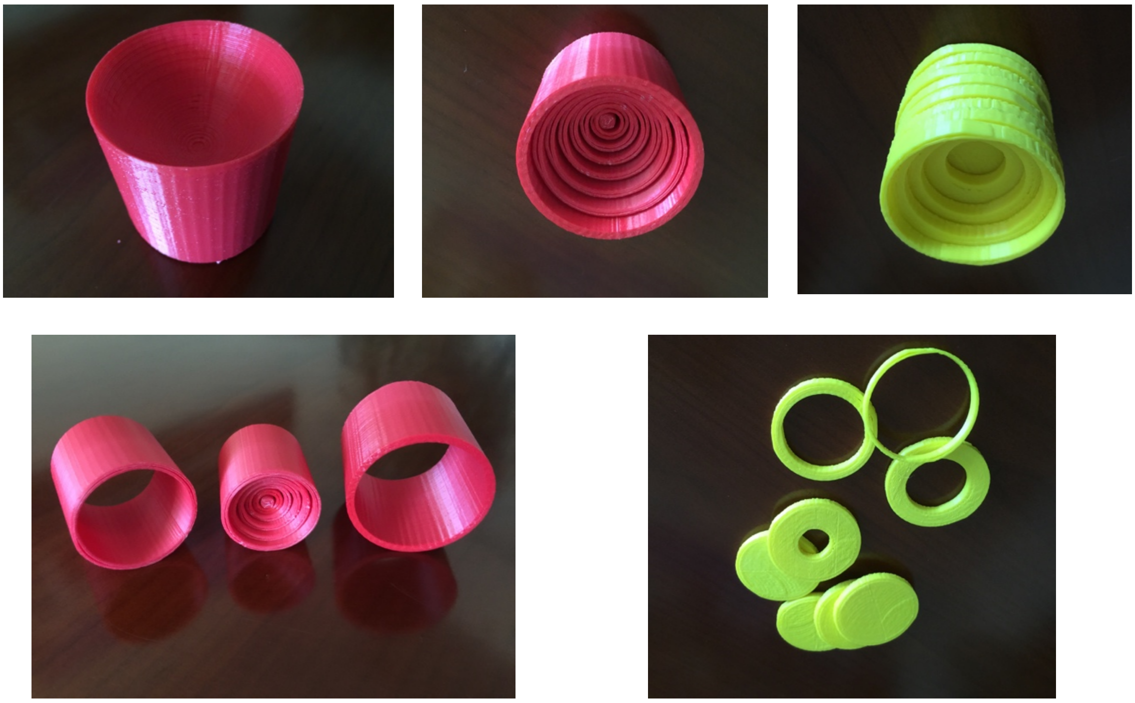


Figure 3.7 3D printed models and approximations of the solid formed by revolving the region bounded by the y-axis, x-axis, $x = 1$, and the graph of $y = x^2 + 1$ about the y axis.

4. You Can't Always Get What You Want—and Other Practical Considerations

Designing models suitable for 3D printing requires, at a minimum, 1) making sure the model is watertight (non-manifold), 2) making sure all surfaces have thickness, 3) paying attention to intersecting surfaces, and 4) making sure all normal face the correct direction [22]. Fortunately, there are plenty of software resources that can help "fix" the STL file prior to sending it to the slicer. Some slicers include built-in options for "fixing." *CalcPlot3D* has an advanced feature to provide some fixing options prior to export of the STL file.

Printing itself requires some patience and occasionally the ability to be a mechanic if there are hardware issues to resolve. Sometimes prints go terribly wrong, as in Figure 4.1. Fatal errors can be caused by bad STL files, build plates that are not level, drafts, bad adhesion to the build plates, among others. Leaving prints unattended gave the fatal results shown in Figure 4.1.

