3D Printed Science Projects Volume 1

Ideas for Your Classroom, Science Fair, or Home

Second Edition

Joan Horvath · Rich Cameron

Apress®

3D Printed Science Projects Volume 1

Ideas for Your Classroom, Science Fair, or Home

Second Edition

Joan Horvath
Rich Cameron

Apress®

3D Printed Science Projects Volume 1: Ideas for Your Classroom, Science Fair, or Home, Second Edition

Joan Horvath
Pasadena, CA, USA

Rich Cameron
Whittier, CA, USA

ISBN-13 (pbk): 979-8-8688-0341-3
ISBN-13 (electronic): 979-8-8688-0342-0
https://doi.org/10.1007/979-8-8688-0342-0

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Miriam Haidara
Development Editor: James Markham
Project Manager: Jessica Vakili

Cover designed by eStudioCalamar

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (https://github.com/whosawhatsis/3DP-Science-Projects). For more detailed information, please visit https://www.apress.com/gp/services/source-code.

In memory of Zillabell “Jane” Friesen

Table of Contents

Chapter 4: Airfoils..101

About the Authors

Joan Horvath and **Rich Cameron** cofounded Nonscriptum LLC in 2015. From their base in Pasadena, California, they consult for educational and scientific users in the areas of 3D printing, maker technologies, and hands-on STEM curriculum development. Joan and Rich are particularly interested in finding ways to use technologies like 3D printing to make math and science easier for everyone, particularly those who learn differently.

Joan and Rich have been prolific authors, with many books from Apress and Make: Community LLC. They have also written for *Make:* magazine and developed online courses for LinkedIn Learning (formerly Lynda.com). Links for all the above are on their website, `www.nonscriptum.com`.

Joan has taught at the university level in a variety of institutions, both in Southern California and online. Before she and Rich started Nonscriptum, she held a variety of entrepreneurial positions, including Vice President of Business Development at a Kickstarter-funded 3D printer company. Joan started her career with 16 years at the NASA/Caltech Jet Propulsion Laboratory, where she worked in programs including the technology transfer office, the Magellan spacecraft to Venus, and the TOPEX/Poseidon oceanography spacecraft. She holds an undergraduate degree from MIT in aeronautics and astronautics and a master's degree in engineering from UCLA.

Rich (known online as “Whosawhatsis”) is an experienced open source developer who has been a key member of the RepRap 3D printer development community for many years. His designs include the original spring/lever extruder mechanism used on many 3D printers, the RepRap Wallace, and the Deezmaker Bukito portable 3D printer. By building and modifying several of the early open source 3D printers to wrestle unprecedented performance out of them, he has become an expert at maximizing the print quality of filament-based printers. When he is not busy making every aspect of his own 3D printers better, from slicing software to firmware and hardware, he likes to share that knowledge and experience online so that he can help make everyone else’s printers better too.

Acknowledgments

The consumer 3D printing ecosystem would not exist in its current form without the open source 3D printing hardware and software community. We are particularly grateful to Marius Kintel, the main developer and maintainer of OpenSCAD software, and his collaborators for their software, which was used to develop the objects in this book.

The Apress production team made this process as seamless as a complex book like this can be. For the first edition, we dealt most directly with Mark Powers, Michelle Lowman, James Markham, Corbin Collins, and Welmoed Spahr. For the second edition, Jessica Vakili, James Markham, and Miriam Haidara guided the day-to-day aspects of the book's development and production.

We picked a lot of scientists' brains as we thought about how to model some of the concepts in this book. We particularly thank high school teacher Michael Cheverie for his insights into teaching chemistry. Joan's long-suffering astronomer husband, Stephen Unwin, was a huge help as we went back into some basic physics or just tried to get past the 3D modeling equivalent of writer's block. For the first edition, Frank Carsey, Dan Berry, Tim Thibault, and many others helped us out by reading a chapter draft or helping us think through alternative ways that we might model something.

We thank the staff, teachers, and students at the Windward School in Los Angeles and the Institute for Educational Advancement in Pasadena for inspiration and discussions of how students learn. We also were inspired to create these models in part by discussions with people in the community of teachers of the visually impaired, notably Mike Cheverie, Lore Schindler, Yue-Ting Siu, and the participants in the Benetech workshop organized by Lisa Wadors.

