M3M6: Applied Complex Analysis (2021)

Dr. Marco Fasondini

s.mfasondini@imperial.ac.uk

Office Hours: 11am Tuesdays, Huxley 6M50

Website: https://github.com/MarcoFasondini/M3M6AppliedComplexAnalysis

References:

1. M.J. Ablowitz & A.S. Fokas, Complex Variables: Introduction and Applications, Second Edition, Cambridge University Press, 2003

0.1 Overview of course

1. Complex analysis, Cauchy's theorem, residual calculus
2. Singular integrals of the form

$$\int_\Gamma \frac{u(\zeta)}{z-\zeta} d\zeta,$$
$$\int_\Gamma u(\zeta) \log|z-\zeta| ds$$

with applications to PDEs, airfoil design, etc.

2. Weiner–Hopf method with applications to integral equations with integral operators

$$\int_0^\infty K(x-y)u(y)dy$$

4. Orthogonal polynomials, with applications to Schrödinger operators, solving differential equations.
5. Conformal mapping and Schwartz–Christoffel maps.

Central themes:

1. Finding "nice" formulae for problems that arise in applications (physics and elsewhere). These can be closed form solution, sums, integral representations, special functions, etc.
2. Computational tools for approximate solutions to problems that arise in applications.

Applications (not necessarily discussed in the course):

1. Ideal fluid flow
2. Acoustic scattering
3. Electrostatics (Faraday cage)
4. Fracture mechanics
5. Schrödinger equations
6. Shallow water waves

0.2 The Project

There is a project worth 10%. This project is *open ended*: you propose a topic. This could be computational based (possibly based on the slides), theoretical based (possibly looking at material from Ablowitz & Fokas), or otherwise. If you are having difficulty coming up with a proposal, please attend the office hours for advice.

Timeline:

- 11 Feb: Office hours to discuss potential projects
- 20 Feb: Turn in short description of proposed project (max 2 paragraphs)
- 20 March: Project due

0.3 Problem sheets

There will be 4 problem sheets during the term.

0.4 Feedback on course work

I'll set aside four office hours to walk through individual solutions to the problem sheets on a 1-on-1 basis. Please email me ahead of time so I can schedule.

1 Lecture 1: Complex analysis review

The first couple lectures will review the basics of complex analysis. We will use plots to help explain the material, and so this first lecture will focus on getting comfortable with plots of analytic functions.

1.1 Plotting functions in the complex plane

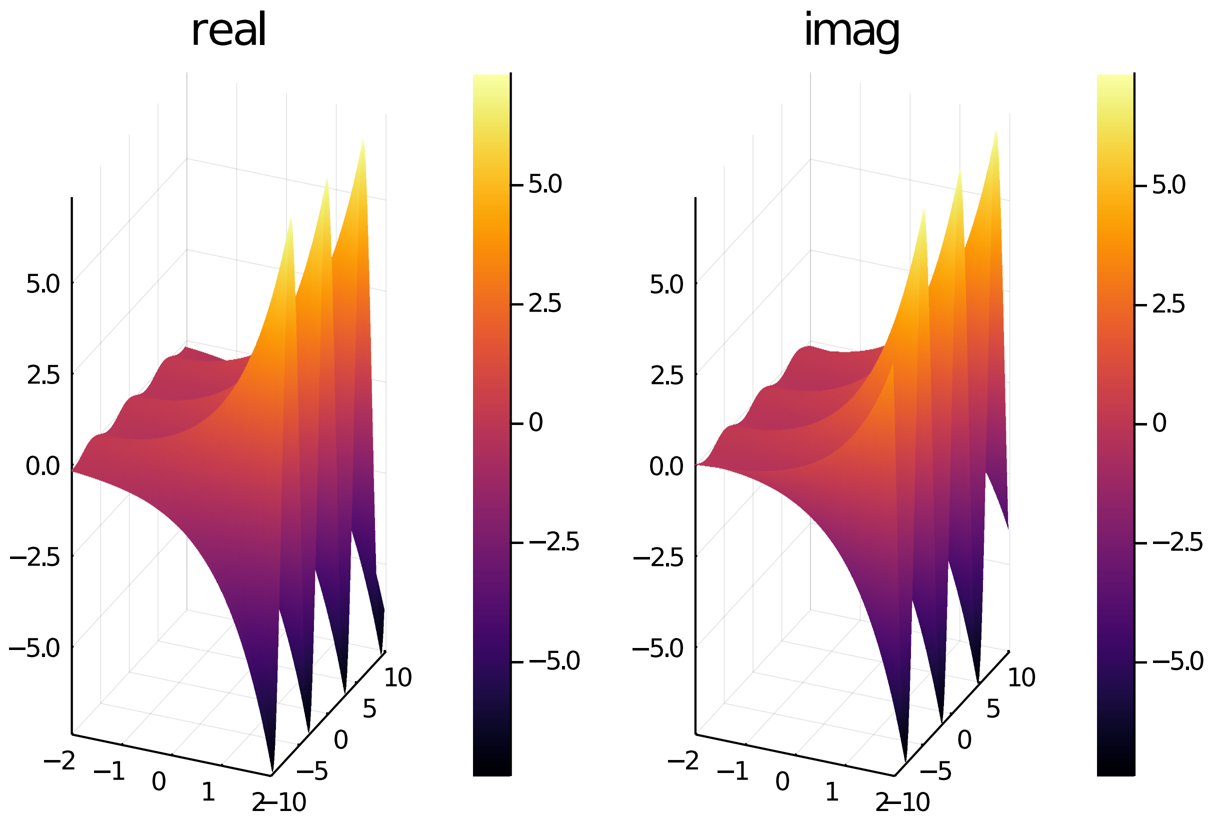
Consider a complex-valued function $f : D \to \mathbb{C}$ where $D \subset \mathbb{C}$. To help understand such functions it is useful to plot them. But how?

Method 1: real and imaginary parts

Every complex-valued function can be written as $f(z) = u(z) + iv(z)$ where $u : D \to \mathbb{R}$ and $v : D \to \mathbb{R}$ are real-valued. These are easy to plot separately. We can do a surface plot:

