Discontinuity and Angles

In this note, we will use Mathematica to

- Visualize discontinuous behavior of complex functions along a curve.
- Visualize the angle between two curves.

1 Discontinuous functions

Let us consider the function \( f(z) = \log(z^2 + 1) \). From the Mathematica practice last time, we know that \( f \) is continuous everywhere excepts for points on the line \( x = 0, y \geq 1 \) and the line \( x = 0, y \leq -1 \). We want to see how \( f(z) \) jumps as \( z \) crosses the imaginary axis (Figure 1). Let us consider the line \( y = 2 \), which has complex form \( z = t + 2i \). This line is mapped to some curve by \( f \), which we call an image curve. We expect that when \( t \) moves from \(-2\) to \( 2 \), the image curve is drawn out continuously, except at \( t = 0 \). At \( t = 0 \), the function \( f(z) = \ln|z^2 + 1| + i\text{Arg}(z^2 + 1) \) jumps by \( 2\pi i \). This is a jump of distance \( 2\pi \) on the vertical direction. In Mathematica (Figure 2),

```mathematica
f[z_] := Log[z^2 + 1]
p[s_] := ParametricPlot[ReIm[t + 2*I], {t, -2, s}, PlotRange -> {{-2, 2}, {0, 4}}]
q[s_] := ParametricPlot[ReIm[f[t + 2*I]], {t, -2, s}, PlotRange -> {{0, 2.5}, {-3, 3}}]
Manipulate[{p[s], q[s]}, {s, -1.9, 2}]
```

Note that there is no mystery about the numbers \(-2, 2, 0, 4, 2.5, -3, 3\) which we put in \textbf{PlotRange}. One should omit option \textbf{PlotRange} from the above commands at the first time of running. This will cause the frame of the plot to vary as one varies \( s \). Once we know the maximum range of the plot, we can specify the \textbf{PlotRange} to fix the frame of the plot.
2 Visualize angles between two curves

Put \( z_0 = 1 + i \). There are infinitely many curves on the complex plane that pass through \( z_0 \). To make vivid our experiment, let us consider two families of curves that pass through \( z_0 \).

\[
\sigma_s(t) = z_0 + e^{is}(1 - e^{it}) \\
\lambda_s(t) = z_0 + t(s + i \cos t).
\]

For each value of \( s \), the curve \( \sigma_s \) and \( \lambda_s \) pass through \( z_0 \) when \( t = 0 \). One can plot both curves together on the complex plane as follows (Figure 3).

```mathematica
z0 = 1 + I
sigma[s_, t_] := z0 + Exp[I*s]*(1 - Exp[I*t])
lambda[s_, t_] := z0 + t*(s + I*Cos[t])
p[s_] := ParametricPlot[{ReIm[sigma[s, t]], ReIm[lambda[s, t]]}, {t, -2, 2}, PlotRange -> {{0, 2.5}, {0, 2}}, PlotLegends -> Automatic]
Manipulate[p[s], {s, -1, 1}]
```

Figure 3

Note that the option `PlotLegends` is for us to distinguish the curves more easily. The blue curve corresponds to the first function (which is \( \sigma_s \)) and the orange curve corresponds to the second function (which is \( \lambda_s \)).

Now let us draw the velocity vectors on \( \sigma_s \) at \( \lambda_s \) at \( z_0 \). That is to draw vectors \( \sigma_s'(0) \) and \( \lambda_s'(0) \) from the point \( z_0 \). The option `Epilog` allows us to annotate the graph. We want to draw two arrows, namely \( \sigma_s'(0) \) and \( \lambda_s'(0) \), at point \( z_0 \). In Mathematica, the command `Arrow[{{a, b}, {c, d}}]` draws an arrow from point \( (a, b) \) to point \( (c, d) \). One can compute

\[
\sigma_s'(0) = e^{is}(-i) \\
\lambda_s'(0) = s + i.
\]

Thus, the first arrow \( \sigma_s'(0) \) can be drawn by the command

\[
\text{Arrow}[\{(1, 1), (1, 1) + \text{ReIm}[\text{Exp}[I*s]*(-1)]\}]
\]

The second arrow can be drawn by the command

\[
\text{Arrow}[\{(1, 1), (1, 1) + \text{ReIm}[s+I]\}]
\]

Don’t execute those commands yet. We put these two commands inside the curly brackets (separated by comma) of the `Epilog -> {...}` command as follows (Figure 4).
The angle between the two curves at the intersection point $z_0$ is defined as the angle between these two velocity vectors. To compute the angle between two curves $\sigma_s$ and $\lambda_s$, for example when $s = -1$ (see Figure 5), we first compute the argument of each velocity vector:

$$\theta_1 = \text{Arg}(\sigma'_s(0)) = \text{Arg}(e^{is}(-i)) = \text{Arg}(e^{-i}(-i))$$
$$\theta_2 = \text{Arg}(\lambda'_s(0)) = \text{Arg}(s + i) = \text{Arg}(-1 + i).$$

Then the angle between the two velocity vectors (swiping from $\sigma'_s(0)$ to $\lambda'_s(0)$) is $\theta = \theta_2 - \theta_1$ (in modulo $2\pi$). In Mathematica,

```mathematica
theta1 = Arg[Exp[-I]*(-I)]
theta2 = Arg[-1+I]
theta = theta2 - theta1
```

Consider the function

$$f(z) = \frac{iz}{z - 3}.$$  

We want to see the angle between the image of the curve $\sigma_s$ and the image of the curve $\lambda_s$ under $f$. The image of the curve $\sigma_s$ under $f$ is $\Sigma_s(t) = f(\sigma_s(t))$. The image of the curve $\lambda_s$ under $f$ is $\Lambda_s(t) = f(\lambda_s(t))$. The image of $z_0$ under $f$ is $w_0 = f(z_0)$. In Mathematica,
\[ f[z_] := I*z/(z-3) \]

\[ w_0 = f[z_0] \]

The velocity of \( \Sigma_s \) at \( w_0 \) is \( \Sigma_s'(0) \). The velocity of \( \Lambda_s \) at \( w_0 \) is \( \Lambda_s'(0) \). In Mathematica, one can compute these velocity vectors by

\[
\begin{align*}
\text{Sigma}[s_\_\_, t_\_] & := f[\text{sigma}[s, t]] \\
\text{Lambda}[s_\_\_, t_\_] & := f[\text{lambda}[s, t]] \\
v1[s_\_] & := \text{D}[\text{Sigma}[s, t], t] \/. t \rightarrow 0 \\
v2[s_\_] & := \text{D}[\text{Lambda}[s, t], t] \/. t \rightarrow 0
\end{align*}
\]

Here the operator \( / . \) is the substitution operator. The third of the above commands means that the velocity vector \( v_1(s) \) is obtained by first taking the derivative of \( \Sigma(s,t) \) with respect to \( t \) and then substituting \( t \) by 0.

Because \( f \) is holomorphic at \( z_0 \) and

\[ f'(z_0) = \left. \frac{-3i}{(z-3)^2} \right|_{z=z_0} = \frac{-3i}{(i-2)^2} \neq 0, \]

we know that \( f \) is angle-preserving (conformal) at \( z_0 \). In Mathematica,

\[
\begin{align*}
p[s_] & := \text{ParametricPlot}[\{\text{ReIm}[\text{sigma}[s, t]], \text{ReIm}[\text{lambda}[s, t]]\}, \\
& \{t,-2,2\}, \text{PlotRange} \rightarrow \{\{0,2.5\},\{0,2\}\}, \\
& \text{Epilog} \rightarrow \{\text{Arrow}[\{\text{ReIm}[z_0], \text{ReIm}[z_0] + \text{ReIm}[\text{Exp}[I*s]*(1-I)]\}], \\
& \text{Arrow}[\{\text{ReIm}[z_0], \text{ReIm}[z_0] + \text{ReIm}[s+I]\}]\}) \\
q[s_] & := \text{ParametricPlot}[\{\text{ReIm}[\text{Sigma}[s, t]], \text{ReIm}[\text{Lambda}[s, t]]\}, \\
& \{t,-2,2\}, \text{PlotRange} \rightarrow \{\{0,3\},\{-2,2\}\}, \\
& \text{Epilog} \rightarrow \{\text{Arrow}[\{\text{ReIm}[w_0], \text{ReIm}[w_0] + \text{ReIm}[v_1[s]]\}], \\
& \text{Arrow}[\{\text{ReIm}[w_0], \text{ReIm}[w_0] + \text{ReIm}[v_2[s]]\}]\}) \\
\text{Manipulate}[\{p[s], q[s]\}, \{s,-1,1\}] \\
\end{align*}
\]

![Figure 6](image)

We can see from the picture that the angle between the image curves \( \Sigma_s \) and \( \Lambda_s \) is the same as the angle between the original curves \( \sigma_s \) and \( \lambda_s \).