1 Implicit Differentiation

1.1 Graphs of equations

An **equation** in y and x is an algebraic expression involving an equality with two (or more) variables. An example might be $x^2 + y^2 = 1$.

The **solutions** to an equation in the variables x and y are all points (x, y) which satisfy the equation.

The **graph** of an equation is just the set of solutions to the equation represented in the Cartesian plane.

With this definition, the graph of a function $f(x)$ is just the graph of the equation $y = f(x)$.

In general, graphing an equation is more complicated than graphing a function. For a function, we know for a given value of x what the corresponding value of $f(x)$ is through evaluation of the function. For equations, we may have 0, 1 or more y values for a given x and even more problematic is we may have no rule to find these values.

To plot such an equation in `Julia`, we can use the `ImplicitEquations` package, which is loaded when `CalculusWithJulia` is:

```
using CalculusWithJulia
using Plots
gr()   # better graphics than plotly() here
```

```
Plots.GRBackend()
```

To plot the circle of radius 2, we would first define a function of *two* variables:

```
f(x,y) = x^2 + y^2
```

```
f (generic function with 1 method)
```

> This is a function of *two* variables, used here to express one side of an equation. `Julia` makes this easy to do - just make sure two variables are in the signature of `f` when it is defined. Using functions like this, we can express our equation in the form $f(x, y) = c$ or $f(x, y) = g(x, y)$, the latter of which can be expressed as $h(x, y) = f(x, y) - g(x, y) = 0$. That is, only the form $f(x, y) = c$ is needed.

Then we use one of the logical operations - `Lt`, `Le`, `Eq`, `Ge`, or `Gt` - to construct a predicate to plot. This one describes $x^2 + y^2 = 2^2$:

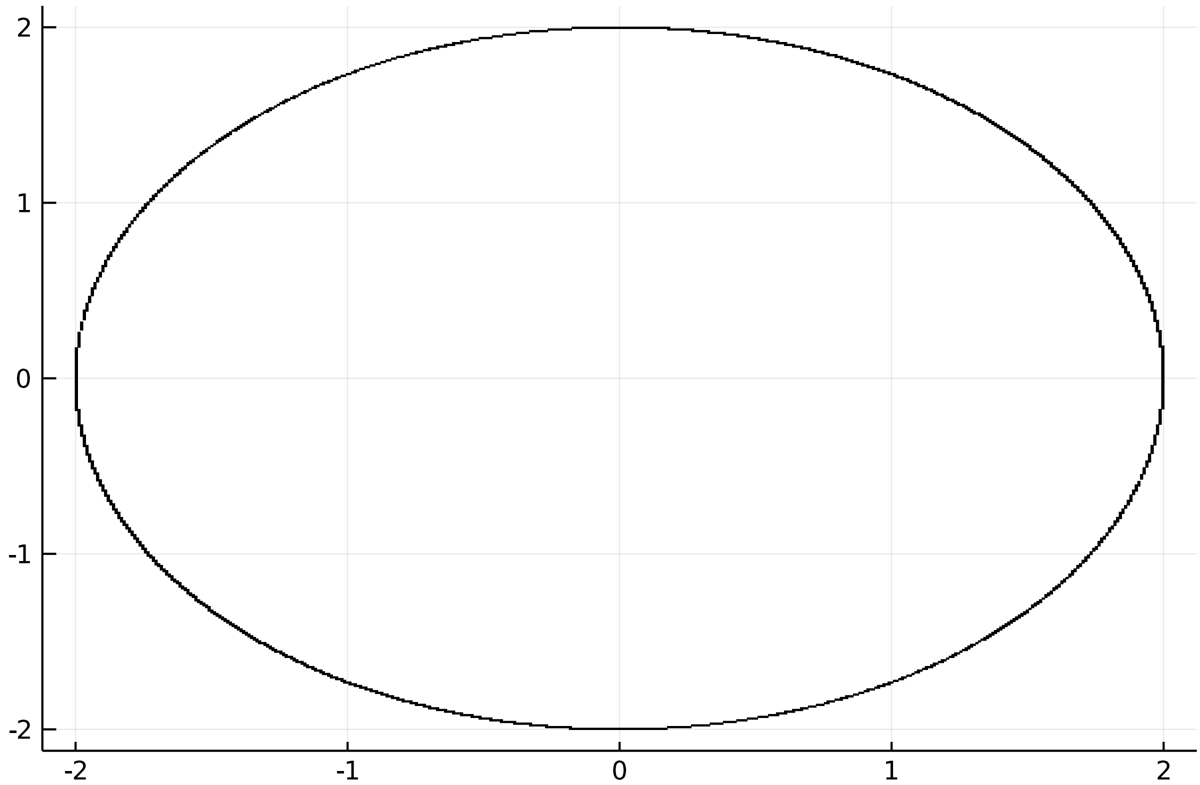
```
r = Eq(f, 2^2)
```

```
ImplicitEquations.Pred(Main.##WeaveSandBox#667.f, ==, 4)
```