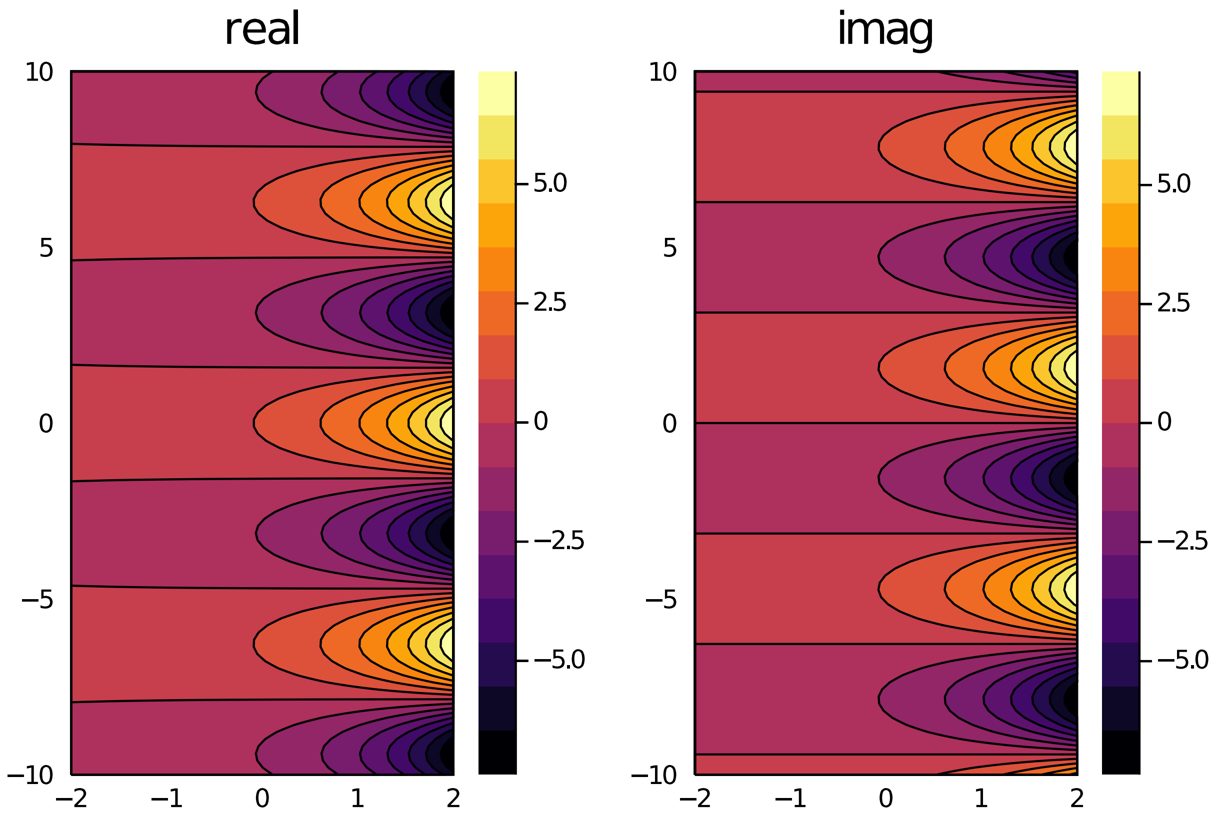
```
using Plots, ComplexPhasePortrait, SpecialFunctions

f = z -> exp(z)
u = z -> real(f(z))
v = z -> imag(f(z))

# set up plotting grid
xx = range(-2 ; stop=2,  length=100)
yy = range(-10; stop=10, length=100)
```

```
plot(surface(xx, yy, u.(xx' .+ im.*yy); title="real"),
     surface(xx, yy, v.(xx' .+ im.*yy); title="imag"))
```



Or a heat plot:

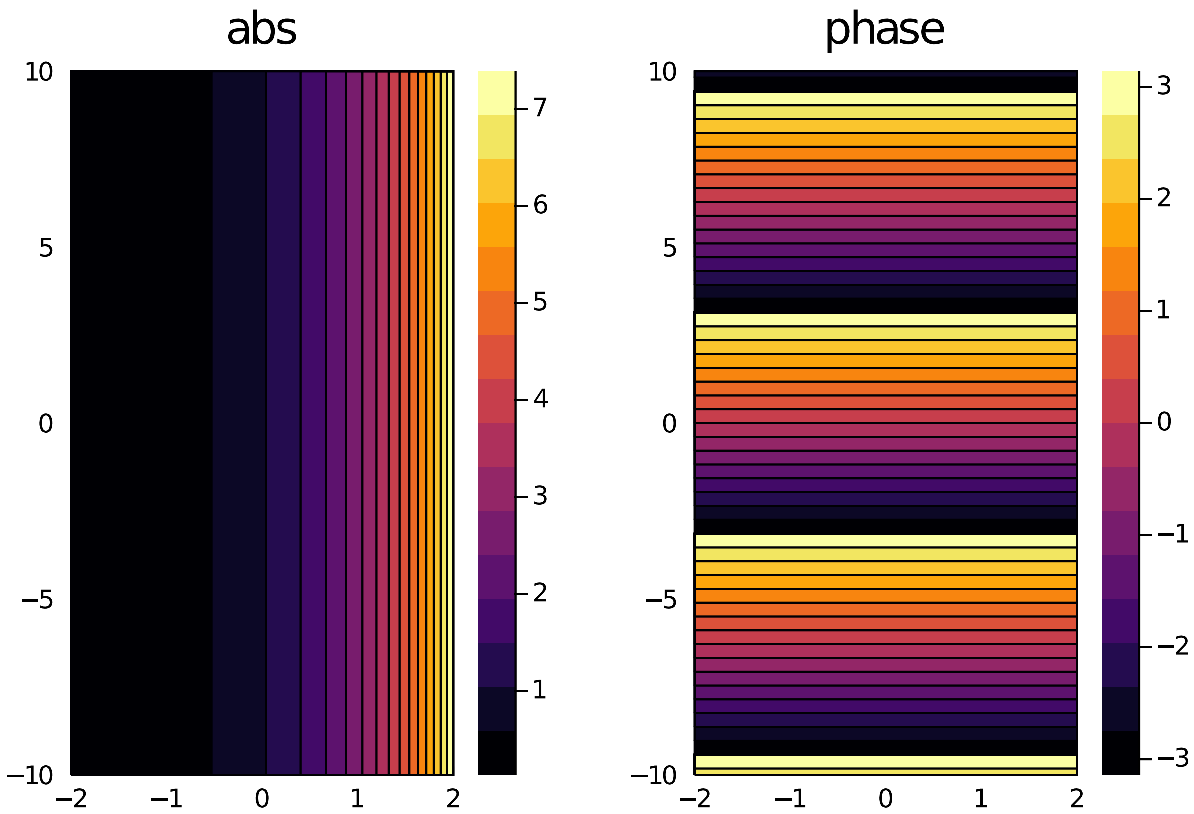
```
plot(contourf(xx, yy, u.(xx' .+ im.*yy); title="real"),
     contourf(xx, yy, v.(xx' .+ im.*yy); title="imag"))
```



Method 2: absolute-value and angle

Every complex number z can be written as $re^{i\theta}$ for $0 \leq r$ and $-\pi < \theta \leq \pi$. $r = |z|$ is called the *absolute value* and θ is called the *phase* or *argument* (or in Julia, *angle*). We can plot these:

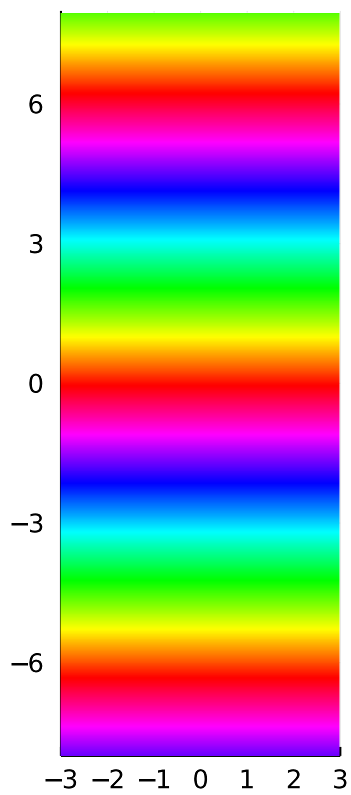
```
xx = -2:0.01:2
yy = -10:0.01:10

f = z -> exp(z)

r = z -> abs(f(z))
θ = z -> angle(f(z))


plot( contourf(xx, yy, r.(xx' .+ im.*yy); title="abs"),
      contourf(xx, yy, θ.(xx' .+ im.*yy); title="phase"))
```



