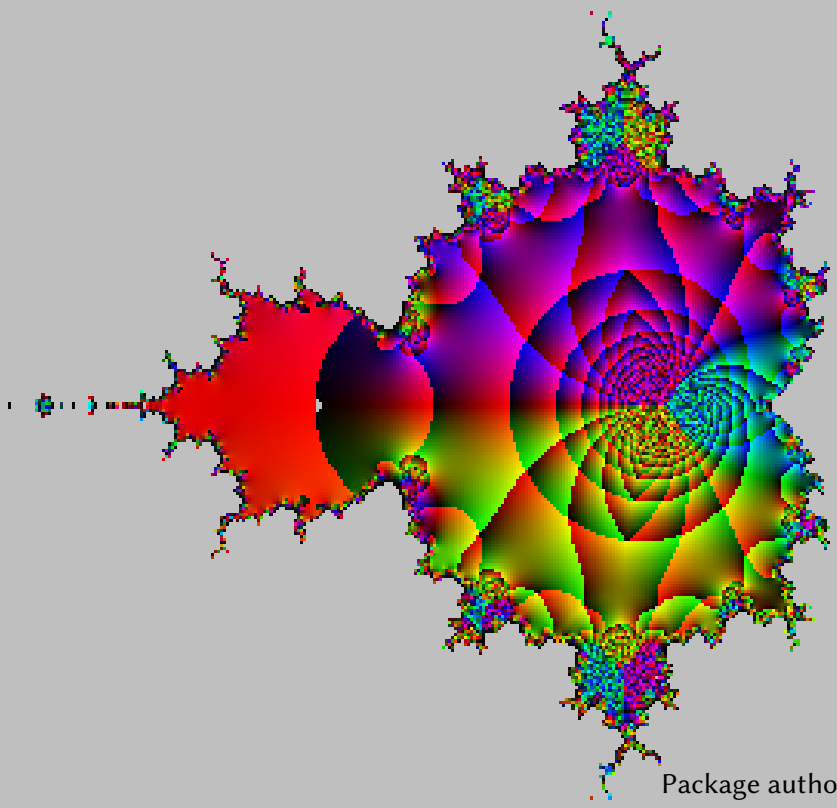
Domain Coloring of complex functions version 0.05

September 2, 2024



Package author(s):
Herbert Voß

*

Contents

1 Introduction

This package works only with `lualatex` and the option `shell-escape`! It creates an intermediate external EPS-file, which is automatically converted with `epstopdf`. The pdf is in the end imported by the macro `\includegraphics`.

1.1 Loading the package

The package `domaincoloring` creates acolored interpretation of the domain of a complex function. The package itself has no options and should be loaded as usual:

```
\usepackage{domaincoloring}
```

The package needs the following Lua modules:

- `domaincoloring.lua` the main module
- `domaincoloring-complex-numbers.lua` for complex math operations
- `domaincoloring-functions.lua` for predefined complex functions

The function module has to be managed by the user himself, if needed.

1.2 Using the macro

There is only one macro which does the external call of the Lua program `domaincoloring.lua`. This program creates the image which is then included into the document. The LaTeX-run needs the `--shell-escape` option to allow the external run of the program to convert the created eps-file into a pdf file, which is then included by the command `\includegraphics` into the document.

```
\DomainColoring[options]{complex function in Lua notation}
```

Every math function has to be preceeded by `cmath` if it has a complex argument. The following complex functions are supported:

```
complex(re,im)        creates a complex number from real and imaginary components.
re(z)                 real part of z
im(z)                 imaginary part of z
arg(z)                argument of z
abs(z)                absolute value of z
sqrt(z)               z^0.5
```

```
pow(x,y)                x^y
exp(z)                  e^z
ln(z)                   e^? = z
log(b,z)                ln(z)/ln(b)
sin(z)
cos(z)
tan(z)
sinh(z)
cosh(z)
tanh(z)
asin(z)
acos(z)
atan(z)
atan2(y,x)
asinh(z)
acosh(z)
atanh(z)
polar(z)                returns r,phi = cmath.abs(z),cmath.arg(z)
rect(r,phi)             returns a complex number from polar = r*e^(i*phi)
diff(z)                 returns re(z)^2-im(z)^2
zeta(z[,accuracy=1e4]) riemann zeta function
lintegrate(f,H,[L=0,n=sensible for H])
cx(string)              creates a complex number from a string (e.g. "1.1-i" -> 1.1+-1i)
cx(re,im)               same as complex(re,im)
```

