Appendix J

Complex Numbers and `cmath`

The square root of a negative number (e.g., $\sqrt{-1}$, $\sqrt{-5}$, $\sqrt{-9}$) is called an imaginary number, and it's convenient to introduce the symbol $i = \sqrt{-1}$ and adopt $\sqrt{-n} = i\sqrt{n}$ as its standard form (e.g. $\sqrt{-5} = i\sqrt{5}$). i has the property $i^2 = -1$, and for higher powers:

$$i^3 = i^2 \cdot i = (-1)i = -i, \quad i^4 = (i^2)^2 = (-1)^2 = 1, \quad i^5 = i^4 \cdot i = i, \text{ etc.}$$

$a+bi$, where a and b are reals, is called a complex number. The first term a is its real part, and the second term bi is the imaginary part. Two complex numbers, $a + bi$ and $c + di$, are equal if and only if $a = c$ and $b = d$.

The conjugate of $a + bi$ is $a - bi$. Multiplying a number to its conjugate produces a real number, the square of its magnitude:

$$(a + bi)(a - bi) = a^2 + b^2$$

The basic algebraic operations:

- **Addition**. Separately add the real parts and imaginary parts. For example,

$$(2 + 3i) + (4 - 5i) = (2 + 4) + (3 - 5)i = 6 - 2i.$$

- **Subtraction**. Separately subtract the real parts and imaginary parts. For example,

$$(2 + 3i) - (4 - 5i) = (2 - 4) + [3 - (-5)]i = -2 + 8i.$$

- **Multiplication**. Perform a multiplication as if each number was a tuple of two values, and replace i^2 by -1 in the result.

$$(a + bi)(c + di) = ac - bd + (ad + bc)i$$

For example,

$$(2+3i)(4-5i) = 8 - 15(-1) + (-10+12)i = 23 + 2i.$$

- **Division.** Multiply the fraction's numerator and denominator by the conjugate of the denominator. For example,

$$\frac{2+3i}{4-5i} = \frac{(2+3i)(4+5i)}{(4-5i)(4+5i)} = \frac{(8-15)+(10+12)i}{16+25} = -\frac{7}{41} + \frac{22}{41}i.$$

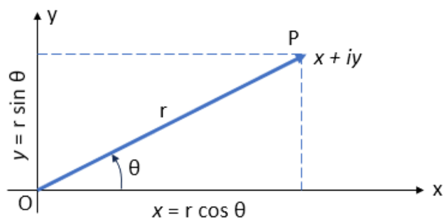

Figure J.1. A Complex Number Graphically

In Fig. J.1, $x+yi$ is represented by the point P at (x, y). It can also be encoded as the directed line segment $\overrightarrow{OP}$. This vector (and hence the complex number) can be described in terms of its length r and the angle θ it makes with the positive x-axis. $r = \sqrt{x^2+y^2}$ is the modulus (magnitude or absolute value) of the complex number, while θ is the phase (amplitude or arg), and usually the smallest, positive angle for which $\tan\theta = y/x$.

From Fig. J.1, we see that $x = r\cos\theta$ and $y = r\sin\theta$, so $z = x+yi = r\cos\theta + ir\sin\theta = r(\cos\theta + i\sin\theta)$. We call this the polar or trigonometric form, and $z = x+yi$ the rectangular form.

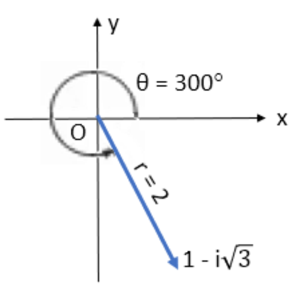

Figure J.2. $1 - i\sqrt{3}$ in Polar Form

Consider $z = 1 - i\sqrt{3}$ in Fig. J.2. Its modulus $r = \sqrt{(1)^2 + (-\sqrt{3})^2} = 2$, and since $\tan\theta = y/x = -\sqrt{3}/1 = -\sqrt{3}$, the phase θ is either 120° or 300°. However, since z lies in quadrant IV then $\theta = 300°$, and its polar form is

$$z = r(\cos\theta + i\sin\theta) = 2(\cos 300° + i\sin 300°).$$

Note that z may also be represented by

$$z = 2[\cos(300° + n360°) + i\sin(300° + n360°)]$$

where n is any integer. For instance, we could write $z = 2(cos(-60°)+isin(-60°)$.

