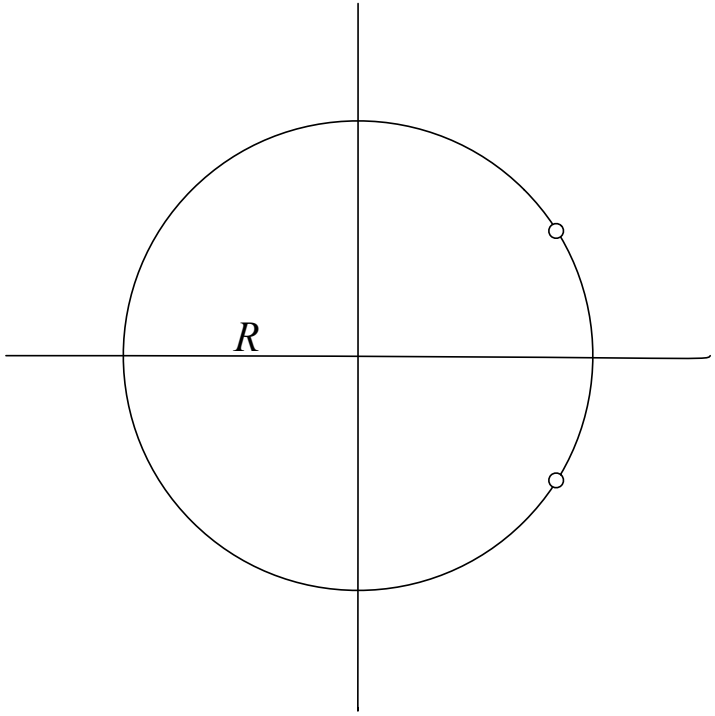
CMSC427
Drawing circles and more

Drawing a circle



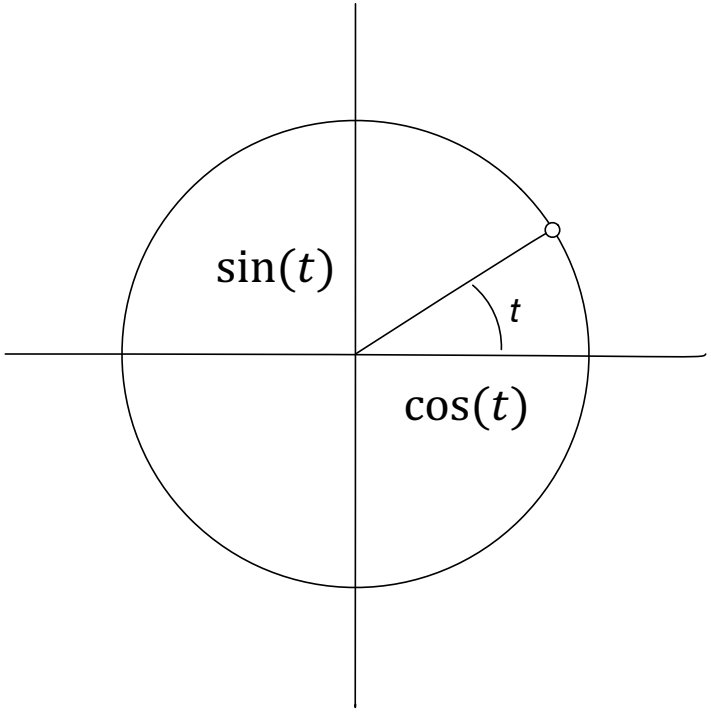
Implicit equation

$$x^2 + y^2 = R^2$$

"Functional" equation
(multi-valued)