The default trigonometrical functions can be used without a preceeding `math`:

```
e=math.exp(1)
pi=math.pi
abs=math.abs
exp=math.exp
log=math.log
cos=math.cos
sin=math.sin
cosh=math.cosh
sinh=math.sinh
sqrt=math.sqrt
atan2=math.atan2
```

1.3 Options

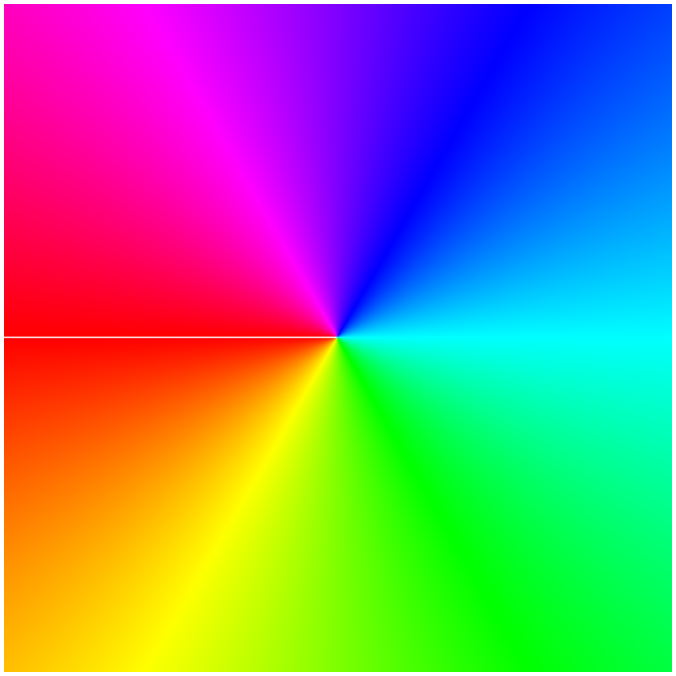
name	*value*	*meaning*
`domain`	-2,2,-2,2	the (re,im)-coordinates for the complex system
`resolution`	500	the number of steps for the re,im interval. One value will be for both axes. Two values like `{500,600}` for real axis and imaginary axis,
`Rmax`	0	forces a circle output if Rmax > 0
`funcName`	{}	corresponding to external file
`grfOptions`	scale=0.5	optional arguments for `\includegraphics`

hsvrgb	phi+pi,1,r	for the conversion into rgb
bgcolor	{0,0,0}	change color to value as background
invers	false	inverted colors with $color = \|color - 255\|$
force	true	With force=false an existing pdf file will be used without calculating a new one.
grid	false	draw a grid with one dashed subgrid at 0.5

2 Examples

2.1 The default with function $f(z) = z$ and $f(z) = 1/z$

```
\DomainColoring{z}   % default filename \jobname-tmp0.png
```



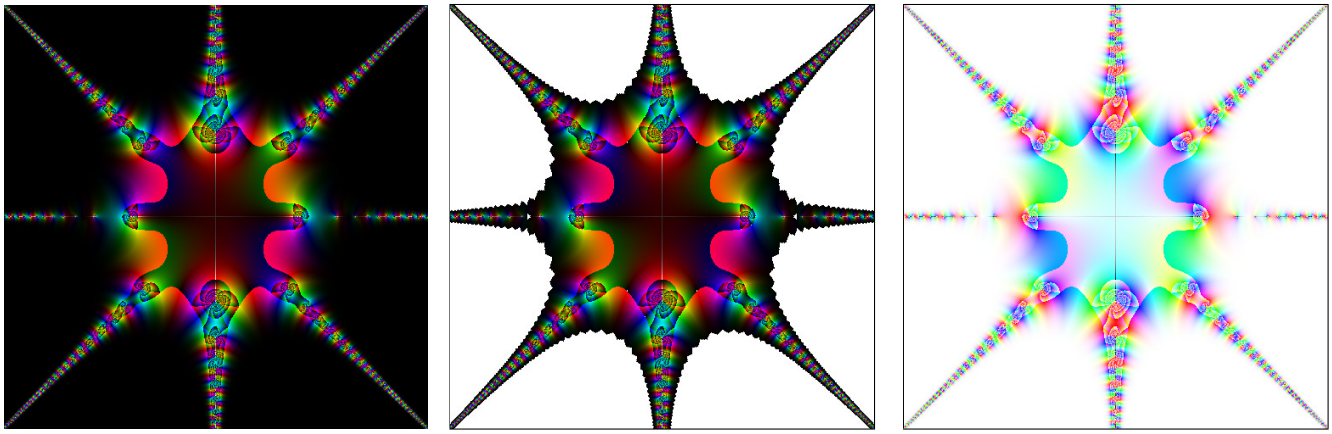
2.2 Defining domain, color mode, resolution and Rmax

$$f(z) = \cos(z)/\sin(z^4 - 1) \tag{1}$$

in Lua-notation: `cmath.cos(z)/cmath.sin(z^4-1)`. All complex functions must be preceeded by `cmath.`. For real functions the prefix `math.` can be omitted.

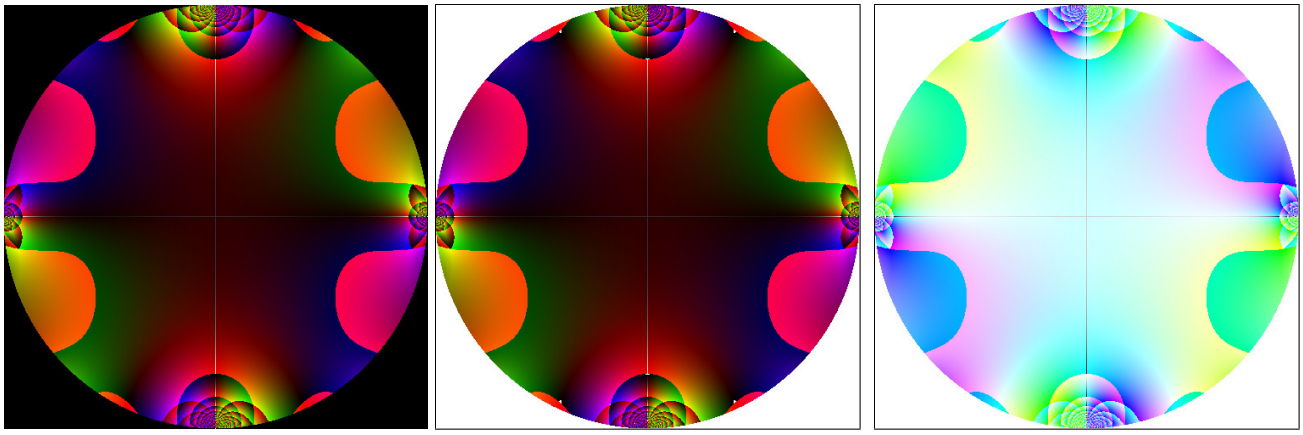
The backgroundcolor can be set with `bgcolor={R,G,B}` (values between 0 and 255). This color will replace the default backgroundcolor `{0,0,0}`. With a negative value for R, eg. -10, it replaces all colors which have a sum of $R+G+B < 10$ with the color defined by the value for G, eg. 255. A setting of `bgcolor{=-8,255,255}` is the same as `bgcolor{=-8,255,0}`, becaus the last value is not used. A given color `{=3,3,1}` will be replaced by `{=255,255,255}`, because $3 + 3 + 1 < 8$ and a color `{=6,2,3}` will be unchanged, it is greater than 8.

```
\DomainColoring[domain={-2.5,2.5,-2.5,2.5},resolution=500,hsvrgb={phi,1,r},
grfOptions={width=0.32\linewidth}]{cmath.cos(z)/cmath.sin(z^4-1)}
\hfill
\frame{\DomainColoring[domain={-2.5,2.5,-2.5,2.5},resolution=500,bgcolor={-8,255,255},
  hsvrgb={phi,1,r},grfOptions={width=0.32\linewidth}]{cmath.cos(z)/cmath.sin(z^4-1)}}
\hfill
\frame{\DomainColoring[domain={-2.5,2.5,-2.5,2.5},resolution=500,invers,
   hsvrgb={phi,1,r}, grfOptions={width=0.32\linewidth}]{cmath.cos(z)/cmath.sin(z^4-1)}}
```



