The `CATools.jl` guide

Evert Provoost
evert.provoost@kuleuven.be

1 Installation

`CATools.jl` (short for complex analysis tools) is a collection of plotting tools to aid the study of complex analysis. It is implemented in Julia, a modern and—at least in the author's opinion—a rather pleasant language to use. No prior knowledge is needed to go through this document and concepts will be introduced where needed. If Julia is not yet installed on your system, you can use the usual method on your operating system or download it from `https://julialang.org`. We strongly recommend you use at least version 1.9 or later!

To install the plotting routines, you enter the following commands in the Julia REPL (read eval print loop). **Note:** Depending on the speed of your machine, this might take a while. Grab a cup of tea or coffee and take a minute to enjoy the weather. :-)

```
using Pkg
pkg"add https://gitlab.kuleuven.be/u0158737/catools.git"
```

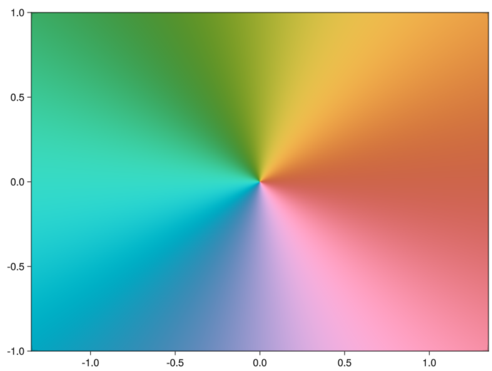
2 A first phase plot

To start producing plots, you first have to load the package you just installed with the `using` keyword.

```
using CATools
```

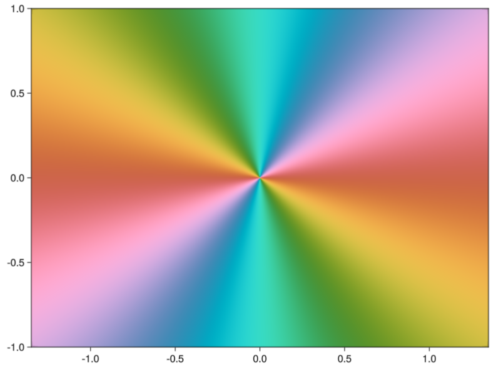
In the remainder of this tutorial we will discuss several plots, starting with the phase plot (we will explain the name `domaincolor` shortly). On such a plot we display the phase, or argument, where $\arg z = 0$ is painted red, $\arg z = \frac{\pi}{2}$ lime green, $\arg z = \pi$ cyan, $\arg z = \frac{3\pi}{2}$ purple, etc.

```
domaincolor(z -> z)
```



We note that a zero (anticlockwise) goes from red through green to blue. A pole on the other hand (below one of second order) gives red, blue and then green.

```
domaincolor(z -> 1/z^2)
```

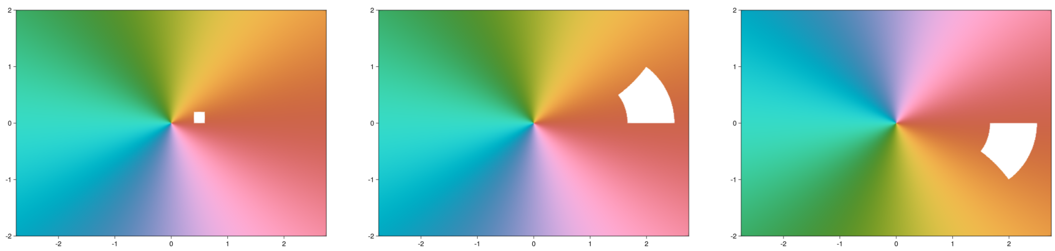


Note the first bit of Julia syntax you will need: an anonymous function (a function you can easily pass as an argument) is written as `z -> f(z)`, analogous to $z \mapsto f(z)$ in mathematics.

It is perhaps interesting to know that this reversal of the phase relates to the complex conjugate. The function $z \mapsto \frac{1}{z}$ is namely the composition of $z \mapsto \frac{1}{\bar{z}} = \frac{1}{|z|}e^{i \arg z}$ and $z \mapsto \bar{z} = |z|e^{-i \arg z}$. Where, in the first step, the unit circle is turned inside out. It is the second step which mirrors the image along the real axis.