As one might expect, our library of 3D prints has evolved over the eight years or so between these two editions. We have drawn on one of the open source math models developed for our *Make: Calculus* book (Make: Community LLC, 2022), which in turn had been somewhat inspired by the math surface model in this book. We note the details in Chapter 1. We also appreciate the many people who had comments and critiques of the models and text in the first edition.

Finally, we are grateful to our families for putting up with our endless brainstorming, kitchen table commandeering, and test runs of explanations both times around.

Introduction to the Second Edition

Almost eight years have gone by since the first edition of *3D Printed Science Projects*. During that time, 3D printers and their software have improved so much that we wound up nearly completely rewriting this book. We are disappointed that the environment in schools has not changed that much. 3D printing largely remains something relegated to key chain fobs.

We hope that this book, in its improved and expanded form, will coax more people into trying to be a bit more ambitious. It is still true that students, parents, and teachers get excited about using a 3D printer, maybe download a model, print it, and then wonder what to do next. Or, they might get into creating models from scratch and get discouraged by the limitations of easier 3D modeling programs or the learning curves of the more capable ones.

We are trying to create a middle path: models that you could just print, but that would be reasonably easy to alter if you wanted to do more. Further, we designed the models so that they would be useful for learning science or math principles while you were changing their features. We wanted to create some seeds of science fair or extra-credit projects—that is, open-ended, meaty explorations that could be explored at a variety of levels.

We were surprised at how hard this turned out to be. Most textbooks and online sites endlessly recycle versions of the same 2D projection of models of science concepts. In each chapter, we have a "Learning Like a Maker" section where we talk about our adventures in defining and implementing the models—which in some cases involved finding online copies of 1935 manuscripts (signed off by a Wright brother and Charles Lindbergh!). In others, it meant figuring out what to do when scientific experts said that everyone teaches a subject one way, but unlearns it in grad school anyway.

This book presumes that you know a little bit about 3D printing already. If not, Chapter 1 and the resources linked there should get you up to speed. The models are all written using the OpenSCAD free and open source 3D modeling program. If you know how to program in a language like C, Java, or Python, that will help, but it is not strictly necessary to alter the models. Chapter 1 and the OpenSCAD materials linked there will help you out with that too.

We have found that teachers use 3D printers in one of two fundamental ways. Either they want to create a model to pass around in class to help students visualize a concept, or they want students to use a printer either to learn engineering and design per se or to cement physical concepts like levers and gears. Since most of these models would lend themselves to being used either way, we have not included a grade level or explicit lesson plans.

To show our readers who are teachers (in the United States) what we had in mind, though, at the end of most chapters, we suggest Next Generation Science Standards (NGSS) that we think might benefit from these models. These science standards, from the group NGSS Lead States, are documented in *Next Generation Science Standards: For States, By States* (The National Academies Press, 2013). Links are given at the end of relevant chapters. If you are a teacher, you may want to check with your state or school standards as well to see the best fit.

The models span a variety of topics, and we tried to cover as many disciplines as possible. Briefly, here is what you can look forward to:

- Chapter 1 gives you a few options to print many different types of mathematical surface. This ability underlies some of the other models. We also discuss the process of 3D printing in this chapter and how to download the models and review the software the models are written in—the free program OpenSCAD. (The 3D printing material that was in Appendix in the first edition is now integrated into Chapter 1.)

- Chapter 2 creates models of waves to allow you to explore what happens when waves overlap and interfere with each other. You can print yourself a model of Young's famous double-slit experiment to see how light from two slits can interfere.

- Chapter 3 takes us back to Newton and Kepler to learn about planets and stars and how they speed up and slow down in their orbits.

- Chapter 4 allows you to create wings with classic airfoil shapes from the early days of flight. You will be able to make yourself a very simplistic test stand that you can use to measure the lift from the wing with just a fan and a postal scale.

- Chapter 5 lets you create basic models of all the "simple machines"—wedge, inclined plane, lever, pulleys, and screws.

- Chapter 6 allows you to model plants and their ecosystems and to design plants for different environments. Maybe you can create a garden for another planet (or for the Earth after another few hundred years of climate change).

- Chapter 7 lets you begin to explore carbon atoms and how water molecules come together to make two different types of ice crystals.
- Chapter 8 explores 2D and 3D trusses and how you can use them in various explorations.
- Chapter 9 (new in the second edition) builds on Chapter 5's simple machines to explore gears.
- Finally, an appendix aggregates all the links in the book.