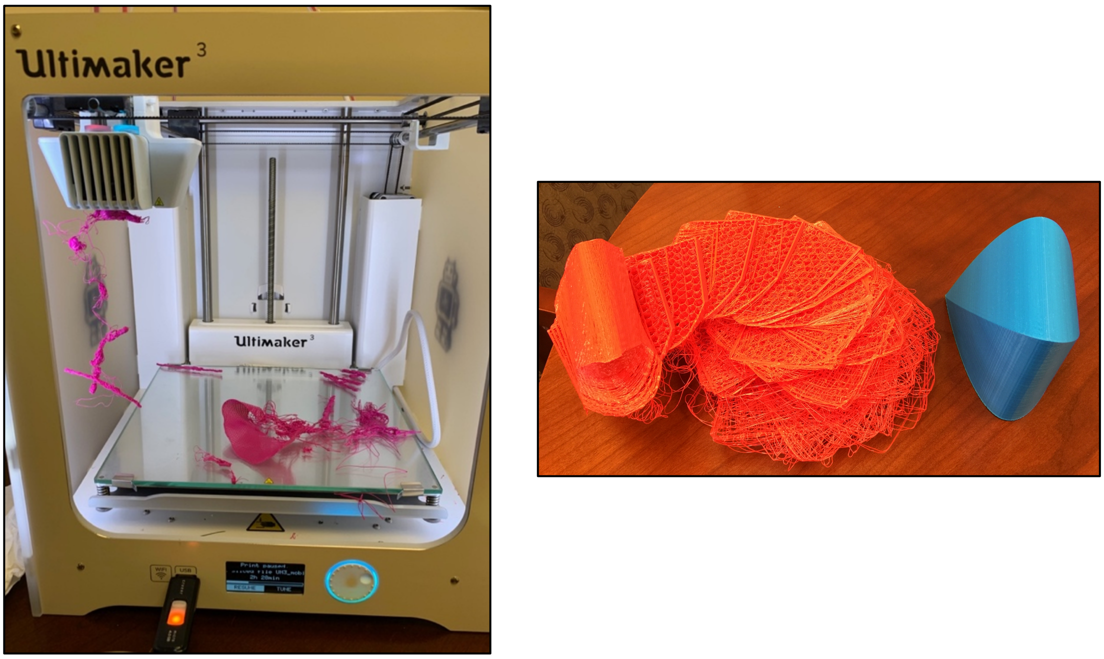
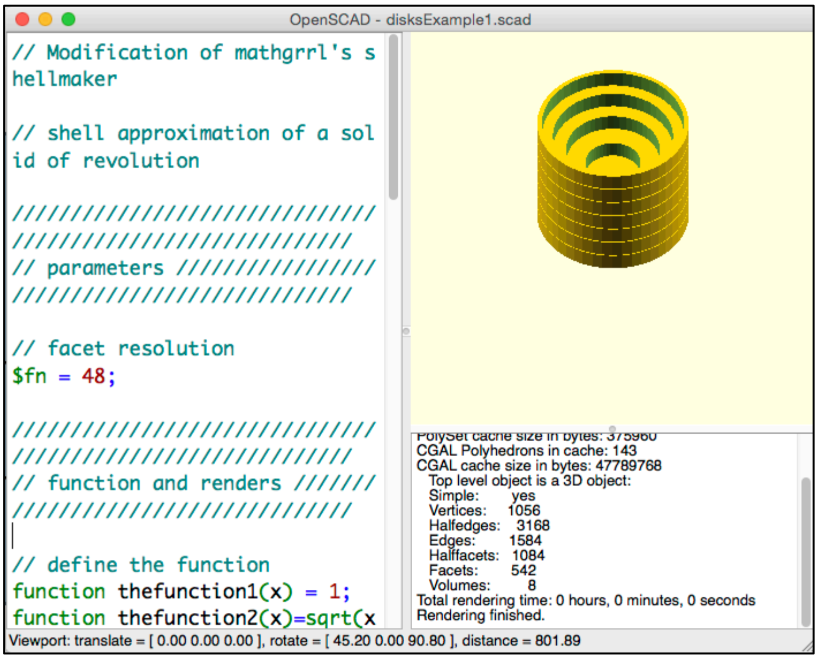


Figure 4.1 3D Prints gone wrong. Left: result of filament misfeed.
Right: result of bad adhesion to print bed.