__ Method 3: Phase portrait __

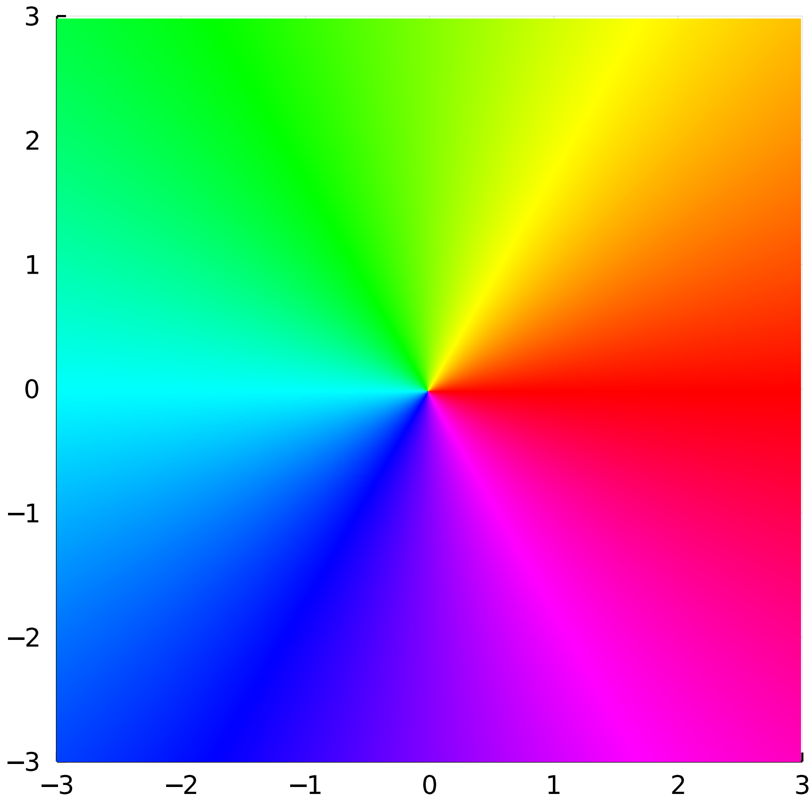
This method is essentially the same, as before, but to *only* plot the phase, and use a "colour wheel" to reflect the topology of θ.

```
phaseplot(-3..3, -8..8, z -> exp(z))
```



This is seen most clearly with $f(z) = z$: in this case, letting $z = r\mathrm{e}^{\mathrm{i}\theta}$ we are simply plotting θ:

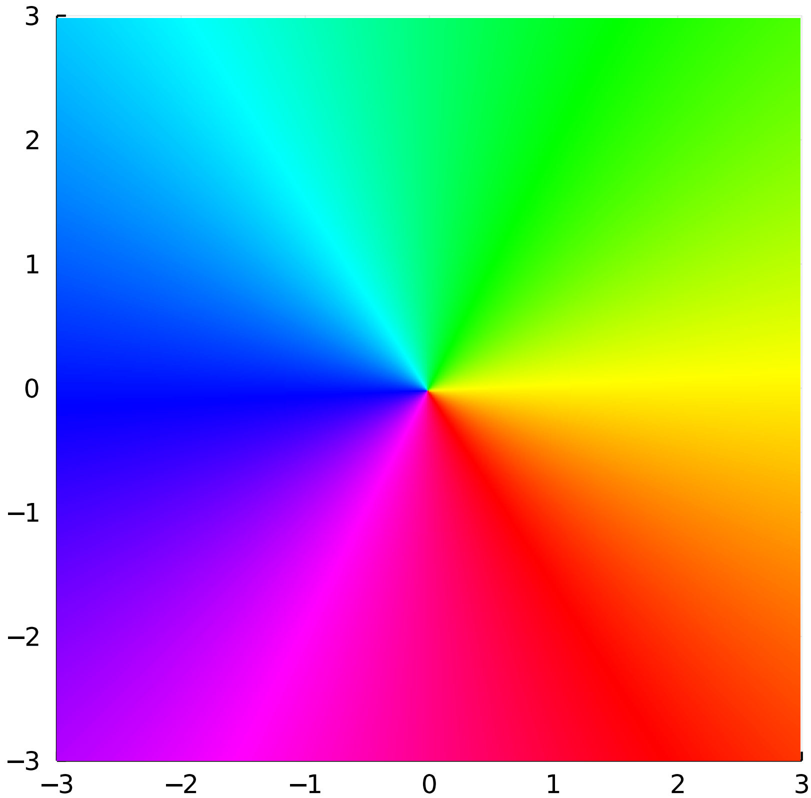
```
phaseplot(-3..3, -3..3, z -> z)
```



In other words, the colour red corresponds to $\arg f(z) \approx 0$, green to $\arg f(z) \approx \frac{2\pi}{3}$ aqua to $\arg f(z) \approx \pi$ and so on.

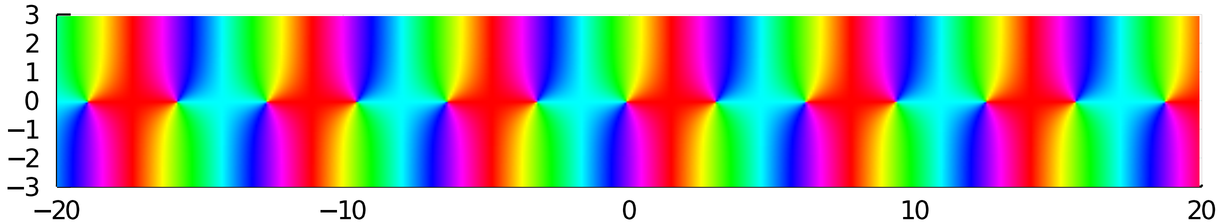
Note that multiplying z by a complex number $R\mathrm{e}^{\mathrm{i}\varphi}$ will rotate the wheel, but the colours still appear in the same order when read counter clockwise:

```
phaseplot(-3..3, -3..3, z -> exp(1.0im)*z)
```



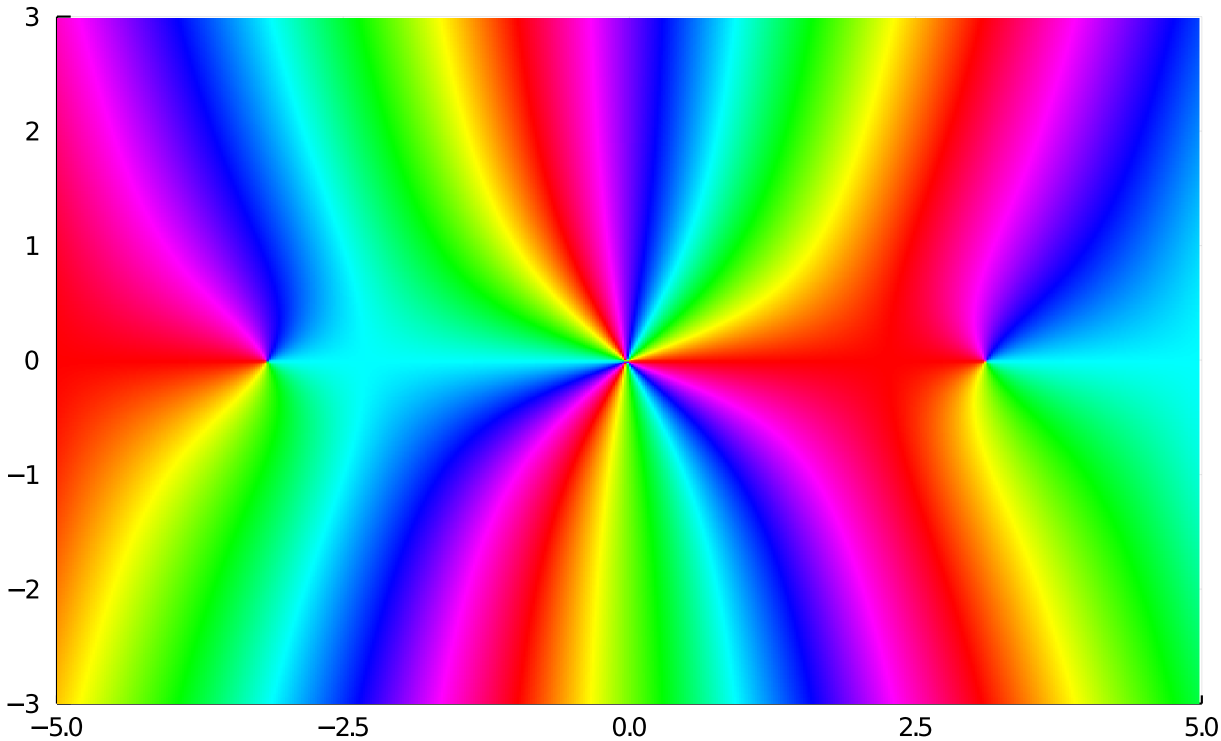
Therefore, if $f(z)$ has a zero at z_0, since it behaves like $f(z) = f'(z_0)(z - z_0)$, we will have the full colour wheel always in the counter clockwise order red–green–blue–red:

```
phaseplot(-20..20, -3..3, z -> sin(z))
```



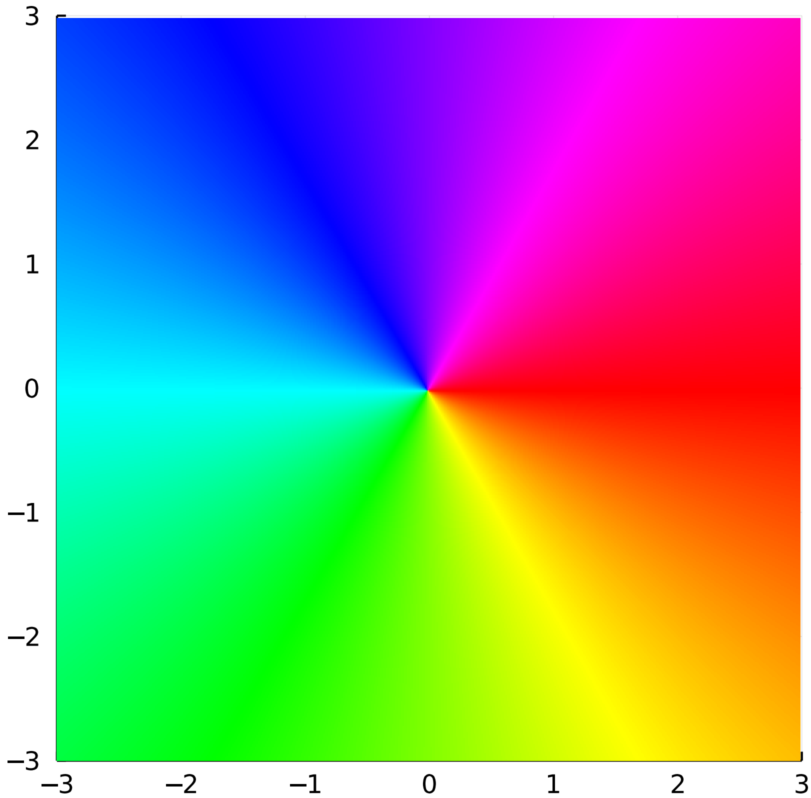
In the case of a double root like $f(z) = z^2 = r\mathrm{e}^{2\mathrm{i}\theta}$, we have the wheel appearing twice, but still in the same order. Thus the order of a zero can be seen by the number of times we go around the colour wheel: here we see that the function has a triple root at zero since it goes red–green–blue–red–green–blue–red–green–blue–red:

```
phaseplot(-5..5, -3..3, z -> z^2*sin(z))
```



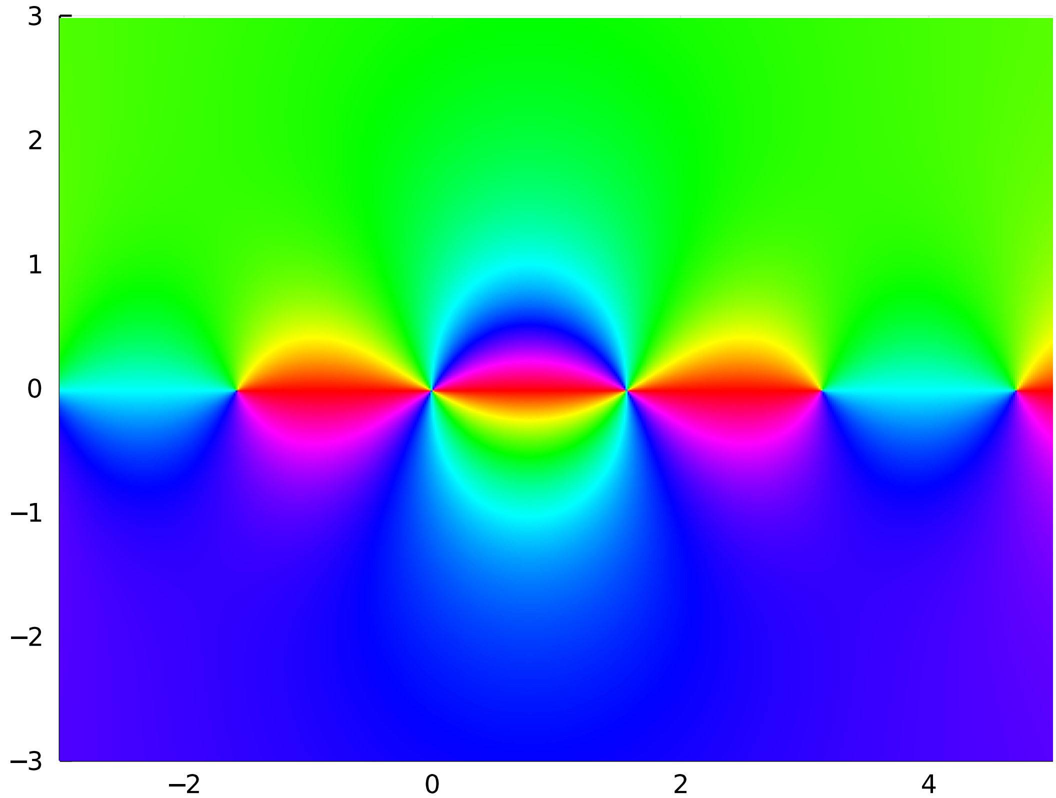
On the other hand, if we plot $f(z) = z^{-1} = r^{-1}e^{-i\theta}$ the wheel is reversed to be red–blue–green–red:

```
phaseplot(-3..3, -3..3, z -> 1/z)
```



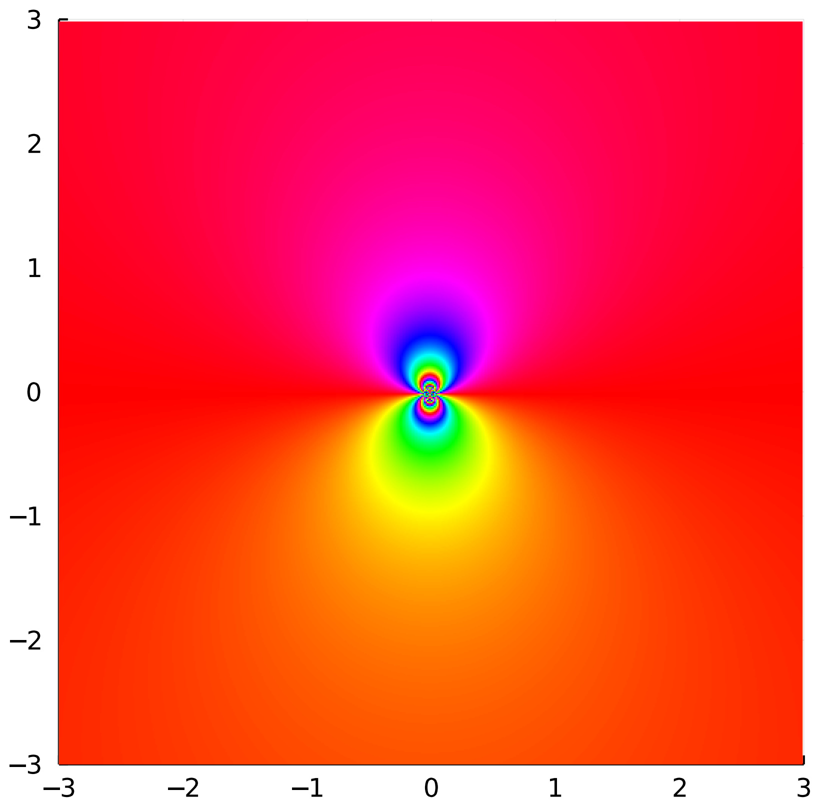
Thus we can see from a phase plot where the poles and zeros are, and the order of the poles and roots: the following function has a double pole at 0 (red–blue–green–red–blue–green), a zero at $-\frac{\pi}{2}$ (red–green–blue–red) and a double zero at $\frac{\pi}{2}$ (red–green–blue–red–green–blue–red) :

```
phaseplot( -3..5, -3..3, z -> cot(z)/z*(π/2-z))
```