We are making the 3D printable models used in this book (although not the book itself!) open source, licensed under a Creative Commons Attribution 4.0 International License (`https://creativecommons.org/licenses/by/4.0/`). That means you can use them for any purpose (including selling prints yourself) and alter and remix them if you credit us. Chapter 1 has some notes about where to find the repositories if you would like to add to these models. We hope these models are the beginning of a set that students everywhere can play with and learn from.

CHAPTER 1

Math Modeling with 3D Prints

Scientific visualization starts with the underlying mathematics. Thus, we are beginning this book on 3D printing for science projects with a chapter on 3D printing mathematical functions. The basic models in this chapter are intended to be a starter set that you alter to 3D print whatever function you like within the boundaries we will get to in a later section.

First, we need to introduce you to the open source program we used to create our 3D printable models, OpenSCAD. Then we use OpenSCAD to create a 3D printed mathematical surface defined by an equation. Then, we cover the 3D printing basics you will need to know to create the models. Finally, we conclude with a brief introduction to 3D printing and resources to help you go farther.

J. Horvath and R. Cameron, *3D Printed Science Projects Volume 1*,
https://doi.org/10.1007/979-8-8688-0342-0_1

MODELS USED IN THIS CHAPTER

This chapter uses two different OpenSCAD models. Select 3D printable STL example files are included in the repository as well, which we explain in the chapter text. The OpenSCAD models are

- `BlockyMath.scad`: This model creates a blocky 3D surface, shown in Listing 1-1.
- `surfaceprint.scad`: This model creates a smooth 3D mathematical surface based on an equation of the form $z = f(x, y)$. It is shown in Listing 1-2.

You also need to install the OpenSCAD program, as we describe in the chapter. As of this writing, OpenSCAD does not work on a Chromebook or similar tablet. A MacOS, Windows, or Linux computer is required.

Math Modeling for 3D Printing

It seems like it should be easy to just put an equation into a program and have a 3D printer “draw” it, like some sort of pen plotter. However, if a 3D printer head just tried to follow an equation, it would have no way of knowing how to avoid material that had already been laid down, so we go about it in a bit more roundabout way.

OpenSCAD

3D printers require a several-step process from the first idea to a finished print. First, you need to develop a 3D model. Models in this book are built using OpenSCAD (`www.openscad.org`), a free, open source 3D solid modeling program. Then, other software takes this file and slices it into layers, which the printer will then create one at a time, typically from the bottom up. This software is called a "slicer," and we will discuss common features of these programs in the "3D Printing" section of this chapter. The output of the slicer is what the 3D printer needs to make something.

Note Models in this book were optimized for printers that use filament and were designed to work well even on a printer that might not be tuned perfectly. If you have a printer that uses liquid resin or powder, you might have to make some adjustments.

OpenSCAD Basics

OpenSCAD allows you to encode geometrical models in a programming language that is like those in the C/Java/Python family. Good documentation is available by clicking the `Documentation` button on the OpenSCAD site's home page. We will denote OpenSCAD model code by showing it in `code font`. We want to acknowledge and thank Marius Kintel and the many other contributors to and maintainers of the program.

Downloading OpenSCAD

You can download OpenSCAD from `www.openscad.org`. Install the program per the instructions on the download site. OpenSCAD is available in versions for MacOS, Windows, and Linux. The models in this book were tested with version 2021.01 for MacOS. If you are a longtime OpenSCAD user and have an older version than that, you may need to update to the current version to be able to run the models in this book, which take advantage of some recently added features.

If you scroll down further on the downloads page, you can find OpenSCAD's nightly builds. These include some optimizations that dramatically increase rendering speed on some of our models, but they are not fully tested, and depending on when you download them, there may be significant software bugs. OpenSCAD has an excellent user manual at `www.openscad.org/documentation.html`. Note that there are block-coding-based versions of OpenSCAD, but as of this writing, they do not support all the functionality needed for our models.