J.1 Multiplication in Polar Form

The polar representation is particularly useful when multiplying complex numbers together. If

$$z = \rho(\cos\phi + i\sin\phi),$$

and

$$z' = \rho'(\cos\phi' + i\sin\phi'),$$

then

$$zz' = \rho\rho'\left[(\cos\phi\cos\phi' - \sin\phi\sin\phi') + i(\cos\phi\sin\phi' + \sin\phi\cos\phi')\right]$$

Using the addition theorems for sine and cosine,

$$\begin{aligned}\cos\phi\cos\phi' - \sin\phi\sin\phi' &= \cos(\phi + \phi'),\\ \cos\phi\sin\phi' + \sin\phi\cos\phi' &= \sin(\phi + \phi').\end{aligned}$$

the equation can be rewritten as:

$$zz' = \rho\rho'\left[\cos(\phi + \phi') + i\sin(\phi + \phi')\right].$$

This form has modulus $\rho\rho'$ and phase $\phi + \phi'$. In other words, to multiply two complex numbers, we multiply their moduli and *add* their angles, as visualized in Fig. J.3.

J.2 De Moivre's Formula

The polar multiplication formula has a particularly important consequence when $z = z'$, for then we have

$$z^2 = \rho^2(\cos 2\phi + i\sin 2\phi).$$

Multiplying again by z we obtain

$$z^3 = \rho^3(\cos 3\phi + i\sin 3\phi),$$

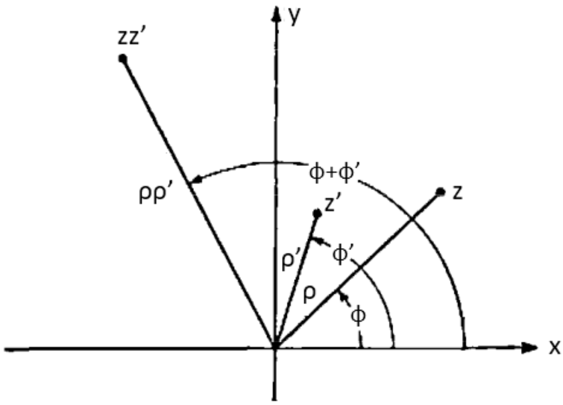

Figure J.3. $1 - i\sqrt{3}$ in Polar Form

and continuing indefinitely in this way,

$$z^n = \rho^n (\cos n\phi + i \sin n\phi), \quad \text{for any integer } n.$$

For example,

$$\begin{aligned}(\sqrt{3} - i)^{10} &= \{2(\cos 330^\circ + i \sin 330^\circ)\}^{10} \\ &= 2^{10}(\cos(10 \cdot 330^\circ) + i \sin(10 \cdot 330^\circ)) \\ &= 1024(\cos 60^\circ + i \sin 60^\circ) = 1024(1/2 + i\sqrt{3}/2) \\ &= 512 + 512i\sqrt{3}.\end{aligned}$$

If z is a point on the unit circle, with $\rho = 1$, we obtain the formula discovered by A. De Moivre (1667–1754):

$$(\cos\phi + i \sin\phi)^n = \cos n\phi + i \sin n\phi.$$

J.3 The nth Root of a Number

A complex number $a + bi = r(\cos\theta + i \sin\theta)$ has exactly n distinct nth roots, which can be readily obtained by applying De Moivre's formula. As an example, let's find the fifth roots of $(4 - 4i)$ by considering its polar version $4\sqrt{2}(\cos 315^\circ + i \sin 315^\circ)$ in its most general form

$$4\sqrt{2}[\cos(315^\circ + k360^\circ) + i \sin(315^\circ + k360^\circ)],$$

where k is any integer, including zero. Using De Moivre's theorem, the fifth root of $(4 - 4i)$ is

$$\begin{aligned}
&\{4\sqrt{2}[\cos(315° + k360°) + i\sin(315° + k360°)]\}^{1/5} \\
&= (4\sqrt{2})^{1/5}\left(\cos\frac{315° + k360°}{5} + i\sin\frac{315° + k360°}{5}\right) \\
&= \sqrt{2}[\cos(63° + k72°) + i\sin(63° + k72°)].
\end{aligned}$$

Assigning in turn the values $k = 0, 1, 2, \cdots$, we find

$$\begin{aligned}
k = 0: &\quad \sqrt{2}(\cos 63° + i\sin 63°) = R_1 \quad \text{(see Fig. J.4)} \\
k = 1: &\quad \sqrt{2}(\cos 135° + i\sin 135°) = R_2 \\
k = 2: &\quad \sqrt{2}(\cos 207° + i\sin 207°) = R_3 \\
k = 3: &\quad \sqrt{2}(\cos 279° + i\sin 279°) = R_4 \\
k = 4: &\quad \sqrt{2}(\cos 351° + i\sin 351°) = R_5 \\
k = 5: &\quad \sqrt{2}(\cos 423° + i\sin 423°) = \sqrt{2}(\cos 63° + i\sin 63°) = R_1
\end{aligned}$$

The five fifth roots were obtained with $k = 0, 1, 2, 3, 4$, and in the general nth root case by assigning $0, 1, 2, 3, \ldots, n - 1$ to k.

The modulus of each of the roots is $\sqrt{2}$ so they all lie on a circle of radius $\sqrt{2}$ centered at the origin. The difference in phase of two consecutive roots is 72° so the roots are equally spaced out around the circle, as in Fig. J.4.

J.4 Euler's Formula

In what follows, the expression $\cos\theta + i\sin\theta$ is so common that I'll abbreviate it to $\operatorname{cis}\theta$, so that, for example,

$$\operatorname{cis}\frac{4}{3}\pi = \operatorname{cis} 240° = \cos 240° + i\sin 240° = -\frac{1}{2} - \frac{1}{2}i\sqrt{3}.$$

When

$$z = \operatorname{cis}\theta = \cos\theta + i\sin\theta,$$

then

$$dz/d\theta = -\sin\theta + i\cos\theta = iz$$

We often see differential equations of this sort in physics: $dy/dt = ky$, and the solution is $y = Ae^{kt}$ (where A is the value of y when $t = 0$). By analogy, it isn't too unreasonable to hope that the solution of our derivative might be something like

$$z = Ae^{i\theta}, \quad \text{i.e.,} \quad \operatorname{cis}\theta = Ae^{i\theta}$$

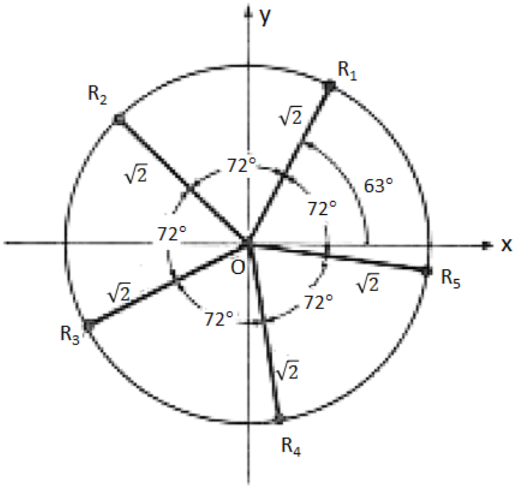

Figure J.4. The Fifth Roots of $4 - 4i$

Putting $\theta = 0$, we get $A = 1$, so

$$\operatorname{cis}\theta = e^{i\theta}.$$

To express what we mean by a power a^z when z is complex, we need a definition that produces the standard value of a^x when z is real, but also gives $a^{z_1} \times a^{z_2} = a^{z_1+z_2}$ for all complex numbers z_1 and z_2. A suitable definition utilizes an infinite series of complex terms:

$$e^z = 1 + z + z^2/2! + z^3/3! + z^4/4! + \cdots + z^n/n! + \cdots$$

which can be shown to converge for all values of z.

It's also possible to prove that $e^{z_1} \cdot e^{z_2} = e^{z_1+z_2}$, so that $e^z = e^{x+iy} = e^x \cdot e^{iy}$, where x and y are real. By definition, e^{iy} is

$$\begin{aligned} e^{iy} &= 1 + iy + (iy)^2/2! + (iy)^3/3! + \cdots \\ &= 1 + iy - y^2/2! - iy^3/3! + y^4/4! + \cdots \\ &= (1 - y^2/2! + y^4/4! - \cdots) + i(y - y^3/3! + y^5/5! - \cdots) \end{aligned}$$

It can be shown by Maclaurin's expansion that for all y,

$$\cos y = 1 - y^2/2! + y^4/4! - \cdots \quad \text{and} \quad \sin y = y - y^3/3! + y^5/5! - \cdots$$

so we have

$$e^{iy} = \cos y + i \sin y$$

or

$$e^{i\theta} = \cos\theta + i\sin\theta,$$

a formula due to the great mathematician Euler (1707-1783). Note that when $\theta = \pi$, we have $e^{i\pi} + 1 = 0$, a remarkable equation connecting the mathematical Big Five – e, π, i, 1, and 0.

The formula

$$\operatorname{cis}\theta_1 \cdot \operatorname{cis}\theta_2 = \operatorname{cis}(\theta_1 + \theta_2)$$

can now be rephrased as:

$$e^{i\theta_1} \cdot e^{i\theta_2} = e^{i(\theta_1+\theta_2)}$$

It's trivial to derive De Moivre's formula using Euler's formula and the exponential law for integer powers

$$(e^{ix})^n = e^{inx},$$

since Euler's formula implies that the left side of this equality is equivalent to $(\cos x + i \sin x)^n$ while the right side is equal to $\cos nx + i \sin nx$.

Using the r, θ polar forms, three useful consequences are

$$\begin{aligned} r_1 e^{i\theta_1} \cdot r_2 e^{i\theta_2} &= r_1 r_2 e^{i(\theta_1+\theta_2)} \\ (re^{i\theta})^n &= r^n e^{in\theta} \\ (re^{i\theta})^{1/n} &= r^{1/n} e^{i\theta/n} \end{aligned}$$

The angle θ is periodic — a complete rotation of 360° or 2π radians leaves the trigonometric functions unchanged, so that:

$$\begin{aligned} e^{i\theta} &= e^{i(\theta+2\pi)} = e^{i(\theta+4\pi)} = e^{i(\theta+6\pi)} = \ldots \\ (re^{i\theta})^{1/n} &= r^{1/n} e^{i\theta/n} = r^{1/n} e^{i(\theta+2\pi)/n} = \ldots \end{aligned}$$

and so every complex number has n n-th roots:

$$\sqrt[n]{z} = z^{1/n} = (re^{i\theta})^{1/n} = r^{1/n} e^{i(\theta+2\pi k)/n}, \quad k = 0, 1, 2, \ldots, n-1$$

J.5 Multi-valued Exponentiation

e^z is not a one-to-one function for complex numbers, which is a problem because a function can only have an inverse if it maps distinct values to distinct values. We need to exclude enough complex numbers from z's domain so that any e^z result can only be generated by a single z. The problem is that a complex number's polar angle can be represented by $\theta + 2\pi k$, where k is any integer. The most common choice for restricting its range is to require $-\pi < \theta \leq \pi$, which is called a *principal branch*.

J.5.1 The Complex Logarithm, $\log z$**.** Once e^z is defined to be one-to-one, it makes sense to find its inverse, $\log z$. We can define it by considering the polar form $z = re^{i\theta}$. Taking logs of both sides results in

$$\log z = \log(re^{i\theta}) = \log r + i\theta = \log r + i(\theta + 2\pi k)$$

As usual, the angle is θ plus any multiple of 2π, which poses a familiar problem. The left side of Fig. J.5 shows the imaginary part of $\log z$ spiralling up the z-axis, highlighting how a single $(x + iy)$ point can generate an infinite number of log values depending on how its polar angle is interpreted.

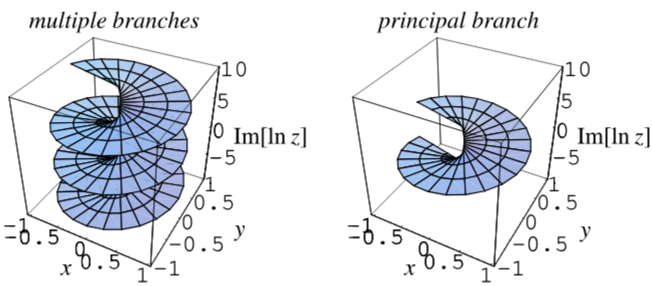

Figure J.5. 3D Surface Plot of $\log(z)$ Showing its Imaginary Part

This is avoided by applying angle-based branch cuts to the log function similar to those imposed on e^z. The angles are constrained so that

$$\log(z) = \log r + i\theta \quad \text{where } -\pi < \theta \leq \pi$$

The right-hand side of Fig. J.5 shows what this does to the imaginary part of $\log z$. When $\theta \approx \pi$, z is just above the negative real x-axis, and $\theta \approx -\pi$ is just below the axis.

Although this constrains $\log z$ to be single-valued, it introduces a discontinuity at the boundaries of the branch cuts along the negative x-axis – the gap visible in the figure.

J.5.2 Other Complex Math Functions. The exponential function is the foundation for most of the other complex math functions. For example,

$$\sin z = \frac{e^{iz} - e^{-iz}}{2i}, \qquad \cos z = \frac{e^{iz} + e^{-iz}}{2},$$

and

$$\sinh z = \frac{e^{z} - e^{-z}}{2}, \qquad \cosh z = \frac{e^{z} + e^{-z}}{2}.$$

Of course, this implies that

$$\sin(iz) = i \sinh z, \qquad \cos(iz) = \cosh z.$$

The inverse complex functions are defined using logarithms:

$$\begin{aligned} \arcsin z &= -i \log(iz + \sqrt{1 - z^2}), & \arccos z &= -i \log(z + i\sqrt{1 - z^2}), \\ \sinh^{-1} z &= \log(z + \sqrt{z^2 + 1}), & \cosh^{-1} z &= \log(z + \sqrt{z^2 - 1}). \end{aligned}$$

The multi-valued nature of the angles used in $z = re^{i\theta}$ mean that these functions must also utilize branch cuts to limit their range of inputs and outputs.

J.6 Complex Numbers in Python

The imaginary unit symbol is written as `j` in Python, so $3 + 4i$ becomes `3+4j`, but i on its own is written as `1j`, not `j`. Below is a sample session involving complex numbers and simple arithmetic:

```
>>> a = 3 + 4j
>>> b = 1 - 2j
>>> a + b
(4+2j)
>>> a - b
(2+6j)
>>> a * b
(11-2j)
>>> a / b
(-1+2j)
>>> a + 2
(5+4j)
>>> a * 2
(6+8j)
>>> a / 2
(1.5+2j)
>>> a**2
(-7+24j)
>>> a ** b      # complex powers
(-21.083139690689016-24.00021070941257j)
>>> a**(1/2)    # roots
(2+1j)
>>> a.conjugate()
(3-4j)
>>> abs(a)      # the modulus
5.0
>>> a.real
3.0
>>> a.imag
4.0
>>> import math
>>> math.degrees(
        math.atan2(a.imag, a.real))
53.13010235415598
```

I didn't need to import `cmath` for any of these basic operations, but `math` is necessary if I want to get a number's polar angle.

The following session shows some of the common errors a user may encounter when first using complex numbers in Python:

```
>>> c = 3 + 4i
             ^
SyntaxError: invalid syntax
>>> (1 + 2j) * (3 + 4j)
(-5+10j)
>>> 1 + 2j * (3 + 4j)     # parenthesis needed
```

```
(-7+6j)
>>> (3 + 4j) // 2
TypeError: can't take floor of complex number.
>>> math.sqrt(1 + 2j)
TypeError: can't convert complex to float
>>> math.sin((3 + 4j))
TypeError: can't convert complex to float
```

J.6.1 Using `cmath`. A common misunderstanding is that the functions in the `math` module will work with complex numbers, which isn't the case as the session above illustrates. For that capability, the `cmath` module (`https://docs.python.org/3/library/cmath.html`) adds an assortment of exponential, logarithmic, trigonometric, and hyperbolic functions, as listed in Table J.1.

Exponential and Logarithmic Functions	
`exp(z)`	Return e^z
`log(z[, base])`	Return the logarithm of z to the given base (default: natural log)
`log10(z)`	Return the base-10 logarithm of z
`sqrt(z)`	Return the square root of z
Trigonometric Functions	
sin(z), cos(z), tan(z), asin(z), acos(z), atan(z)	
Hyperbolic Functions	
sinh(z), cosh(z), tanh(z), asinh(z), acosh(z), atanh(z)	
Complex Number Conversion	
`phase(z)`	Return the phase (angle) of complex number z in radians
`polar(z)`	Convert complex number z to polar coordinates (r, ϕ)
`rect(r, phi)`	Convert polar coordinates (r, ϕ) to complex number
Classification Functions	
`isfinite(z)`	Return True if both real and imaginary parts are finite
`isinf(z)`	Return True if either real or imaginary part is infinity
`isnan(z)`	Return True if either real or imaginary part is NaN
`isclose(a, b)`	Return True if values a and b are close to each other
Constants	
cmath.pi, cmath.e, cmath.tau, cmath.inf, cmath.infj, cmath.nan, cmath.nanj	

Table J.1. Python `cmath` Module

The following session shows a few uses of cmath:

```
>>> a = 3 + 4j
>>> import cmath
>>> r, theta = cmath.polar(a)
>>> import math
>>> math.degrees(theta)
53.13010235415598
>>> r
5.0
>>> cmath.rect(r, theta)
(3.0000000000000004+3.9999999999999996j)
>>> cmath.sin(a)
(3.853738037919377-27.016813258003936j)
>>> cmath.exp(a)
(-13.128783081462158-15.200784463067954j)
>>> cmath.exp(cmath.log(a))
(3+3.999999999999999j)
>>> cmath.exp(1j*math.pi) + 1      # Euler's equation e^{i pi) + 1 = 0
1.2246467991473532e-16j
```

J.6.2 The *n*th Root using Python. Let's return to the *n*th root example from earlier, but this time coded in Python. The aim is to find the fifth roots of $(4-4i)$. The following code creates the variable:

```
>>> a = 4 - 4j
>>> cmath.polar(a)
(5.656854249492381, -0.7853981633974483)   # r, theta
>>> 4 * math.sqrt(2)      # this is r
5.656854249492381
>>> math.radians(315)     # this is not theta
5.497787143782138
>>> math.radians(315-360) # this is theta
-0.7853981633974483
```

The polar angle for $(4-4j)$ has been normalized to fall between $-\pi$ and π.

The next fragment shows the *single* value returned for $a^{1/5}$:

```
>>> root = a ** (1/5)
>>> root
(1.3968022466674208-0.22123174208247431j)
>>> cmath.polar(root)
(1.4142135623730954, -0.15707963267948966)  # r, theta
>>> cmath.phase(root)
-0.15707963267948966      # theta again
>>> math.degrees( cmath.phase(root))   # in degrees
-9.0
```

If we look back at Fig. J.4, it turns out that Python is giving us R_5, but the angle is $-9°$ rather than $351°$.

We could get the other roots by adding and subtracting multiples of 360/5° to this result, but a simpler approach is to use my getRoots() function, imported from complexUtils.py:

```
>>> import complexUtils
>>> rs = complexUtils.getRoots(a, 5)     # a^(1/5)
>>> for r in rs: print(r)
(1.3968022466674208-0.22123174208247431j)
(0.6420395219202063+1.2600735106701009j)
(-1+1.0000000000000002j)
(-1.260073510670101-0.642039521920206j)
(0.22123174208247404-1.3968022466674208j)
```

Once again referring to Fig. J.4, we can see that the roots are listed in counter-clockwise order starting with R_5. The code for getRoots() implements the formula:

$$z^{1/n} = (re^{i\theta})^{1/n} = r^{1/n} * e^{i(\theta+2\pi k)/n}, \quad \text{for } k = 0, 1, 2, \dots, n-1$$

```
def getRoots(z, n):
  # Handle zero specially
  if z.real == 0 and z.imag == 0:
    return [complex(0, 0)]
  # Get principal root
  rRoot = abs(z)**(1/n)
  theta = cmath.phase(z)
  # Calculate all n roots
  roots = []
  for k in range(n):
    ang = (theta + 2*math.pi * k)/n
    roots.append( rRoot*cmath.exp(1j*ang))
  return roots
```

Going one step better, showRoots.py plots these roots in the style of Fig. J.4. In the following call, the user supplies three integers representing $(4-4i)^{1/5}$:

```
> python showRoots.py
Real part: 4
Imaginary part: -4
Root number (n): 5
```

The image in Fig. J.6 is generated.

J.6.3 Investigating Branch Cuts. The branch cut for `cmath.log()` restricts a complex number's angle to be in the range $-\pi < \theta \leq \pi$, which corresponds to either side of the negative real x-axis. This can be confirmed by testing some values which are just above and just below the branch cut:

```
>>> above = cmath.rect(1, math.pi-0.00001)  # e^(i pi - eps)
>>> above
```

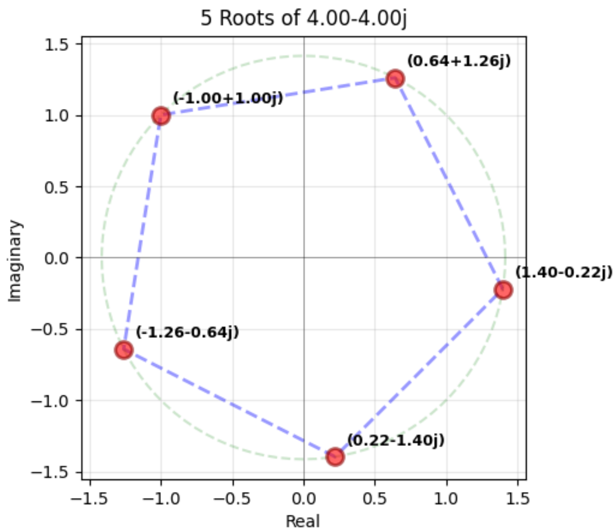

Figure J.6. The Fifth Roots of 4-4i using `showRoots.py`

```
(-0.99999999995+1.000000000002131e-05j)     # (-1 + <small>i)
>>> below = cmath.rect(1, -math.pi+0.00001)  # e^(i -pi + eps)
>>> below
(-0.99999999995-1.000000000002131e-05j)      # (-1 - <small>i)
>>> cmath.log(above)
(-4.1355554465750615e-18+3.141582653589793j)  # 0 + i pi
>>> cmath.log(below)
(-4.1355554465750615e-18-3.141582653589793j)  # 0 - i pi
```

`above` and `below` have different polar coordinates, but their rectangular forms are just above and below $x = -1$. Fig. J.7 shows these two points on the principal branch for `cmath.log()` from Fig. J.5.

The discontinuity in the log curve becomes apparent when the logarithms of `above` and `below` are compared – their imaginary parts differ by almost 2π. In other words, two complex numbers in close proximity have been mapped to very different places on the log curve.

This discontinuity issue exists for all of the complex math functions, so care has to be taken when using them. viewCMath.py offers a way to visualize these curves by creating four plots of the `cmath` function supplied by the user. For instance, in the following call the user has typed `log`:

```
> python viewCMath.py
Enter function name: log
Plotting log(z)
```

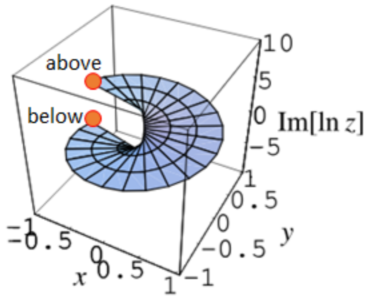

Figure J.7. The `above` and `below` points on the Principal Branch for `cmath.log()`

```
Real range: [-3.0, 3.0]
Imaginary range: [-3.0, 3.0]
```

The program reads the ranges for the real and imaginary axes from the text file cmFuncs.txt, and generates the four plots of `cmath.log(z)` shown in Fig. J.8.

The plots can be individually rotated, and the 'Imaginary Part' graph in Fig. J.8 has been turned so it corresponds to the image in Fig. J.5. The branch cut isn't rendered as a gap but rather as a very abrupt jump in the values plotted on the z-axis.

Each graph is based on a series of (x, y, z) points generated by calling the specified function on $(x + iy)$ to obtain z. Each of these z complex numbers is converted in a different way appropriate for a particular graph. For example, the imaginary plot utilizes `z.imag` values.

The drawback of 3D plots is that they only represent three dimensions of data. Across the base of the graph, I'm plotting the real and imaginary parts of each complex number input to the function ($\log z$ in Fig. J.8), but only *part* of the complex number returned by the function along the z-axis.

For example, the $(x + iy)$ output by a call to $\log z$ is divided into the top-right plot for each x and the bottom-left plot for each y. Argueably this is useful because the branch cut related to the polar angle range becomes clearer when the imaginary output is separated out.

Another very common approach is to draw phase plots enhanced with domain coloring and contour lines. For example, Fig. J.9 is generated by phaseCMath.py when the user enters `log`:

```
> python phaseCMath.py
Enter function name: log
Plotting log(z)
```

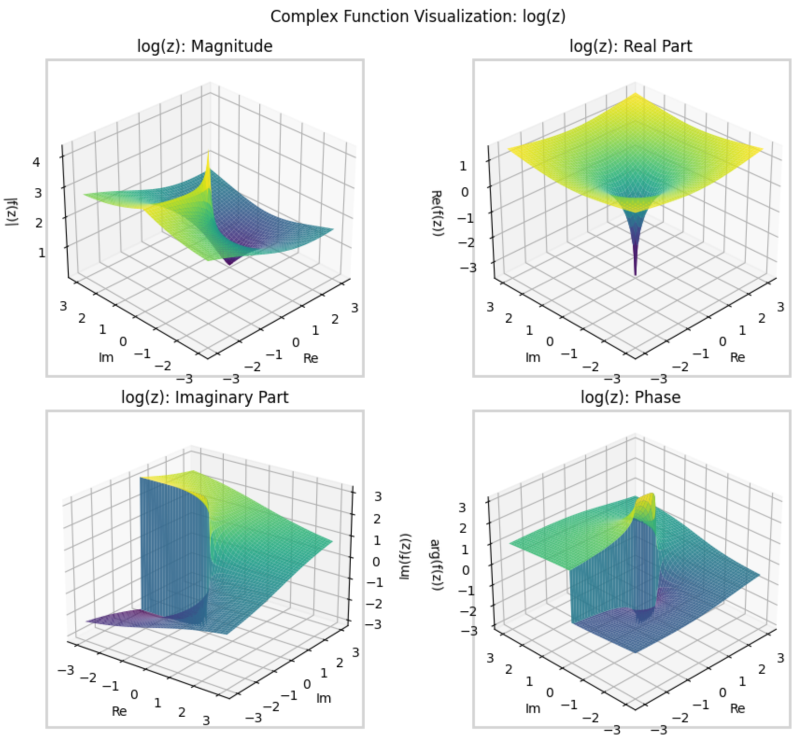

Figure J.8. Visualizing `cmath.log(z)`

```
Real range: [-3.0, 3.0]
Imaginary range: [-3.0, 3.0]
```

Fig. J.9 corresponds most closely to the bottom-right plot of the phase information in Fig. J.8, but the colors, and their ordering, have additional meaning.

Positive numbers are colored red, negative numbers are colored blue, and zeros and poles occur at the points where all the colors meet. A zero is distinguished by the change from red, yellow, green, to blue in a counter-clockwise order around the point. A pole, which is when the output extends off to infinity, has its colors change in a clockwise order. This means that a zero is present on the middle right of Fig. J.9.

A sudden jump from one color to another, like the one that occurs in the middle left of Fig. J.9 is an indication of a discontinuity caused by a branch cut.

There are two types of contour line drawn over the surface – the black lines are contours of equal phase, and are labelled in degrees. The white lines are

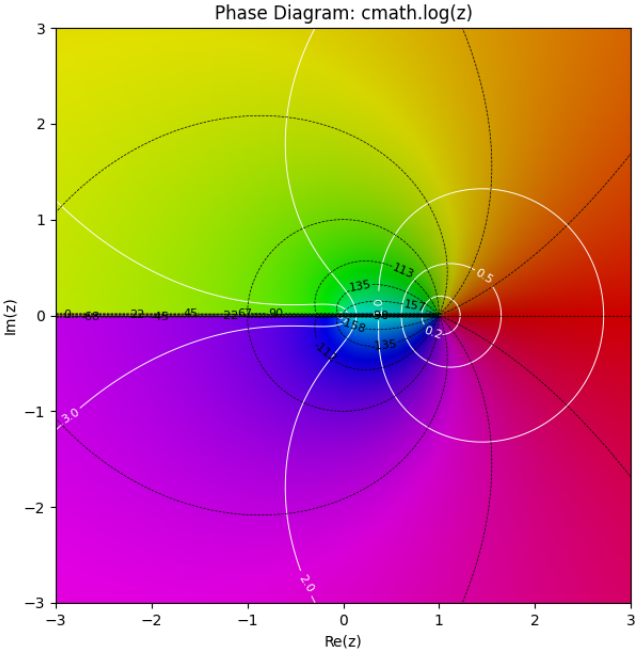

Figure J.9. Visualizing `cmath.log(z)` with a Phase Plot

contours of equal magnitude (modulus), and so are showing the data that is presented in the top-left graph of Fig. J.8. In Fig. J.9 these reinforce the fact that a zero occurs on the middle right, and that the magnitudes increase to the middle left.

Section 7.11.7 on solving cubic equations uses much of the same code (which is located in complexUtils.py). Alternatively, a very nice online tool that does much the same job can be found at `https://www.dynamicmath.xyz/complex/function-plotter/hsv.htm`. A good overview of the approach can be found in [**Weg16**].