

Fonctions d'une variable complexe (II)

Exponentielle, logarithmes, puissances

Marc Lorenzi

7 avril 2019

```
Entrée [1]: import matplotlib.pyplot as plt
import cmath
import math
import colorsys
import random
```

```
Entrée [2]: plt.rcParams['figure.figsize'] = (10, 10)
```

Avertissement : Avant de lire ce notebook, regardez celui intitulé Fonctions d'une variable complexe (I).

Nous allons poursuivre ce que nous avons commencé dans le notebook précédent et nous intéresser aux fonctions usuelles bien connues sur \mathbb{R} : exponentielle, logarithme, puissances. Dans un troisième notebook, nous explorerons la trigonométrie.

Voici tout d'abord la fonction de tracé introduite dans le premier notebook. Je ne la ré-expliquerai pas.

```
Entrée [3]: def arg01(z):
    theta = cmath.phase(z) / (2 * math.pi)
    if theta < 0: return theta + 1
    else: return theta
```

```
Entrée [4]: def psi(x, m, d):
    if x == 0: return 0
    else:
        return abs(math.sin(math.pi * math.log(x, 2) / m)) ** d
```

```
Entrée [5]: def plot_complex(f, extent=[-1.5, 1.5, -1.5, 1.5], m=1, d=0.2, nx=301
xmin, xmax, ymin, ymax = extent
ws = [[0 for i in range(nx)] for j in range(ny)]
plt.hsv()
for i in range(nx):
    for j in range(ny):
        x = xmin + j * (xmax - xmin) / nx
        y = ymax - i * (ymax - ymin) / ny
        z = x + 1j * y
        try:
            w = f(z)
            r = psi(abs(w), m, d)
            ph = arg01(w)
            ws[i][j] = colorsys.hsv_to_rgb(ph, 1, r)
        except:
            ws[i][j] = (0., 0., 0.)
plt.imshow(ws, interpolation='bicubic', extent=extent, aspect='au
plt.grid()
```

1. L'exponentielle

1.1 C'est quoi ?

La fonction exponentielle est la mère de toutes les fonctions usuelles. Voici sa définition.

Définition : Pour tout $z \in \mathbb{C}$,

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!}$$

Lorsqu'on maîtrise la théorie des séries entières, il est alors assez facile de prouver les résultats suivants.

Proposition : Pour tous $z, z' \in \mathbb{C}$, $e^{z+z'} = e^z e^{z'}$.

Proposition : La fonction exponentielle est holomorphe sur \mathbb{C} , et $\exp' = \exp$.

Rappelons que pour $z = x + iy \in \mathbb{C}$, on a $e^z = e^x(\cos y + i \sin y)$, où pour l'instant, nous appelons \sin et \cos les fonctions de \mathbb{R} vers \mathbb{R} dont vous avez sans doute déjà entendu parler. Nous reviendrons sur ces fonctions dans le prochain notebook.

1.2 Un algorithme de calcul de l'exponentielle

Nous allons examiner dans cette section comment calculer une valeur approchée de e^z à partir de la série ci-dessus. Le terme général de la série qui définit l'exponentielle est $\frac{z^n}{n!}$. La bonne nouvelle est le dénominateur $n!$ qui tend très vite vers l'infini. Le numérateur z^n , en revanche, est une fort mauvaise nouvelle si $|z|$ est grand.

1.2.1 Majorons le reste d'une série convergente

Commençons par établir une inégalité.

Proposition : Soit $n \in \mathbb{N}^*$. On a

$$\sum_{k=n+1}^{\infty} \frac{1}{k!} \leq \frac{1}{n!n}$$

Démonstration : Soit $k \geq n + 1$. On a

$$\frac{1}{k!} = \frac{1}{(n+1)!} \frac{1}{(n+2)(n+3)\dots k} \leq \frac{1}{(n+1)!} \frac{1}{(n+1)^{k-n-1}}$$

De là,

$$\sum_{k=n+1}^{\infty} \frac{1}{k!} \leq \frac{1}{(n+1)!} \sum_{k=n+1}^{\infty} \frac{1}{(n+1)^{k-n-1}} = \frac{1}{(n+1)!} \sum_{k=0}^{\infty} \frac{1}{(n+1)^k}$$

La série ci-dessus est une série géométrique convergente, dont la somme est

$$\sum_{k=0}^{\infty} \frac{1}{(n+1)^k} = \frac{1}{1 - \frac{1}{n+1}} = \frac{n+1}{n}$$

D'où le résultat.

1.2.2 Le cas $|z| \leq 1$

Soit maintenant $z \in \mathbb{C}$ tel que $|z| \leq 1$. Pour tout entier $k \geq n + 1$, on a $|z^k| \leq |z|^{n+1}$. Ainsi,

$$\left| \sum_{k=n+1}^{\infty} \frac{z^k}{k!} \right| \leq \frac{|z|^{n+1}}{n!n}$$

Pour calculer e^z à la précision 10^{-p} il suffit donc de calculer $\sum_{k=0}^n \frac{z^k}{k!}$ où l'entier n est tel que

$$\left| \frac{1}{z} \right|^{n+1} n!n \geq 10^p$$

Prenons par exemple $|z| \leq \frac{1}{8}$. Quel entier n choisir ? Allons, un peu de Python. Voici tout d'abord la fonction factorielle.

```
Entrée [6]: def fact(n):  
             p = 1  
             for k in range(1, n + 1): p = p * k  
             return p
```

Puis une petite boucle.

Entrée [7]:

```
n = 0
while 8 ** n * n * fact(n) <= 1e17: n = n + 1
print(n)
```

11

Si $|z| \leq \frac{1}{8}$, $n = 11$ est donc suffisant pour obtenir une précision de 10^{-17} . Sachant que les flottants Python ont moins de 17 chiffres significatifs, cela nous suffira plus qu'abondamment.

1.2.3 Le cas général

Que faire si nous ne savons rien de $|z|$? Ce qui nous sauve c'est que la fonction exponentielle est un morphisme. On a pour tout entier p

$$\exp\left(\frac{z}{2^p}\right)^{2^p} = e^z$$

L'idée est de diviser z par 2, puis par 2, puis par 2, etc. jusqu'à ce que le module de $\frac{z}{2^p}$ soit assez petit, disons inférieur à $\dots \frac{1}{8}$. On se retrouve alors dans le cas précédent. On calcule une valeur approchée de

$$\exp\left(\frac{z}{2^p}\right)$$

puis on élève cette valeur à la puissance 2^p . Pour cela, on remarque que p élévations au carré suffisent :

$$x^{2^p} = (\dots ((x^2)^2)^2 \dots)^2$$

Écrivons donc notre fonction exponentielle et comparons la à la fonction `cmath.exp` de Python !

1.2.4 La fonction

```

Entrée [8]: def exp(z):
            t = 0
            while abs(z) > 1/8:
                z = z / 2
                t = t + 1
            n = 12
            s = 0
            f = 1
            p = 1
            for k in range(1, n + 1):
                s = s + p / f
                f = f * k
                p = p * z
            for k in range(t): s = s ** 2
            return s

```

```

Entrée [9]: z = 1.2345 + 2.245 * 1j
            print(exp(z))
            print(cmath.exp(z))
            print('différence : ', abs(exp(z) - cmath.exp(z)))

```

```

(-2.1454222185312273+2.6847334724848904j)
(-2.1454222185312273+2.6847334724848886j)
différence : 1.7763568394002505e-15

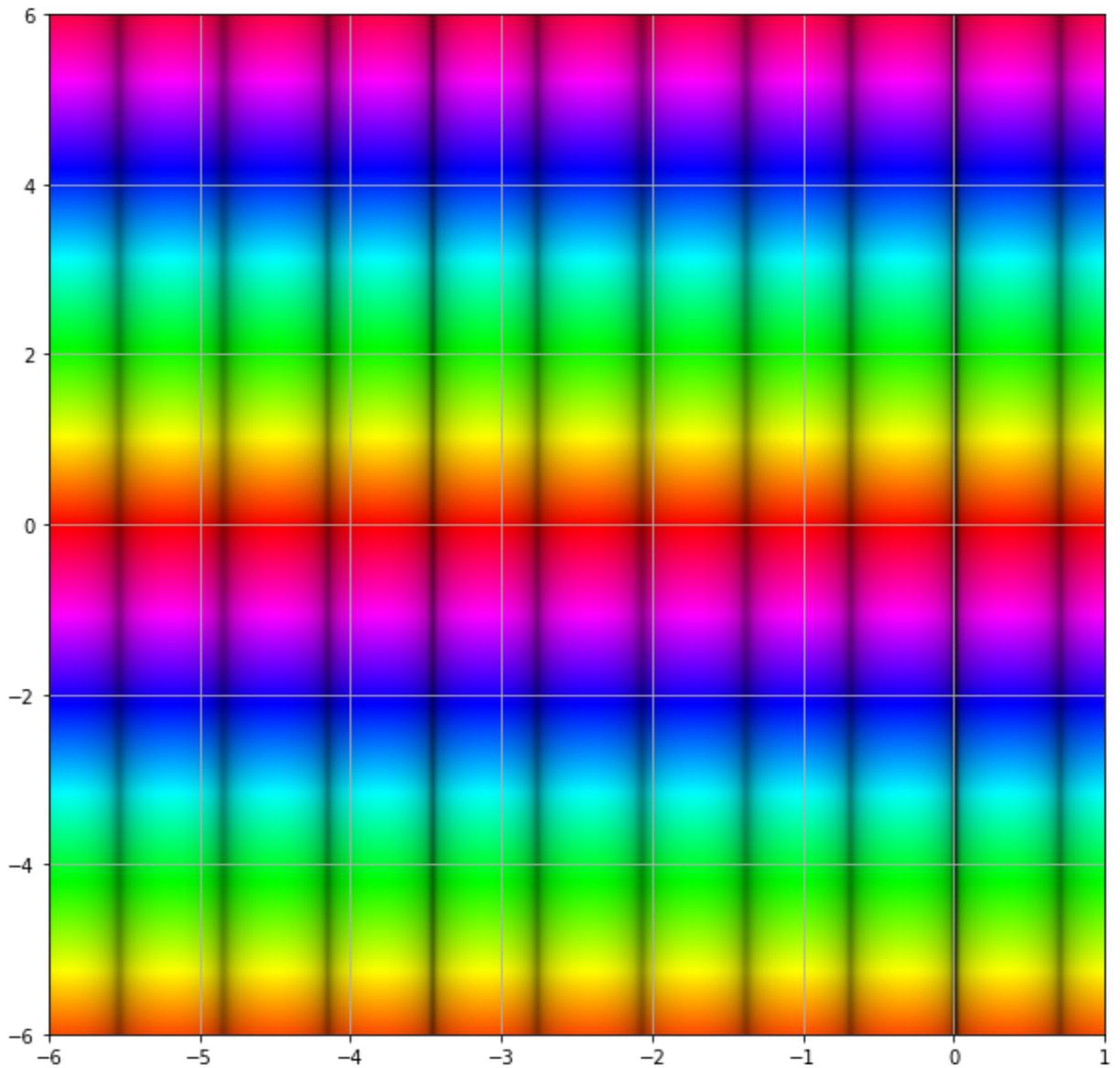
```

Nous espérons 10^{-17} mais il ne faut pas rêver. Il est normal que les derniers chiffres puissent différer entre les deux fonctions, ils sont perturbés par les erreurs d'arrondi dans les calculs. Seuls des calculs avec une plus grande précision pourraient régler ce problème (ce que nous ne ferons pas).

