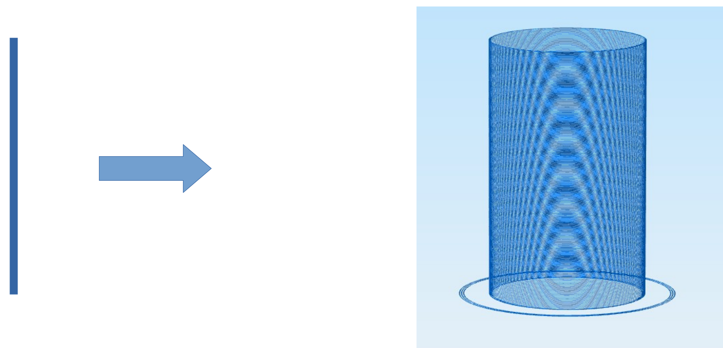
# Spinner

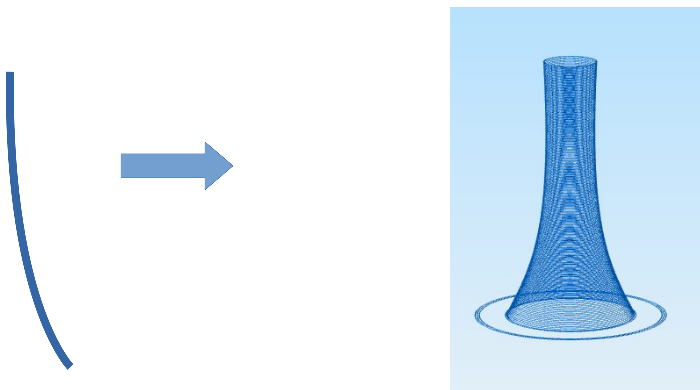
a tool for printing cylindrical skins.

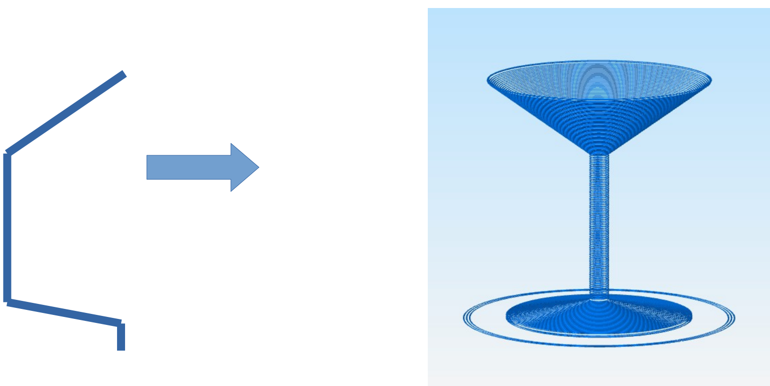
**Spinner** is a program for generating “idealized” gcode for “vase” printing of cylindrical shapes following a programmable “profile.” Spinner will evaluate the profile and set an “ideal” layer height for each position in the profile. What we mean by “ideal” is the largest layer height which will permit the next layer to overlap the previous layer by a target “threshold”, usually one half the nozzle diameter.

To illustrate, let’s take a couple profiles, and see what happens when we “spin” them:

**PROFILE** **“SPUN” MODEL**







## How to use Spinner

Spinner is a command line tool which takes these areguments:

spinner [-c *config.json* ] [ -p *profile.data* ] [ s start.gcode ] [ -e end.gcode ]

-c config.json

> This file defaults to the '**spinner_config.json**' and describes both characteristics of your printer (bed size, filament diameter, nozzle size, etc) and paramters for the print job like temperature, speed, and cooling.

-p profile.data

> This is the profile data expressed as a series of 2D points. These points are essentially [x, z], where [x,z] is the radius of the model (x) at height (z). This data should start with z=0; The file defaults to **spinner_profile.data**.

-s start.gcode

> This file contains your favorite startup code for you printer, it will be printed first. Following this gcode will be instructions to set bed and extruder temp and to wait for them to reach temperature. The default file is **spinner_start.gcode.**

-e end.gcode

> This file contains gcode for ending the print. The default file is **spinner_end.gcode**.

**The config file**:

Below is a sample *config* file:

```
// spinner global printer and print settings
{
    "nozzleSize"        : 0.80,
    "filamentDiameter ": 1.75,
    "bedsizeX"          : 400,
    "bedsizeY"          : 400,
    "bedTemp"           :  65,
    "eTemp"             : 215,
    "travelSpeed"       : 4000,
    "printSpeed"        : 700,
    "retractLength"     : 2,
    "retractSpeed"      : 3000,
    "printSpeeds"       : [ 1100,  350, 1100, 350 ],
    "fanStarts"         : [  0.5,   45,  105, 145 ],
    "fanSpeeds"         : [  150,  255,  150, 255 ],
    "layerHeight"       : 0.4,
    "extrusionFactor"   : 1.0
};
```

Most of these settings are self explanatory, but I'll highlight a few of the non-obvious points:

**1)** There is a table of data formed by the elements *fanStarts*, *fanSpeeds*, and *printSpeeds*. The table expresses how cooling and print speed vary with along the Z-axis. *FanStarts* is a list of points along the Z-axis expressed in mm.

In the example above, the print will proceed at printSpeed (700) until it reaches 0,5mm and then will increase to 1100, and turn the fan on to 150. (Fan speeds are expressed over the range of 0 – 255 inclusive). These settings remain constant until the model reaces 45mm in height, when the speed decreases to 300, and the fan increases to its maximum.

**2)** layerHeight is the Target Layer Height described in the previous section.

# The profile file

The profile data file is a list of 2D points in the following format:

`[ x, y ] …`

example: the straight vase data is like this:
 cat simple.data

[30, 0]

[30, 100]

$

newlines and whitespace are optional.

**How does spinner decide on the layer height?**

Spinner establishes a maximum xy offset (called the “Overhang Threshold” or OT) between layers, and uses this to “solve” for layer height. We supply the spinner with a several values needed to compute this:

1) The nozzle diameter (ND)
2) overhang tolerance (OT, usually half the ND)
3) The “Target Layer Height” (TLH)

The TLH is the maximum layer height which will be used when the adjacent layers have no overhang challenge. Specifically when:

```
abs ( radius_at(z) - radius_at (z+TLH) ) <  OT.
```

As long as the profile presents no overhang challenges, the model will print at the target layer height. As overhang challenges present themselves, meaning the above expression is false, then we attempt to find the ideal layer height via a this process:

**1) Compute the layer height from the slope implied by the line:**

```
  [ radius_at_(z), z], [ radius_at(z+TLH), z+TLH]
```

This step will compute a new layer height (H1) achieving an overhang of exactly OT. Unless, that is, the profile had more detail between z and z+TLH. Which is why we introduce the next step.

**2) Test the computed value.**
By computing the radius_at (z+H1) we should get exactly OT difference from radius_at(z). If the difference is still >OT, then **goto step 1** with a new TLH set to (H1)/2

**3) return H1.**