There are unicode infix operators for each of these which make it easier to read at the cost of being harder to type in. This predicate would be written as `f  2^2` where is **not** two equals signs, but rather typed with `\Equal[tab]`.)

These "predicate" objects can be passed to `plot` for visualization:

```
plot(r)
```

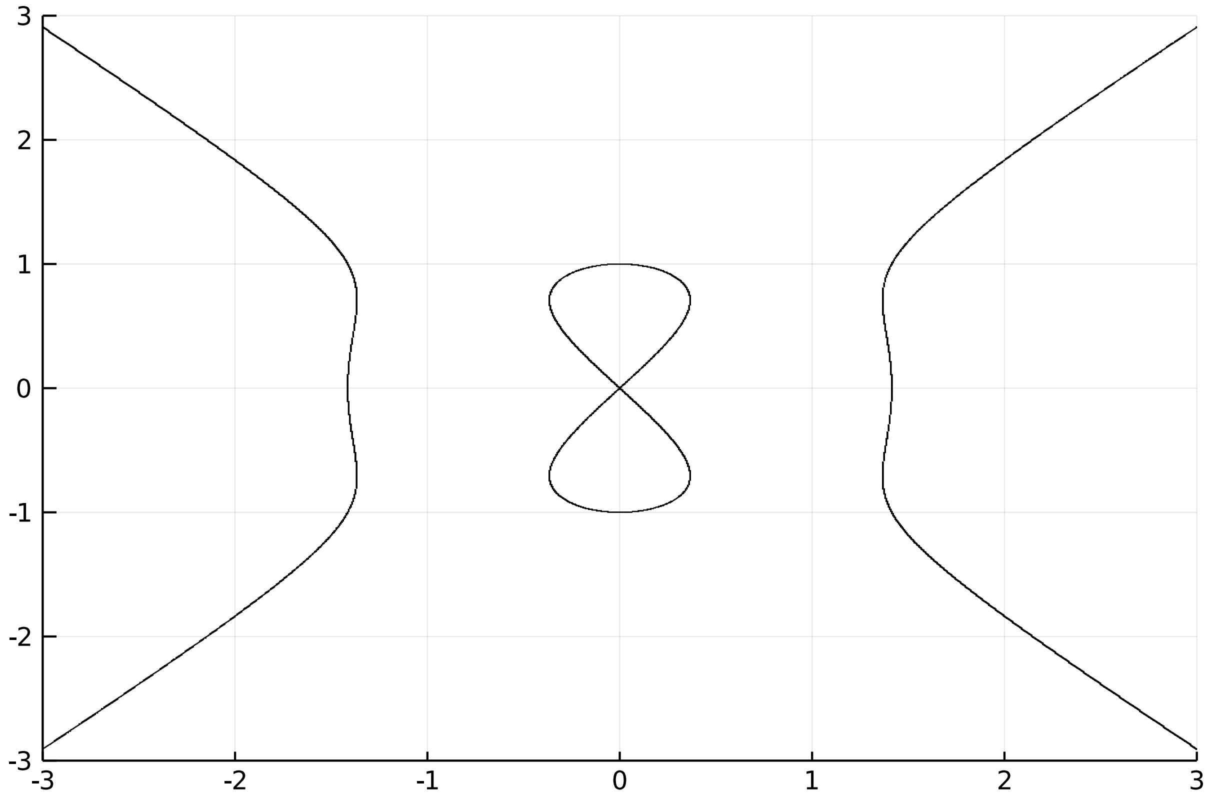


Of course, more complicated equations are possible and the steps are similar - only the function definition is more involved. For example, the Devils curve has the form