1.3 Le graphe

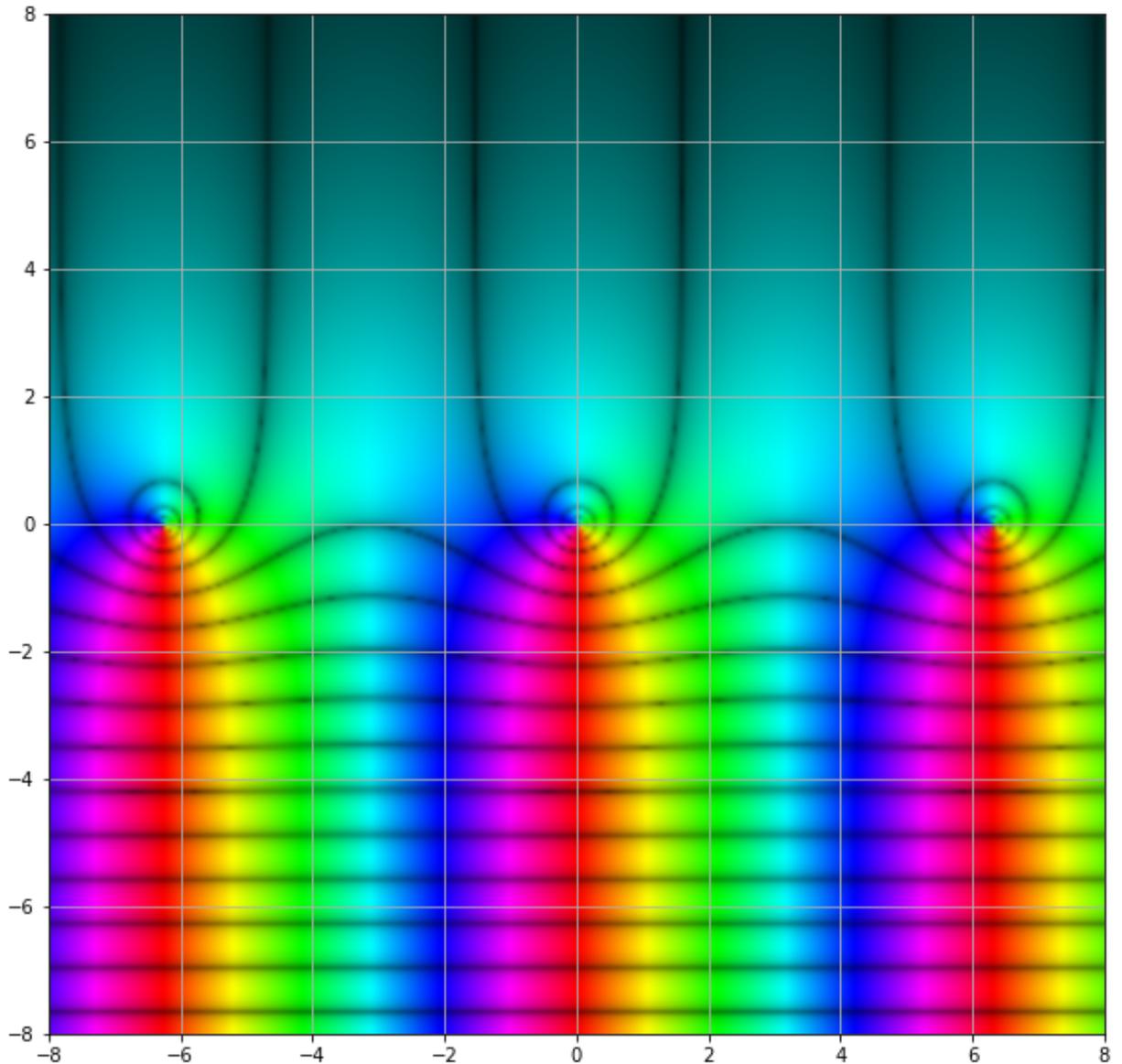
À quoi ressemble graphiquement l'exponentielle complexe ? Remarquons que $|e^z| = e^x$ et $\arg e^z \equiv y[2\pi]$. Nous voyons donc que le module de e^z ne dépend que de x , et son argument ne dépend que de y . Le graphe de l'exponentielle va donc être une suite de bandes verticales, et ces bandes auront des couleurs périodiques.

```
Entrée [10]: plot_complex(lambda z: cmath.exp(z), extent=[-6, 1, -6, 6])
```



Ce serait presque décevant :-). Tentons maintenant $z \mapsto e^{iz} - 1$. Cette fonction a une infinité de racines, qui sont les multiples de 2π .

```
Entrée [11]: M = 8  
plot_complex(lambda z: cmath.exp(1j * z) - 1, extent=[-M, M, -M, M])
```



Remarquez les cycles de couleurs autour des racines dans le sens trigonométrique.
Pourquoi le haut du graphique est-il tout noir ? Eh bien, posons $z = x + iy$. On a $e^{iz} = e^{ix} e^{-y}$ et donc $|e^{iz}| = e^{-y}$. Lorsque y tend vers $+\infty$, ce module tend donc vers 0. Conséquence : $e^{iz} - 1$ tend vers -1 et notre convention de couleurs affiche des pixels cyans de plus en plus sombres.

Passons au logarithme !

2. Le logarithme

2.1 Introduction

Le module `cmath` possède une fonction `log`. Mais qu'est-ce que $\log z$ pour z complexe ?

Définition : Soit $z \in \mathbb{C}^*$. **UN** logarithme de z est un nombre complexe w tel que $e^w = z$.

Nous avons supposé $z \neq 0$ parce que $e^w = 0$ est une équation perdue d'avance. Posons donc $z = re^{it}$, où $r = |z|$ et $t = \arg z$ est un argument de z . Soit $w = x + iy$. On a

$$e^w = z \Leftrightarrow e^x e^{iy} = re^{it}$$

Ce qui équivaut à $x = \ln |z|$ et $y = t + 2k\pi$ où $k \in \mathbb{Z}$. Ainsi, les logarithmes de z sont les nombres complexes de la forme

$$w = \ln |z| + i \arg z + 2ik\pi$$

où $\arg z$ est un argument de z et $k \in \mathbb{Z}$.

Voici la documentation du module `cmath` :

```
cmath.log(x[, base])
```

Returns the logarithm of `x` to the given base. If the base is not specified, returns the natural logarithm of `x`. There is one branch cut, from 0 along the negative real axis to $-\infty$, continuous from above.

Traduction : pour Python,

$$\log z = \ln |z| + i \arg z$$

où $\arg z$ est l'unique argument de z appartenant à $]-\pi, \pi]$. C'est ce que l'on appelle la **détermination principale du logarithme**.

Proposition : \log est holomorphe sur $\Omega = \mathbb{C} \setminus \mathbb{R}_-$. Et pour tout $z \in \Omega$, $\log' z = \frac{1}{z}$.

Démonstration : Nous admettrons ce résultat.

La fonction \log , bien que définie sur \mathbb{C}^* pose des problèmes pour les réels négatifs. Pour voir quel genre de problèmes, le mieux à faire est de la tracer. Mais avant, intéressons nous à la façon de calculer une valeur approchée de $\log z$. En cas d'impatience, sautez directement au paragraphe 3.3.

2.2 Un algorithme de calcul du logarithme

Proposition : Soit $z \in \mathbb{C}$ tel que $|z| < 1$. On a

$$\log(1 + z) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{z^n}{n}$$

$\log(1 + z)$ est donc la somme d'une série convergente. Mais remarquons que cette série converge moins rapidement que la série qui converge vers e^z .

2.2.1 Le cas réel

Soit $x \in \mathbb{R}_+^*$. Si $x < 1$, alors $\frac{1}{x} > 1$ et $\ln \frac{1}{x} = -\ln x$. Supposons donc dorénavant que $x \geq 1$.

Proposition : Il existe un unique entier naturel t tel que $\frac{1}{2} < \frac{x}{2^t} \leq 1$.

Démonstration : L'inégalité souhaitée équivaut à

$$2^{t-1} < x \leq 2^t$$

L'existence d'un tel entier t en découle. Précisément,

$$t = 1 + \lceil \log_2 x \rceil$$

Si nous voulons calculer $\ln x$, il nous suffit de savoir calculer $\ln \frac{x}{2^t}$, puisque

$$\ln x = \ln \frac{x}{2^t} + t \ln 2$$

Il faut évidemment aussi connaître une valeur approchée de $\ln 2$. On peut donc se ramener au calcul de $\ln x$ pour $\frac{1}{2} < x \leq 1$. Ce que nous supposons maintenant.

Soit $w = \frac{x-1}{1+x}$. On a $\frac{1+w}{1-w} = x$ et on vérifie aisément que $-\frac{1}{3} < w \leq 0$. De là

$$\ln x = \ln(1+w) - \ln(1-w) = \sum_{k=1}^{\infty} (-1)^{k-1} \frac{w^k}{k} + \sum_{k=1}^{\infty} \frac{w^k}{k}$$

ou encore

$$\ln x = \sum_{k=1}^{\infty} ((-1)^{k-1} + 1) \frac{w^k}{k} = 2 \sum_{k=0}^{\infty} \frac{w^{2p+1}}{2p+1}$$

On peut donc approcher $\ln x$ par une somme partielle de la série ci-dessus, quelque chose du genre

$$\sum_{k=0}^n \frac{w^{2p+1}}{2p+1}$$

Que prendre pour n ? Cette série est une série alternée, car $w \leq 0$, et on vérifie aisément que son terme général décroît en valeur absolue vers 0. Le reste des sommes partielles est donc majoré en valeur absolue par son premier terme

$$\frac{|w^{2n+3}|}{2n+3} \leq \frac{1}{3^{2n+3}(2n+3)}$$

Alors, combien pour n ? Mettons que nous voulions une erreur inférieure à 10^{-17} :

```
Entrée [12]: n = 0
while 3 ** (2 * n + 3) * (2 * n + 3) <= 10 ** 17: n = n + 1
print(n)
```

15

$n = 15$ sera parfait. Allons, programmons !

```
Entrée [13]: def ln(x):
ln2 = 0.6931471805599453
if x < 1: return - ln(1 / z)
else:
t = 0
while x >= 1:
x = x / 2
t = t + 1
w = (x - 1) / (1 + x)
w2 = w ** 2
sgn = 1
n = 15
s = 0
for k in range(n):
s = s + w / (2 * k + 1)
w = w * w2
return 2 * s + t * ln2
```

```
Entrée [14]: x = 123.456
print(ln(x))
print(math.log(x))
print(ln(x)-math.log(x))
```

4.815884817283264

4.815884817283264

0.0

2.2.2 Le cas complexe

Soit $z \in \mathbb{C}^*$. On a $\log z = \ln |z| + i \arg z$ où $\arg z \in]-\pi, \pi]$. Il est donc immédiat d'obtenir une valeur approchée de la partie réelle de $\log z$, c'est ce que nous avons fait dans le paragraphe précédent. Que faire pour la partie imaginaire ?