$$y = \pm\sqrt{x^2 + R^2}$$

Circle with trig



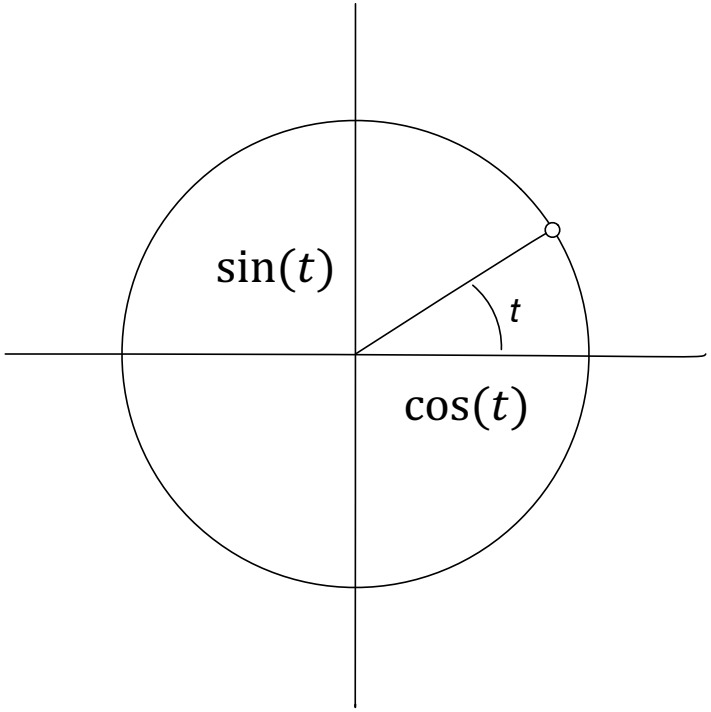
Parametric equation

$$x = R\cos(t)$$

$$y = R\sin(t)$$

$$0 \leq t \leq ??$$

Circle with trig



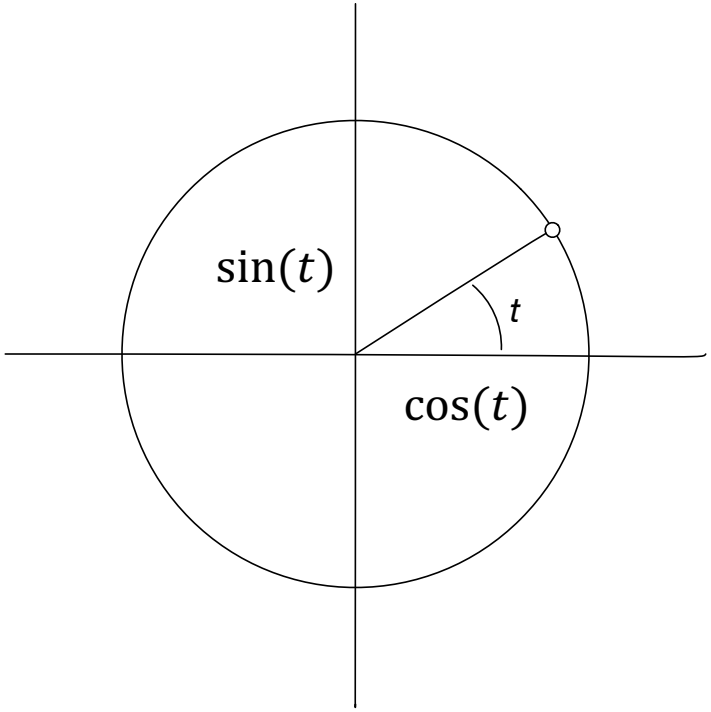
Parametric equation

$$x = R\cos(t)$$

$$y = R\sin(t)$$

$$0 \leq t \leq 2\pi$$

Validating parametric equations



Substitute parametric into implicit

$$x = R\cos(t)$$

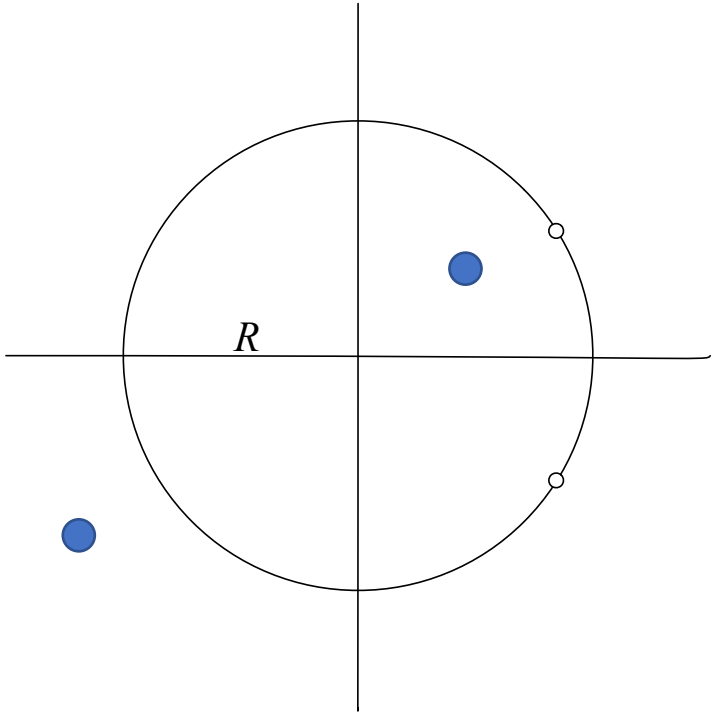
$$y = R\sin(t)$$

$$(R\cos(t))^2 + (R\sin(t))^2 = R^2$$

$$R^2(cos^2(t) + sin^2(t)) = R^2$$

$$R^2(1) = R^2$$

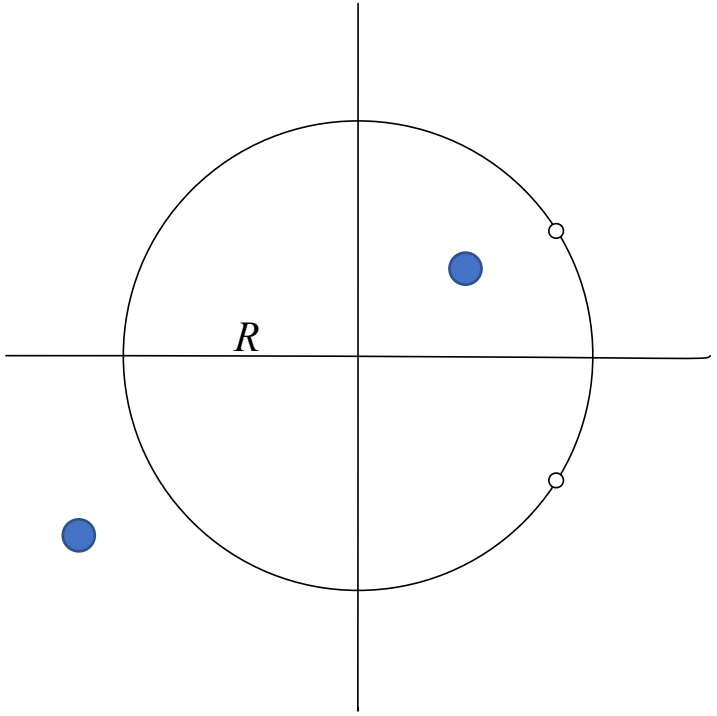
Inside or outside?



Implicit equation

$$x^2 + y^2 = R^2$$

Inside or outside?



Implicit equation

$$x^2 + y^2 = R^2$$

$x^2 + y^2 > R^2$ out

$x^2 + y^2 < R^2$ in

Summary of curve equations