Functions with more complicated singularities do not have such nice phase plots (these are what we will call *essential* singularities):

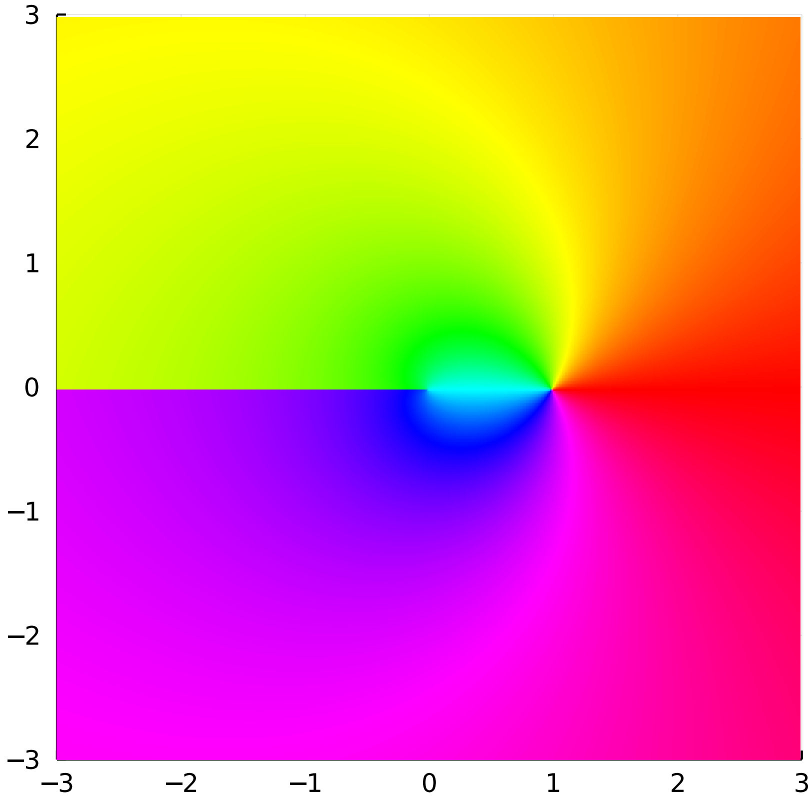
```
phaseplot(-3..3, -3..3, z -> exp(1/z))
```



The jumps of functions like $\log z$ or $\sqrt{z}$ also appear naturally in the phase plot: here we see

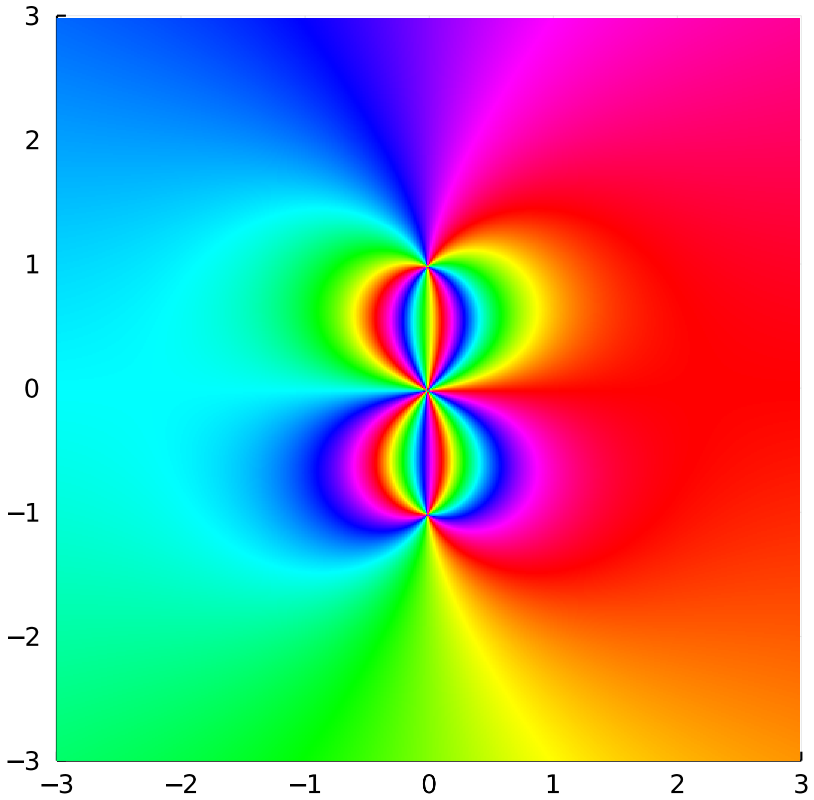
$\log z$ has a zero at 1 (red–green–blue–red) and a jump along $(-\infty, 0]$

```
phaseplot( -3..3, -3..3, z -> log(z))
```

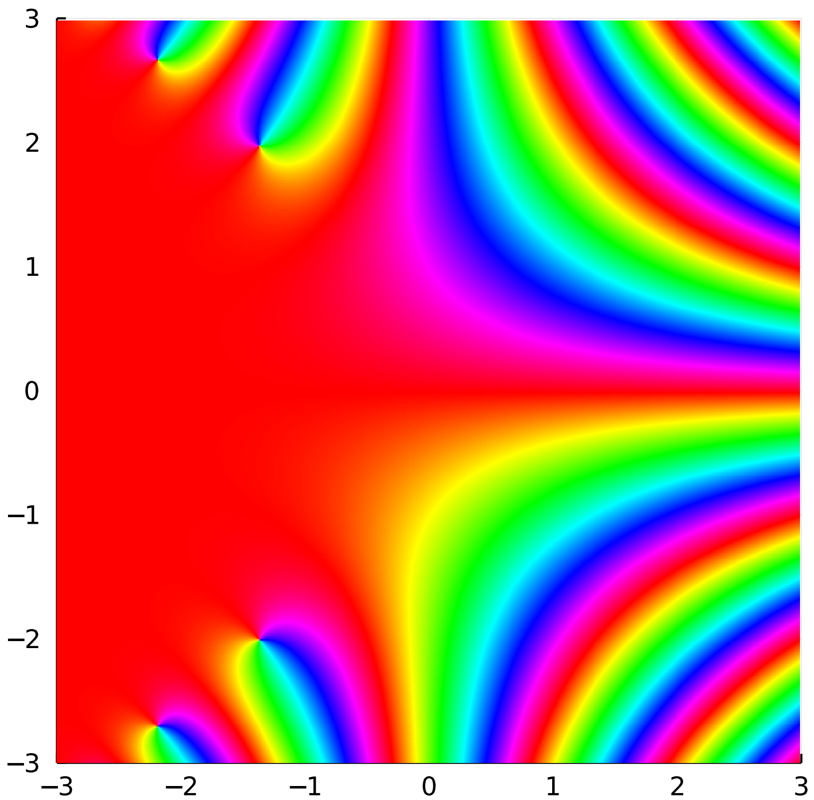


Often, singularities are in the complex plane. can you determine where the zeros and roots, and their orders, of the following function are from the picture?

```
phaseplot(-3..3, -3..3, z -> z^5  / (1 + z^2)^3)
```

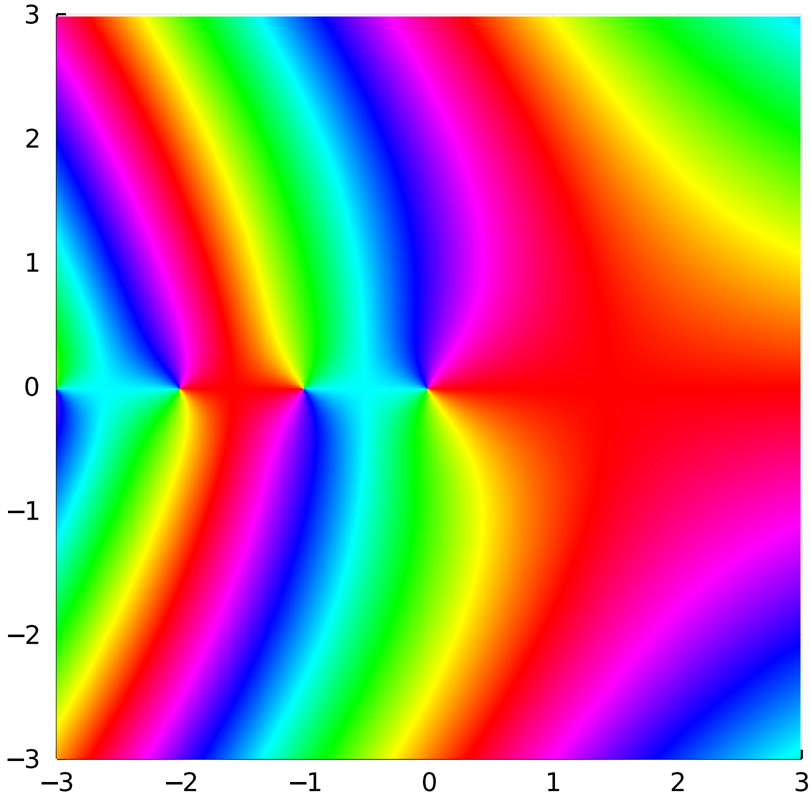



```
phaseplot(-3..3, -3..3, z -> erfc(z))
```



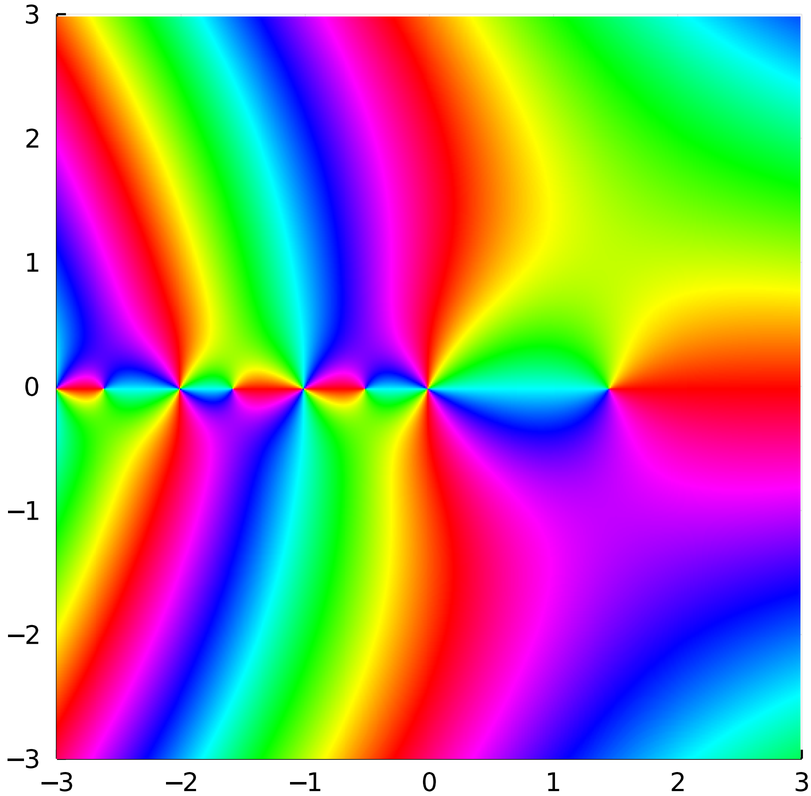
Here's the Gamma function. Are they poles or zeros at 0, -1, -2, ...?

```
phaseplot(-3..3, -3..3, z -> gamma(z))
```



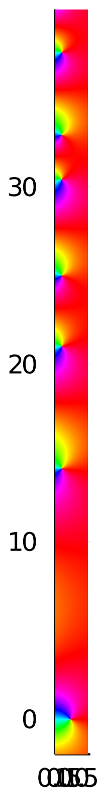
Something strange is happening at -1/2, -3/2, ... What if we differentiate? Here `dual(z,1)` is a dual number, which provides a convenient way to implement automatic differentiation: `gammap(z)` is derivative of `gamma(z)` w.r.t. `z`.

```
using DualNumbers
gammap = z -> epsilon(gamma(dual(z,1)))
phaseplot(-3..3, -3..3, gammap)
```



Finally, the Riemann Hypothesis: Here is a plot of the Riemann zeta function near the critical line $z = 0.5 + \mathrm{i}t$:

```
phaseplot(0:0.011:2, -2:0.011:40, zeta)
```



1.2 Holomorphicity, Cauchy's theorem, and Analyticity

A beautiful feature of complex analysis is one starts with a very weak assumption of holomorphicity, essentially complex-differentiability in an open set, deduce Cauchy's theorem and Cauchy's integral formula, which then imply analyticity. That is, if we assume a function is one-time complex differentiable, then it has an infinite number of derivatives.

These results are built up as follows:

1. Holomorphic functions
2. Cauchy's theorem
3. Deformation of contours
4. Cauchy's integral formula
5. Analyticity and Taylor series

We will review the first three topics this lecture, with the remaining for next lecture.

1.2.1 Holomorphic functions

Definition (Complex-differentiable) Let $D \subset \mathbb{C}$ be an open set. A function $f : D \to \mathbb{C}$ is called *complex-differentiable* at a point $z_0 \in D$ if

$$f'(z_0) = \lim_{z \to z_0} \frac{f(z) - f(z_0)}{z - z_0}$$

exists, for any angle of approach to z_0.

Definition (Holomorphic) Let $D \subset \mathbb{C}$ be an open set. A function $f : D \to \mathbb{C}$ is called *holomorphic* in D if it is complex-differentiable at all $z \in D$.

Definition (Entire) A function is *entire* if it is holomorphic in $\mathbb{C}$

Examples

1. $$1$$
   is entire
2. $$z$$
   is entire
3. $$1/z$$
   is holomorphic in $\mathbb{C}\backslash\{0\}$
4. $$\sin z$$
   is entire

5. $$\csc z$$
   is holomorphic in $\mathbb{C}\backslash\{\ldots, -2\pi, -\pi, 0, \pi, 2\pi, \ldots\}$
6. $$\sqrt{z}$$
   is holomorphic in $\mathbb{C}\backslash(-\infty, 0]$

1.2.2 Cauchy's theorem

Definition (Contour) A *contour* is a continuous & piecewise-continuously differentiable function $\gamma : [a, b] \to \mathbb{C}$.

Definition (Simple) A *simple contour* is a contour that is 1-to-1.

Definition (Closed) A *closed contour* is a contour such that $\gamma(a) = \gamma(b)$

Examples of contours

1. Line segment $[a, b]$ is a simple contour, with $\gamma(t) = t$
2. Arc from re^{ia} to re^{ib} is a simple contour, with $\gamma(t) = re^{it}$
3. Circle of radius r is a closed simple contour, with $\gamma(t) = re^{it}$ and $a = -\pi$, $b = \pi$
4. $$\gamma(t) = \cos(t + i)^2$$
   defines a contour that is not simple or closed
5. $$\gamma(t) = e^{it} + e^{2it}$$
   for $[a, b] = [-\pi, \pi]$ defines a contour that is closed but not simple

Definition (Contour integral) The *contour integral* over γ is defined by

$$\int_\gamma f(z)dz := \int_a^b f(\gamma(t))\gamma'(t)dt$$

An important property of a contour is its *arclength*:

Definition (Arclength) The *arclength* of γ is defined as

$$\mathcal{L}(\gamma) := \int_a^b |\gamma'(t)|dt$$

A very useful result is that we can use the maximum and arclength to bound integrals:

Proposition (ML) Let $f : \gamma \to \mathbb{C}$ and

$$M = \sup_{z\in\gamma} |f(z)|$$

Then

$$\left|\int_\gamma f(z)dz\right| \leq M\mathcal{L}(\gamma)$$

Proposition If $f(z)$ is holomorphic on γ, then $\int_\gamma f'(z)dz = f(\gamma(b)) - f(\gamma(a))$

Theorem (Cauchy) If f is holomorphic inside and on a closed contour γ, then $\oint_\gamma f(z)dz = 0$

1.2.3 Deformation of contours

Definition (Domain) A *domain* is a non-empty, open and connected set $D \subset \mathbb{C}$.

Definition (Homotopic) Two closed contours $\gamma_1 : [a, b] \to D$ and $\gamma_2 \to D$ in a domain D are *homotopic* if they can be continuously deformed to one-another while remaining in D.

Theorem (Deformation of contours) Let $f(z)$ be holomorphic in a domain D. Let γ_1 and γ_2 be two homotopic contours. Then

$$\int_{\gamma_1} f(z)dz = \int_{\gamma_2} f(z)dz$$

Definition (Simply connected) A domain is *simply connected* if every closed contour is homotoic to a point.

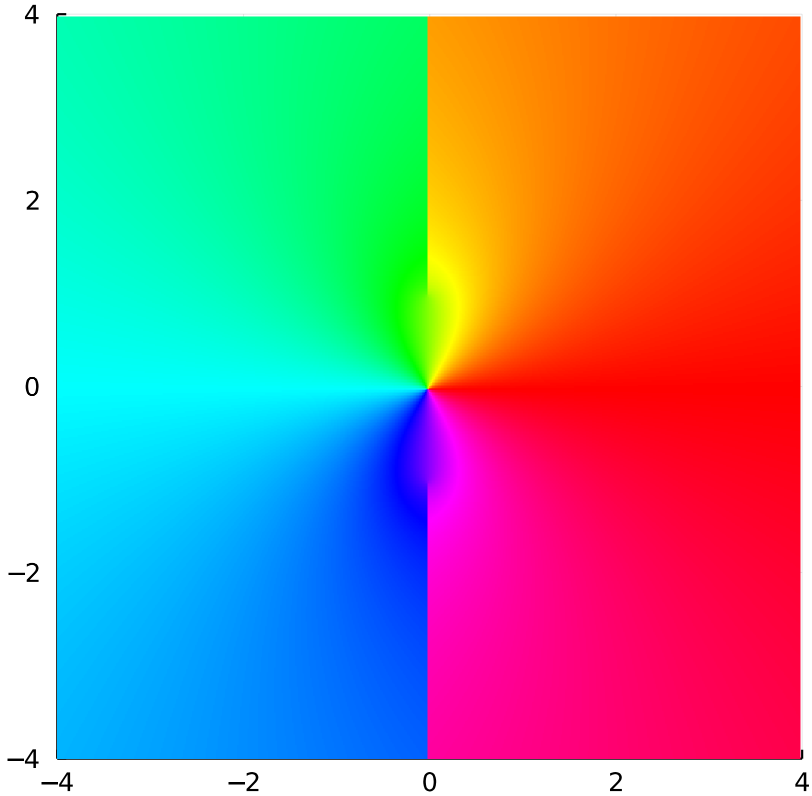
Corollary (Deformation of contours on simply connected domains) Let $f(z)$ be holomorphic in a simply-connected domain D. If γ_1 and γ_2 are two contours in D with the same endpoints, then

$$\int_{\gamma_1} f(z)dz = \int_{\gamma_2} f(z)dz$$

1.2.4 Demonstration

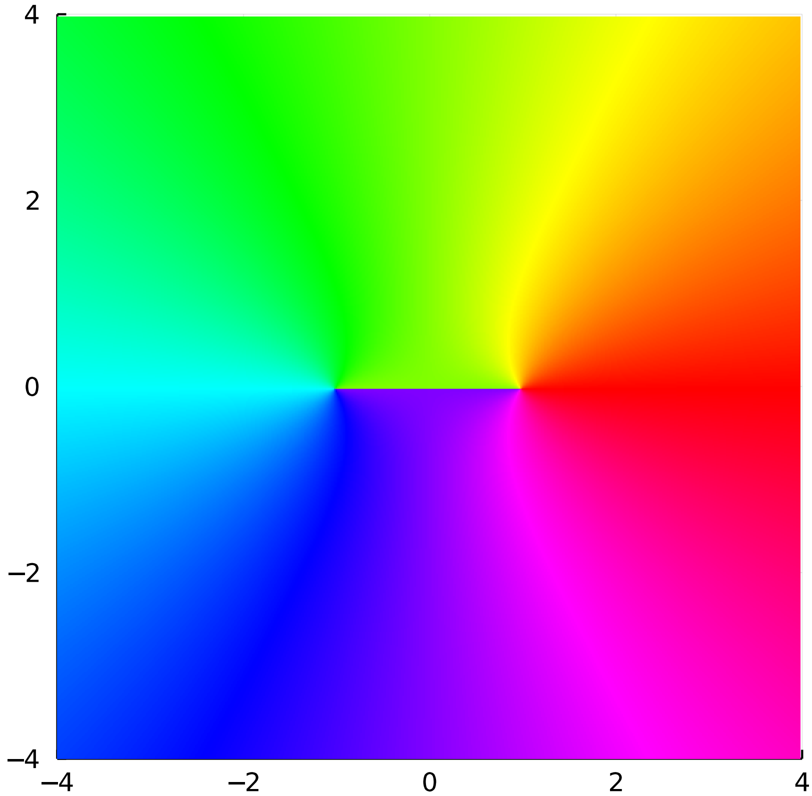
We can usually infer the domain where a function is holomorphic from a phase portrait, here we see that $\operatorname{arcsinh} z$ has cuts on $[\mathrm{i}, \mathrm{i}\infty)$ and $[-\mathrm{i}, -\mathrm{i}\infty)$, and a zero (red–green–blue–red) at zero, hence we can infer that it is holomorphic in $\mathbb{C}\backslash([\mathrm{i}, \mathrm{i}\infty) \cup [-\mathrm{i}, -\mathrm{i}\infty))$.

```julia
using Plots, ComplexPhasePortrait, SpecialFunctions, ApproxFun
phaseplot(-4..4, -4..4, z -> asinh(z))
```



The following example $\sqrt{z-1}\sqrt{z+1}$ is analytic in $\mathbb{C}\backslash[-1,1]$ and will be returned to:

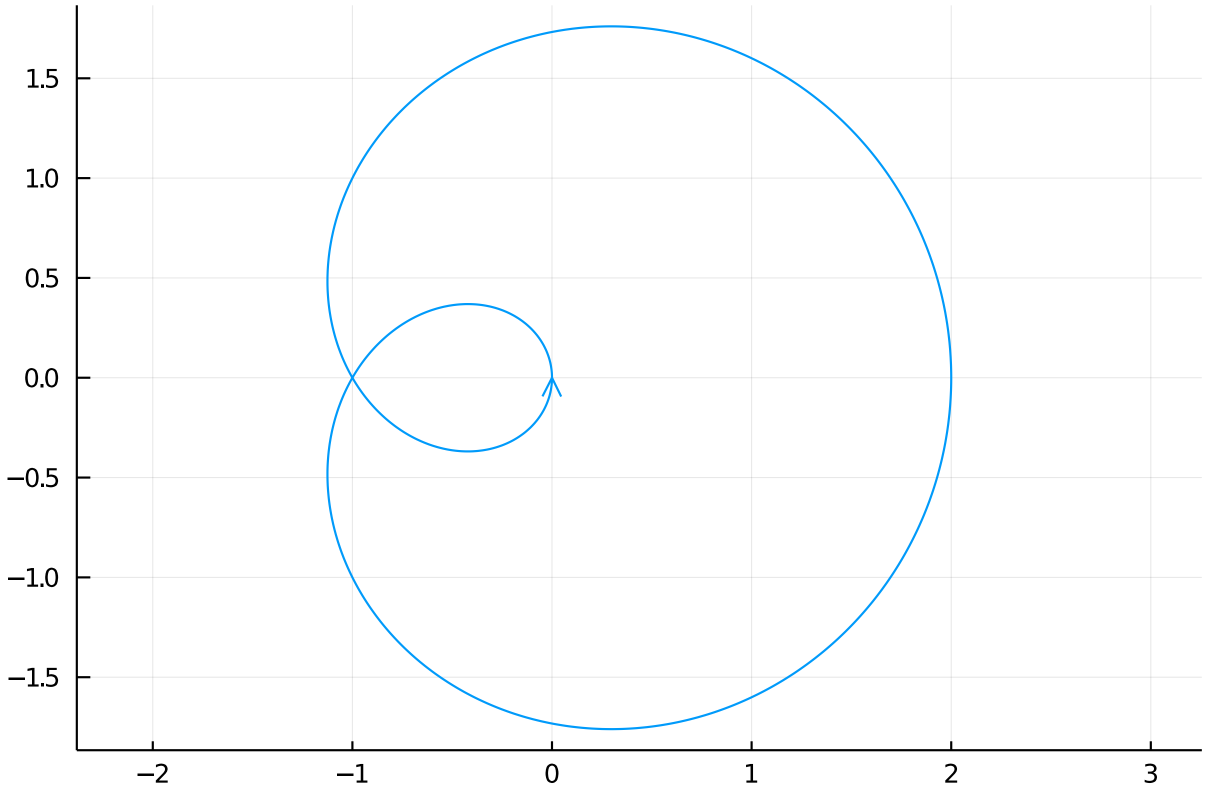
```
phaseplot(-4..4, -4..4, z -> sqrt(z-1)sqrt(z+1))
```



Here's an example of a closed contour that is not simple:

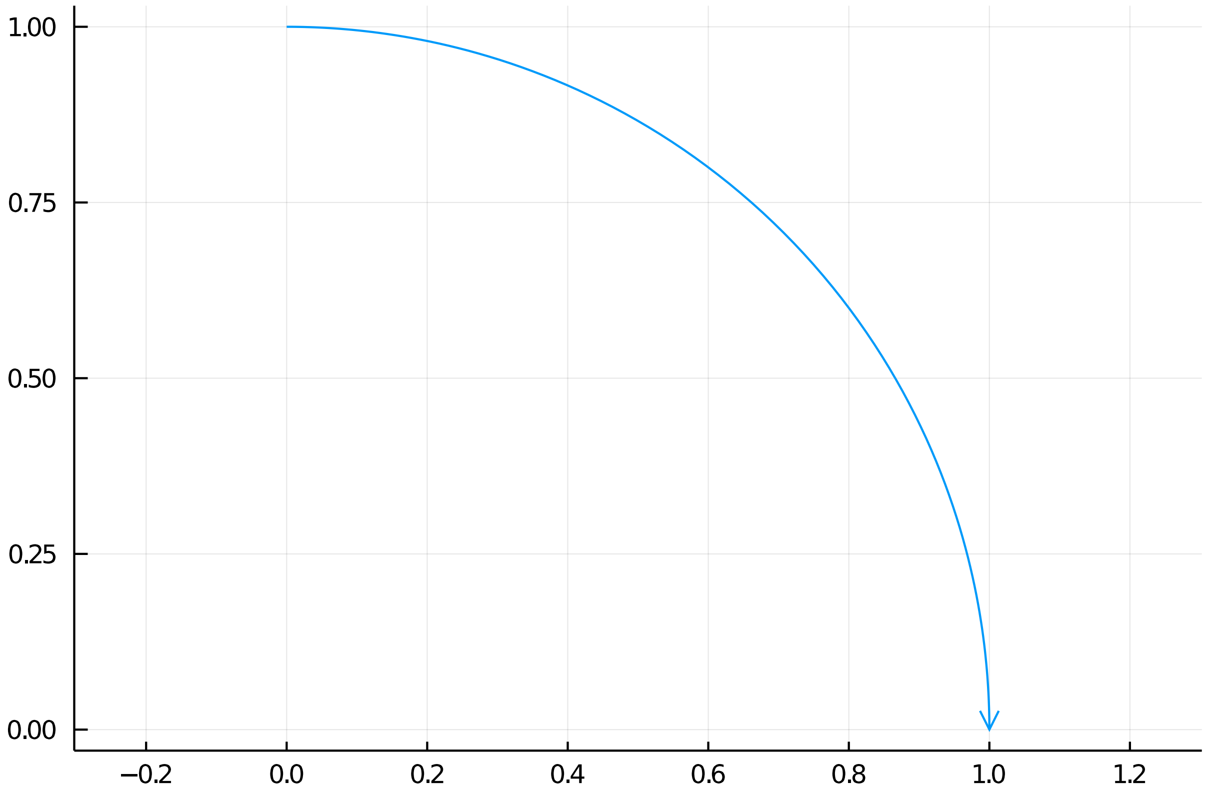
```
a,b = -π, π
```

```
tt = range(a, stop=b, length=1000)

γ = t -> exp(im*t) +exp(2im*t)

plot(real.(γ.(tt)), imag.(γ.(tt)); ratio=1.0, legend=false , arrow=true)
```



The notion of integration over a contour in the complex plane is independent of the notion of holomorphicity. For example, we can happily define the integral for non-holomorphic functions such as $\Re e^z = e^x \cos y$:

```
f = Fun( z -> real(exp(z)), Arc(0.,1.,(0,π/2)))  # Not holomorphic!

plot(domain(f); legend=false , ratio=1.0, arrow=true)
```




```
sum(f)  # this means contour integral
```

```
-1.2485382363935422 + 1.949326343919058im
```

This is defined in terms of a parameterization, and so the following real integral gives the same result:

```
g = im*Fun(t-> f(exp(im*t))*exp(im*t), 0 .. π/2)
sum( g )  # this is standard integral
```

```
-1.248538236393543 + 1.9493263439190578im
```

The following show that the contour of integration does not affect the integral for holomorphic functions:

```
f = Fun( z -> exp(z), Arc(0.,1.,(0,π/2))) #integrate over an arc
sum(f)  , f(im)-f(1)
```

```
(-2.1779795225909053 + 0.8414709848078971im, -2.177979522590906 + 0.8414709
848078968im)
```

```
f = Fun( z -> exp(z), Segment(1,im)) # integrate over a line segment
sum(f)  , f(im)-f(1)
```

```
(-2.1779795225909053 + 0.8414709848078966im, -2.177979522590907 + 0.8414709
848078968im)
```

The following demonstrates Cauchy's theorem, integration over a closed contour equals zero:

```
f = Fun( z -> exp(z), Circle())  #  Holomorphic!
sum(f)
```

```
6.751217953510028e-17 - 1.6526433588101707e-16im
```

If `f` is not holomorphic, it doesn't apply:

```
f = Fun( z -> exp(z)/z, Circle())  # Not holomorphic at zero
sum(f)
```

```
-7.899668397939031e-16 + 6.283185307179588im
```

```
f = Fun( z -> real(exp(z)), Circle())  # Not holomorphic anywhere!
sum(f)
```

```
-3.3420237696193494e-16 + 3.141592653589793im
```

We can test contour deformation experimentally: in the following, the integral (implemented as `sum`) over two different contours returns the same value (up to numerical precision)

```
f = Fun( z -> exp(z), Arc(0.,1.,(0,π/2)))  # Holomorphic!
fs = Fun( z -> exp(z), Segment(1,im))  # Holomorphic!

sum(f)  , sum(fs)
```

```
(-2.1779795225909053 + 0.8414709848078971im, -2.1779795225909053 + 0.841470
9848078966im)
```

Here is a depiction of the two contours:

```
plot(domain(f); ratio=1.0, legend=false , arrow=true)
plot!(domain(fs), arrow=true)
```