Proposition : Soit $z \in \mathbb{C} \setminus \mathbb{R}_-$. On a

$$\arg z = 2 \arctan \frac{\Im z}{\Re z + |z|}$$

Démonstration : Posons $z = x + iy$. Soit $\theta = \arg z$. On a

$$\cos^2 \frac{\theta}{2} = \frac{1}{2}(1 + \cos \theta) = \frac{1}{2}\left(1 + \frac{x}{|z|}\right) = \frac{|z| + x}{2}$$

et de même

$$\sin^2 \frac{\theta}{2} = \frac{1}{2}(1 - \cos \theta) = \frac{1}{2}\left(1 - \frac{x}{|z|}\right) = \frac{|z| - x}{2}$$

Comme $z \notin \mathbb{R}_-$, on a $|z| + x \neq 0$. De là,

$$\tan^2 \frac{\theta}{2} = \frac{\sin^2 \frac{\theta}{2}}{\cos^2 \frac{\theta}{2}} = \frac{|z| - x}{|z| + x} = \frac{(|z| - x)(|z| + x)}{(|z| + x)^2} = \frac{y^2}{(|z| + x)^2}$$

D'où

$$\tan \frac{\theta}{2} = \pm \frac{y}{|z| + x}$$

Comme $\theta \in]-\pi, \pi[$, le signe de θ est celui de y . Le \pm ci-dessus est donc un plus. Pour terminer, il reste à remarquer que $-\frac{\pi}{2} < \frac{\theta}{2} < \frac{\pi}{2}$, on peut donc passer à l'arc tangente dans l'égalité ci-dessus pour obtenir

$$\frac{\theta}{2} = \arctan \frac{y}{|z| + x}$$

Voici donc ci dessous une fonction renvoyant l'argument d'un nombre complexe z , à supposer que l'on dispose d'une fonction arc tangente !

```
Entrée [15]: def argument(z):  
    if z.real == 0 and z.imag < 0: return math.pi  
    else:  
        return 2 * arctan(z.imag / (z.real + abs(z)))
```

La fonction `log` est alors immédiate.

```
Entrée [16]: def log(z):  
    return ln(abs(z)) + 1j * argument(z)
```

Il nous reste à écrire la fonction arc tangente.

Élaguons un peu. Soit $x \in \mathbb{R}$.

- Si $x < 0$, alors $\arctan x = -\arctan(-x)$.
- Si $x > 1$ alors $\arctan x = \frac{\pi}{2} - \arctan \frac{1}{x}$ (et $0 < \frac{1}{x} < 1$).

Il nous suffit donc de savoir calculer $\arctan x$ pour $0 \leq x \leq 1$.

Proposition : Pour tout $x \in]-1, 1]$,

$$\arctan x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}$$

Ce que nous avons raconté sur le logarithme réel reste valable. Si $x \in [0, 1]$ et que nous approchons $\arctan x$ par la somme partielle

$$\sum_{k=0}^n (-1)^k \frac{x^{2k+1}}{2k+1}$$

nous obtenons une erreur majorée en valeur absolue par $\frac{x^{2n+3}}{2n+3}$. Si x est proche de 1, ou, pire, égal à 1, cette erreur est inacceptable. Vite, une solution !

Proposition : Soit $x \in \mathbb{R}$. On a

$$\arctan x = 2 \arctan \frac{x}{1 + \sqrt{1+x^2}}$$

Démonstration : Pour $0 \leq t < 1$ on a (exercice)

$$2 \arctan t = \arctan \frac{2t}{1-t^2}$$

Soit $t = \frac{x}{1+\sqrt{1+x^2}}$. On a (exercice aussi)

$$\frac{2t}{1-t^2} = x$$

d'où le résultat.

Remarquons que si $0 \leq x \leq 1$, on a $0 \leq t \leq \frac{x}{2} \leq \frac{1}{2}$. Et si l'on pose $u = \frac{t}{1+\sqrt{1+t^2}}$, alors, de même, $0 \leq u \leq \frac{1}{4}$, et $\arctan x = 2 \arctan t = 4 \arctan u$. Avec les notations ci-dessus, quelle valeur de n doit-on prendre pour calculer $\arctan u$ avec une erreur inférieure à 10^{-17} ?

Entrée [17]:

```
n = 0
while 4 ** (2 * n + 3) * (2 * n + 3) <= 10 ** 17: n = n + 1
print(n)
```

12 termes suffiront. Voici donc la fonction `arctan` .

```
Entrée [18]: def arctan(x):
               if x < 0: return - arctan(-x)
               elif x > 1: return math.pi / 2 - arctan(1 / x)
               else:
                   t = x / (1 + math.sqrt(1 + x ** 2))
                   u = t / (1 + math.sqrt(1 + t ** 2))
                   n = 12
                   s = 0
                   sgn = 1
                   u2 = u * u
                   for k in range(n + 1):
                       s = s + sgn * u / (2 * k + 1)
                       u = u * u2
                       sgn = - sgn
                   return 4 * s
```

```
Entrée [19]: x = 0.9
               print(arctan(x))
               print(math.atan(x))
```

```
0.7328151017865067
0.7328151017865067
```

Maintenant, notre fonction logarithme devrait fonctionner.

```
Entrée [20]: z = 1.234 + 5.678 * 1j
               print(log(z))
               print(cmath.log(z))
```

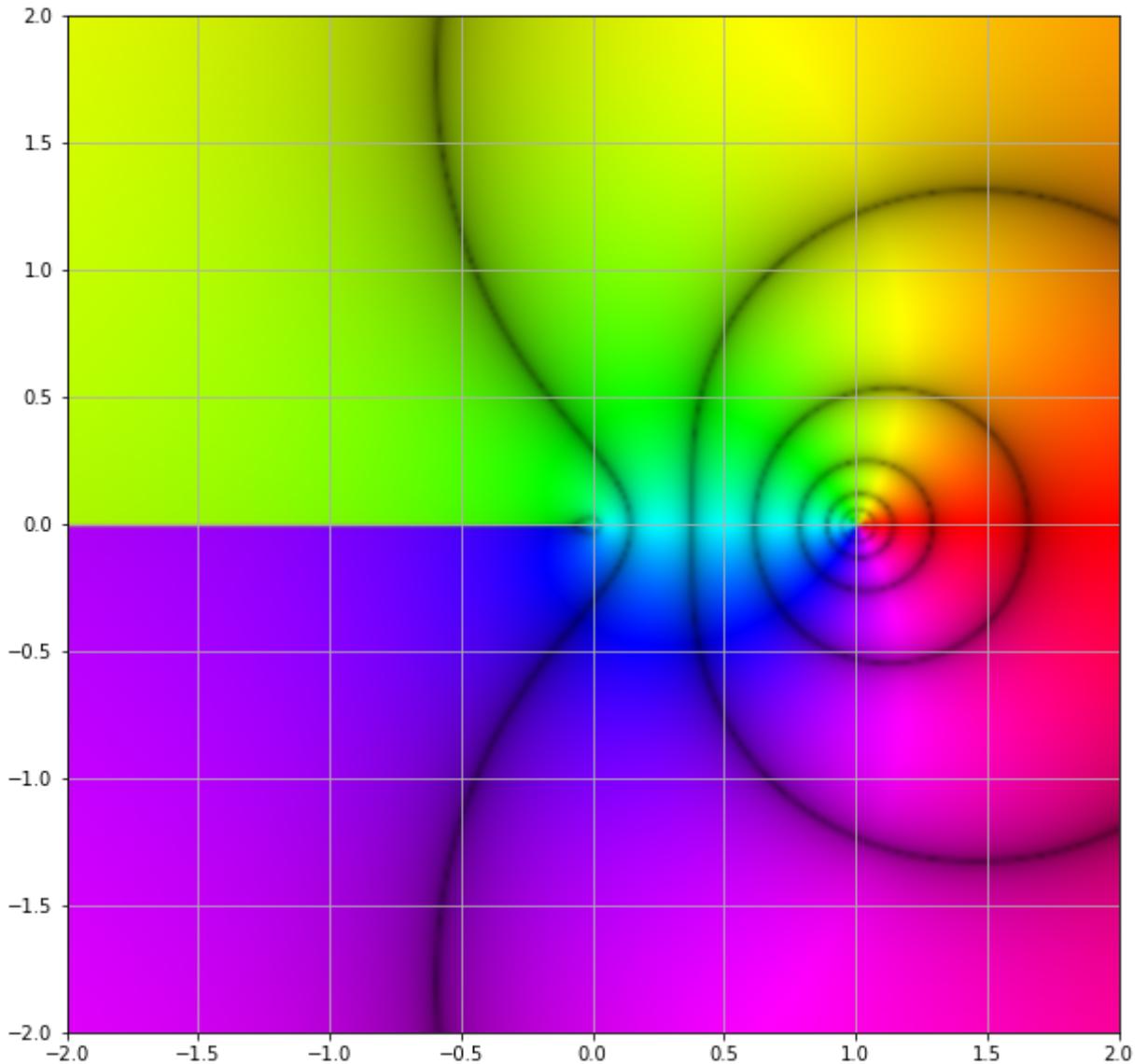
```
(1.7596744709987848+1.3567941381565736j)
(1.7596744709987848+1.3567941381565736j)
```

2.3 Le graphe

Après cette ((trop) longue) digression, traçons enfin le graphe du logarithme complexe.

Entrée [21]:

```
M = 2  
plot_complex(lambda z: cmath.log(z), extent=[-M, M, -M, M])
```



Que voit-on ?

