Using *Mathematica* to study complex numbers (week 3)

■ Basics

Mathematica is set up to deal with complex numbers, although there are some tricks one has to learn. The simplest way to enter *i* (square root of -1) is as I (upper case I).

```
z = 2 + 3 I
```

```
2 + 3 ⅈ
```

Note that *Mathematica* writes I in lowercase in the output. Here's another example:

```
Sqrt[-4]
```

```
2 ⅈ
```

Real & Imaginary parts, Magnitude (=absolute value) & Argument, and Complex Conjugate are obtained as follows:

```
Re[z]
```

```
2
```

```
Im[z]
```

```
3
```

```
Abs[z]
```

$\sqrt{13}$

```
Arg[z]
```

$\mathrm{ArcTan}\left[\frac{3}{2}\right]$

```
N[Arg[z]]
```

```
0.982794
```

```
zbar = Conjugate[z]
```

```
2 - 3 ⅈ
```

Complex numbers can be added, subtracted, multiplied and divided as for reals:

```
z + zbar
```

```
4
```

(which is correct as the sum should be twice the real part of z)

```
z - zbar
```

```
6 ⅈ
```

(again correct as the sum gives twice the imaginary part of z times I)

```
z zbar
```

```
13
```

(which is correct since it is the square of the magnitude of z)

```
z / zbar
```

$-\frac{5}{13} + \frac{12\,i}{13}$

```
{Abs[z / zbar], N[Arg[z / zbar]]}
```

{1, 1.96559}

(which is correct since, as seen in lectures, z/zbar has magnitude 1 and argument twice that of z)

Note that *Mathematica*'s convention for the argument is $-\pi < \text{Arg}[z] \leq \pi$

```
{Arg[1], Arg[I], Arg[-1], Arg[-I]}
```

$\left\{0, \frac{\pi}{2}, \pi, -\frac{\pi}{2}\right\}$

Also, by convention, Arg[0]=0

```
Arg[0]
```

0

■ Simple examples of manipulating complex numbers

Example discussed previously in class: Result is displayed automatically in "x + i y" form.

```
z2 = (3 + I) / (2 + I)
```

$\frac{7}{5} - \frac{i}{5}$

To get result in polar form:

```
{r = Abs[z2], theta = Arg[z2]}
```

$\left\{\sqrt{2}, -\text{ArcTan}\left[\frac{1}{7}\right]\right\}$

One can also enter the complex number in polar form---all *Mathematica* functions take complex arguments.

```
z2polar = r Exp[I theta]
```

$\sqrt{2}\, e^{-i\, \text{ArcTan}\left[\frac{1}{7}\right]}$

To get back in Cartesian form use the useful function "ComplexExpand"

```
ComplexExpand[z2polar]
```

$\frac{7}{5} - \frac{i}{5}$

Another example from previous lectures:

```
z3 = (5 - 2 I) / (5 + 2 I)
```

$\frac{21}{29} - \frac{20\, i}{29}$

```
{Abs[z3], Arg[z3]}
```

$\left\{1, -\text{ArcTan}\left[\frac{20}{21}\right]\right\}$

A final example

```
Abs[(2 + 3 I) / (1 - I)]
```

$\sqrt{\frac{13}{2}}$

Features of ComplexExpand

Mathematica usually assumes that numbers are complex. Thus in

```
z4 = (x + I y) ^2
```

$(x + i\,y)^2$

it assumes that x and y are complex:

```
{Re[z4], Im[z4]}
```

$\{\mathrm{Re}[(x + i\,y)^2], \mathrm{Im}[(x + i\,y)^2]\}$

One nice feature of ComplexExpand is that it assumes that all variables are real (unless you tell it otherwise).

```
ComplexExpand[(x + I y) ^2]
```

$x^2 + 2\,i\,x\,y - y^2$

Roots

Mathematica does not automatically give all complex roots, e.g.

```
(1) ^{1 / 3}
```

{1}

To get all the roots we can use Solve.
(Note that we have to "Clear" z, since it was defined above.)

```
Clear[z]; Solve[z^3 == 1, z]
```

$\{\{z \to 1\}, \{z \to -(-1)^{1/3}\}, \{z \to (-1)^{2/3}\}\}$

To get the result in “x+iy” form use ComplexExpand:

```
root = ComplexExpand[Solve[z^3 == 1, z]]
```

$\left\{\{z \to 1\}, \left\{z \to -\frac{1}{2} - \frac{i\sqrt{3}}{2}\right\}, \left\{z \to -\frac{1}{2} + \frac{i\sqrt{3}}{2}\right\}\right\}$

Note that *Mathematica* gives the results as a list of assignments (which I have labeled "root"). We can use this list with the following construction involving "/."
(Read this as "Evaluate z with the assignment rules in root, one at a time".)

```
z /. root
```

$\left\{1, -\frac{1}{2} - \frac{i\sqrt{3}}{2}, -\frac{1}{2} + \frac{i\sqrt{3}}{2}\right\}$

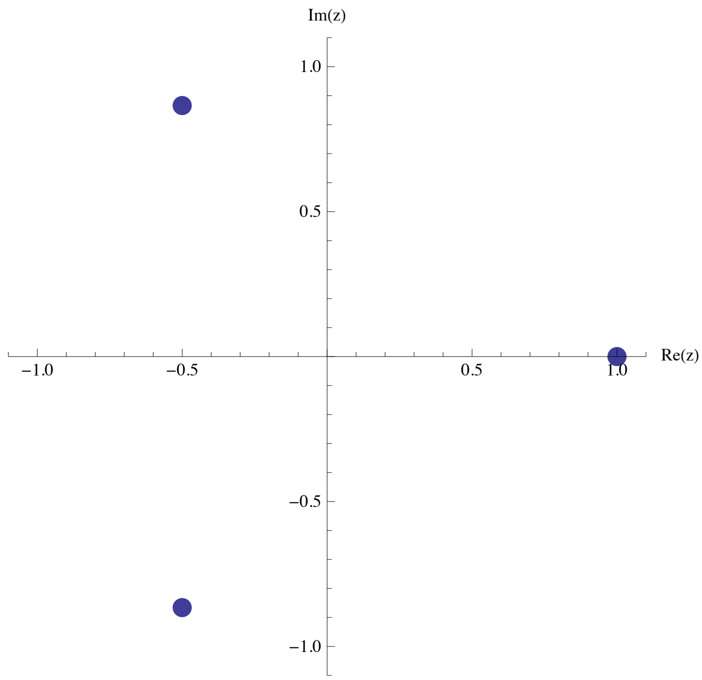
In this way we can check that all 3 roots are really roots:

```
ComplexExpand[z^3 /. root]
```

{1, 1, 1}

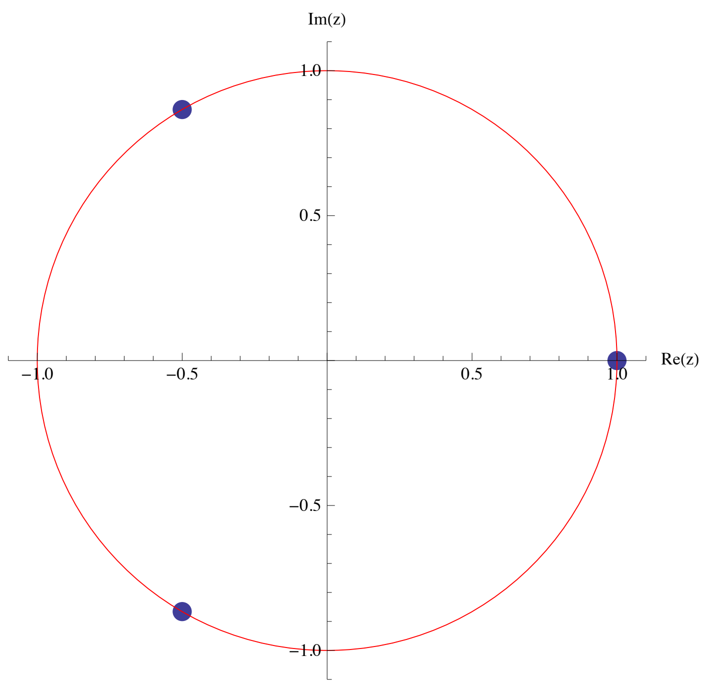
Here is one way to plot the roots

```
rootplot = ListPlot[{Re[z], Im[z]} /. root, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}},
  AxesLabel → {"Re(z)", "Im(z)"}, AspectRatio → 1, PlotStyle -> PointSize[0.03]]
```



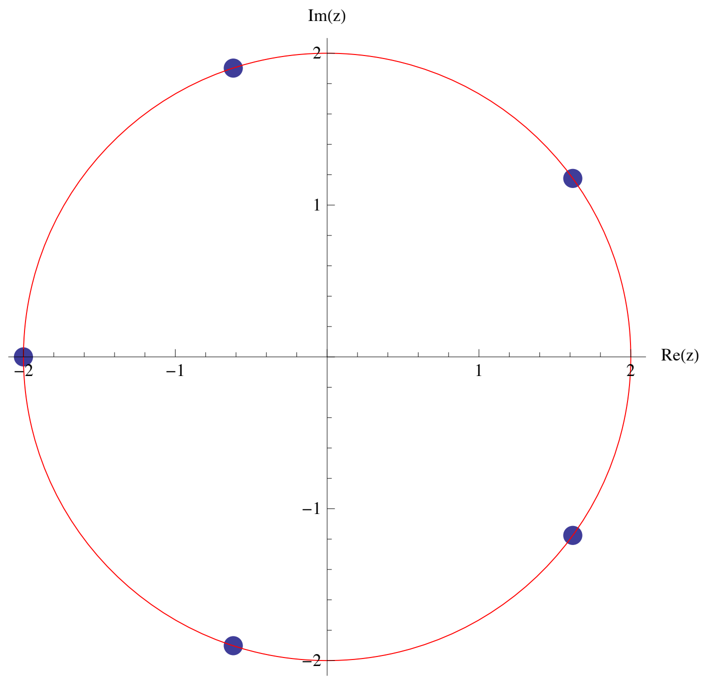
Showing that the roots lie on the "unit circle"

```
Show[rootplot, Graphics[{Red, Circle[{0, 0}, 1]}]]
```



Here is a plot of the fifth roots of -32 (which includes -2). Note the use of "Module" to package all the commands into one unit. The initial parenthesis "{root,rootplot}" lists the local names that are used---these do not get defined outside of the Module, and thus do not overwrite other values.

```
Module[{root, rootplot}, root = Solve[z^5 == -32, z];
 rootplot = ListPlot[{Re[z], Im[z]} /. root, PlotRange → {{-2.1, 2.1}, {-2.1, 2.1}},
   AxesLabel → {"Re(z)", "Im(z)"}, AspectRatio → 1, PlotStyle -> PointSize[0.03]];
 Show[rootplot, Graphics[{Red, Circle[{0, 0}, 2]}]]]
```



■ Complex Series

Everything that works for real series in *Mathematica* (and which we discussed before) was actually working all along for complex series

```
Sum1[z_] = Sum[z^n / Sqrt[n], {n, 1, Infinity}]
```

$\text{PolyLog}\left[\frac{1}{2}, z\right]$

The disk of convergence has radius 1 for this sum. On the boundary, the sum diverges at some point and converges at others:

```
Sum1[1]
```

```
ComplexInfinity
```

```
N[Sum1[I]]
```

```
-0.427728 + 0.667691 i
```

```
N[Sum1[-1]]
```

```
-0.604899
```

Mathematica infact knows how to "analytically continue" the function outside of its disk of convergence (something we may discuss later), e.g.

```
N[Sum1[1 + I]]
```

```
-0.482402 + 1.43205 i
```

■ Complex Functions

Here are some basic examples: everything works for complex arguments.

```
N[Sin[1 + 2 I]]
```

```
3.16578+ 1.9596 ⅈ
```

```
ComplexExpand[Sin[x + I y]]
```

```
Cosh[y] Sin[x] + ⅈ Cos[x] Sinh[y]
```

For Logs and powers *Mathematica* makes standard choices to resolve the ambiguity in the argument of the logarithm. Note that the N[] (for numerically evaluate) is need to get an actual numerical result. Note that //N after a command has the same effect.

```
Log[3 + I]
```

```
Log[3 + ⅈ]
```

```
N[Log[3 + I]]
```

```
1.15129+ 0.321751 ⅈ
```

```
Log[3 + I] // N
```

```
1.15129 + 0.321751 ⅈ
```

```
(1 + 2 I) ^ (3 + I)
```

$(1+2\,i)^{3+i}$

```
N[(1 + 2 I) ^ (3 + I)]
```

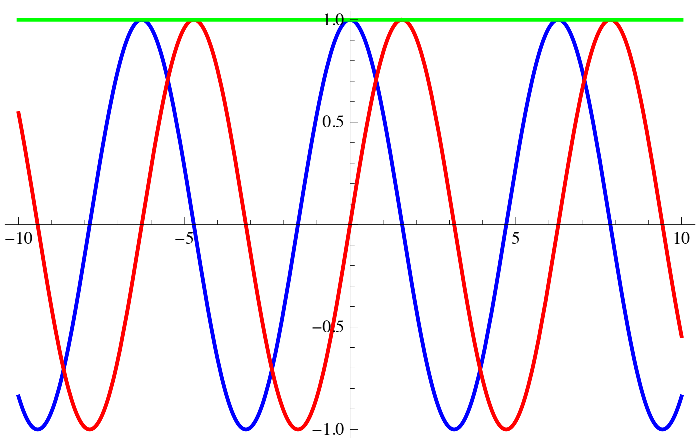
```
-2.0442 - 3.07815 ⅈ
```

■ Plotting Complex Functions

Complex valued function can be difficult to visualize due to depending on multiple variables and functions behaving differently along the imaginary axis. Using *Mathematica*'s 2D plots separately for the real and imaginary parts, contour plots and 3D plots can greatly help. The following are a few examples.

Looking at the exponential function e^z for a purely imaginary argument

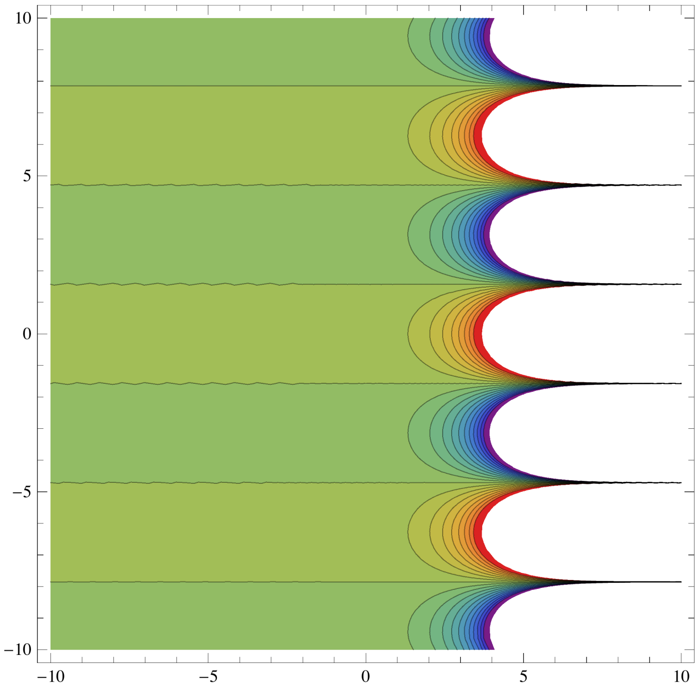
```
Plot[{Re[Exp[I * x]], Im[Exp[I * x]], Abs[Exp[I * x]]},
 {x, -10, 10}, PlotStyle → {{Thick, Blue}, {Thick, Red}, {Thick, Green}}]
```



We can see the real part (blue) is a Cos[] whereas the imaginary part (red) is a Sin[] and the magnitude stays a constant value of 1.

Here is a contour plot with a general complex argument.

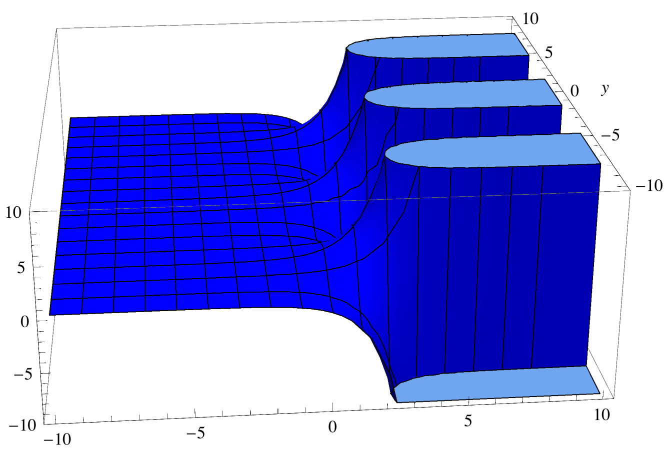
```
ContourPlot[Re[Exp[x + I * y]], {x, -10, 10}, {y, -10, 10},
 Contours → 20, ContourShading → Automatic, ColorFunction → "Rainbow"]
```



The more red the the region is, the larger the function is, the more blue, the smaller.

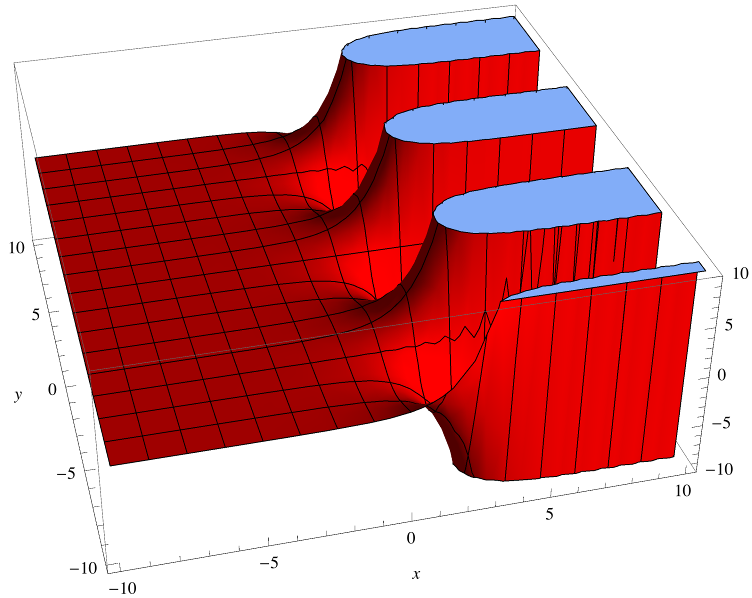
You can see the same plot but with a third dimension showing the value of the function using Plot3D

```
Plot3D[{Re[Exp[x + I * y]]}, {x, -10, 10}, {y, -10, 10},
 PlotStyle → Blue, PlotRange → {-10, 10}, AxesLabel → Automatic]
```



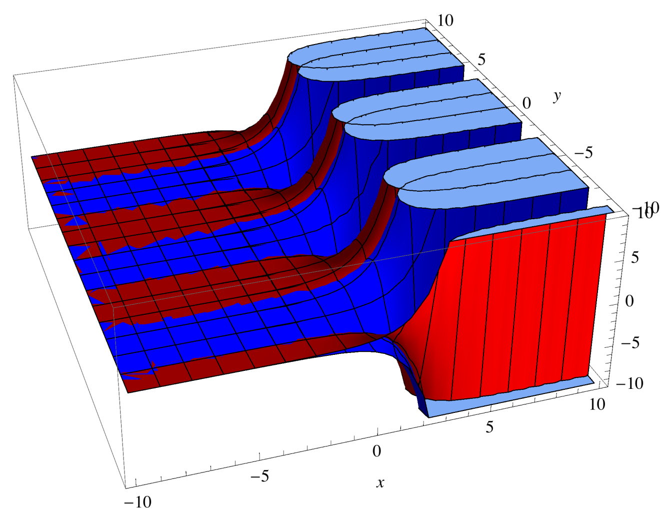
As you would expect, the imaginary part looks the same but shifted by a $\pi/2$ phase

```
Plot3D[{Im[Exp[x + I * y]]}, {x, -10, 10}, {y, -10, 10},
 PlotStyle → Red, PlotRange → {-10, 10}, AxesLabel → Automatic]
```



Here they are together.

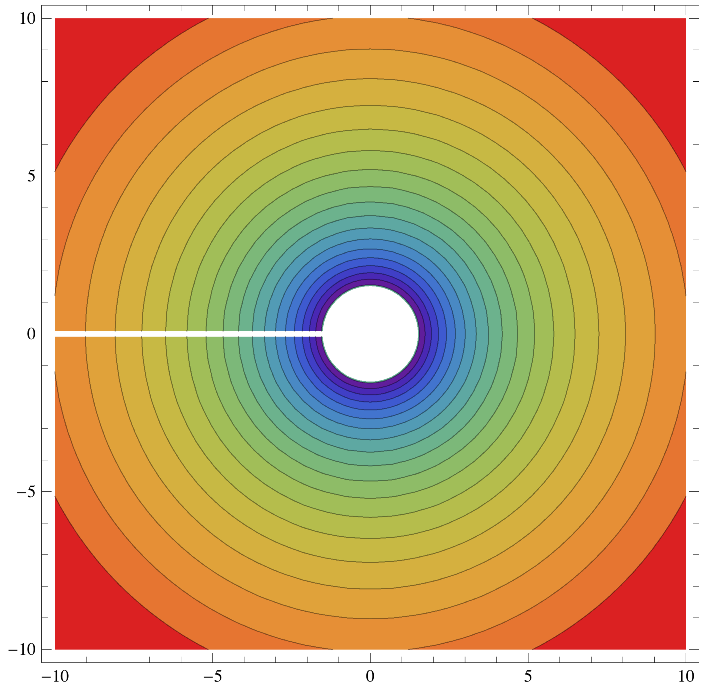
```
Plot3D[{Re[Exp[x + I * y]], Im[Exp[x + I * y]]}, {x, -10, 10}, {y, -10, 10},
 PlotStyle → {Blue, Red}, PlotRange → {-10, 10}, AxesLabel → Automatic]
```



These sorts of plots can be especially useful for visualizing branch cuts, such as the one along the negative real line for the Log[ ] function.

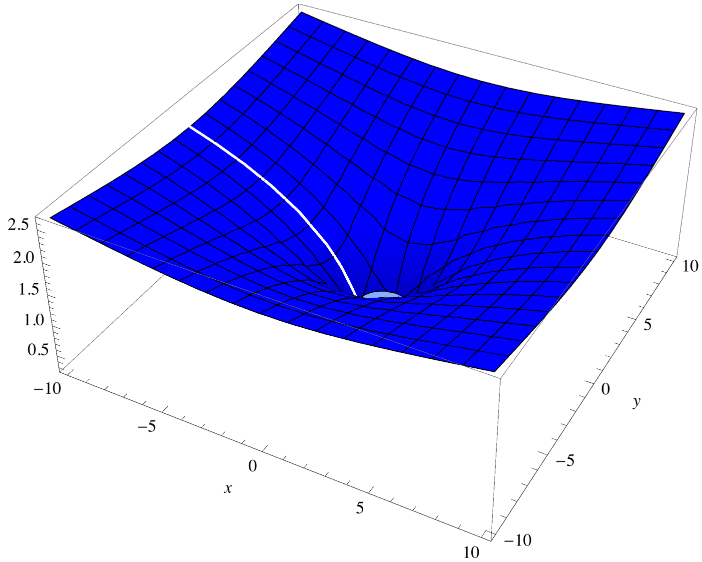
Please note that *Mathematica* chooses to put the discontinuity in the imaginary part of the Logarithm between $-\pi$ and $+\pi$, rather than between 0 and 2π, as discussed in class. This moves the “branch cut” to the negative real axis.

```
ContourPlot[Re[Log[x + I * y]], {x, -10, 10}, {y, -10, 10},
 ContourShading → Automatic, ColorFunction → "Rainbow", Contours → 20]
```

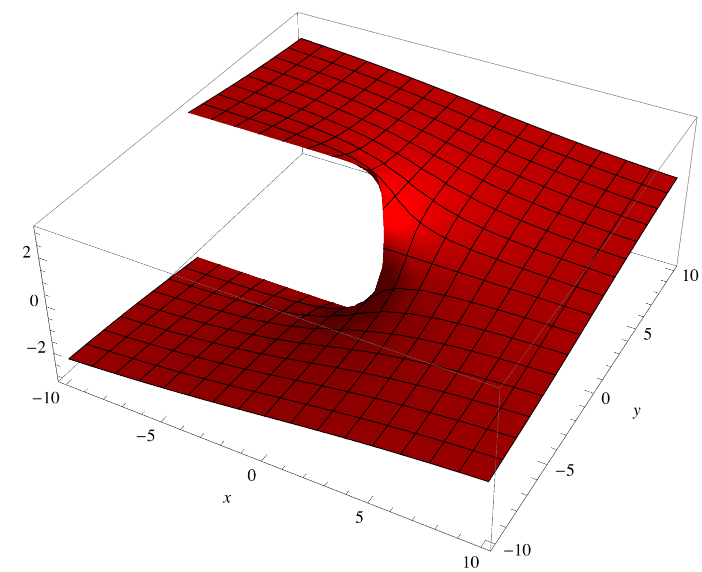



```
ContourPlot[Im[Log[x + I * y]], {x, -10, 10}, {y, -10, 10},
 ContourShading → Automatic, ColorFunction → "Rainbow", Contours → 30]
```

```
Plot3D[Re[Log[x + I * y]], {x, -10, 10}, {y, -10, 10}, PlotStyle → Blue, AxesLabel → Automatic]
```

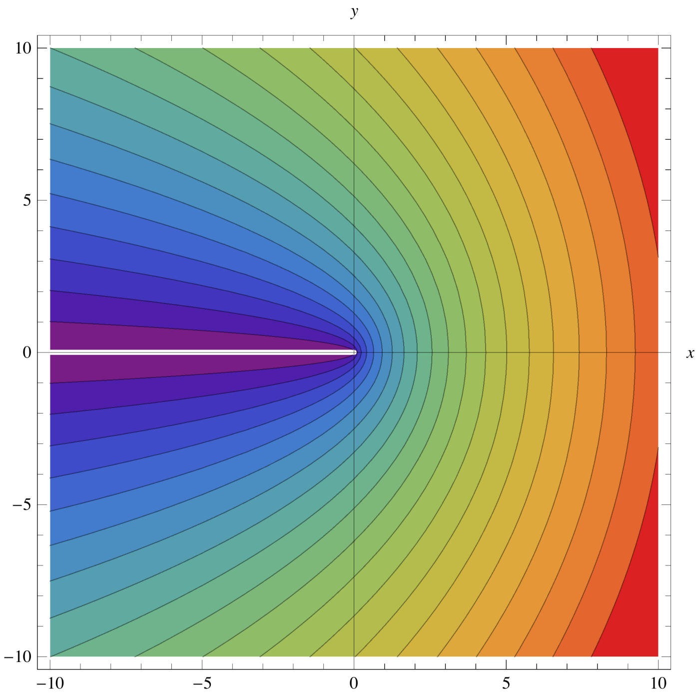



```
Plot3D[Im[Log[x + I * y]], {x, -10, 10}, {y, -10, 10}, PlotStyle → Red, AxesLabel → Automatic]
```

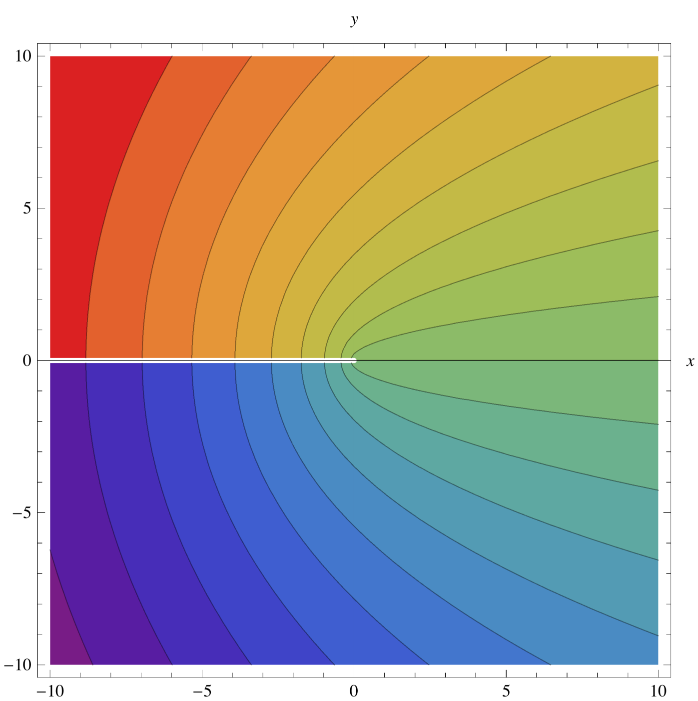


Here is another example with $\sqrt{z}$ which also has a branch cut along the negative real line.

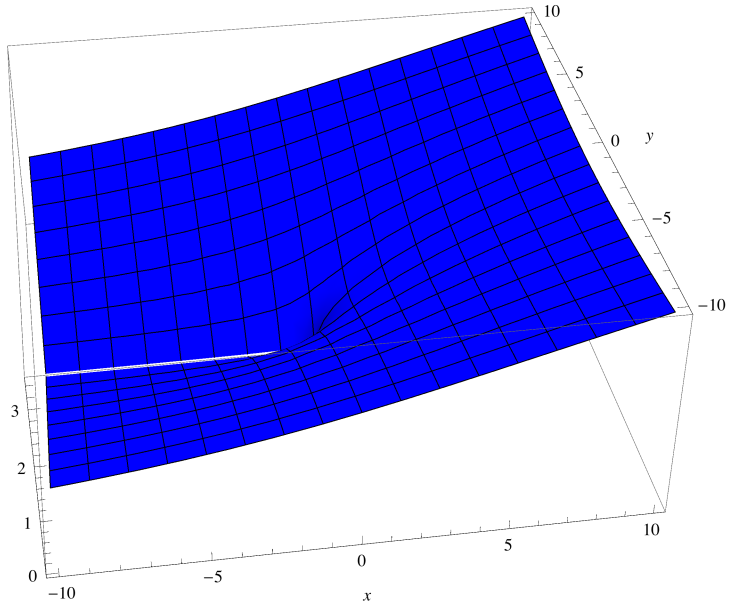
```
ContourPlot[Re[Sqrt[x + I * y]], {x, -10, 10}, {y, -10, 10}, AxesLabel → Automatic,
 ContourShading → Automatic, ColorFunction → "Rainbow", Contours → 20]
```




```
ContourPlot[Im[Sqrt[x + I * y]], {x, -10, 10}, {y, -10, 10}, AxesLabel → Automatic,
 ContourShading → Automatic, ColorFunction → "Rainbow", Contours → 20]
```



```
Plot3D[Re[√(x + I * y)], {x, -10, 10}, {y, -10, 10}, PlotStyle → Blue, AxesLabel → Automatic]
```




```
Plot3D[Im[√(x + I * y)], {x, -10, 10}, {y, -10, 10}, PlotStyle → Red, AxesLabel → Automatic]
```