- Implicit $f(x,y) = 0$
  - Inside/Outside tests
- Parametric $x = fx(t)$ $y = fy(t)$
  - Drawing/Intersection tests
- Functional $y = f(x)$
  - Dual purpose: drawing, testing

Coding parametric curves

Circle

```
for t = 0 to 2π by step 0.1
     x = R*cos(t)
     y = R*sin(t)
     putpixel(x,y)
```

Generic

given

$x = f_x(t)$

$y = f_y(t)$

```
for t = t0 to t1 by step dt
     x = fx(t)
     y = fy(t)
     putpixel(x,y)
```

Processing.org

Circle | Processing 3.3.5

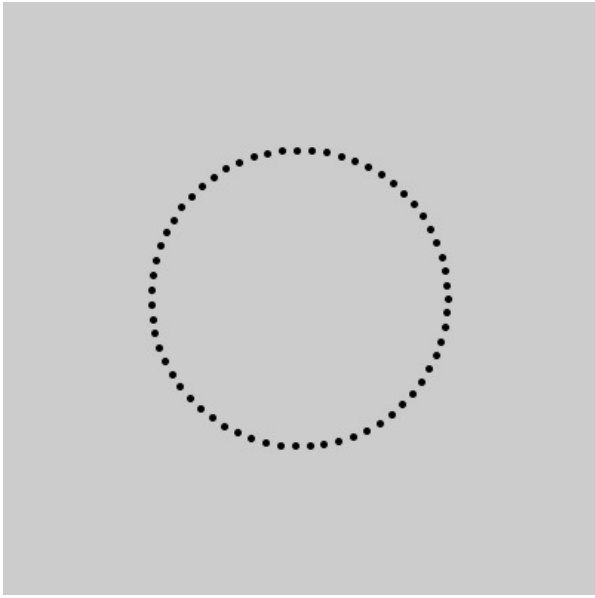
Java

Circle

```
1  // Parametric circle
2
3
4  size(400,400);
5  float r = 100;
6  strokeWeight(5);
7
8  for (float t = 0; t < 2*PI; t += 0.1) {
9    float x = width/2  + r*cos(t);
10   float y = height/2 + r*sin(t);
11   point(x,y);
12 }
13
14 save("curve.jpg");
```

Done saving.