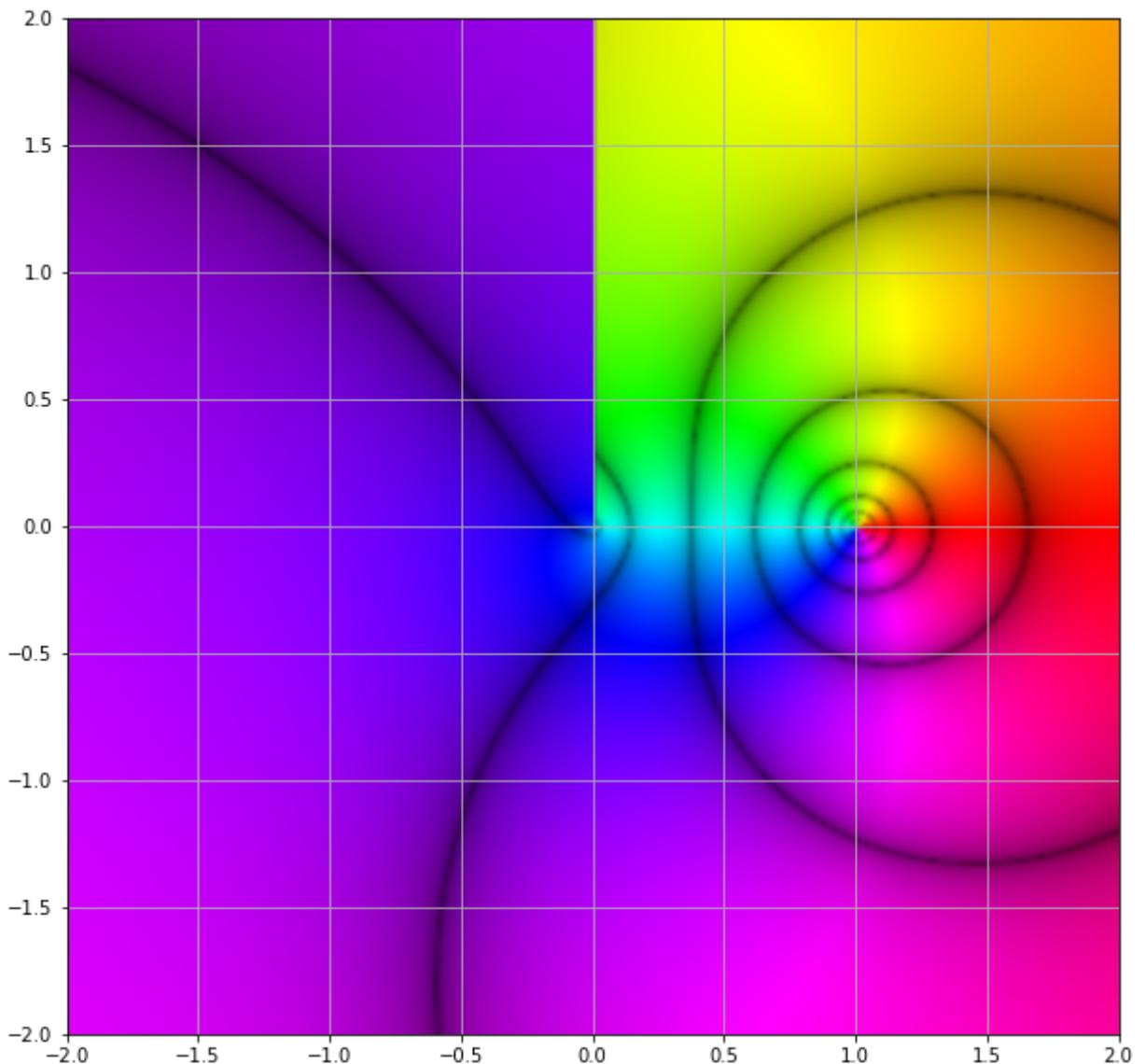
- Une racine en 1. Normal, puisque $\log 1 = 0$. Remarquez le cycle des couleurs. Oui, on peut définir l'ordre de multiplicité d'une racine pour une fonction holomorphe, et nous osons dire que 1 est racine simple de \log .
- Une discontinuité sur \mathbb{R}_-^* . Remarquez le saut des couleurs au passage des réels strictement négatifs.
- Un gros souci en 0. Normal puisque $|\log z|$ tend vers $+\infty$ lorsque z tend vers 0. A-t-on un pôle, comme pour les fractions rationnelles ? Non, c'est plus grave que cela. Car à gauche de 0 on a en plus de la discontinuité. Nous avons là ce que l'on appelle une **singularité non isolée**, l'extrémité de \mathbb{R}_- . La demi-droite \mathbb{R}_- est appelée une **coupure** du logarithme.

2.4 Autres déterminations du logarithme

Il existe bien d'autres **déterminations du logarithme**, c'est à dire des fonctions ℓ holomorphes sur une partie de \mathbb{C} et telles que $e^{\ell(z)} = z$. Par exemple, on peut choisir l'argument de z dans $]-\frac{3\pi}{2}, \frac{\pi}{2}]$, auquel cas nous obtiendrons une fonction logarithme holomorphe sur \mathbb{C} privé des imaginaires positifs. La voici.

```
Entrée [22]: def mylog(z):
    t = cmath.phase(z)
    if math.pi / 2 < t and t <= math.pi: t = t - 2 * math.pi
    return math.log(abs(z)) + 1j * t
```

```
Entrée [23]: M = 2
plot_complex(lambda z: mylog(z), extent=[-M, M, -M, M])
```



Notre nouveau logarithme est holomorphe sur \mathbb{C} privé de la demi-droite $i\mathbb{R}_+$.

On peut hélas montrer qu'il n'existe aucune détermination du logarithme holomorphe sur \mathbb{C}^* . De façon très générale (et très vague), si Ω est un ouvert connexe de \mathbb{C}^* , il existe une détermination du logarithme holomorphe sur Ω si et seulement si on ne peut pas "faire le tour de O " en restant dans Ω . Par exemple, si Γ est une "courbe qui joint O et l'infini", alors il existe une détermination du logarithme holomorphe sur $\mathbb{C} \setminus \Gamma$. Un exemple fou ?

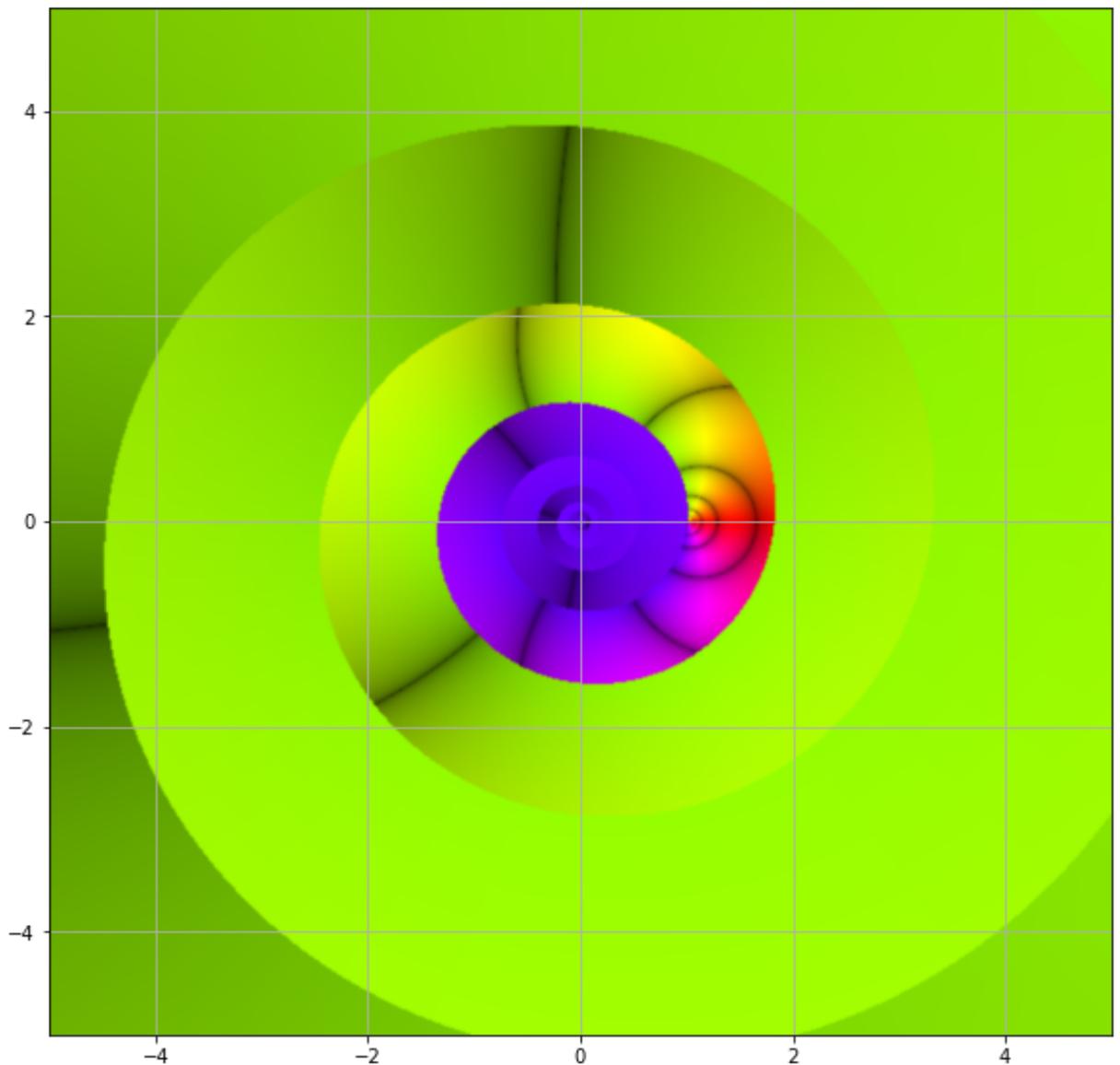
2.4 Le logarithme fou

Soit $a > 1$. Prenons pour Γ la courbe définie par l'équation polaire $\rho = a^\theta$. Plus explicitement, un nombre complexe z appartient à Γ si et seulement si $z = \rho e^{i\theta}$ où $\rho = a^\theta$.

Cette courbe est une spirale qui s'enroule autour de O et "part à l'infini". Il doit donc exister un logarithme fou qui est holomorphe sur $\mathbb{C} \setminus \Gamma$! Sans autre commentaire, le voici avec son graphe.

```
Entrée [24]: def logfou(z, a):  
    r = abs(z)  
    theta = cmath.phase(z)  
    k = math.floor((math.log(r, a) - theta) / (2 * math.pi))  
    return math.log(r) + 1j * (theta + 2 * k * math.pi)
```

```
Entrée [27]: M = 5  
plot_complex(lambda z: logfou(z, 1.1), extent=[-M, M, -M, M], nx=600)
```



Comment fonctionne ce logarithme ? Partez de la zone violette et tournez en ne coupant jamais la spirale. L'argument de z augmente, augmente indéfiniment, et tend vers $+\infty$. En fait, sans entrer dans les détails des calculs, $\arg z$ augmente beaucoup plus vite que $\ln |z|$. Conséquence, $\ln |\ln z|$ tend vers $\frac{\pi}{2}$ lorsque z tend vers l'infini en spiralant. Si vous vous demandiez pourquoi tout est vert vers l'extérieur du graphe vous avez la réponse : le vert est la couleur de l'argument $\frac{\pi}{2}$.

Exercice : Pourquoi le centre du dessin est-il violet ?