This is easily seen by looking at the effect these steps have on a square, or, algebraically, by noting that only the final step modifies the phase.

```
domaincolor(z -> z, 2, box=(.4,.6+.2im,:white))
domaincolor(z -> 1/conj(z), 2, box=(.4,.6+.2im,:white))
domaincolor(z -> 1/z, 2, box=(.4,.6+.2im,:white))
```



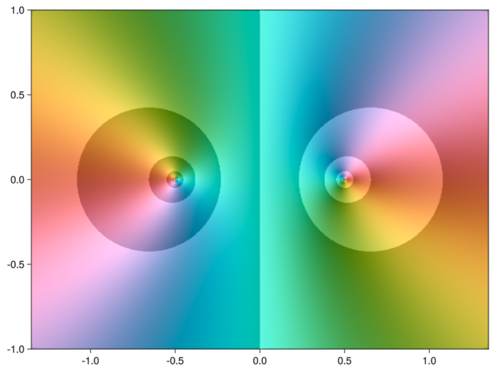
$z \mapsto z$ $\qquad$ $z \mapsto \frac{1}{\bar{z}}$ $\qquad$ $z \mapsto \frac{1}{z}$

In the above code, the second argument fixes the range of the axes. For the specific details about the `box` keyword argument we refer to the documentation (to which we will refer shortly). Also note the use of `im` as imaginary unit in Julia.

3 Domain colouring en modular surfaces

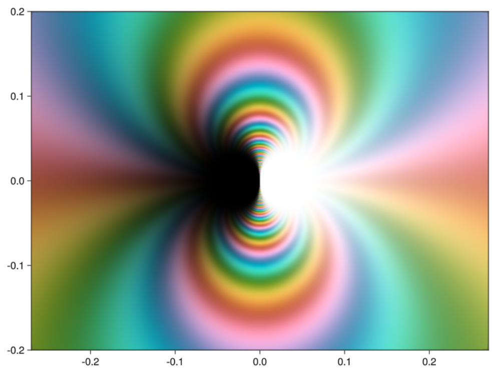
Given that $z \mapsto \frac{1}{z}$ and $z \mapsto \bar{z}$ have identical phase, this in itself is not always sufficient to study a function's behaviour. For this reason it is common to add contours of the magnitude (usually of its logarithm). The resulting figure is called a domain colouring (hence `domaincolor`). We add the contours using the option `abs=true`. This way we can distinguish the pole (left) and zero (right) in the below example.

```
domaincolor(z -> (conj(z) - .5)/(z + .5), abs=true)
```



Of course there are many variations on this idea. One could, for example, paint zero black and infinity white. This is for instance useful to illustrate the Casorati–Weierstraß theorem.

```
domaincolor(z -> exp(1/z), .2, abs=Inf)
```



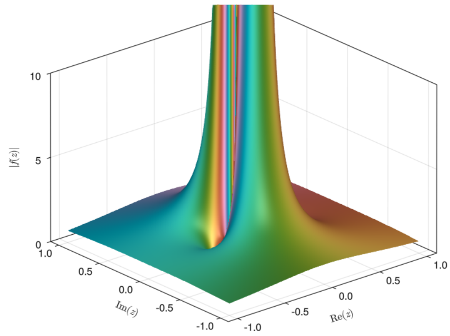
Aside from what is illustrate here, many other options are available, for instance grids. For a list of all the available options you can consult the function documentation using `?`, for example:

```
? domaincolor
```

The 2D plots of this package are from a different Julia package called `DomainColoring.jl`; visit `https://eprovst.github.io/DomainColoring.jl` for many more information and examples.

Yet another way to represent magnitude and phase at the same time is by moving the former to the third dimension, which results in a (painted) modular surface. Another essential singularity we can visualize is $z \mapsto \sin\left(\frac{1}{z}\right)$ near 0.

```
modularsurface(z -> sin(1/z))
```

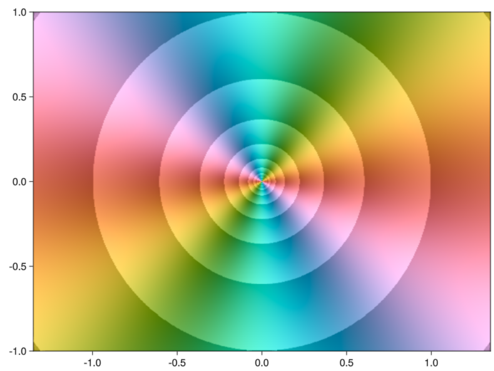


For this example—as with all other 3D plots—it is worthwhile to plot the figure yourself. This additionally allows you to rotate the plot.

4 Branch points and Riemann surfaces

When looking at the domain colouring of $z \mapsto z^2$ we see the same values appear twice in the complex plane.

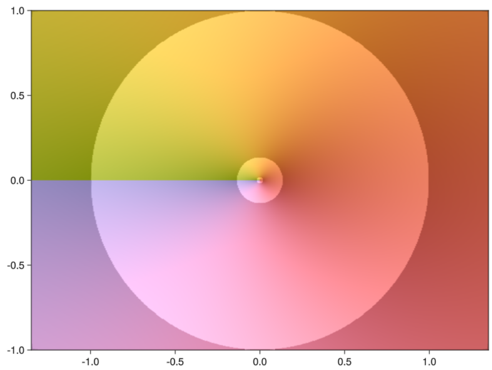
```
domaincolor(z -> z^2, abs=true)
```



The inverse map $z \mapsto z^{\frac{1}{2}}$ will hence take two values in any given point. The exception is 0, where the inverse map is unique. Such a point z_0 where a map has n values (here $n = 1$), but every neighbour has strictly more than n values, we call a branch point (more precisely: every neighbourhood of z_0 has at least one point where the map takes at least $n + 1$ values).

If we want to arrive a single valued function, we will have to pick one of the two values everywhere. The usual approach for a continuous function is to pick an arbitrary curve, starting from the branch point, and to require continuity everywhere except when crossing this curve. This results in a set of single valued functions (so called branches) which can be attached at this curve to form continuous functions. For $z \mapsto z^{\frac{1}{2}}$ the usual choice as branch is the negative real axis, as is also done in Julia. (This is a result of the underlying convention to use $|z|^{\frac{1}{2}} e^{i\frac{\operatorname{Arg} z}{2}}$, with $\operatorname{Arg} z \in (-\pi, \pi]$ the so called principal value of the argument.)

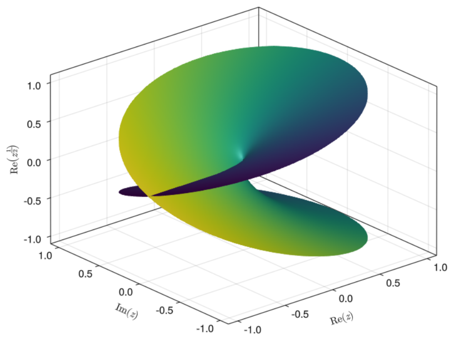
```
domaincolor(z -> z^(1/2), abs=true)
```



However, we could also try to make a plot of all the values at the same time. Note that this is a four dimensional object. For $w = f(z)$ we namely have $(\operatorname{Re} z, \operatorname{Im} z, \operatorname{Re} w, \operatorname{Im} w)$ as graph. (Alternatively, one cal of course also look at the phase and magnitude.)

When we project this orthogonally on the first three components and colour the resulting surface according to the dropped component, we get the following for $z \mapsto z^{\frac{1}{2}}$.

```
riemannpow(1//2)
```

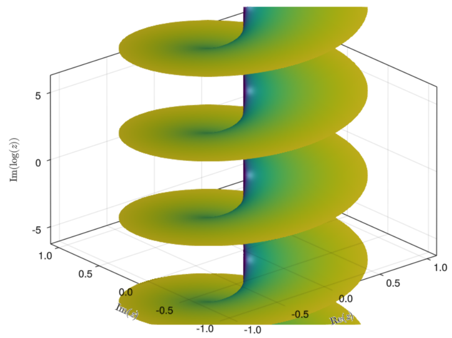


Here 1//2 is Julia notation for the exact rational number $\frac{1}{2}$. On the negative real axis this surface seems to intersect itself, this is merely an artefact of the chosen projection, the colour is in fact distinct. Note that indeed each point, except for the branch point 0, has two distinct values.

We encourage you to play around with different rational numbers and to look what does (and what does not) change the number of values. Also try an irrational power; note that it never reattaches to itself, *it takes an infinite number of values!*

A 'simpler' example of a function that takes infinitely many values is $\log z = \log |z| + i \arg z$. Using a slightly different projection, we get the following 'staircase' picture.

```
riemannlog()
```



5 Riemann sphere

Finally, `CATools.jl` is also able to plot a function on the Riemann sphere. Let us again look at the sine, with a zero at the origin (south pole) and an essential singularity at infinity (north pole).

```
riemannsphere(sin)
```