Console Errors



Processing.org

```
size(400,400);
float r = 100;
strokeWeight(5); // Set point size in pixels

for (float t = 0; t < 2*PI; t += 0.1) {
   float x = width/2  + r*cos(t);
   float y = height/2 + r*sin(t);
   point(x,y);
   }

save("curve.jpg");
```

Playing with the code …

```
for (float t = 0; t < 2*PI; t += 0.3) { // Change increment
```

```
for (float t = 0; t < PI; t += 0.3) { // Change limits
```

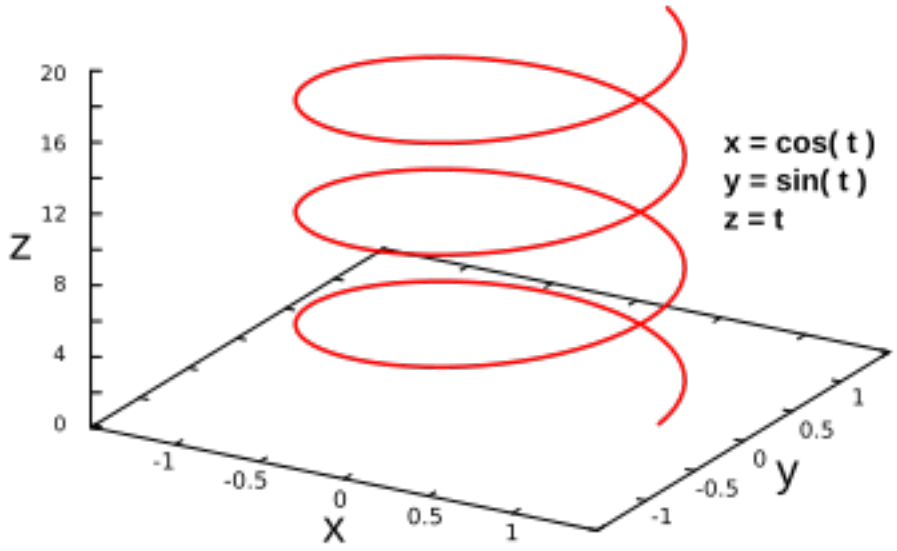
```
float x = width/2  + 1.5*r*cos(t);    // Unequal r
float y = height/2 + r*sin(t);
```

```
float x = width/2  + 10*t*cos(t);    // Varying r
float y = height/2 + 10*t*sin(t);
```

```
strokeWeight(random(1,10)); // Random point size
point(x,y);
```

Next steps

- Animate
- Collect more parametric curves
  - Cartesian
  - Polar
- Add third dimension
- Move to surfaces

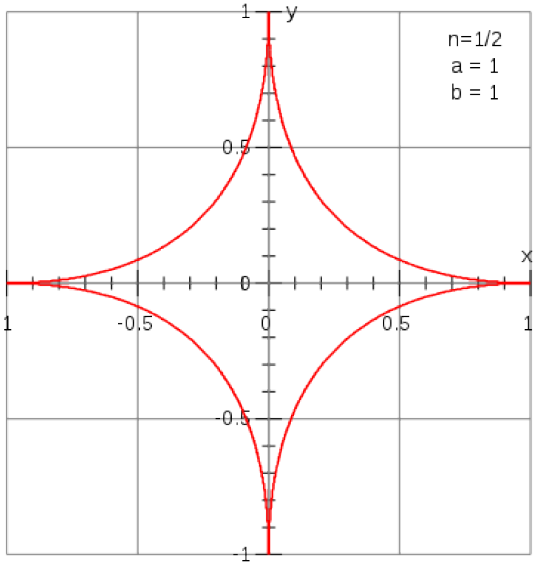


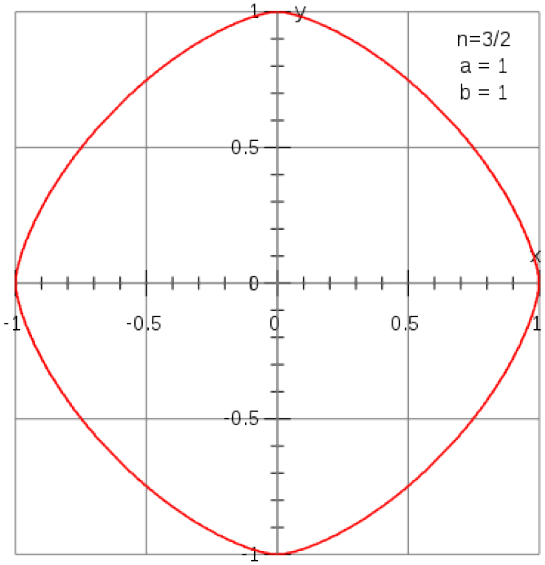
Animate

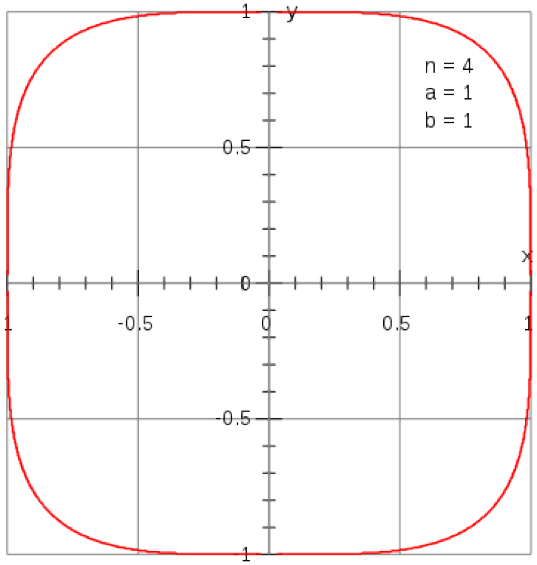
```
// Infinity Rainbow
// from Jason Labbe (jasonlabbe3d.com)
void setup() {
      size(600,600);
      colorMode(HSB,255); // Hue/Saturation/Brightness
      background(255);
      noStroke();
}

void draw() {
      fill(frameCount % 255,255,255);
      float x = (width/2)+100*sin(frameCount/20.0);
      float y = (height/2)+200*sin(frameCount/10.0);
      ellipse(x,y,20,20);
      }
```

Superellipses

Generalized ellipse that can take different shapes based on exponent n







$$|x|^n + |y|^n = R^2$$

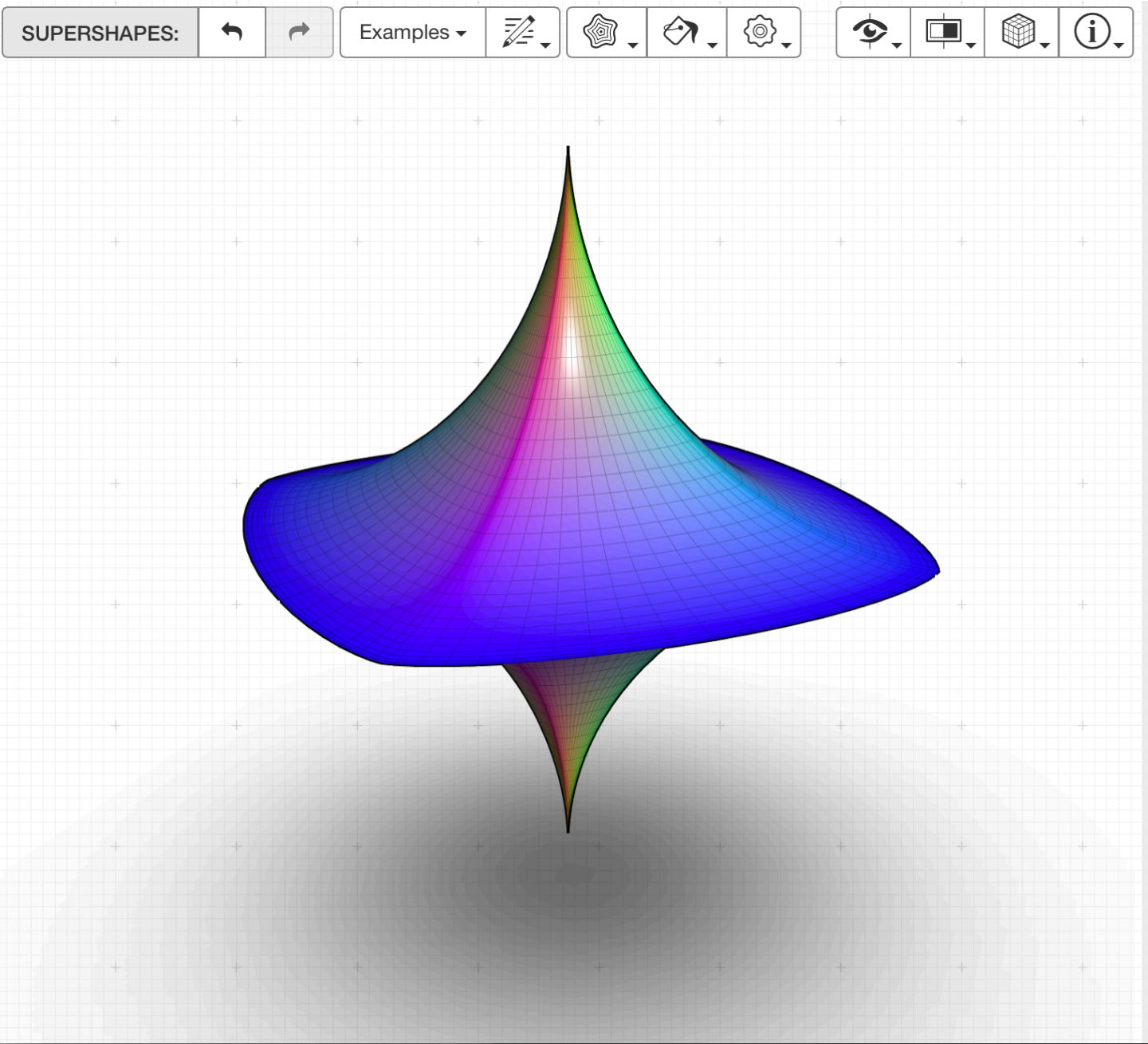
$$x = R\cos^{\frac{2}{n}}(t)sgn(\cos(t))$$

$$y = R\sin^{\frac{2}{n}}(t)sgn(\sin(t))$$

https://www.desmos.com/calculator

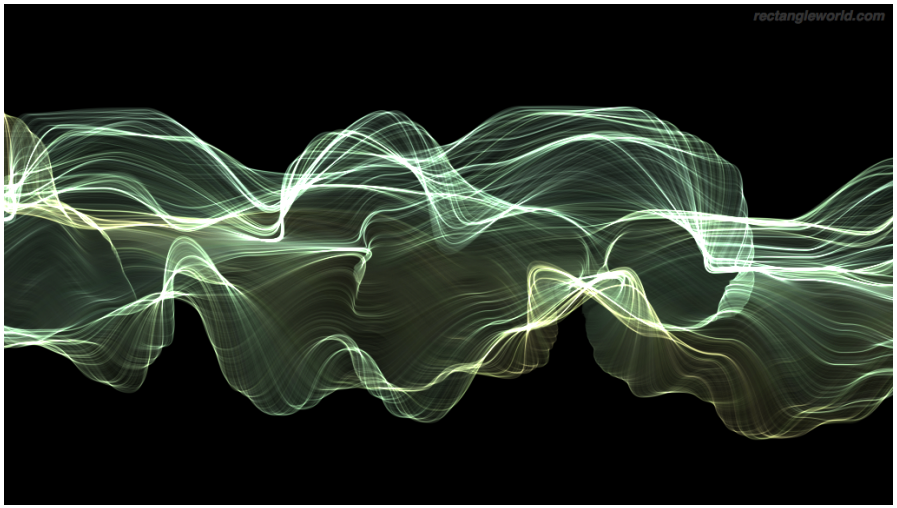
https://en.wikipedia.org/wiki/Superellipse

Andrew March supershapes



What you should know

1. How to draw circles and more with parametric equations
2. Using powers of fractions to bias curves
3. Using parametric equations for curves, shapes, animations and other purposes
4. How to play with a parametric curve to get variations and randomness



Resources

- Desmos
  - https://www.desmos.com/calculator
  - 2D graphing with parametric curves
- Andrew Marsh's supershapes
  - http://andrewmarsh.com/apps/staging/supershapes.html
  - Super ellipsoids and similar shapes
- Processing
  - https://processing.org
  - Resource for quick program "sketches", concepts
  - Sketches Circle.pde, InfinityRainbow.pde, HelixFly.pde, RaggedCircles.pde, CircleToSpiral.pde