Exercice : Expliquez la fonction `logfou`.

3. Les puissances

3.1 C'est quoi ?

Comment définir z^α pour $z, \alpha \in \mathbb{C}$?

Définition : Pour $z \in \mathbb{C}^*$ et $\alpha \in \mathbb{C}$ on pose

$$z^\alpha = e^{\alpha \log z}$$

où $\log z$ est **un** logarithme de z . De plus, pour $\alpha \in \mathbb{R}_+^*$ on pose $0^\alpha = 0$ et on pose également $0^0 = 1$. En revanche, 0^α n'est pas défini si $\alpha \in \mathbb{C} \setminus \mathbb{R}_+$. Abandonnez donc tout espoir de calculer 0^i .

Proposition : Si \log est la détermination principale du logarithme, alors $z \mapsto z^\alpha$ est holomorphe sur $\mathbb{C} \setminus \mathbb{R}_-$. Sa dérivée est la fonction $z \mapsto \alpha z^{\alpha-1}$.

Démonstration : C'est évident, on compose des fonctions holomorphes. La formule de dérivation des composées nous donne une valeur de la dérivée identique à celle que l'on trouve pour les fonctions de variable réelle.

Les puissances complexes vérifient-elles les propriétés "classiques" ? Oui pour certaines. En voici une qui fonctionne :

$$z^{\alpha+\beta} = e^{(\alpha+\beta) \log z} = e^{\alpha \log z} e^{\beta \log z} = z^\alpha z^\beta$$

En revanche,

$$(zz')^\alpha = e^{\alpha(\log zz')} = e^{\alpha(\log z + \log z' + 2ik\pi)} = z^\alpha z'^\alpha e^{2iak\pi}$$

où $k \in \mathbb{Z}$ dépend de z et z' . Si $\alpha \in \mathbb{Z}$ alors, oui, $(zz')^\alpha = z^\alpha z'^\alpha$. Sinon, la formule classique est valable pour certains z, z' mais pas pour d'autres.

3.2 "La" racine carrée

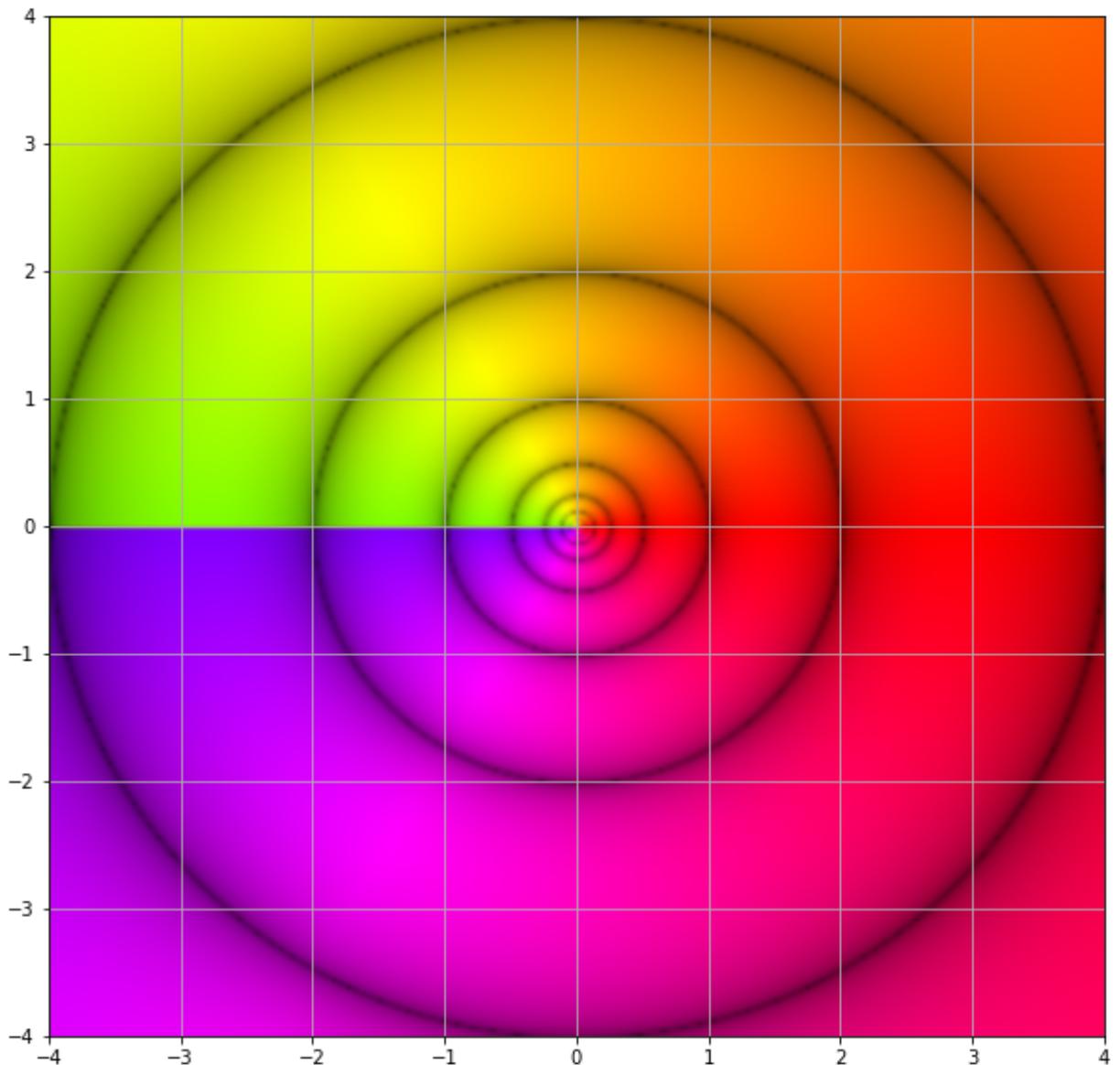
Définissons "la" **racine carrée** de z comme étant $z^{\frac{1}{2}}$, c'est à dire $\exp(\frac{1}{2}\log z)$. Cette fonction est holomorphe sur $\mathbb{C} \setminus \mathbb{R}_-$. Elle est discontinue en tout réel négatif.

```
Entrée [28]: print(cmath.sqrt(-1))  
print(cmath.sqrt(-1 - 0.0001 * 1j))  
print(cmath.sqrt(-1 + 0.0001 * 1j))
```

```
1j  
(4.999999993750001e-05-1.0000000012499999j)  
(4.999999993750001e-05+1.0000000012499999j)
```

Traçons son graphe. Lorsque z tourne autour de O , \sqrt{z} fait de même, mais deux fois moins vite puisque $\arg \sqrt{z} = \frac{1}{2} \arg z$.

```
Entrée [29]: M = 4  
plot_complex(lambda z: z ** (1 / 2), extent=[-M, M, -M, M], m=0.5)
```



Que voit-on ? Une discontinuité de la fonction sur \mathbb{R}_- . Quand l'argument de z passe de 0 à π , celui de $z^{1/2}$ va de 0 (rouge) à $\frac{\pi}{2}$ (vert). En revanche, quand l'argument de z passe de 0 à $-\pi$, celui de $z^{1/2}$ va de 0 (rouge) à $-\frac{\pi}{2}$ (violet).

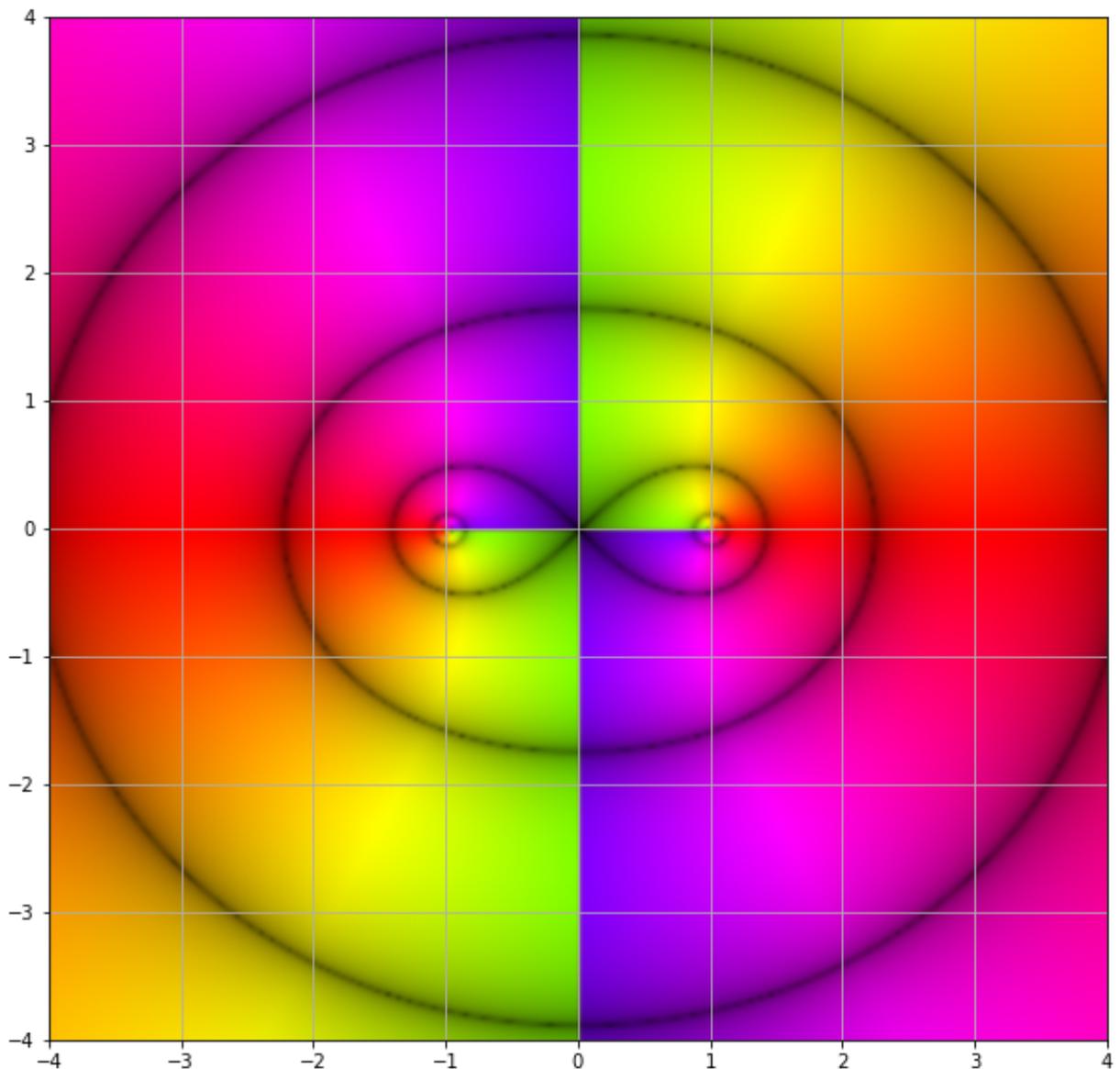
Exercice : Et la fonction racine cubique ? Et $z^{\frac{2}{3}}$? Expérimentez.

3.3 Toujours des racines carrées

Considérons la fonction $z \mapsto \sqrt{z^2 - 1}$.

```
Entrée [30]: def f(z): return cmath.sqrt(z ** 2 - 1)
```

```
Entrée [31]: M = 4  
plot_complex(f, extent=[-M, M, -M, M])
```



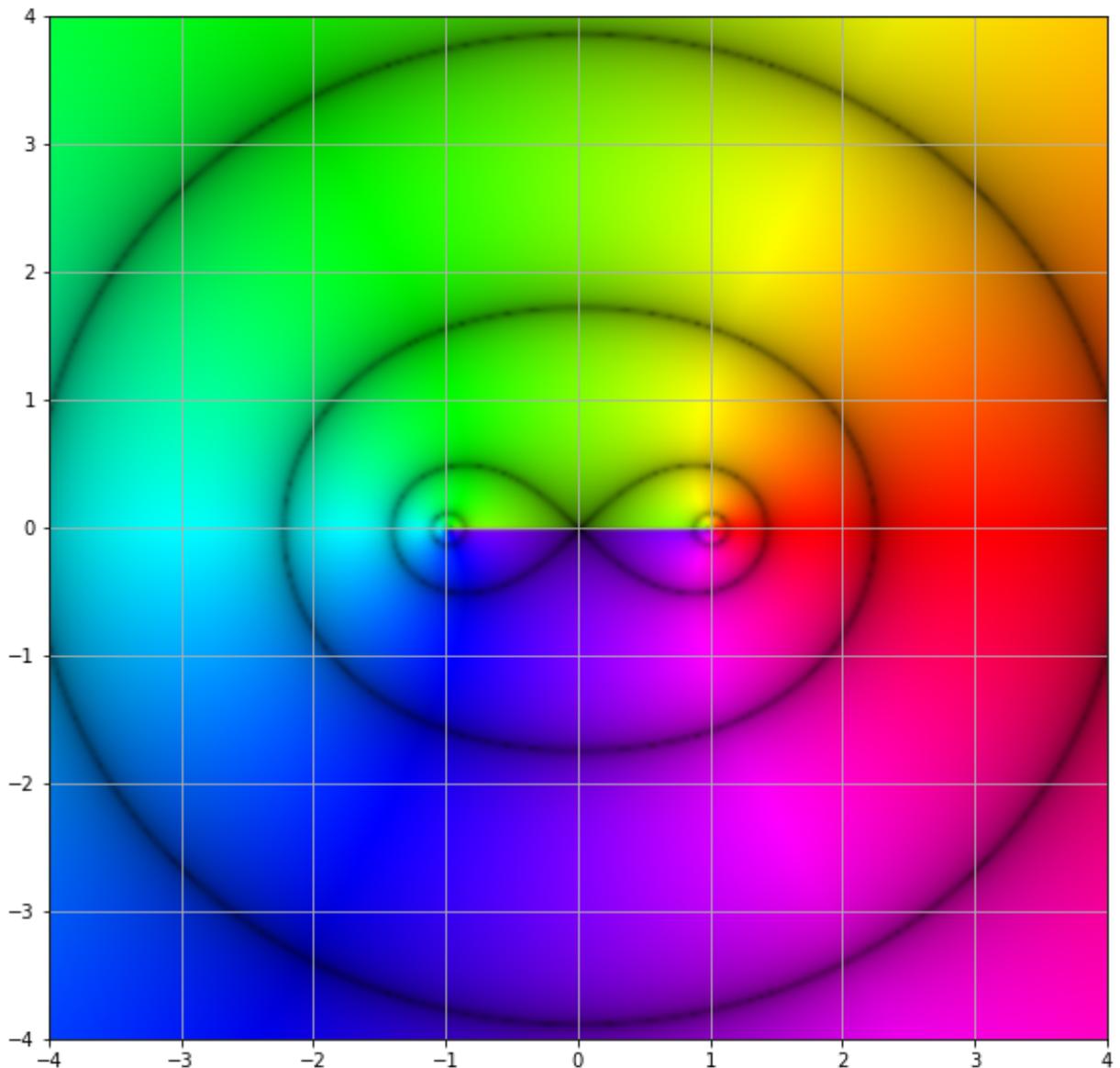
La fonction f pose "un problème" en tout z tel que $z^2 - 1 \in \mathbb{R}_-$, c'est à dire :

- Si $z \in \mathbb{R}$ et $-1 \leq z \leq 1$, ou alors
- Si $z \in i\mathbb{R}$ est imaginaire pur.

Peut-on espérer diminuer le nombre de points problématiques, tout en conservant la propriété que $f(z)$ est **une** racine carrée de $z^2 - 1$? Oui, si on remplace $f(z)$ par $-f(z)$ dans le demi-plan gauche.

```
Entrée [32]: def g(z):  
             w = cmath.sqrt(z ** 2 - 1)  
             if z.real < 0: return -w  
             elif z.real == 0 and z.imag > 0: return -w  
             else: return w
```

```
Entrée [33]: M = 4  
            plot_complex(g, extent=[-M, M, -M, M])
```

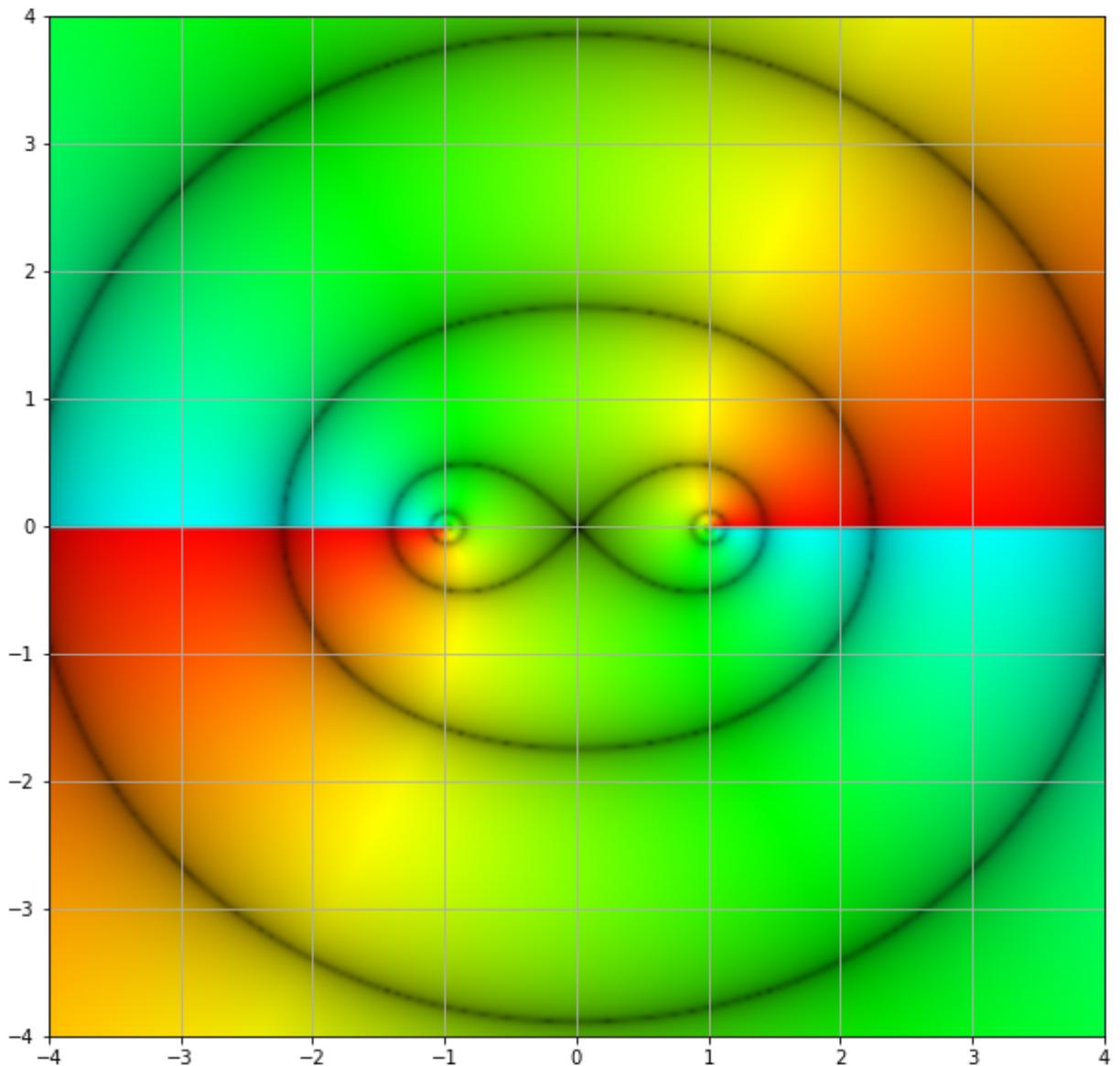


Quelques couleurs sur un graphique ne sont pas une preuve, mais il semble que nous ayons amélioré la situation. La fonction g est une racine carrée de $z^2 - 1$ qui est (peut-être ?) holomorphe sur $\mathbb{C} \setminus [-1, 1]$. Tentons une autre modification de f . Est-il possible de rendre la fonction régulière sur $] -1, 1[$ et de repousser les problèmes aux demi-droites $] -\infty, -1]$ et $[1, +\infty[$? Oui. Considérons la fonction h ci-dessous.

```
Entrée [34]: def h(z):  
    w = cmath.sqrt(z ** 2 - 1)  
    if z.imag < 0 and z.real > 0: return -w  
    elif z.imag > 0 and z.real < 0: return -w  
    else: return w
```