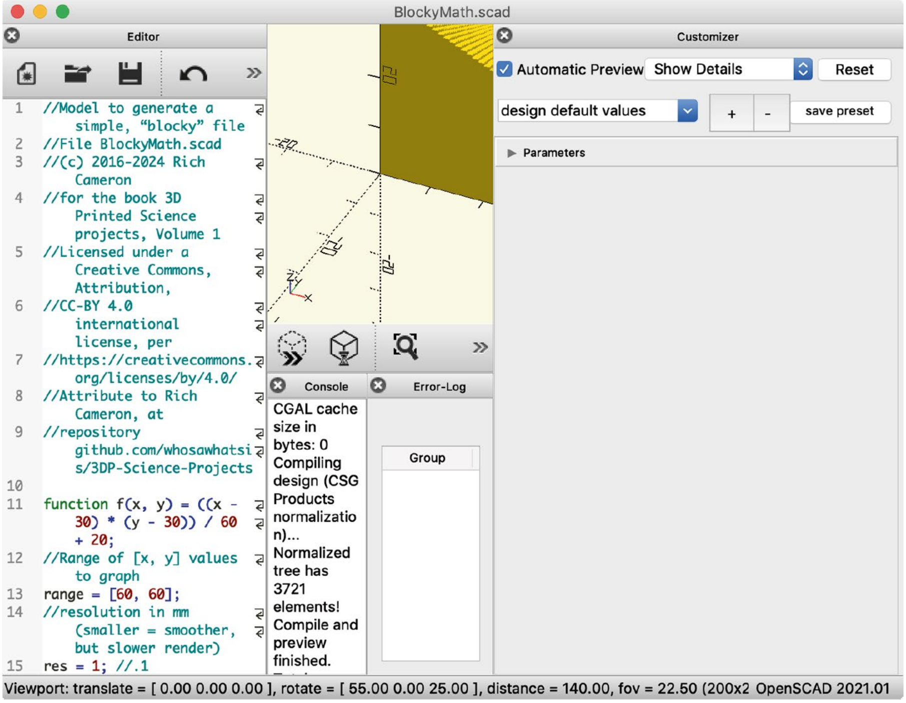
Editing the Models

Although the models in this book can be printed with their default values, the intent is that you use them as a starting point and make them your own. To edit one of the models in this book, first you would obtain the relevant OpenSCAD (`.scad`) file for the model you are interested in. See the "Archives and Repositories" section of this chapter for how that works.

Once you have a file you want to use, open OpenSCAD, click `File` ➤ `Open`, and open the `.scad` file. You may see four panes: one with code in it (the `Editor`), two labeled `Customizer` and `Console` that we will get to later, and one that starts off with just coordinate axes, which will in due course display a preview of our math model. (You might have to explicitly open one or more of these windows depending on your settings.)

Each of these panes can be resized, moved around, or hidden within the window, so your window may look a little different than the ones in our screenshots. If there are any of those panes that you do not see, go to the `View` menu, and uncheck the relevant `Hide` line (e.g., uncheck `Hide Editor` if you do not see the `Editor` window).

Figure 1-1 shows a very simple model that builds a surface out of blocks, which we explore in the next few sections. By default, OpenSCAD also previews the model for you. This can be turned off by unselecting `Design` ➤ `Automatic Reload and Preview` if you are working with models that take a while to render and you want to control it.



```
//Model to generate a simple, “blocky” file
//File BlockyMath.scad
//(c) 2016-2024 Rich Cameron
//for the book 3D Printed Science projects, Volume 1
//Licensed under a Creative Commons, Attribution,
//CC-BY 4.0 international license, per
//https://creativecommons.org/licenses/by/4.0/
//Attribute to Rich Cameron, at
//repository github.com/whosawhatsis/3DP-Science-Projects

function f(x, y) = ((x - 30) * (y - 30)) / 60 + 20;
//Range of [x, y] values to graph
range = [60, 60];
//resolution in mm (smaller = smoother, but slower render)
res = 1; //.1
```

```
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Normalized tree has 3721 elements!
Compile and preview finished.
```

```
Viewport: translate = [ 0.00 0.00 0.00 ], rotate = [ 55.00 0.00 25.00 ], distance = 140.00, fov = 22.50 (200x2 OpenSCAD 2021.01
```

Figure 1-1. *Opening a model in OpenSCAD*

Math Background

We have tried to avoid too many equations in this book. We do assume that you understand basic algebra, Cartesian coordinates, what it means to raise a number to a power or take a root of it, and trigonometric functions like sine, cosine, and tangent. But, with that said, science often requires some math to describe it, and that math then is used to create models.

We also presume you know what a mathematical function is—a relationship among a number of variables. In this case, we are dealing with functions using three variables, which we will call x, y, and z. Function notation looks like this: $z = f(x, y)$. All that means is that our variable, z, can be computed for any given pair of values for the x and y variables.

Having three variables means we can define shapes in three dimensions, with one variable corresponding to each dimension. Normally these three-dimensional shapes would be shown on a page with two-dimensional projections. This is often fine, and you can see what is going on. Sometimes, however, it really helps to hold a 3D model in your hand and turn it this way and that. This chapter will get you started on doing that for many types of functions.

3D printing convention holds that x and y are in the plane of the platform that your model is being built up on, and z is the vertical height above that (Figure 1-2). The bottom of the surfaces generated in this chapter is usually the $z = 0$ plane. In this convention, you always transform what you are printing to have z greater than or equal to zero since you cannot build under the platform.

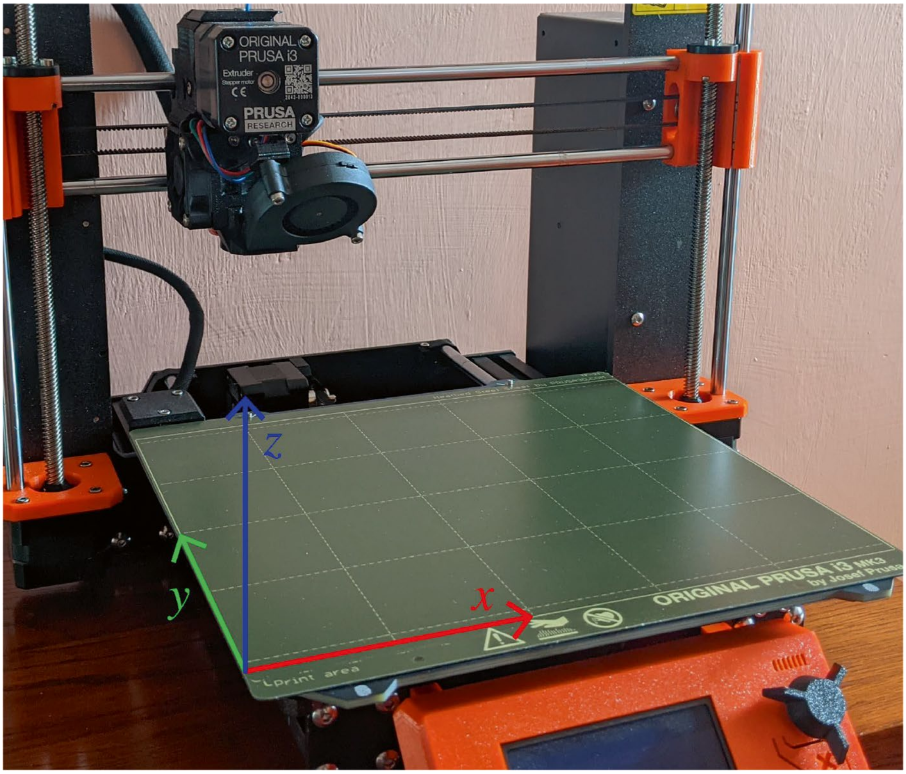


Figure 1-2. *Coordinate axes of a 3D printer*

In other words, if you know that z would be negative for some values of x and y that you want to use, you may have to add an offset to your equation so that z is always greater than zero and remember that the offset is there when you think about what your model represents. In the next section, we get you started with a model entirely in OpenSCAD that creates surfaces of functions

`z = f(x,y),`

where x and y are the plane of the 3D printer's build platform and z is the height of the surface above that plane.

Tip The maintainer of OpenSCAD has co-authored a book on learning to code with OpenSCAD, *Programming with OpenSCAD: A Beginner's Guide to Coding 3D Printable Objects* by Justin Gohde and Marius Kintel (2021: No Starch Press). You might find that a good resource for more depth on OpenSCAD's ins and outs. We also have a more in-depth review of OpenSCAD as a tool for learning math in our book *Make: Geometry* (2021: Make: Community LLC).

Simple "Blocky" Model

The simplest way to print a 3D surface is to compute the height of the surface on a regular grid of points. Then, we create a rectangular solid that is a small square at the base, with the height equal to the value of the function at the center of each of these small squares. The model in Listing 1-1 is an extremely simple one that will create an STL file with a surface of small rectangular solid pieces. The function in this example is

`z = f(x, y) = 0.01 (x - 50) (y - 50) + 30,`

and the 3D print will go from `x = 0` to `x = 60` and `y = 0` to `y = 60`. This creates a "saddle point" structure, as shown in Figure 1-6. The `x` and `y` dimensions of the model are determined by the `range = [x, y]` variable in millimeters, with `z` height computed, also in millimeters. If the resulting structure is too big (by default, 60 mm by 60 mm on the bottom, or a bit over 2 inches square), then you can scale the whole piece in your 3D printing software.

The values of `x` and `y` both start at 0 and go to the values specified by the range vector (read as `range[0]` for `x` and `range[1]` for `y`). The res variable, also in millimeters, lets you control the spacing of the calculated points, allowing you to make a smoother surface at the cost of additional processing time.

Table 1-1 lists the model parameters that can be changed. Figure 1-3 shows a surface generated this way. Notice a pattern on the top surface; this is a printing artifact that we will talk about in the "3D Printing" section.

Table 1-1. *Blocky Surface Model Variables*

Variable	**Default Value and Units**	**Meaning**
`f(x,y)`	mm	Function to graph.
`range`	[60, 60] mm	Maximum x/y values to graph. Determines the size of the print in those dimensions.
`res`	1 mm	Size of the boxes in the graph. Making this smaller produces a smoother surface but takes longer to process.

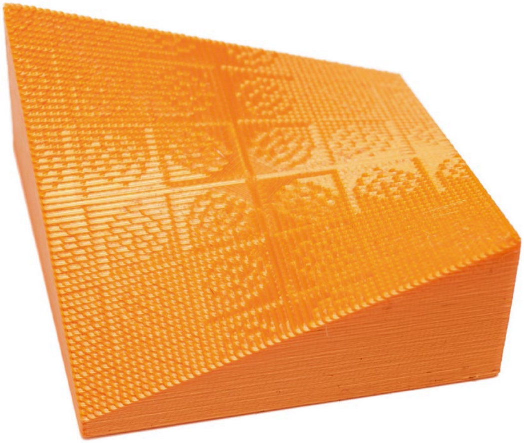


Figure 1-3. *The blocky surface, as printed*

Listing 1-1. OpenSCAD Model to Generate a "Blocky" Surface (file `blocky.scad`)

```
//Model to generate a simple, "blocky" file
//File BlockyMath.scad
//(c) 2016-2024 Rich Cameron
//for the book 3D Printed Science projects, Volume 1
//Licensed under a Creative Commons, Attribution,
//CC-BY 4.0 international license, per
//https://creativecommons.org/licenses/by/4.0/
//Attribute to Rich Cameron, at
//repository github.com/whosawhatsis/3DP-Science-Projects

function f(x, y) = ((x - 30) * (y - 30)) / 60 + 20;
//Range of [x, y] values to graph
range = [60, 60];
//resolution in mm (smaller = smoother, but slower render)
res = 1; //.1

for(x = [0:res:range[0]], y = [0:res:range[1]])
  translate([x, y, 0])
    cube([res + .001, res + .001, f(x, y)]);
```

To see this surface in OpenSCAD, use the menu item `Design` ➤ `Preview`. You can also do this by clicking on the little box outlined with dashed lines with a double arrow on it. Make any changes you feel you need to make by editing the text in the `Editor` and click `Preview` again to see if you have created what you intended. Repeat until you think you are done.

Note You cannot make any changes other than by changing the model code in the Editor. There is no drag and drop capability in OpenSCAD.

In OpenSCAD, `Preview` creates an object you can view but cannot export. It is a lot faster than a full render, which can take a long time for some of the models in this book. Use this to preview models as you are making changes. When you have your final model, go to `Design` ➤ `Render` (or click the button with the solid-outlined cube next to the preview button) to create a model that can be exported for 3D printing. This is called an *STL file* (more on this in the "3D Printing" section). You can export an STL file by clicking `File` ➤ `Export` ➤ `Export as STL`. The editor pane also has an STL export button that you can use.

There is another way to make changes to some of the models in this book. This is by using the `Customizer`. In Figure 1-1, you can see the `Customizer` pane has a pull-down menu labeled `Parameters`. If we expand that pull-down menu (Figure 1-4), we see the `range` and `res` boxes. We can change those values with the pull-down menu or type a new one. These values will then supersede the ones in the code until you close and reopen the file.

Many of the models in later chapters are designed so that you only need to alter parameters for the model in the `Customizer`, which avoids needing to touch the code that makes up the OpenSCAD file. In other cases, we talk you through how to make those changes. Note that the `Customizer` does not support mathematical expressions, so you still need to use the code editor for changing the function.