One of the best resources for the 3D printing novice is a blog [23] by Dr. Laura Taalman (@mathgrrl) of James Madison University in the United States. Dr. Taalman's extensive collection of 3D models, tips and tricks for successful models, and sharing project fails can help build confidence in one's ability to successfully print models. A particularly useful resource for this author was the "One 3D Print A Day for a Year" [24] in which Dr. Taalman experimented with various 3D prints, including prints suitable for use in mathematics classrooms. Day 238 [25] of the project involved OpenSCAD [26] modeling solids of revolution using shells. It was the shellmaker OpenScad code that inspired the 3D prints in Figure 3.7; with some modest modifications the code can generate various volume approximations by changing the function definitions. Changing some of the other parameters makes it possible to approximate using disks and washers (Figure 4.2).

Another of Dr. Taalman's useful resources is a blog entry on selecting 3D modeling software [27]. In this blog article, she discusses different combinations of software and gives examples. This is very useful for the novice who is just starting as well as for intermediate and advanced 3D modelers alike.



OpenSCAD - disksExample1.scad

```
// Modification of mathgrrl's s
hellmaker

// shell approximation of a sol
id of revolution

///////////////////////////////
/////////////////////////////
// parameters /////////////////
/////////////////////////////

// facet resolution
$fn = 48;

///////////////////////////////
/////////////////////////////
// function and renders ///////
/////////////////////////////

// define the function
function thefunction1(x) = 1;
function thefunction2(x)=sqrt(x
```

```
PolySet cache size in bytes: 375960
CGAL Polyhedrons in cache: 143
CGAL cache size in bytes: 47789768
   Top level object is a 3D object:
   Simple:      yes
   Vertices:    1056
   Halfedges:   3168
   Edges:       1584
   Halffacets:  1084
   Facets:      542
   Volumes:       8
Total rendering time: 0 hours, 0 minutes, 0 seconds
Rendering finished.
```

Viewport: translate = [0.00 0.00 0.00], rotate = [45.20 0.00 90.80], distance = 801.89

Figure 4.2 Modeling solid of revolution using disks and washers.

5. Summary

As mathematics educators, we want to provide students opportunities to put contextual meaning to abstract ideas. The evolution of advanced 3D modeling with various readily available software tools that can create visualizations helps us to bring difficult to imagine objects to life through interactive and dynamic applications. We can extend our reach to students who have visual impairments using 3D printing and by doing so, we also improve experience for sighted students. Open source 3D graphing tools and decreasing prices of 3D printers make these enhanced opportunities broadly available to instructors and students.

Acknowledgements The author would like to acknowledge the efforts of my good friend and colleague, Dr. David R. Hill (Temple University, retired) to the Demos with Positive Impact Project. The project was his brainchild and it was great fun working with him and together learning the technology of the time. We continue to be amazed that many tasks with which we struggled then are today so much easier than they were in the early 2000s due to advancement and availability of software and hardware. The initial Demos with Positive Impact project was partially funded by the National Science Foundation under grant DUE-9952306 and was the first effort of its kind to create a repository of peer-reviewed ready-to-use demos, most of which involved visualizations. We, and the contributors to the project, provided suggestions for instructors for use in their mathematics classrooms. While the project is no longer active, the demos housed on the site continue to attract both instructors and students from high school to collegiate level.

References

[1] Knill, O. and Slavkovsky, E. (2013) Illustrating Mathematics using 3D Printers. ArXiv.org.

[2] Geelan, D. (2012). Effectiveness of Scientific Visualizations in Year 11 Chemistry and Physics Education. In TBC (Ed.), *Australian Computers in Education Conference 2012: Refereed Conference Proceedings* (pp. 1-10). Australia: University of Southern Queensland.

[3] Crouch, C. H., Fagen, A. P., Callan, J. P., and Mazur, E. (2004). Classroom Demonstrations: Learning Tools or Entertainment? *American Journal of Physics*. Volume 72, Issue 6. 835-838. https://works.swarthmore.edu/fac-physics/203.

[4] *Demos with Positive Impact*. http://www.mathdemos.org.

[5] Wolfram Research, Inc., Mathematica, Version 2.2, Champaign, IL (1993).

[6] Roberts, L. F. and Hill, D. R. (2002). Washer Method Demo Gallery. *Demos with Positive Impact*, http://www.mathdemos.org.

[7] Roberts, L. F. and Hill, D. R. (2002). Shell Method Demo Gallery. *Demos with Positive Impact*, http://www.mathdemos.org.

[8] Wolfram Research, Inc., Mathematica, Version 12.2, Champaign, IL (2019).

[9] Seeburger, Paul. *Visualizing Mathematics in 2D & 3D*, Monroe Community College, sites.monroecc.edu/multivariablecalculus/.

[10] Roberts, L. F. (2003). Applying Calculus to Approximate Volumes of Physical Objects. *Demos with Positive Impact*, http://mathdemos.org/mathdemos/volumeapps/.

[11] O'Neal, J. (2003). Enlightening Volumes: Curve Fitting to Approximate Volumes. *Demos with Positive Impact*, https://mathdemos.org/mathdemos/bulbdemoframes/.

[12] Hohenwarter, M. (2001) *Geogebra Online*, GeoGebra, https://www.geogebra.org.

[13] GeoGebra Team. (2016) GeoGebra Books Editor. https://www.geogebra.org/m/P5Zrj0Su.

[14] Roberts, L. F. (2016) GeoGebraBook: Visualizing Solids of Revolution and Volumes. https://www.geogebra.org/m/dT474KzV.

[15] Brzezinski, T. (2018) How to Create a Custom Surface of Revolution in GeoGebra. https://youtu.be/euHh-ClVSrk.

[16] Thurber, A. and Bandy, J. (2018). Creating Accessible Learning Environments. Retrieved 8/19/2019 from http://cft.vanderbilt.edu/guides-sub-pages/creating-accessible-learning-environments.

[17] Siu, Y-T. (2014) 3D Printing for Accessible Materials in Schools, Final Report. UC Berkeley and San Francisco State University. Retrieved 8/19/2019 from http://diagramcenter.org/wp-content/uploads/2014/06/3D_FinalReport_SIU_3.docx.

[18] Chakravorty, D. (2019) STL File Format (3D Printing) – Simply Explained. Retrieved 8/19/2019 from https://all3dp.com/what-is-stl-file-format-extension-3d-printing/.

[19] Moebius Strip. WolframMathworld. http://mathworld.wolfram.com/MoebiusStrip.html.

[20] Simplify3D Software, https://www.simplify3d.com.

[21] Ultimaker 3, https//www.ultimaker.com.

[22] Aura (2018) Preparing Files for 3D Printing: File Fixing Terminology Explanation and Checklist. Retrieved 8/21/2019 from https://i.materialise.com/blog/en/preparing-files-for-3d-printing/.

[23] Taalman, L. https://mathgrrl.com/hacktastic/

[24] Taalman, L. (2014) One 3D Print a Day for a Year. http://makerhome.blogspot.com.

[25] Taalman, L. (2014) http://makerhome.blogspot.com/2014_04_21_archive.html.

[26] OpenSCAD, The Programmers Solid 3D CAD Modeller. http://www.openscad.org.

[27] Taalman, L. (2017) What 3D Design Software Should I Use? https://mathgrrl.com/hacktastic/2017/03/what-3d-design-software-should-i-use/.