Traçons h .

```
Entrée [35]: M = 4  
plot_complex(h, extent=[-M, M, -M, M])
```

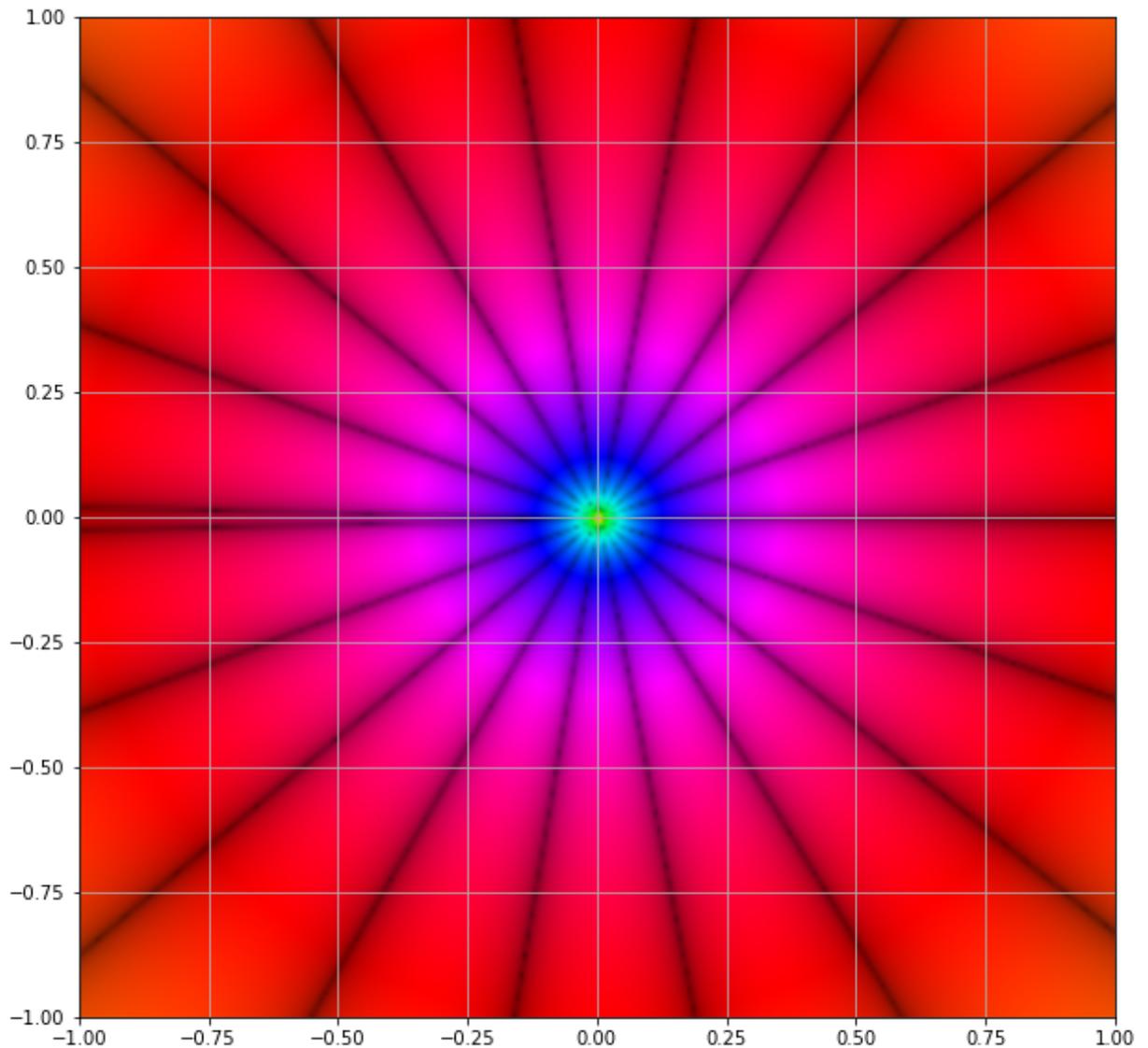


Gagné ! Enfin, peut-être. Nos dessins nous suggèrent une probable continuité, mais ils ne prouvent certainement pas l'holomorphicité. Souvenez-vous de h , elle reviendra au premier plan lorsque nous parlerons de la fonction arc cosinus.

3.4 Les puissances bizarres

Tentons $z \mapsto z^i$.

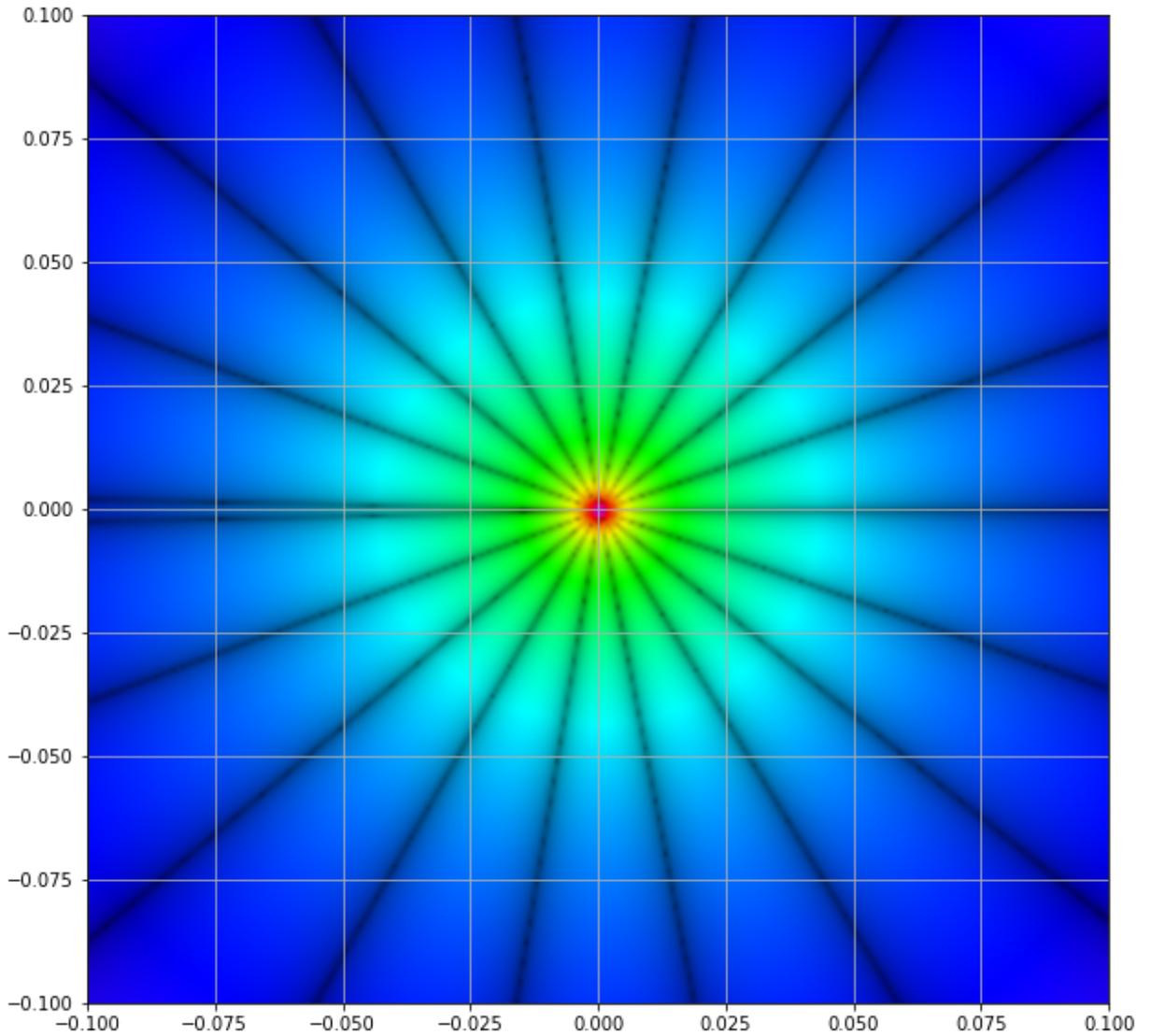
```
Entrée [36]: M = 1  
plot_complex(lambda z: z ** 1j, extent=[-M, M, -M, M], m=0.5)
```



Cela ne ressemble à rien de ce que nous avons déjà vu. Changeons la valeur de M :

Entrée [37]:

```
M = 0.1  
plot_complex(lambda z: z ** 1j, extent=[-M, M, -M, M], m=0.5)
```



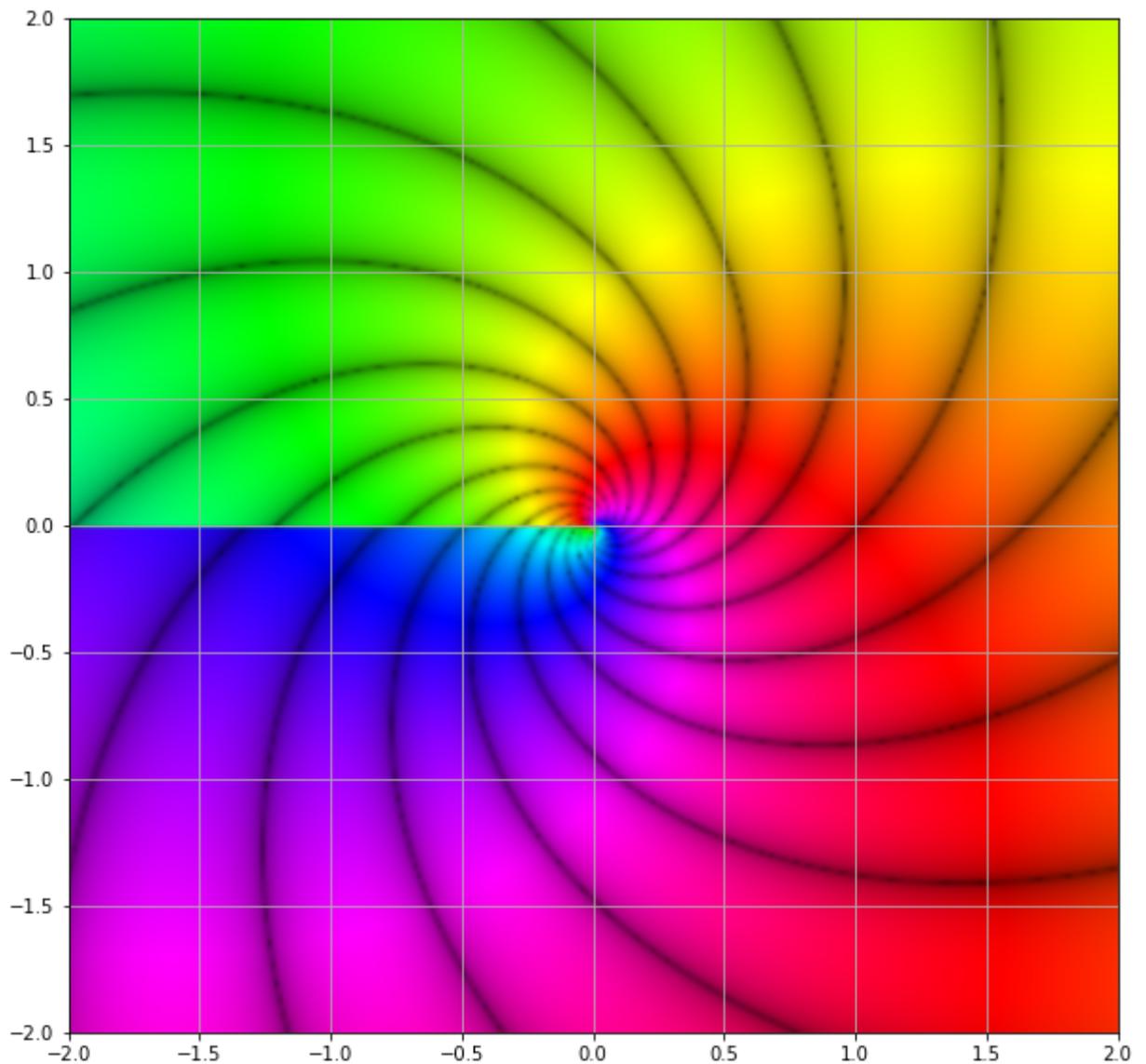
Les couleurs ont changé complètement. En fait, les cycles de couleurs n'ont pas lieu, comme nous en avons pris l'habitude, sur des cercles de centre O , mais sur des **demi_droites** passant par O . Bigre, qu'est-ce que cela signifie ? Posons $z = r e^{i\theta}$ où $r > 0$ et $-\pi < \theta \leq \pi$. On a

$$z^i = e^{i \log z} = e^{i \ln r - \theta} = e^{i \ln r} e^{-\theta}$$

Ainsi, $|z^i| = e^{-\theta}$ et $\arg z^i = \ln r$. Élever à la puissance i intervertit en quelque sorte le module et l'argument. Je vous laisse méditer les conséquences d'une telle chose.

