Parametric Valentine

We are about to create an image that students will encounter in virtually all math curricula. It is so common, though, because it is a perfect combination of mathematical elegance and a very recognizable result.

The Graph

We usually graph y as a function of x. In this case, though, we will define our function in terms of two different parameters: r and θ. Like x and y, they define a point in space, but in a different way: r refers to the distance from the origin, while θ indicates the angle of elevation from the horizontal of r. We can relate our x-y coordinate system to our r-θ cordinate system with the equations

```
x = r * Cos[θ]
y = r * Sin[θ]
```

Our cardioid is defined by the relationship:

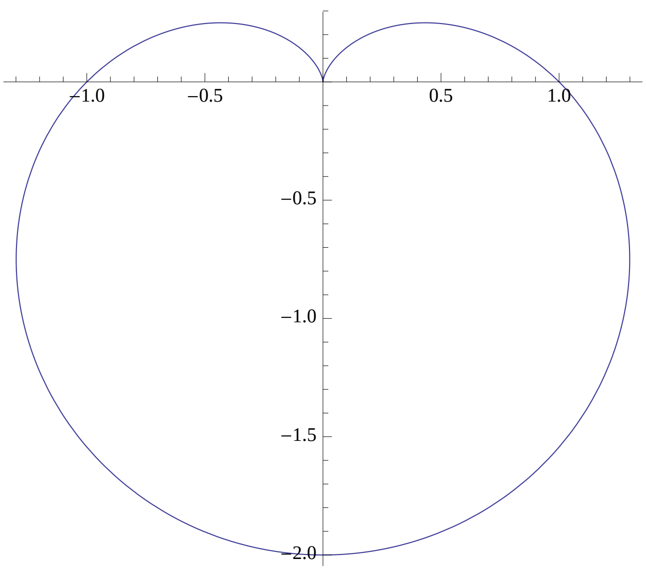
```
r = 1 - Sin[θ]
```

So, converting to x and y, we get the parametric equations

```
x = (1 - Sin[θ]) Cos[θ]
y = (1 - Sin[θ]) Sin[θ]
```

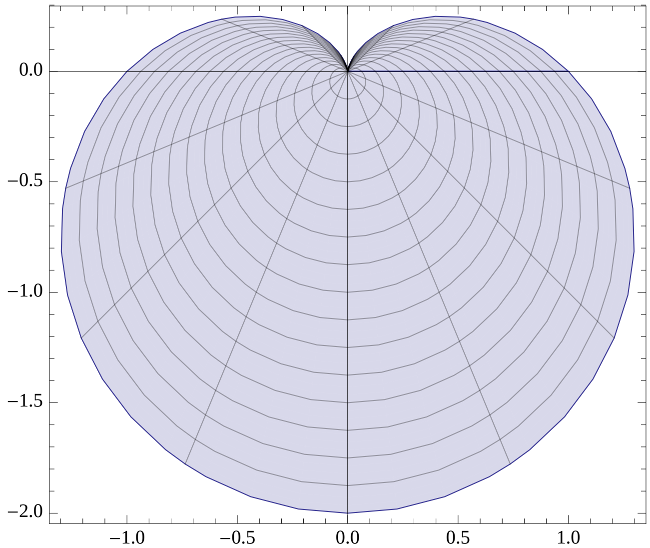
So let's graph it!

```
ParametricPlot[{(1 - Sin[θ]) Cos[θ], (1 - Sin[θ]) Sin[θ]}, {θ, 0, 2 π}]
```



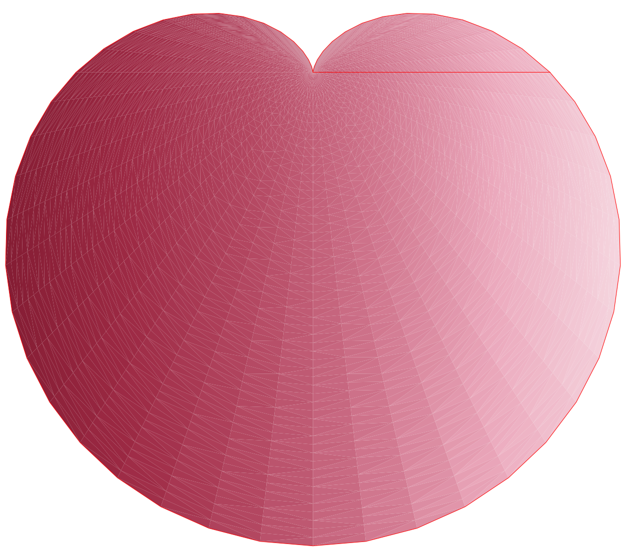
Aww. To pretty it up, let's add another parameter, s, which will vary from 0 to 1. Essentially, this will allow us to create an infinite number of concentric hearts, resulting in a solid-looking shape.

```
ParametricPlot[{s (1 - Sin[θ]) Cos[θ], s (1 - Sin[θ]) Sin[θ]}, {θ, 0, 2 π}, {s, 0, 1}]
```



And finally, let's apply some visual tweaks:

```
ParametricPlot[{s (1 - Sin[θ]) Cos[θ], s (1 - Sin[θ]) Sin[θ]},
 {θ, 0, 2 π}, {s, 0, 1}, ColorFunction → "ValentineTones", Axes → False,
 Mesh → False, Frame → False, BoundaryStyle → Directive[Thin, Red]]
```



Let's get nerdier

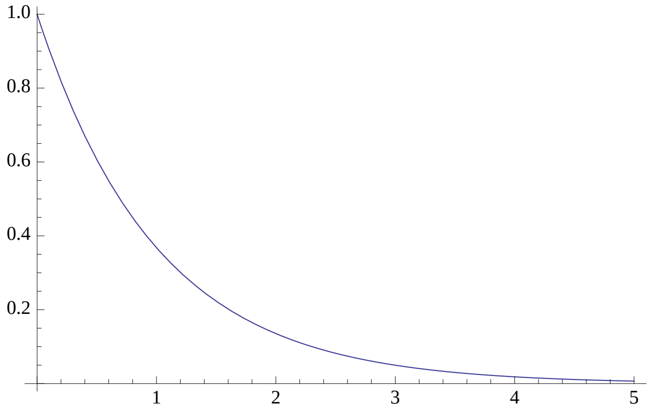
So that's the classic cardioid. It's simple and good-looking, but it isn't a perfect heart. And since we're celebrating Valentine's day, not learning polar coordinates, let's try to improve it.

Exponential decay functions are generally of the form

$$y = A * e^{-Bx}$$

and they tend to look something like this:

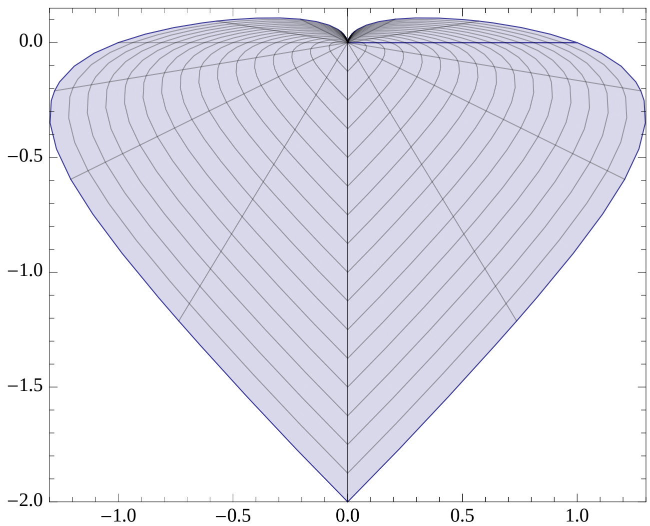
```
Plot[ⅇ^-x, {x, 0, 5}]
```



The end of a heart is typically pointed, not rounded and bulky like our cardioid. Perhaps, if we multiply the "y" values of our cardioid by some exponential decay function, we'll be able to generate a pointed end.

In[6]:=
```
ParametricPlot[{s (1 - Sin[θ]) Cos[θ], s (1 - Sin[θ]) Sin[θ] ⅇ^-Abs[Cos[θ]]},
 {θ, 0, 2 π}, {s, 0, 1}, PlotRange → {{-1.3, 1.3}, {-2, .15}}]
```

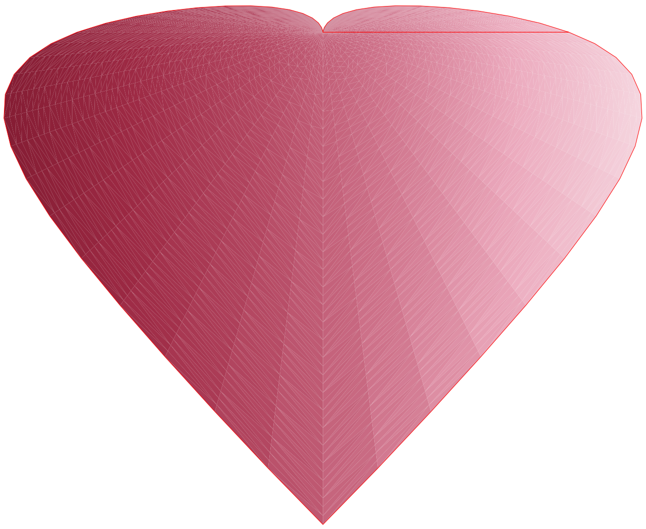
Out[6]=



So we pretty it up again:

In[7]:=
```
ParametricPlot[{s (1 - Sin[θ]) Cos[θ], s (1 - Sin[θ]) Sin[θ] e^(-Abs[Cos[θ]])}, {θ, 0, 2 π},
  {s, 0, 1}, PlotRange → {{-1.3, 1.3}, {-2, .15}}, ColorFunction → "ValentineTones",
  Axes → False, Mesh → False, Frame → False, BoundaryStyle → Directive[Thin, Red]]
```

Out[7]=



We could make this even prettier by tweaking constants here and there to make some things bulkier and some things pointier, but this is left as an exercise to the reader.

Even more impressive

For a truly astounding array of heart-themed math (in 3D!), we direct the curious user to http://mathworld.wolfram.com/HeartSurface.html.