ComplexRegions.jl: A Julia package for regions in the complex plane

Tobin A. Driscoll[1]

1 University of Delaware

Summary

Complex variables offer a unique and powerful way to represent planar regions and maps between them. Since the work of Riemann and others in the 19th century, complex variables have been used to derive quantities and properties of importance to a wide range of applications in science and engineering, in addition to their inherent mathematical importance. The advent of computing accelerated and extended this trend to applications in fluids, electromagnetics, queueing theory, computer graphics, and many other fields; e.g., Dias & Elcrat (1992); Driscoll (1994); Gaier (1979); Greengard (1990); Martens, Brown, & Haacke (1992); Trefethen & Williams (1986); Versnel (1982); Gu, Wang, Chan, Thompson, & Yau (2004).

Software packages for working with complex variables, particularly conformal maps, have been in the public domain for decades (Bjørstad & Grosse, 1987; Crowdy, Kropf, Green, & Nasser, 2016; Driscoll, 1996; Hough & Papamichael, 1983; Hu, 1995; Marshall & Rohde, 2007; Trefethen, 1980). There are additional papers based on research codes that apparently have not made it into the public domain, e.g. DeLillo & Elcrat (1993); Kerzman & Trummer (1986); O'Donnell & Rokhlin (1989). These packages tend to take the traditional form of library routines that require some degree of facility with custom data structures used to represent geometry and functions. Moreover, computational methods are mostly specialized to particular types or classes of regions, leading to parallel and duplicated functionality within a package, and incompatible representations between them.

The `ComplexRegions` package for Julia provides a software framework for the front end of computations over regions in the extended complex plane. It defines abstract types for curves, paths (i.e., piecewise smooth contours), and regions, defining minimal interfaces for them and providing default behavior based only on the interfaces. These are then implemented as concrete subtypes; e.g., a `Circle` is a subtype of `AbstractClosedCurve`, and a `Polygon` is a subtype of `AbstractClosedPath`. These types define data structures and methods to operate on them.

Julia's multiple dispatch facility enables some convenient uses of this basic framework. For example, the `Base.intersect` method is extended to have definitions for many possible pairings of curve and path arguments. In the future, a method for constructing conformal maps, say, could have one method for arguments of types `AbstractDisk` and `PolygonalRegion` that would call upon a Schwarz-Christoffel mapping code. The end user need not know what sort of mapping algorithm is needed for a particular problem, and the master method could easily be extended to numerous other contexts without the need for a "switchyard" portal that could become difficult to maintain.

The `ComplexRegions` package builds on the `ComplexValues` package that defines `Polar` and `Spherical` types for working with polar and Riemann sphere representations of complex numbers. It also provides recipes for plotting the major abstract types with the popular `Plots.jl` package. For example:

```
using ComplexRegions,Plots
c = Circle(0,1)
t = n_gon(3)
s = n_gon(4)
plot( ExteriorRegion([c,3+s,6+t]),leg=:none )
```
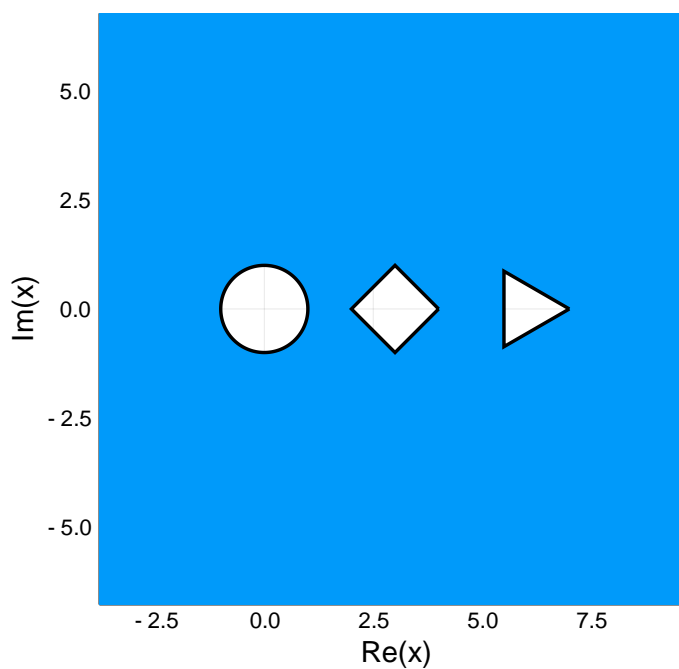

Figure 1: A multiply connected exterior region

```
julia> li = Line(1/2,1/2+1im)   # line through 0.5 and 0.5+1i
Line{Complex{Float64}} in the complex plane:
   through (0.5 + 0.0im) parallel to (0.0 + 1.0im)

julia> c = 1 / li            # a circle
Circle{Complex{Float64}} in the complex plane:
   centered at (1.0 + 0.0im) with radius 1.0, negatively oriented

julia> intersect(li,c)
   2-element Array{Complex{Float64},1}:
    0.5 + 0.8660254037844386im
    0.5 - 0.8660254037844386im

julia> plot(Spherical(li),leg=:none);  plot!(Spherical(c))
```
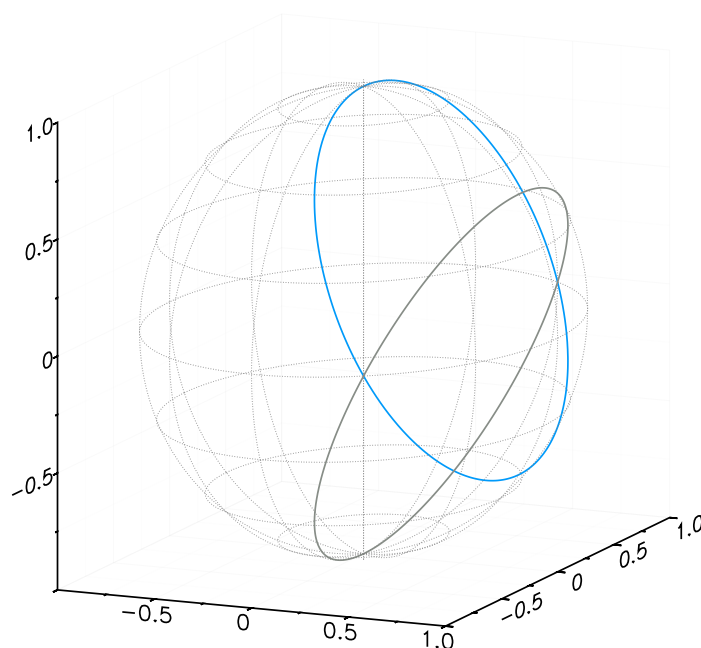
Figure 2: Line and circle on the Riemann sphere

`DomainSets` ("DomainSets.jl," 2017) and `IntervalSets` ("IntervalSets.jl," 2016) are a pair of packages with similar goals and interface, but oriented toward the representation of function domains in real spaces. Although `ComplexRegions` was not consciously patterned after them, there is enough similarity in approach to consider merging the functionality in the future.

One future use case for this package is to implement numerical methods for conformal maps. Another is for the fast solution of special cases of the Laplace equation that are especially amenable to solution using complex-variable methods (Gopal & Trefethen, 2019).

References

Bjørstad, P., & Grosse, E. (1987). Conformal mapping of circular-arc polygons. *SIAM J. Sci. Stat. Comput.*, *8*, 19–32. doi:10.1137/0908003

Crowdy, D. G., Kropf, E. H., Green, C. C., & Nasser, M. M. S. (2016). The Schottky–Klein prime function: A theoretical and computational tool for applications, *81*, 589–628. doi:10.1093/imamat/hxw028

DeLillo, T. K., & Elcrat, A. R. (1993). Numerical conformal mapping methods for exterior regions with corners. *J. Comput. Phys.*, *108*, 199–208. doi:10.1006/jcph.1993.1175

Dias, F., & Elcrat, A. R. (1992). Ideal jet flow with a stagnation streamline. *European J. Mech. B*, *11*, 23–247.

DomainSets.jl. (2017). Julia package at https://github.com/JuliaApproximation/DomainSets.jl.

Driscoll, T. A. (1994). Conformal mapping and convergence of Krylov iterations. In. Colorado conference on iterative methods.

Driscoll, T. A. (1996). A MATLAB toolbox for Schwarz–Christoffel mapping. *ACM Trans. Math. Soft.*, *22*, 168–186.

Gaier, D. (1979). Capacitance and the conformal module of quadrilaterals. *J. Math. Anal. Appl.*, *70*, 236–239. doi:10.1016/0022-247X(79)90086-6

Gopal, A., & Trefethen, L. N. (2019). Solving Laplace problems with corner singularities via rational functions. *SIAM Journal on Numerical Analysis*, *57*(5), 2074–2094. doi:10.1137/19M125947X

Greengard, L. (1990). Potential flow in channels. *SIAM J. Sci. Stat. Comput.*, *11*, 603–620. doi:10.1137/0911035

Gu, X., Wang, Y., Chan, T. F., Thompson, P. M., & Yau, S.-T. (2004). Genus zero surface conformal mapping and its application to brain surface mapping, *23*, 949–958. doi:10.1109/tmi.2004.831226

Hough, D. M., & Papamichael, N. (1983). An integral equation method for the numerical conformal mapping of interior, exterior, and doubly-connected domains. *Numer. Math.*, *41*, 287–307. doi:10.1007/bf01418327

Hu, C. (1995). *User's guide to DSCPACK* (Nat. Inst. Aviation Res. No. 95-1). Wichita State Univ.

IntervalSets.jl. (2016). Julia package at https://github.com/JuliaMath/IntervalSets.jl.

Kerzman, N., & Trummer, M. R. (1986). Numerical conformal mapping via the Szegő kernel. *J. Comput. Appl. Math.*, *14*, 125–142. doi:10.1016/0377-0427(86)90133-0

Marshall, D. E., & Rohde, S. (2007). Convergence of a variant of the Zipper algorithm for conformal mapping. *SIAM J. Numer. Anal.*, *45*, 2577–2609. doi:10.1137/060659119

Martens, M. A., Brown, R. W., & Haacke, E. M. (1992). Conformal mapping analyses of microstrips with circular and elliptic cross-sections. *IEEE Trans. Microw. Theory*, *40*, 1836–1840. doi:10.1109/22.156612

O'Donnell, S. T., & Rokhlin, V. (1989). A fast algorithm for the numerical evaluation of conformal mappings. *SIAM J. Sci. Stat. Comput.*, *10*, 475–487. doi:10.1137/0910031

Trefethen, L. N. (1980). Numerical computation of the Schwarz–Christoffel transformation. *SIAM J. Sci. Stat. Comput.*, *1*, 82–102.

Trefethen, L. N., & Williams, R. J. (1986). Conformal mapping solution of Laplace's equation on a polygon with oblique derivative boundary conditions. *J. Comput. Appl. Math.*, *14*, 227–249. doi:10.1016/0377-0427(86)90141-x

Versnel, W. (1982). The geometric correction factor for a rectangular Hall plate. *J. Appl. Phys.*, *53*, 4980–4986. doi:10.1063/1.331373