The optional argument Rmax allows to crop everything around the circle with radius Rmax:

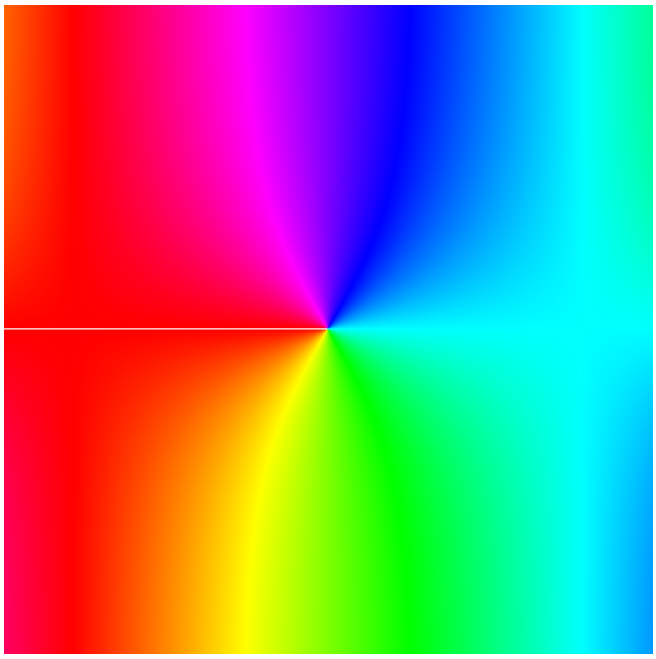
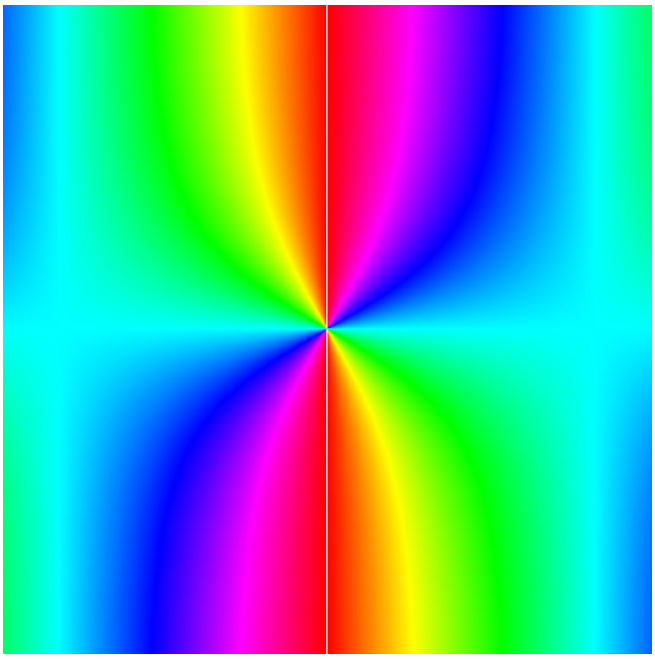
```
\DomainColoring[domain={-1,1,-1,1},resolution=500,hsvrgb={phi,1,r},
grfOptions={width=0.32\linewidth}, Rmax=1, bgcolor={1,1,1}]{cmath.cos(z)/cmath.sin(z^4-1)}
\hfill
\frame{\DomainColoring[domain={-1,1,-1,1},resolution=500,bgcolor={-8,255,1},hsvrgb={phi,1,r},
grfOptions={width=0.32\linewidth},Rmax=1]{cmath.cos(z)/cmath.sin(z^4-1)}}
\hfill
\frame{\DomainColoring[domain={-1,1,-1,1},resolution=500,invers=true,hsvrgb={phi,1,r},
grfOptions={width=0.32\linewidth},Rmax=1]{cmath.cos(z)/cmath.sin(z^4-1)}}
```



2.3 Option for \includegraphics

With grfOptions one can define optional arguments for \includegraphics:

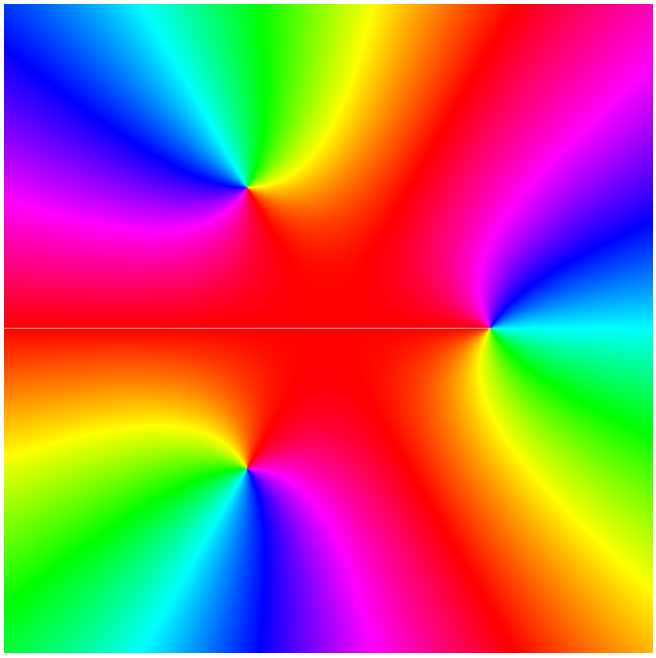
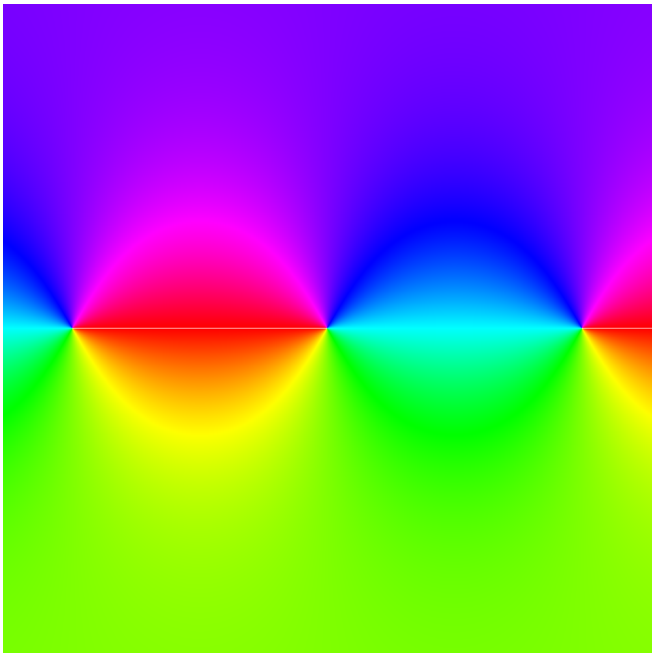
```
\DomainColoring[grfOptions={width=0.49\linewidth}]{cmath.sin(z)}\hfill
\hfill
\DomainColoring[grfOptions={width=0.49\linewidth}]{cmath.sin(0.9*z)*cmath.sin(z)}
```

2.4 Higher resolution

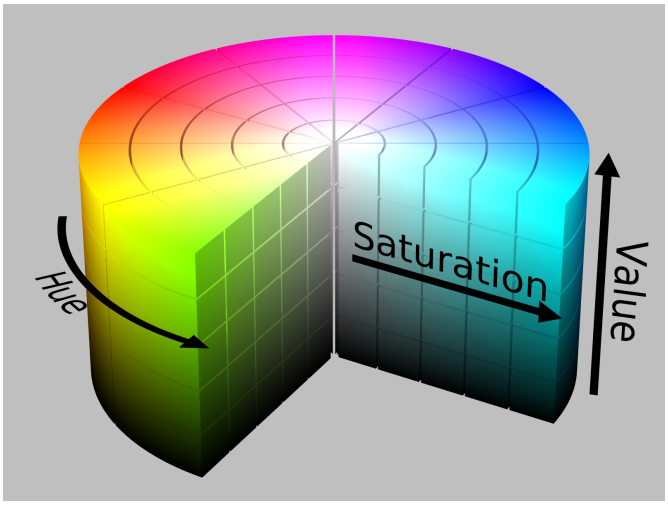
The resolution is more or less the number of pixels for the given domain. It is possible to have different values for the two coordinates. Is only one value for `resolution` given, then it is for both axes.

```
\DomainColoring[resolution=1000,grfOptions={width=0.49\linewidth}]{z^3-1}
\hfill
\DomainColoring[resolution=1000,
  grfOptions={width=0.49\linewidth}]{(z+1)^2*(z-1)/((z+i)*(z-i)^2)}
```

2.5 hsv to rgb conversion

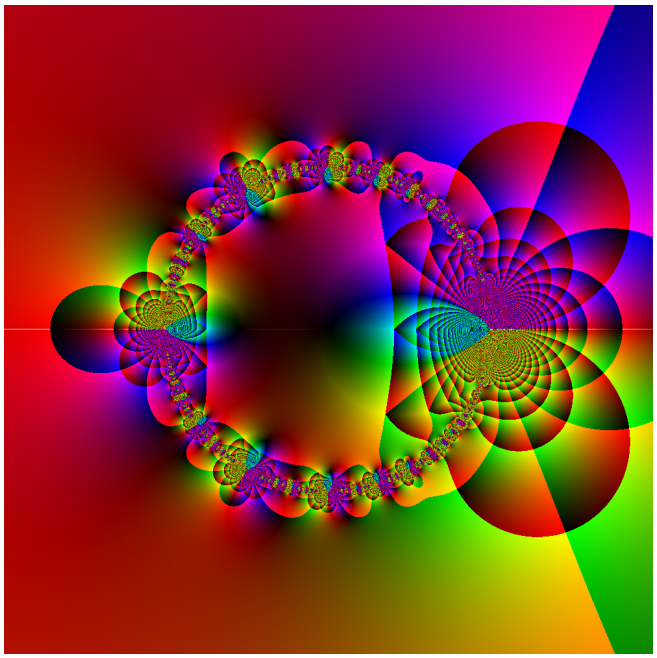
The color model (Wikipedia):



http://en.wikipedia.org/wiki/File:HSV_color_solid_cylinder_alpha_lowgamma.png

The complex number $z = x+iy$ is converted into its trogonometrical representation $x = r\cdot\cos\phi$ and $y = r\cdot\sin\phi$ with $r = \sqrt{x^2+y^2}$. The values r and ϕ are now taken as values for the hsv color model with a constant second value for saturation. ϕ is used for hue. For example: `hsvrgb=phi,1,r`, which gives

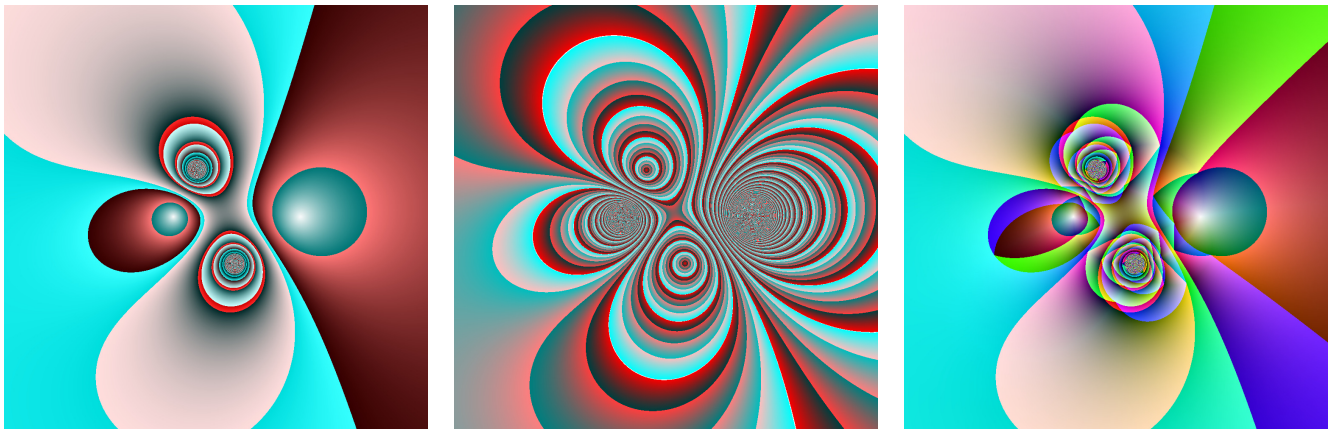
```
\DomainColoring[resolution=1000,grfOptions={width=0.49\linewidth},domain={-2,2,-2,2},
   funcName=f10,hsvrgb={phi,1,r}]{}
\hfill
\DomainColoring[resolution=1000,hsvrgb={phi,1,r^2/(1+r^2)},Rmax=2,domain={-2,2,-2,2},
   grfOptions={width=0.49\linewidth}]{cmath.sin(z)*cmath.sin(0.99*z)}
```



The optional argument `hsvrgb` must define three values which can use the arguments phi and r in any mathematical combination. It must only be compatible to the Lua math conventions, e.g. `hsvrgb={phi+2,0.5,2/r}`

```
\DomainColoring[resolution={1000,1000},
   domain={-5,5,-5,5}, grfOptions={width=0.32\linewidth},
   hsvrgb={0.5,r/5,5-r},funcName=f16]{}
\hfill
\DomainColoring[resolution={1000,1000},
   domain={-5,5,-5,5}, grfOptions={width=0.32\linewidth},
```

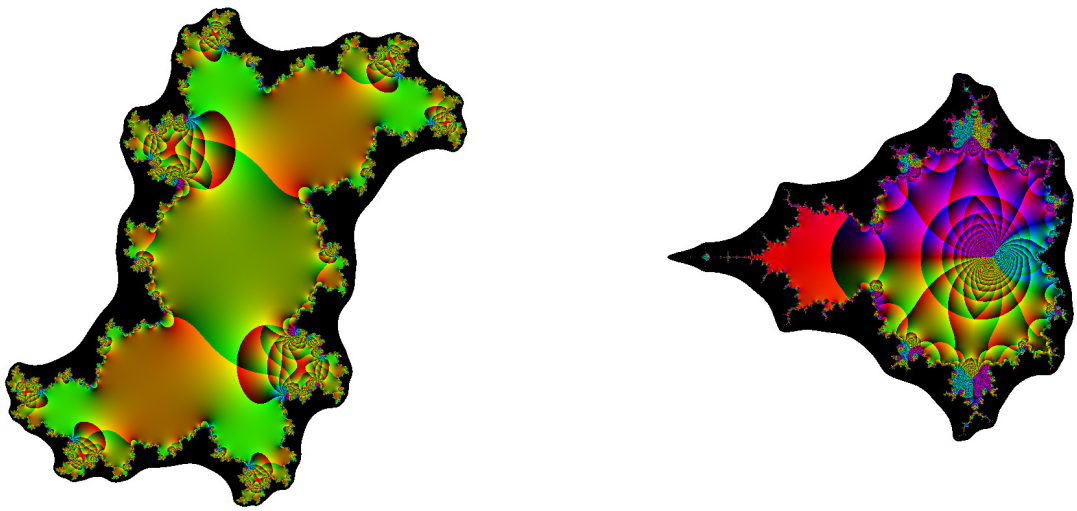
```
   hsvrgb={0.5,5-r,5/r},funcName=f16]{}
\hfill
\DomainColoring[resolution={1000,1000},
   domain={-5,5,-5,5}, grfOptions={width=0.32\linewidth},
   hsvrgb={phi,r/5,5-r},funcName=f16]{}
```



2.6 External function definition

The already existing file `domaincoloring-functions.lua` collects sone definitions of complex functions $f(z)$, which can be used from inside LaTeX with the optional argument `funcName<Lua function name>`. In this case the mandatory argument of `\DomainColoring` has no meaning and can be empty.

```
\DomainColoring[domain={-1.5,1.5,-1.5,1.5},resolution={1001,1001},hsvrgb={phi,1,1/r},
  grfOptions={width=0.49\linewidth},funcName=f12,bgcolor={1,1,1}]{}
\hfill
\DomainColoring[domain={-2.5,1.5,-2,2},resolution={1001,1001},hsvrgb={phi,1,1/r},
  grfOptions={width=0.49\linewidth},funcName=f13,bgcolor={1,1,1}]{}
```



The contents of the function file of the current version of `domaincoloring` is::

```
-- $Id: domaincoloring-functions.lua 978 2024-09-02 15:27:30Z herbert $

kpse.set_program_name("luatex")
```

```
function f0(z)
   return cmath.sin(1/z)*cmath.cos(1/z)/z^3+1/z
end

function f1(z)
    return cmath.cos(z)/cmath.sin(z^4-1)
end

function f2(z)
  local c = complex(1,-1)
  local d = complex(0,0.28)
  return cmath.cos(c^2*z^2)/cmath.cos(c*(z-d))
end

function f3(z)
  return z*(z+i)^2/(z-i)^2
end

function f4(z)
  if abs(z) < 0.1 then
    return complex(0.001,0.001)
  else
    return cmath.sin(1/(z*z))
  end
end

function f5(z)
  return cmath.sqrt(1-1/(z*z)+z^3)
end

function f9(z)
  local c = complex(1,-1)
  local d = complex(1,1)
  return z^2*c^2*(z*c-1-i)/(z*c-2*d)
end

function f10(z)
  local sum = complex(0,0)
  for n=1,20 do
      sum = sum + z^n/(1-z^n)
  end
  return sum
end

function f11(z)
  local iterateNo = 3
  for n=1,iterateNo do
    z = z^2
  end
  return z
end

function f12(z)   -- julia
  local iterateNo = 15
  for n=1,iterateNo do
    z = z^2 + complex(0.25,-0.5)
```

```
  end
  return z
end

function f13(z)   -- mandelbrot
  local iterateNo = 15
  local c = z
  z = complex(0,0)
  for n=1,iterateNo do
    z = z^2 + c
  end
  return z
end

function f14(z)
  local iterateNo = 5
--  local c = z
--  z = complex(0,0)
  for n=1,iterateNo do
    z = cmath.sin(z)*cmath.sin(0.8*z) -- + c
  end
  return z
end

function f15(z)
  local alpha = 4
  local C0 = complex(1,0)
  local C1 = 2 * alpha * z
  for n = 2,20 do
    C = (2*z*(n+alpha-1)*C1 - (n+2*alpha-2)*C0)/n
    C0 = C1
    C1 = C
  end
  return C
end

function f16(z)
   local A = z^2 - z -2
   local B = z^2 + complex(1,1)
   return A/B
end

function f17(z)
  return e^(z^cmath.sin(z^cmath.tan(z^cmath.cos(z))))
end

function f18(z)
  return (z+0.5)*(z-0.5)/z
end
```

References

[1] Juan Carlos Ponce Campuzano. *Dynamic Mathematics. Domain Coloring – Visualizing Complex Functions.* July 15, 2018. URL: https://www.dynamicmath.xyz/domain-coloring/ (visited on 08/23/2024).

[2] Konstantin Poelke and Konrad Polthier. *Domain Coloring of Complex Functions*. Aug. 18, 2024. URL: https://www.mi.fu-berlin.de/en/math/groups/ag-geom/publications/db/ieee_article_old_low_v3_1.pdf (visited on 08/18/2024).

[3] vismath. *Thema Domain Coloring*. Aug. 18, 2024. URL: https://www.vismath.eu/de/blog/domain-coloring/ (visited on 08/18/2024).

[4] WikipediA. *DOmain Coloring*. Aug. 18, 2024. URL: https://en.wikipedia.org/wiki/Domain_coloring (visited on 08/18/2024).

Index