$$y^4 - x^4 + ay^2 + bx^2 = 0$$

Here we draw the curve for a particular choice of a and b. For illustration purposes, a narrower viewing window than the default of $[-5, 5] \times [-5, 5]$ is specified below using `xlims` and `ylims`:

```
a,b = -1,2
f(x,y) =  y^4 - x^4 + a*y^2 + b*x^2
plot(Eq(f, 0), xlims=(-3,3), ylims=(-3,3))
```



The rendered plots look "blocky" due to the algorithm used to plot the equations. As there is no rule defining (x, y) pairs to plot, a search by regions is done. A region is initially labeled undetermined. If it can be shown that for any value in the region the equation is true (equations can also be inequalities), the region is colored black. If it can be shown it will never be true, the region is dropped. If a black-and-white answer is not clear, the region is subdivided and each subregion is similarly tested. This continues until the remaining undecided regions are smaller than some threshold. Such regions comprise a boundary, and here are also colored black. Only regions are plotted - not (x, y) pairs - so the results are blocky. Pass larger values of $N = M$ (with defaults of 8) to `plot` to lower the threshold at the cost of longer computation times.

1.1.1 The IntervalConstraintProgramming package

The `IntervalConstraintProgramming` package also can be used to graph implicit equations. For certain problem descriptions it is significantly faster and makes better graphs. The usage is slightly more involved:

We specify a problem using the `@constraint` macro. Using a macro allows expressions to involve free symbols, so the problem is specified in an equation-like manner:

```
using IntervalArithmetic, IntervalConstraintProgramming
S = @constraint x^2 + y^2 <= 2
```

```
Separator:
  - variables: x, y
  - expression: x ^ 2 + y ^ 2 ∈@*( [-(*@∞@*(, 2]
```

The right hand side must be a number.

The area to plot over must be specified as an `IntervalBox`, basically a pair of intervals. The interval $[a, b]$ is expressed through `a..b`.

```
X = IntervalBox(-3..3, -3..3)
```

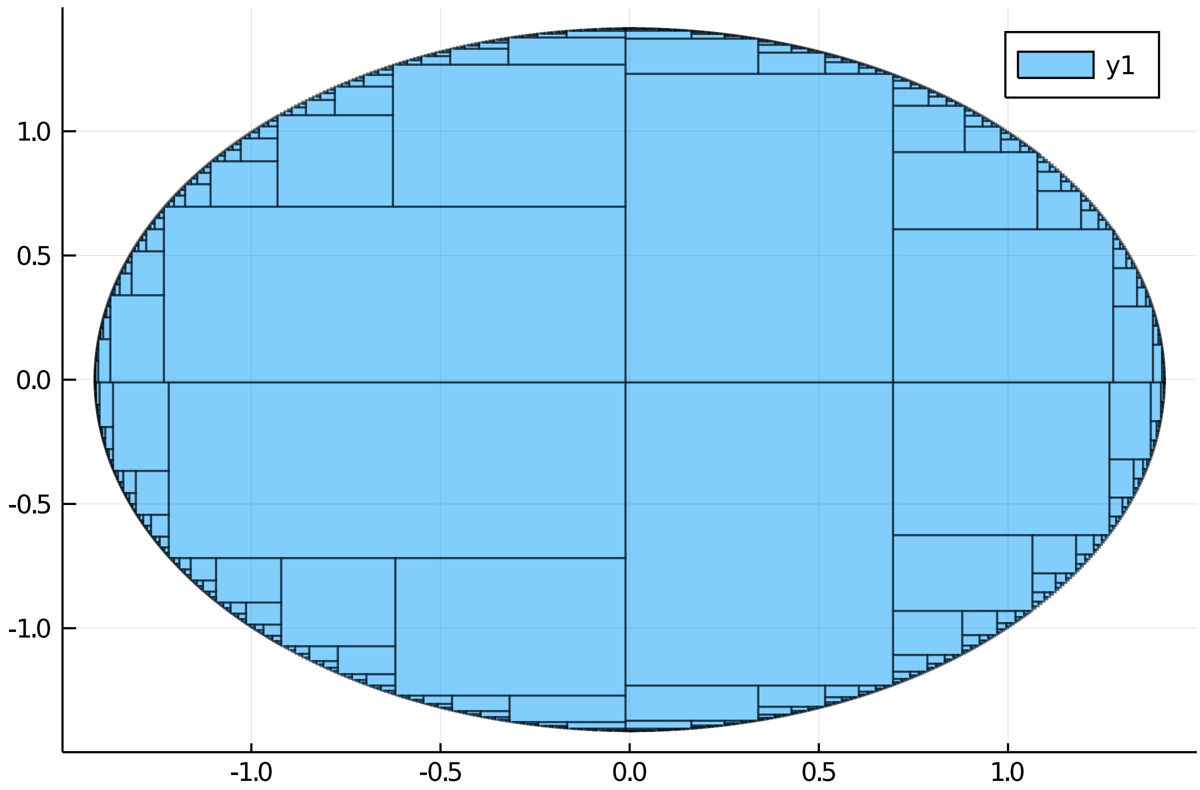
```
[-3, 3] ×@*( [-3, 3]
```

```
r = pave(S, X)
```

```
Paving:
- tolerance ϵ@*( = 0.01- inner approx. of length 1152- boundary approx. of length 1156
```

We can plot either the boundary, the interior, or both.

```
plot(r.inner)       # plot interior; use r.boundary for boundary
```



1.2 Tangent lines, implicit differentiation

The graph $x^2 + y^2 = 1$ has well-defined tangent lines at all points except $(-1, 0)$ and $(0, 1)$ and even at these two points, we could call the vertical lines $x = -1$ and $x = 1$ tangent lines. However, to recover the slope would need us to express y as a function of x and then differentiate that function. Of course, in this example, we would need two functions: $f(x) = \sqrt{1-x^2}$ and $g(x) = -\sqrt{1-x^2}$ to do this completely.

In general though, we may not be able to solve for y in terms of x. What then?

The idea is to *assume* that y is representable by some function of x. This makes sense, moving on the curve from (x, y) to some nearby point, means changing x will cause some change in y. This assumption is only made *locally* - basically meaning a complicated graph is reduced to just a small, well-behaved, section of its graph.

With this assumption, asking what dy/dx is has an obvious meaning - what is the slope of the tangent line to the graph at (x, y).

The method of implicit differentiation allows this question to be investigated. It begins by differentiating both sides of the equation assuming y is a function of x to derive a new equation involving dy/dx.

For example, starting with $x^2 + y^2 = 1$, differentiating both sides in x gives:

$$2x + 2y \cdot \frac{dy}{dx} = 0.$$

The chain rule was used to find $d/dx(y^2) = 2y \cdot dy/dx$. From this we can solve for dy/dx (the resulting equations are linear in dy/dx, so can always be solved explicitly):

$$dy/dx = -x/y$$

This says the slope of the tangent line depends on the point (x, y) through the formula $-x/y$.

As a check, we compare to what we would have found had we solved for $y = \sqrt{1-x^2}$ (for (x, y) with $y \geq 0$). We would have found: $dy/dx = 1/2 \cdot 1/\sqrt{1-x^2} \cdot -2x$. Which can be simplified to $-x/y$. This should show that the method above - assuming y is a function of x and differentiating - is not only more general, but can even be easier.

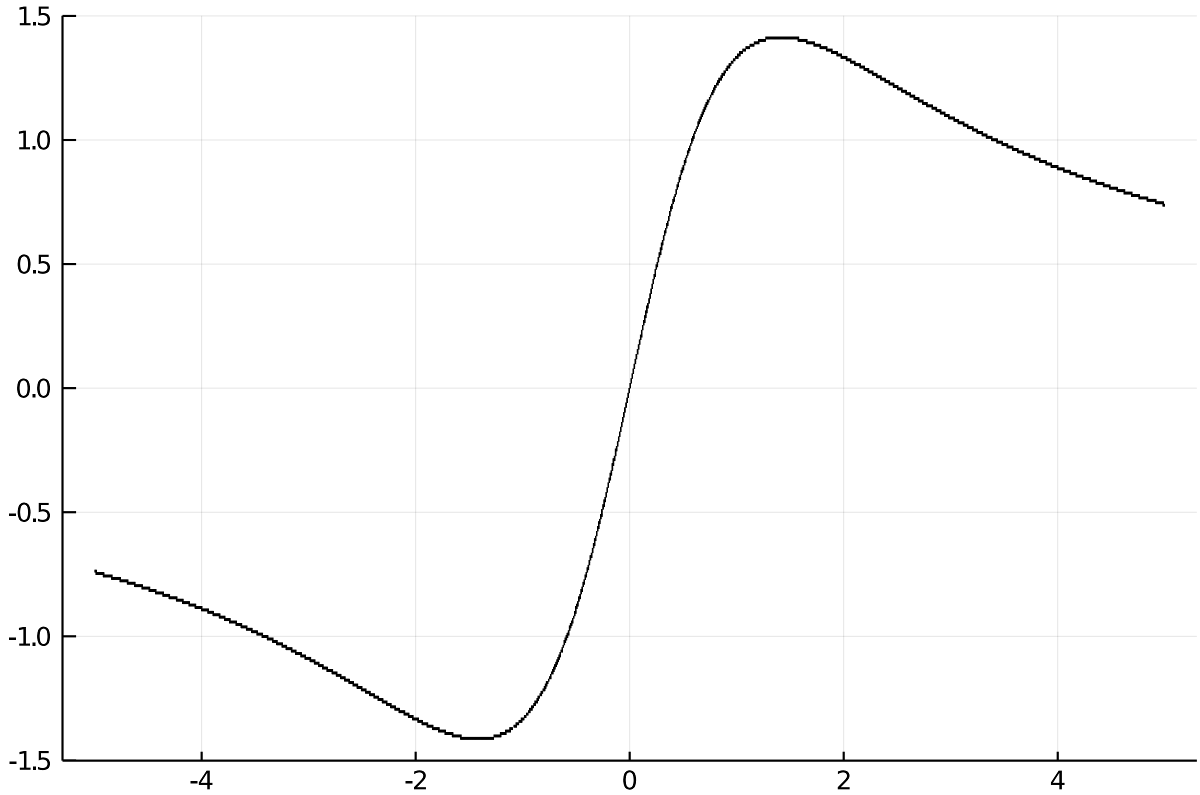
The name - *implicit differentiation* - comes from the assumption that y is implicitly defined in terms of x. According to the Implicit Function Theorem the above method will work provided the curve has sufficient smoothness near the point (x, y).

Examples Consider the serpentine equation

$$x^2y + a \cdot b \cdot y - a^2 \cdot x = 0, \quad a \cdot b > 0.$$

For $a = 2, b = 1$ we have the graph:

```
a, b = 2, 1
f(x,y) = x^2*y + a * b * y - a^2 * x
plot(Eq(f, 0))
```



We can see that at each point in the viewing window the tangent line exists due to the smoothness of the curve. Moreover, at a point (x, y) the tangent will have slope dy/dx satisfying:

$$2xy + x^2 \frac{dy}{dx} + a \cdot b \frac{dy}{dx} - a^2 = 0.$$

Solving, yields:

$$\frac{dy}{dx} = \frac{a^2 - 2xy}{ab + x^2}.$$

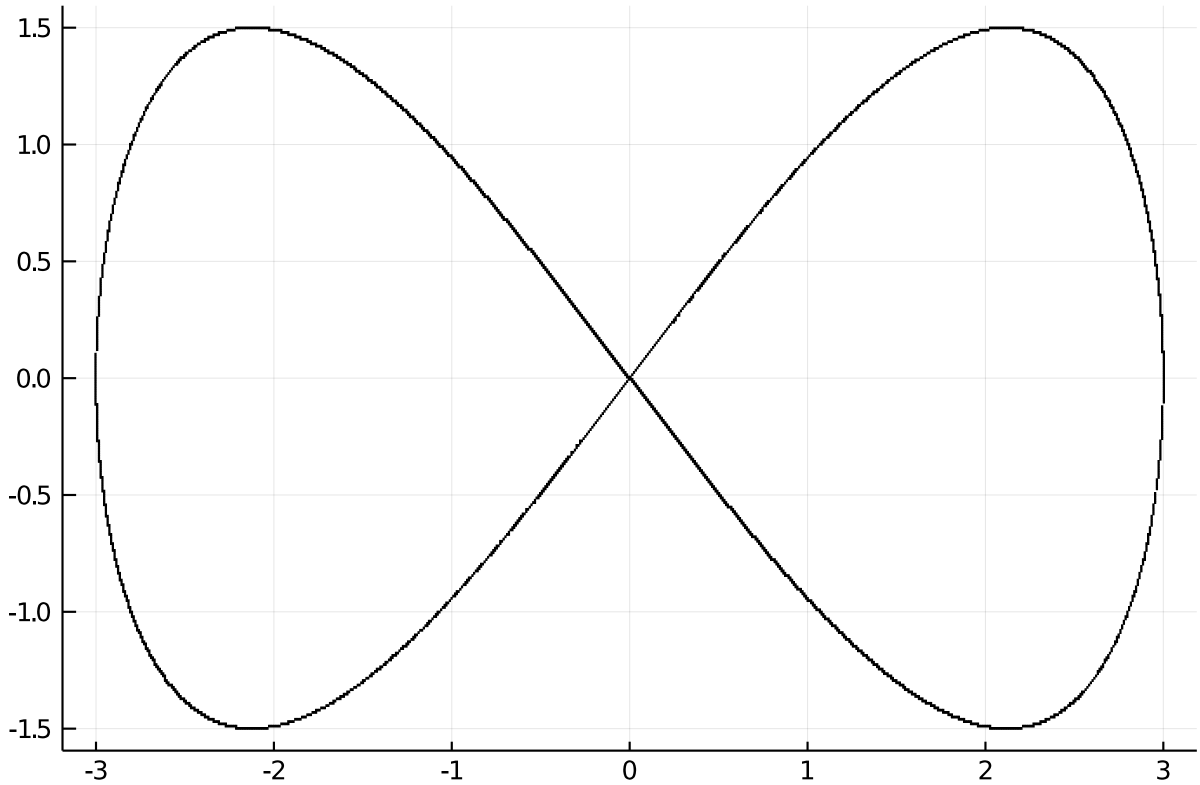
In particular, the point $(0, 0)$ is always on this graph, and the tangent line will have positive slope $a^2/(ab) = a/b$.

The eight curve has representation

$$x^4 = a^2(x^2 - y^2), \quad a \neq 0.$$

A graph for $a = 3$ shows why it has the name it does:

```
a = 3
f(x,y) = x^4 - a^2*(x^2 - y^2)
plot(Eq(f, 0))
```



The tangent line at (x, y) will have slope, dy/dx satisfying:

$$4x^3 = a^2 \cdot (2x - 2y\frac{dy}{dx}).$$

Solving gives:

$$\frac{dy}{dx} = -\frac{4x^3 - a^2 \cdot 2x}{a^2 \cdot 2y}.$$

The point $(3, 0)$ can be seen to be a solution to the equation and should have a vertical tangent line. This also is reflected in the formula, as the denominator is $a^2 \cdot 2y$, which is 0 at this point, whereas the numerator is not.

Example The quotient rule can be hard to remember, unlike the product rule. No reason to despair, the product rule plus implicit differentiation can be used to recover the quotient rule. Suppose $y = f(x)/g(x)$, then we could also write $yg(x) = f(x)$. Differentiating implicitly gives:

$$\frac{dy}{dx}g(x) + yg'(x) = f'(x).$$

Solving for dy/dx gives:

$$\frac{dy}{dx} = \frac{f'(x) - yg'(x)}{g(x)}.$$

Not quite what we expect, perhaps, but substituting in $f(x)/g(x)$ for y gives us the usual formula: