Mapping properties of inversion function

In this note, we will use Mathematica to visualize some mapping properties of the inversion function $f(z) = \frac{1}{z}$. The methodology explained below is applicable to *any* complex functions. The inversion function plays a key role in the understanding of Möbius transformation $\frac{az+b}{cz+d}$. Recall that Möbius transformation is a combination of four following transformations: the translation, scaling, rotation, and inversion. Indeed, one can express

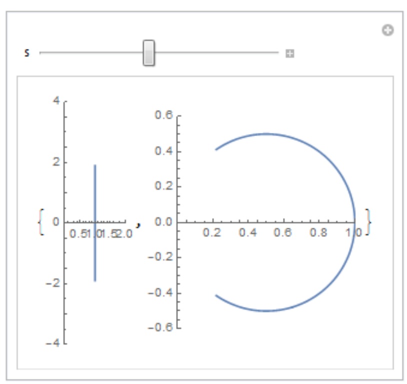
$$\frac{az+b}{cz+d} = \frac{a}{c} + \frac{b - ad/c}{cz+d} = \alpha + \frac{\beta}{z+\gamma}$$

where $\alpha = a/c$, $\beta = b/c - ad/c^2$, $\gamma = d/c$ and look at the following chain:

$$z \xrightarrow{\text{translation}} z + \gamma \xrightarrow{\text{inversion}} \frac{1}{z+\gamma} \xrightarrow[\text{rotation}]{\text{scaling}} \frac{\beta}{z+\gamma} \xrightarrow{\text{translation}} \alpha + \frac{\beta}{z+\gamma}.$$

Let us start by sketching the image of a line, say the vertical line $x = 1$, under the inversion function. The line $x = 1$ has complex parametrization $z = 1 + it$ where $t \in \mathbb{R}$. To plot, we need to restrict the range of t to a finite interval, for example $t \in [-1, 1]$. To get better visualization, one can create an "in-motion" plot by altering the range of t. The idea is that for each $s > 0$, we sketch the image of line $z = 1 + it$ for $t \in [-s, s]$, under inversion. Then vary s to see how the images are drawn out.

```
p1[s_] :=
 ParametricPlot[ReIm[1 + t*I], {t, -s, s}, AxesOrigin -> {0, 0},
  PlotRange -> {{0, 2}, {-4, 4}}]
q1[s_] :=
 ParametricPlot[ReIm[1/(1 + t*I)], {t, -s, s}, AxesOrigin -> {0, 0},
  PlotRange -> {{0, 1}, {-.6, .6}}]
Manipulate[{p1[s], q1[s]}, {s, .1, 4}]
```



The option **AxesOrigin→{0,0}** is to make sure that the axes intersect each other at origin $(0, 0)$. The option **PlotRange → {{a,b},{c,d}}** indicates that we want to see graph in the window $a \le x \le b$, $c \le y \le d$. These two options can be removed. They are used only to fix the window. Without them, the window may change as s varies, which can cause annoyance.

It seems that the image path of line $z = 1 + it$ is the circle $C_{1/2}(1/2)$ with the origin excluded. This fact can be verified rigorously as follows. The vertical line $x = a$ has complex parametrization $z = a + it$.

$$f(z) = \frac{1}{a+it} = \frac{a - it}{(a+it)(a-it)} = \underbrace{\frac{a}{a^2+t^2}}_{u} + i \underbrace{\frac{-t}{a^2+t^2}}_{v}.$$

It follows that

$$u^2 + v^2 = \frac{1}{a^2 + t^2} = \frac{u}{a},$$

which can be rewritten (by completing square) as

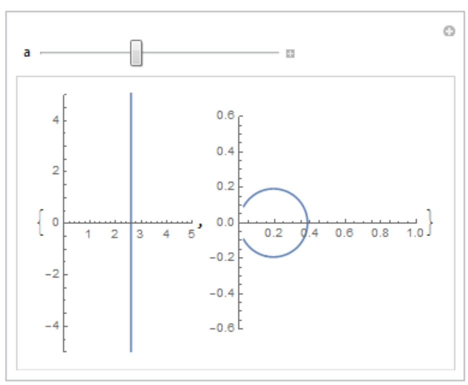
$$\left(u - \frac{1}{2a}\right)^2 + v^2 = \left(\frac{1}{2a}\right)^2.$$

Therefore, $f(z) = u + iv$ indeed lies on the circle $C_{\frac{1}{2a}}(\frac{1}{2a})$. Now that we know the image of a vertical line is a circle centered at a point on the real line and passing through the origin (the origin itself being excluded), our next question is:

What is the image of a half-plane, say $\{z : \ x > 1\}$, under the inversion function?

The half-plane $\{z : \ x > 1\}$ can be viewed as the union of vertical lines $x = a$ where $a > 1$. If we sketch the image of each line, which is the circle $C_{\frac{1}{2a}}(\frac{1}{2a})$, and look at the family of those circles, we can realize the image of the half-plane under inversion.

```
p2[a_] :=
 ParametricPlot[ReIm[a + I*t], {t, -10, 10}, AxesOrigin -> {0, 0},
  PlotRange -> {{0, 5}, {-5, 5}}]
q2[a_] :=
 ParametricPlot[ReIm[1/(a + I*t)], {t, -10, 10}, AxesOrigin -> {0, 0},
   PlotRange -> {{0, 1}, {-.6, .6}}]
Manipulate[{p2[a], q2[a]}, {a, 1, 5}]
```



We see that the image of the half-plane is the open disk $D_{1/2}(1/2)$. Next, we attempt to visualize the angle-preserving (conformality) nature of the inversion function. Note that conformality is a local property, i.e. a property held at a given point and/or its neighborhood. Function $f(z) = 1/z$ is holomorphic on $\mathbb{C}\backslash\{0\}$ and

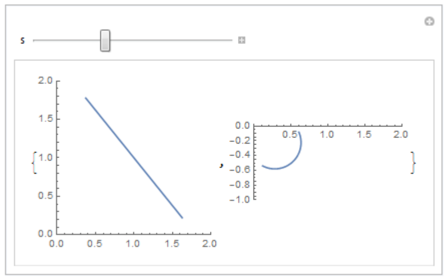
$$f(z) = -\frac{1}{z^2} \neq 0.$$

Thus, f is conformal on $\mathbb{C}\backslash\{0\}$. Fix a point on the complex plane, say $z_0 = 1 + i$. Through z_0, we draw many straight lines and their images under f. The line γ_s passing through z_0 with slope $s \in [0, 2\pi]$ has complex parametrization

$$\gamma_s(t) = z_0 + te^{is} = (1 + t\cos s) + i(1 + t\sin s).$$

Its image under f is a curve η_s with complex parametrization $\eta_s(t) = f(\gamma_s(t)) = \frac{1}{\gamma_s(t)}$. We use Mathematica to sketch the curve γ_s together with its image η_s for different the values of $s \in [0, \pi]$.

```
p3[s_] :=
 ParametricPlot[ReIm[1 + I + t*Exp[I*s]], {t, -1, 1},
  AxesOrigin -> {0, 0}, PlotRange -> {{0, 2}, {0, 2}}]
q3[s_] :=
 ParametricPlot[ReIm[1/(1 + I + t*Exp[I*s])], {t, -1, 1},
  AxesOrigin -> {0, 0}, PlotRange -> {{0, 2}, {-1, 0}}]
Manipulate[{p3[s], q3[s]}, {s, 0, 2*Pi}]
```



We know that

- $\gamma_s(0) = z_0$,
- $\gamma_s'(0) = e^{is}$ is a tangent vector of γ_s at z_0,
- $\eta_s'(0)$ is a tangent vector of η_s at $f(z_0) = \frac{1}{1+i} = \frac{1}{2} - \frac{1}{2}i$.

The chain rule gives $\eta_s'(0) = f'(\gamma_s(0))\gamma_s'(0) = f'(z_0)\gamma_s'(0)$. Thus, the $\operatorname{Arg}\eta_s'(0) = \operatorname{Arg} f'(z_0) + \operatorname{Arg}\gamma_s'(0)$ in modulo 2π. This means the difference between $\operatorname{Arg}\gamma_s'(0)$ and $\operatorname{Arg}\eta_s'(0)$ is unchanged as s varies. This difference is equal to

$$\operatorname{Arg} f'(z_0) = \operatorname{Arg}\left(-\frac{1}{z_0^2}\right) = \operatorname{Arg}\left(\frac{i}{2}\right) = \frac{\pi}{2}.$$

To draw the tangent vector on the curves γ_s and η_s, we first express $\gamma_s'(0)$ and $\eta_s'(0)$ in complex standard form: $\gamma_s'(0) = \cos s + i \sin s$ and $\eta_s'(0) = \frac{i}{2}e^{is} = -\frac{1}{2}\sin s + \frac{1}{2}i\cos s$. The tangent vector of γ_s at z_0 is vector $(\cos s, \sin s)$ based at $(1,1)$. The tangent vector of η_s at $f(z_0)$ is vector $(-1/2\sin s, 1/2\cos s)$ based at $(1/2, -1/2)$. One can draw these vectors by adding the **Epilog** option to the previous commands

```
p3[s_] := ParametricPlot[...,
Epilog -> {Arrow[{{1, 1}, {1, 1} + {Cos[s], Sin[s]}}]}]]
q3[s_] := ParametricPlot[...,
Epilog -> {Arrow[{{.5, -.5}, {.5, -.5} + {-.5*Sin[s], .5*Cos[s]}}]}]]
```