Encore un petit essai, fort intéressant : $z \mapsto z^{e^{i\pi/4}}$.

Entrée [38]: `M = 2`
`plot_complex(lambda z: z ** ((1 + 1j) / math.sqrt(2)), extent=[-M, M, -M, M])`



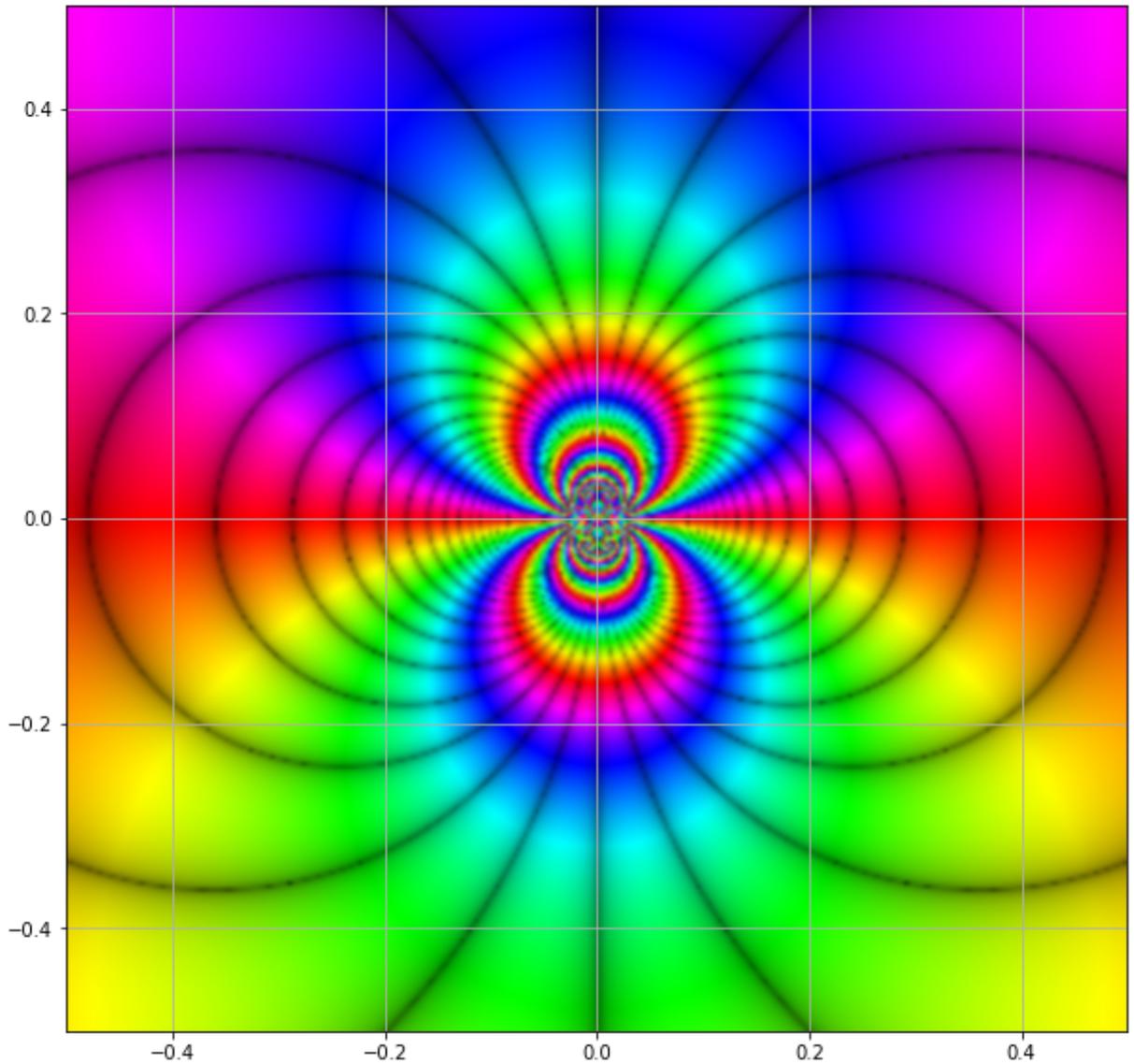
Exercice : Trouver une équation des courbes $|f(z)| = cte$. Ce sont des spirales, quel genre de spirales ?

4. Singularités essentielles

4.1 Une fonction au comportement compliqué

Considérons la fonction $f : z \mapsto \exp \frac{1}{z}$.

```
Entrée [39]: M = 0.5  
plot_complex(lambda z: cmath.exp(1 / z), extent=[-M, M, -M, M])
```



La situation en 0 est terriblement compliquée. On ne voit pas des cercles qui "tournent autour de O ". Au contraire :

- Concernant les cercles où le module est constant, on en "voit" une infinité, ils ont l'air de passer par O et de tous avoir en ce point une tangente verticale.
- Il semblerait que nous ayons aussi des cercles où l'argument est constant, on en "voit" une infinité, ils ont l'air de passer par O et de tous avoir en ce point une tangente horizontale.

Et puis d'ailleurs est-ce que ce sont vraiment des cercles ? Que se passe-t-il exactement en 0 ?

4.2 Singularité essentielle

Proposition : Soit $f : z \mapsto e^{1/z}$. Soit $\varepsilon > 0$. Pour tout nombre complexe $w \neq 0$, il existe $z \in \mathbb{C}^*$ tel que $|z| \leq \varepsilon$ et $f(z) = w$.

Dit autrement, dans tout voisinage de 0, aussi petit soit-il, la fonction f est une surjection sur \mathbb{C}^* .

Démonstration : Soit $z \in \mathbb{C}^*$. Un calcul très simple montre que $f(z) = w$ si et seulement si $\frac{1}{z} = \ln |w| + i(\arg w + 2k\pi)$ où $k \in \mathbb{Z}$. Appelons z_k ce nombre. On a

$$|z_k| = \frac{1}{\sqrt{\ln^2 |w| + (\arg w + 2k\pi)^2}}$$

Et on peut rendre cette quantité inférieure à ε en prenant k suffisamment grand.

Le comportement de f au voisinage de 0 est donc particulièrement compliqué. Nous avons à l'origine ce que l'on appelle une **singularité essentielle**.

4.3 Les cercles tangents

Ce que nous avons vu sur le graphe de f est-il bien la réalité ? Montrons-le.

Soit $z = x + iy$. On a

$$\exp\left(\frac{1}{z}\right) = \exp \frac{x - iy}{x^2 + y^2} = \exp \frac{x}{x^2 + y^2} \exp \frac{-iy}{x^2 + y^2}$$

Ainsi,

$$\left| \exp\left(\frac{1}{z}\right) \right| = \exp \frac{x}{x^2 + y^2} \quad \text{et} \quad \arg \exp\left(\frac{1}{z}\right) = \frac{-iy}{x^2 + y^2}$$

Soit $c > 0$. On a $\left| \exp\left(\frac{1}{z}\right) \right| = c$ si et seulement si $x = \ln c(x^2 + y^2)$. Posant $k = \frac{1}{\ln c}$, ceci équivaut à $x^2 + y^2 - kx = 0$, ou encore

$$\left(x - \frac{k}{2}\right)^2 = y^2 = \frac{k^2}{4}$$

qui est bien l'équation d'un cercle passant par O . Le lecteur consciencieux se convaincra facilement que ce cercle a une tangente verticale en O .

Calculs totalement analogue pour les courbes $\arg f(z) = c$, où $c \in \mathbb{R}$. Si $c \neq 0$, on obtient le cercle d'équation

$$x^2 + \left(y + \frac{1}{2c}\right)^2 = \frac{1}{4c^2}$$

Si $c = 0$ on obtient l'axe Ox .

4.4 À gauche de l'origine

Soit $c > 0$ un réel fixé. Soit \mathcal{U} l'ensemble des nombres complexes $z = x + iy$ où $x < 0$ et $|\frac{y}{x}| \leq c$. De tels z sont des nombres complexes de partie réelle négative dont on astreint l'argument à ne pas trop s'approcher de $\pm \frac{\pi}{2}$. Si vous préférez, \mathcal{U} est la partie du demi-plan à gauche de l'axe Oy limitée par deux demi-droites faisant un angle aigu avec \mathbb{R}_- . On a pour un tel z :

$$|f(z)| = \exp \frac{x}{x^2 + y^2} \exp \frac{1}{x} \frac{1}{1 + \frac{y^2}{x^2}}$$

On a $1 + \frac{y^2}{x^2} \geq k = \frac{1}{1+c^2}$. Comme $x < 0$, on en déduit que

$$|f(z)| \leq \exp \frac{k}{x}$$

et le majorant tend vers 0 lorsque x tend vers 0. Ainsi, $f(z) \rightarrow 0$ lorsque $z \rightarrow 0, z \in \mathcal{U}$.

4.5 À droite de l'origine

Si vous reprenez les calculs précédents en supposant cette fois que $x > 0$ vous obtiendrez que $|f(z)|$ tend vers $+\infty$. Comportement de f totalement différent, donc, à gauche et à droite de O . Ceci ne se voit pas du tout sur le graphique, notre représentation n'est pas suffisante pour détecter ce genre de chose.

4.6 L'axe Oy

Soit $t \in \mathbb{R}^*$. On a $f(it) = \exp(-\frac{i}{t})$. $f(it)$ est donc de module 1, constant. En revanche, l'argument de $f(z)$ est $-\frac{1}{t}$. Cet argument tend vers 0 lorsque t tend vers l'infini. En revanche, lorsque t tend vers 0, l'argument de $f(it)$ tend vers l'infini. Il y a donc une infinité de cycles de couleurs sur l'axe Oy dans tout voisinage de O .

5. Et maintenant ?

Passez au notebook numéro 3, nous allons y explorer les fonctions trigonométriques.

Entrée []: