

Fonctions_Complexes_1

April 12, 2019

1 Fonctions d’une variable complexe (I)

Marc Lorenzi

7 avril 2019

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

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

Nous allons explorer “visuellement” quelques fonctions $\mathbb{C} \rightarrow \mathbb{C}$. Après une petite discussion sur le modèle de couleurs HSV, nous verrons une méthode pour représenter les fonctions de variable complexe. But totalement avoué de ce notebook : mettre en évidence graphiquement quelques propriétés remarquables d’une classe particulière de fonctions de la variable complexe : les fonctions **holomorphes**.

1.1 1. représenter graphiquement une fonction d’une variable complexe

1.1.1 1.1 Le modèle de couleurs HSV

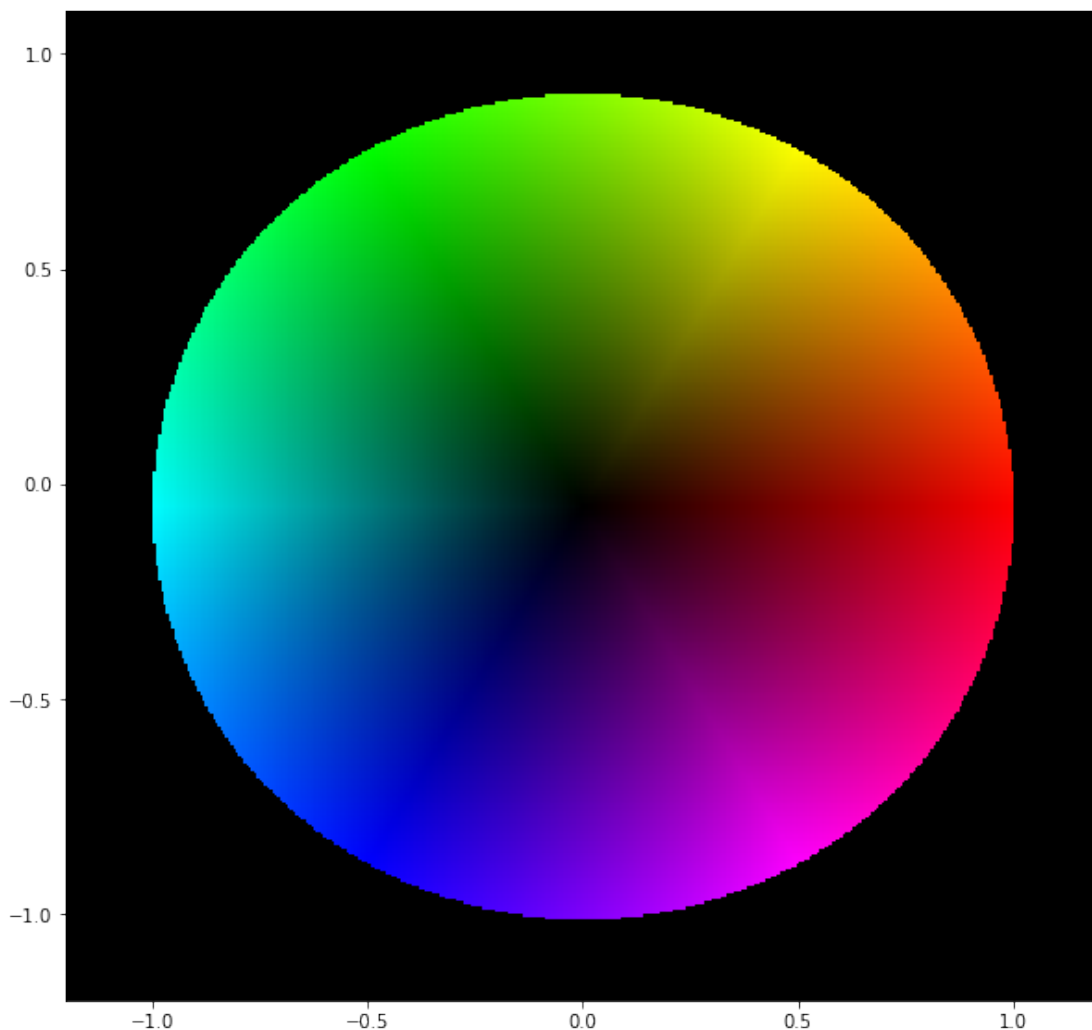
Le modèle **HSV (Hue, Separation, Value)** est un modèle de couleurs plus intuitif à manipuler que le modèle standard RGB (Red, Green, Blue). Chaque couleur est représentée par un triplet (h, s, v) où

- h (“hue”, “teinte” en français) est la couleur au sens “naïf”. $h = 0$ correspond au rouge. Lorsque h augmente, on passe par les couleurs de l’arc en ciel : rouge, orange, jaune, vert, bleu, violet ... puis, lorsque h tend vers 1, on passe du violet au rouge. Nous avons donc un cycle de couleurs.
- s est la saturation. plus s est petit, plus les couleurs sont “fades” (pour $s = 0$ on obtient les niveaux de gris). Dans le système RGB, augmenter s consiste à augmenter le minimum des trois valeurs r, g, b .
- v (“value”, “brightness”, “luminance” ou “brillance” en français) est la brillance de la couleur. Dans le système RGB, augmenter v consiste à augmenter simultanément r, g, b en gardant leurs proportions respectives constantes.

Ci-dessous, le disque de centre O et de rayon 1. Pour tout nombre complexe $z = re^{i\theta}$ où $0 \leq r \leq 1$ et $0 \leq \theta \leq 2\pi$ on colorie le point correspondant en prenant $(h, s, v) = (\frac{\theta}{2\pi}, 1, r)$. La fonction `cmath.phase` renvoyant un argument de z dans l'intervalle $]-\pi, \pi]$, on adapte évidemment la valeur de θ .

```
In [3]: N = 400
m = [[(0, 0, 0) for i in range(N)] for j in range(N)]
for i in range(N):
    for j in range(N):
        x = -1.2 + j * 2.4 / N
        y = 1.2 - i * 2.4 / N
        z = x + 1j * y
        theta = cmath.phase(z)
        if theta < 0: theta = theta + 2 * math.pi
        r = abs(z)
        if r <= 1: m[i][j] = colorsys.hsv_to_rgb(theta / (2 * math.pi), 1, r)
plt.imshow(m, extent=[-1.2, 1.2, -1.2, 1.1])
```

```
Out[3]: <matplotlib.image.AxesImage at 0x11535f2e8>
```



Remarquez comment

- Les couleurs deviennent plus brillantes du centre vers le bord du disque.
- Lorsqu'on tourne autour de l'origine, les couleurs décrivent les couleurs de l'arc en ciel.

Nous allons maintenant utiliser le modèle HSV pour dessiner des fonctions $\mathbb{C} \rightarrow \mathbb{C}$. Et expliquer comment se comportent ces fonctions.

1.1.2 1.2 Introduction

Comment dessiner une fonction $f : \mathbb{C} \rightarrow \mathbb{C}$? Évidemment, l'idéal serait que mes lecteurs puissent voir dans un espace de dimension 4. Mais comme il est possible que ce ne soit pas le cas pour tous, nous allons procéder autrement. Diverses méthodes existent, nous allons nous concentrer sur l'une d'entre-elles. Pour tout $z \in \mathbb{C}$, nous colorions le point du plan d'affixe z avec une couleur qui dépend de $f(z)$. Comme $f(z)$ est essentiellement un objet bidimensionnel, nous utiliserons des couleurs qui dépendent de deux paramètres.

À partir d'ici, des choix stratégiques s'imposent. Rien ne vous empêche bien entendu d'essayer d'autres possibilités. Nous allons colorier le point d'affixe z avec la couleur (h, s, v) (dans le modèle HSV, donc) où

$$h = \arg f(z), s = 1, v = |f(z)|$$

Enfin, c'est à peu près cela. Car, rappelons-nous, les paramètres h, s, v doivent appartenir à $[0, 1]$. Pour h , c'est facile. La fonction `cmath.phase` renvoie l'argument du nombre complexe z , entre $-\pi$ et π . Ramener la valeur à l'intervalle $[0, 1]$ se fait aisément.

```
In [4]: def arg01(z):  
        theta = cmath.phase(z) / (2 * math.pi)  
        if theta < 0: return theta + 1  
        else: return theta  
  
In [5]: for z in [4, 1j, -1j, 1+1j, -2, 2 - 2* 1j]:  
        print(z, arg01(z))
```

```
4 0.0  
1j 0.25  
(-0-1j) 0.75  
(1+1j) 0.125  
-2 0.5  
(2-2j) 0.875
```

Le module de $f(z)$, en revanche, varie sur $\mathbb{R}_+$. J'ai choisi ici de prendre

$$v = \left| \sin \left(\frac{\pi}{m} \log_2(1 + |f(z)|) \right) \right|^d$$

où m et d sont deux paramètres. J'imagine que vous attendez quelques explications :-)

- Grâce au logarithme, lorsque le module de $f(z)$ double, la quantité augmente environ de 1
- Le sinus permet d’obtenir des variations périodiques de période m , où m peut être choisi comme on le désire.
- L’exposant d est un paramètre de “dureté”. Plus d est grand, plus les contrastes sont importants. La valeur $d = 1$ est tout à fait acceptable dans la plupart des cas, mais on se réserve la possibilité de la modifier.

```
In [6]: def psi(x, m, d):
        if x == 0: return 0
        else:
            return abs(math.sin(math.pi * math.log(x, 2) / m)) ** d
```

Un petit dessin vaut mieux qu’un long discours, alors écrivons la fonction de tracé.

1.1.3 1.3 La fonction de tracé

La fonction `plot_complex` prend en paramètres :

- Une liste extent de 4 flottants : les abscisses minimale et maximale et les ordonnées minimale et maximale des points à tracer.
- Un paramètre m dont nous avons déjà parlé.
- Un paramètre d dont nous avons aussi parlé.
- deux paramètres nx et ny qui sont le nombre de lignes et de colonnes de l’image que nous voulons afficher.

```
In [7]: def plot_complex(f, extent=[-1.5, 1.5, -1.5, 1.5], m=1, d=0.2, nx=400, ny=400):
        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='auto')
        plt.grid()
```

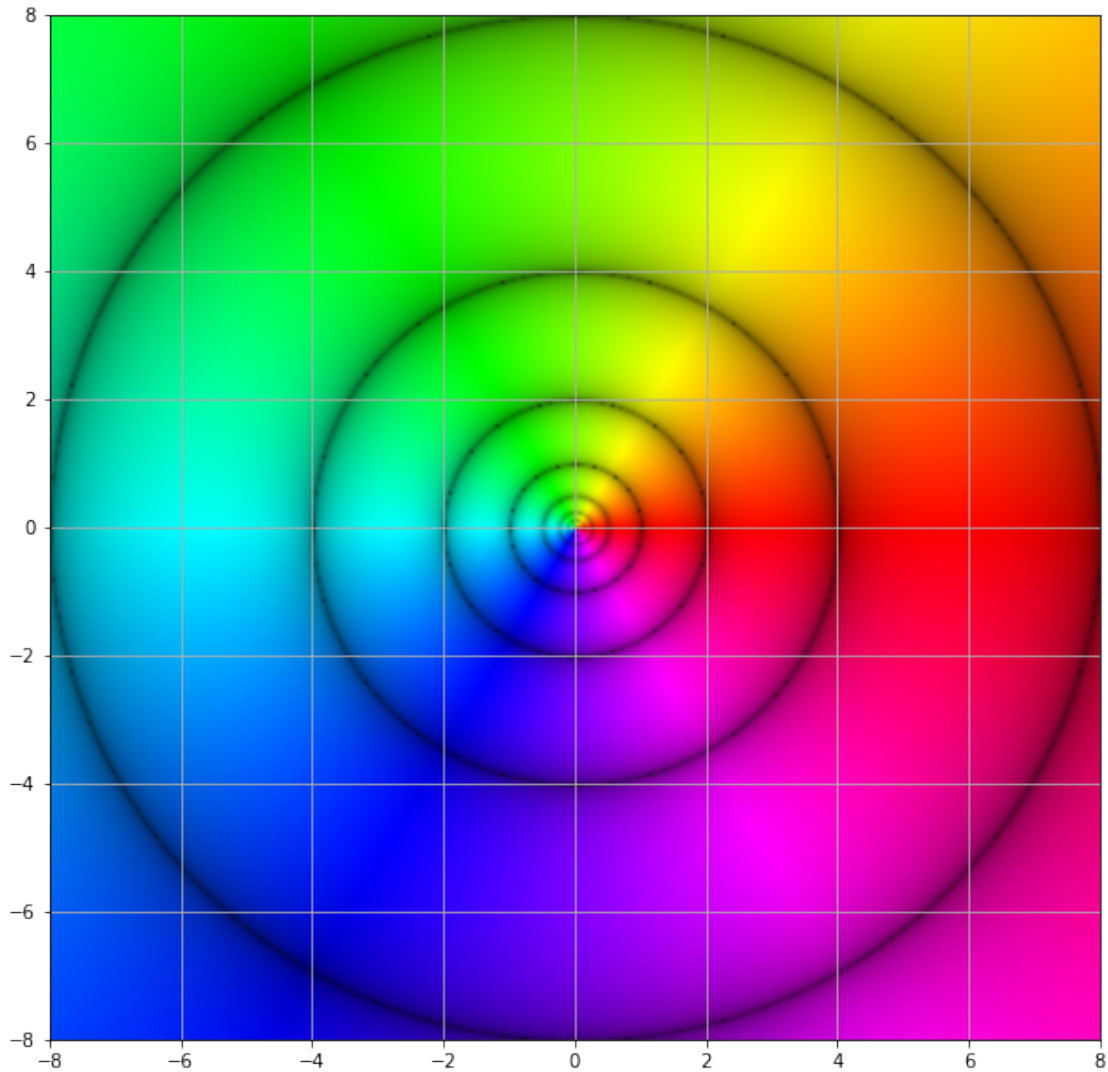
Le `try ... except` permet de ne pas se préoccuper d’éventuelles divisions par 0, ou autres calamités liées au calcul numérique. Si, en un point z , $f(z)$ n’est pas défini, le pixel correspondant est colorié en noir.

1.1.4 1.4 Un premier test

Commençons par le plus simple possible (hormis les fonctions constantes) : la fonction $z \mapsto z$.

In [8]: $M = 8$

```
plot_complex(lambda z: z, extent=[-M, M, -M, M])
```



Que voit-on ?

- Des anneaux concentriques, centrés à l'origine, d'épaisseurs de plus en plus grandes. En comptant les anneaux à partir du centre vous obtenez une idée de la valeur de $|f(z)|$. Rappelez-vous que d'un anneau à l'autre, le module est environ multiplié par 2.
- Les anneaux sont multicolores. Très précisément, les couleurs passent par toutes les couleurs de l'arc en ciel *dans le sens trigonométrique*. Ces couleurs vous donnent l'argument de $f(z)$.

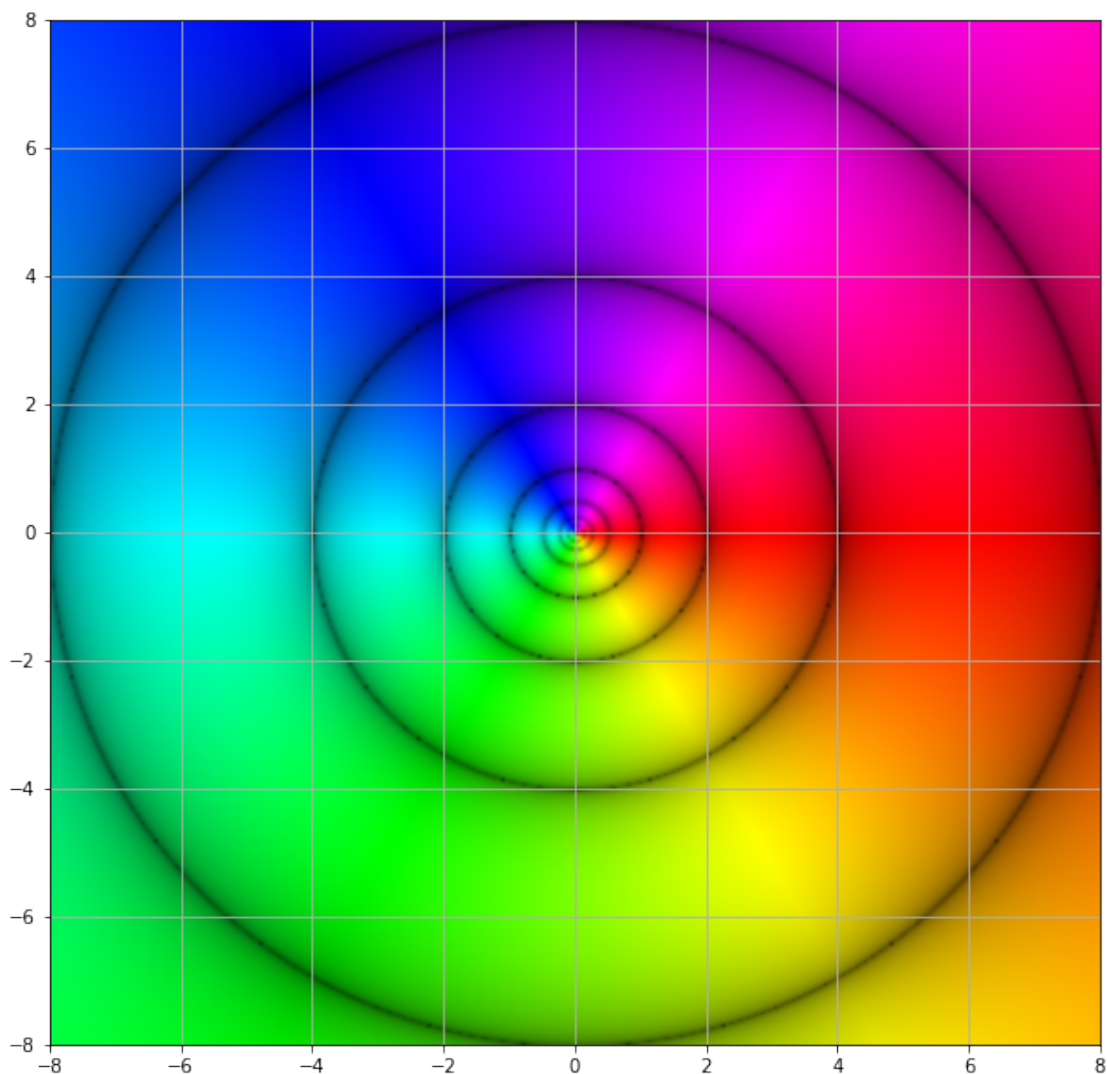
Exercice : Essayez les fonctions $z - 1$, $z + i$, $3z + 2$, $iz - 1$, $-z$. Bien, vous avez compris que toutes les fonctions affines se ressemblent. Cercles concentriques autour de l'unique racine, coloriés dans le sens trigonométrique.

1.1.5 1.5 Un second test

Avant d'entrer dans le vif du sujet, essayons aussi $z \mapsto \bar{z}$.

In [9]: $M = 8$

```
plot_complex(lambda z: z.conjugate(), extent=[-M, M, -M, M])
```



C'est pareil que $z \mapsto z$, me direz-vous ? Oui, sauf que c'est le contraire ! Cette fois-ci les couleurs de l'arc en ciel sont parcourues dans le sens trigonométrique **inverse**.

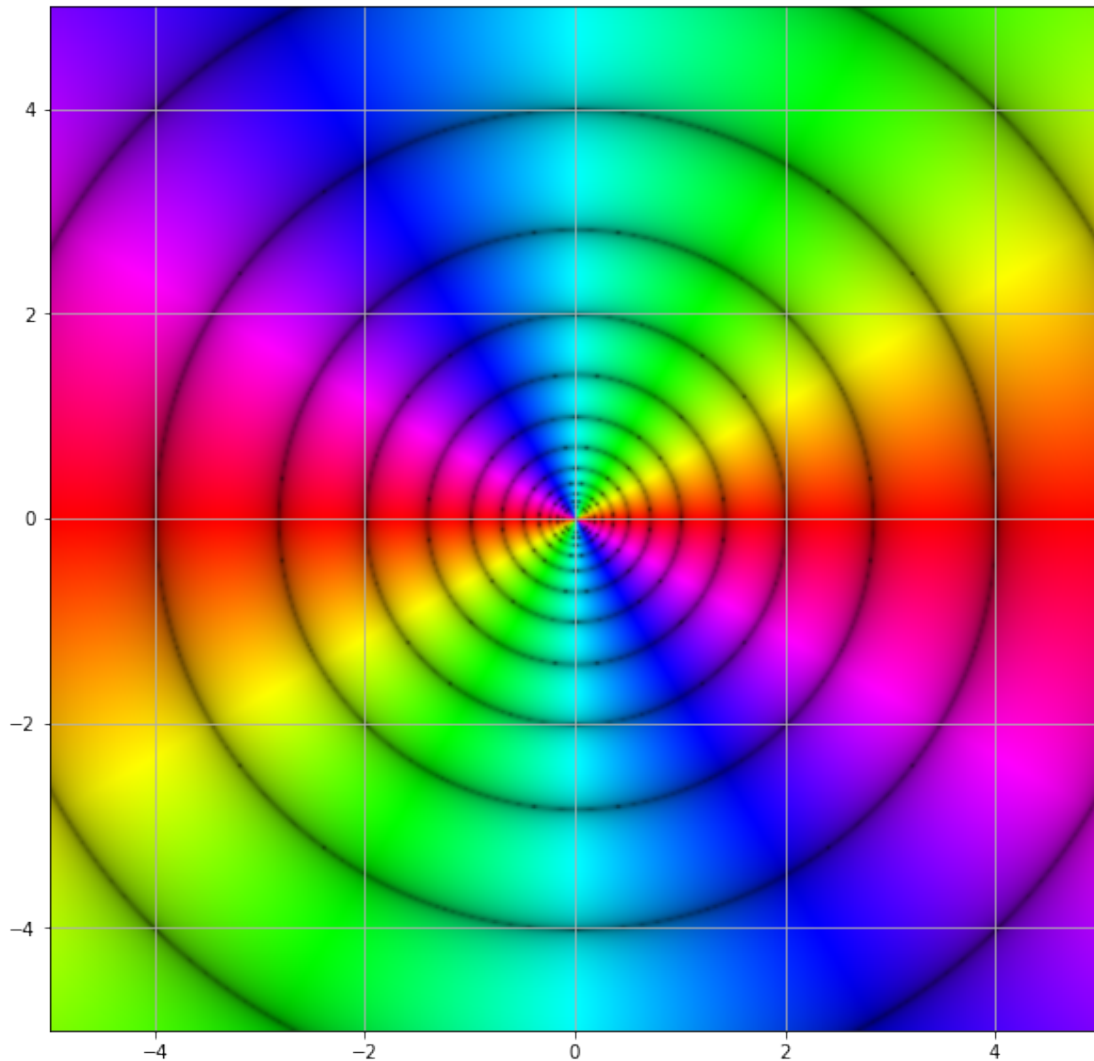
Moralité : Dans ce qui va suivre, regardez toujours dans quel sens varient les couleurs, c'est essentiel.

1.2 2. Polynômes

1.2.1 2.1 Commençons par le commencement

Quel est le polynôme le plus simple dont le degré soit au moins 2 ? X^2 ? Ou $X^2 - 1$? Ou ... Essayons les deux. À vous d'en tester d'autres.

```
In [10]: M = 5  
         plot_complex(lambda z: z ** 2, extent=[-M, M, -M, M])
```

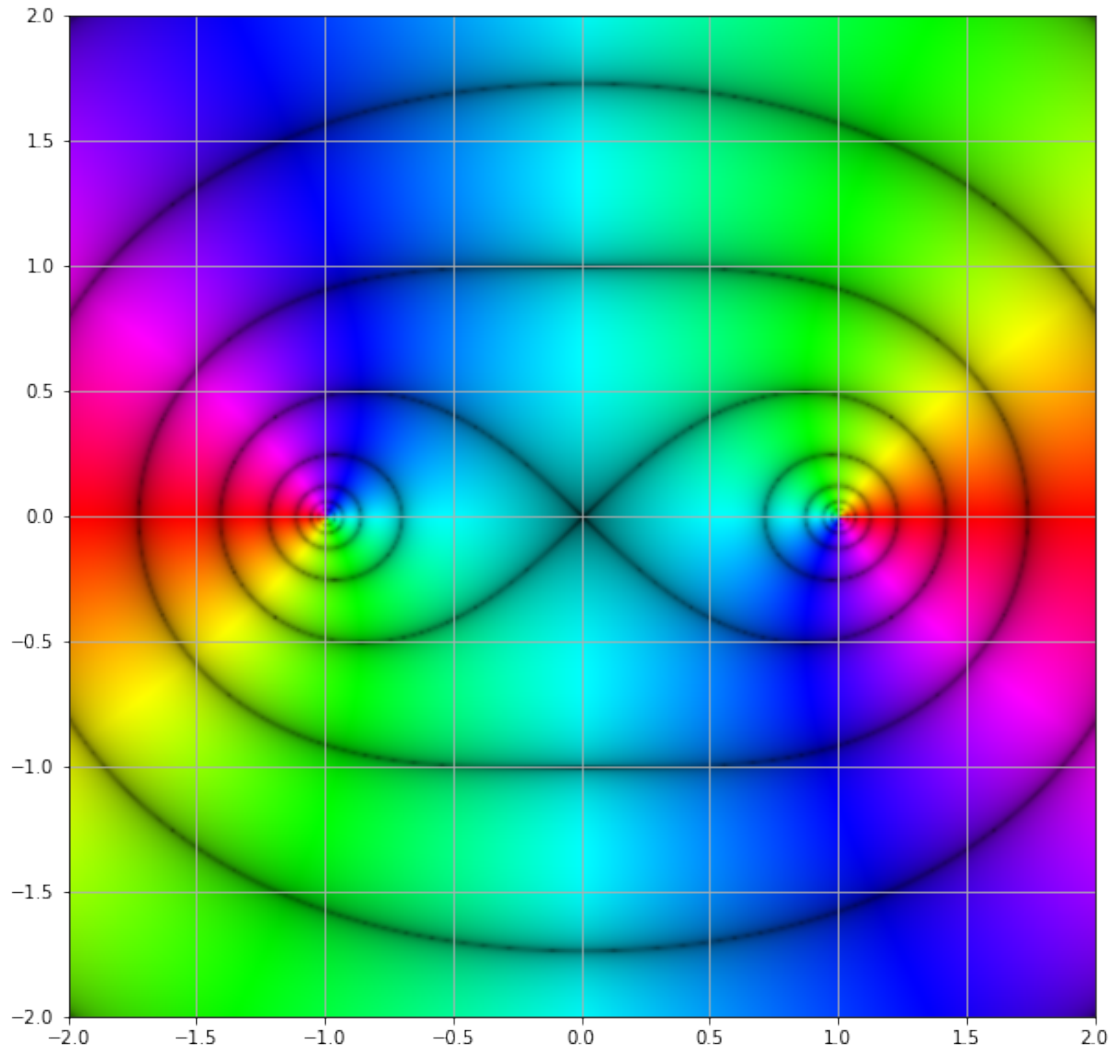


Voyez-vous la racine double ??? Le cycle des couleurs est parcouru **DEUX** fois à chaque tour autour de l'origine : imaginez un nombre complexe z qui se promène sur un cercle de centre O . Quand z a fait **UN** tour, z^2 a fait **DEUX** tours. Pourquoi ? Tout simplement parce que $\arg(z^2) = 2 \arg z$.

Essayons maintenant $X^2 - 1$.

1.2.2 2.2 Deux racines simples

```
In [11]: M = 2  
         plot_complex(lambda z: z ** 2 - 1, extent=[-M, M, -M, M])
```



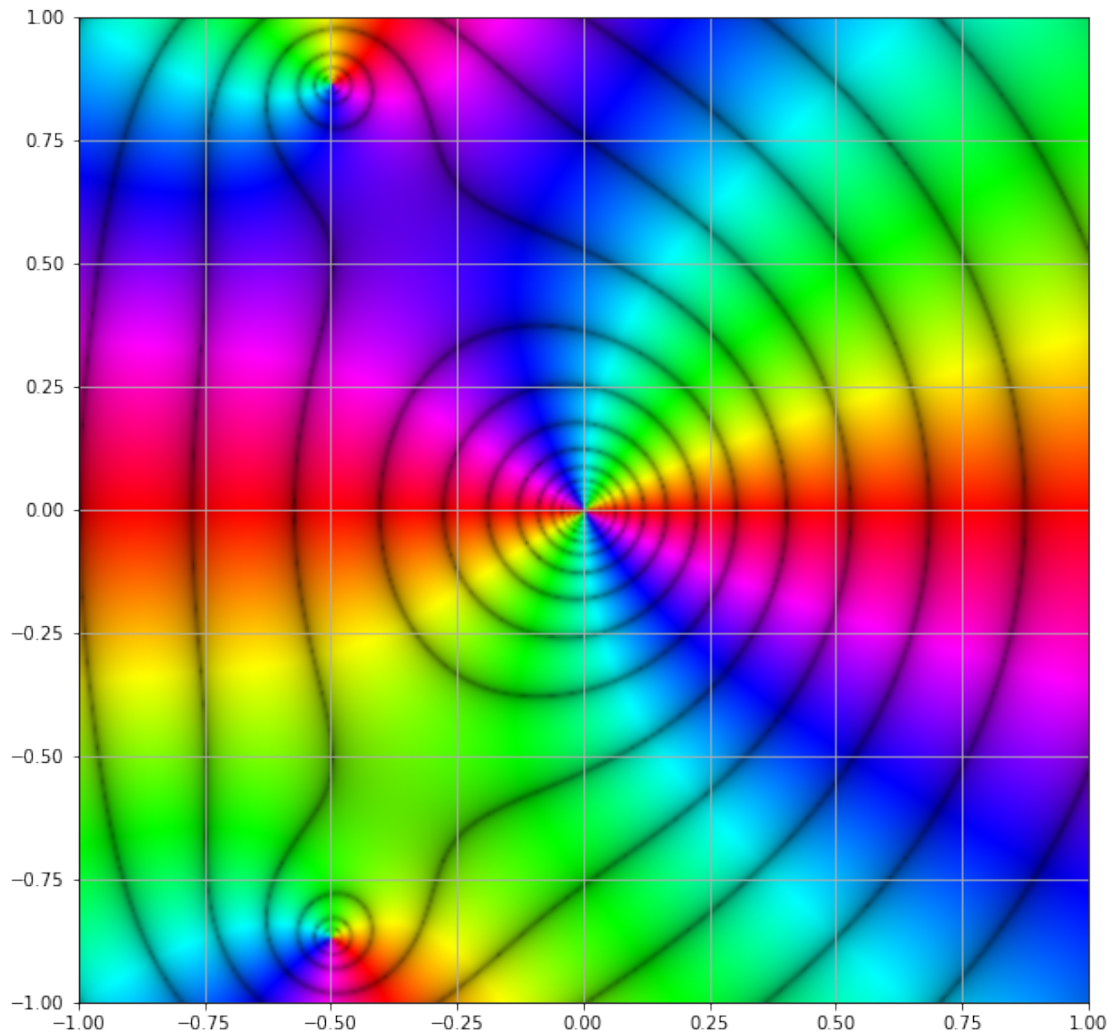
Vous devriez voir les deux racines simples (un cycle de couleurs) -1 et 1 . Remarquez aussi les ellipses entourant les deux racines simultanément : **DEUX** cycles de couleurs !

Exercice : Mettez M à 100 ci-dessus. Alors, X^2 et $X^2 - 1$ sont-ils si différents, vus de loin ?

1.2.3 2.3 Une racine double, deux racines simples

Essayons un peu plus compliqué, le polynôme $X(X^2 + X + 1)$.

```
In [12]: M = 1  
         plot_complex(lambda z: z ** 2 * (z ** 2 + z + 1), extent=[-M, M, -M, M])
```



Exercice : Combien de cycles de couleurs sur les courbes entourant chaque racine ? Et sur les “cercles” qui entourent l’ensemble de toutes les racines ?

Exercice : À vous de jouer avec vos polynômes préférés. Racines simples, racines multiples, $X^n - 1$, polynômes aléatoires, il y a de quoi faire.

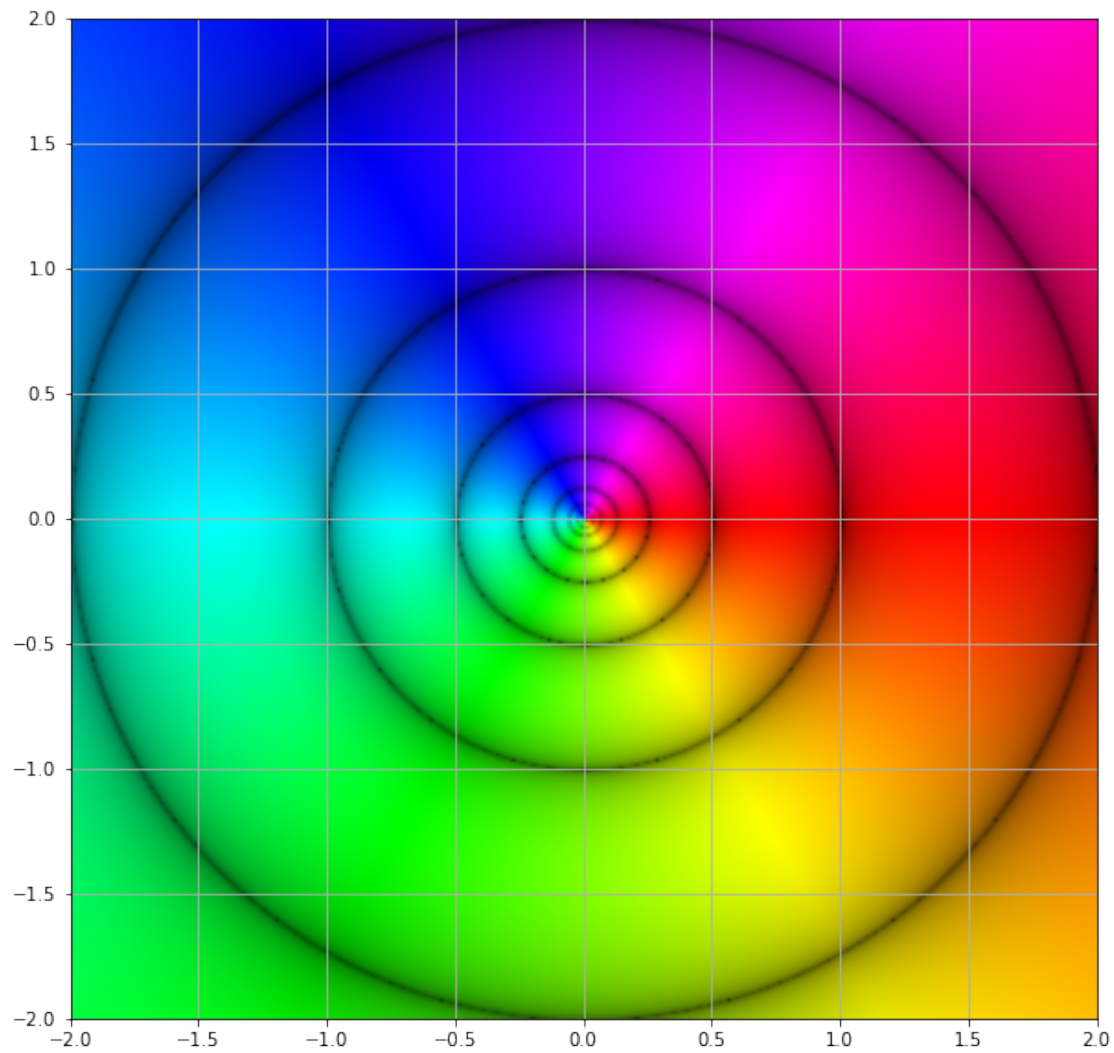
1.3 3. Fractions rationnelles

1.3.1 3.1 La fraction la plus simple

Commençons par $\frac{1}{z}$.

In [14]: $M = 2$

```
plot_complex(lambda z: 1 / z, extent=[-M, M, -M, M])
```

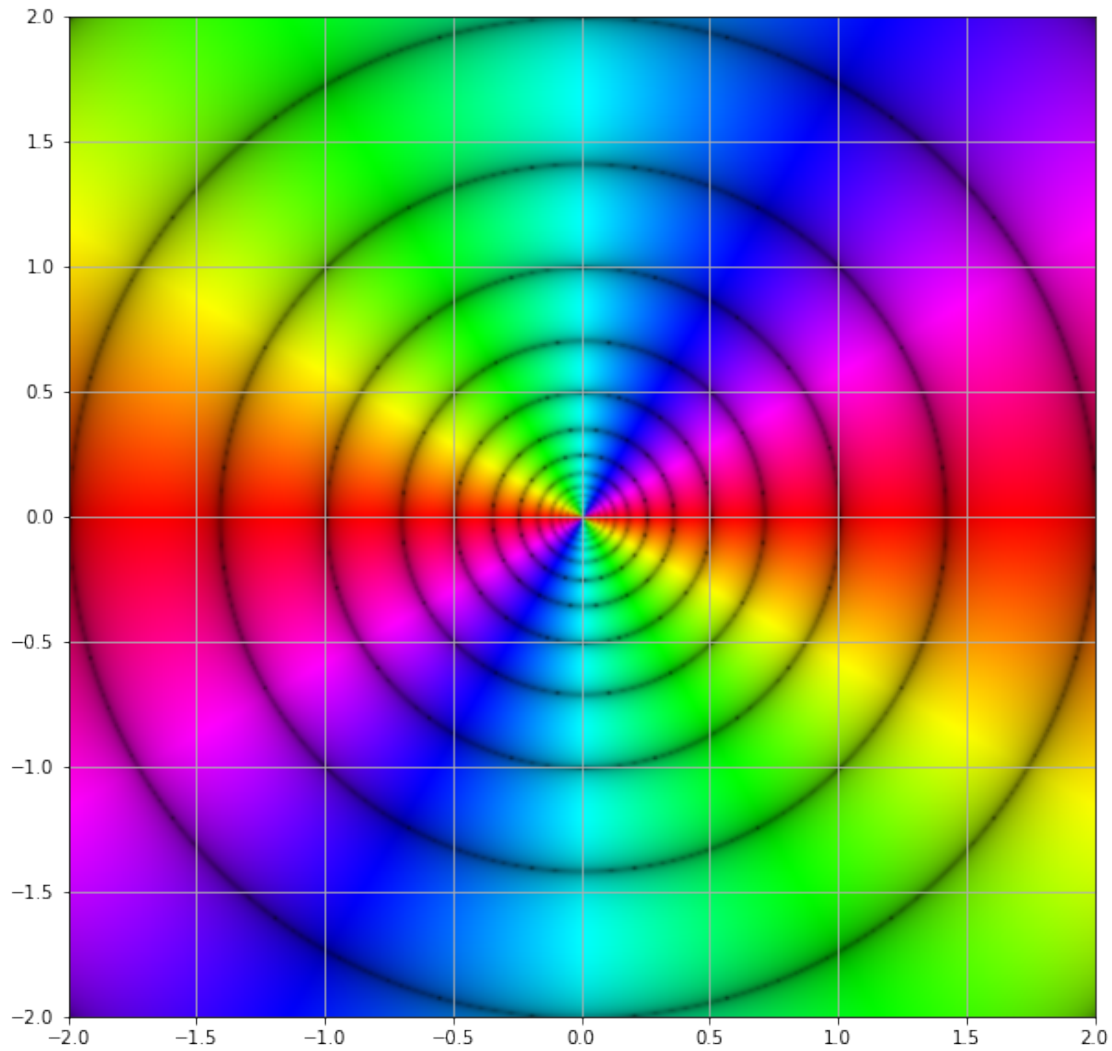


Bon, c'est comme $z \mapsto z$? Non, car les couleurs tournent en sens inverse. Pourquoi ? Parce que $\arg \frac{1}{z} = -\arg z$.

1.3.2 3.2 Pôle double

Traçons maintenant $\frac{1}{z^2}$.

```
In [15]: M = 2
          plot_complex(lambda z: 1 / z ** 2, extent=[-M, M, -M, M])
```



Deux cycles de couleurs par tour, en sens inverse. Sans preuve :

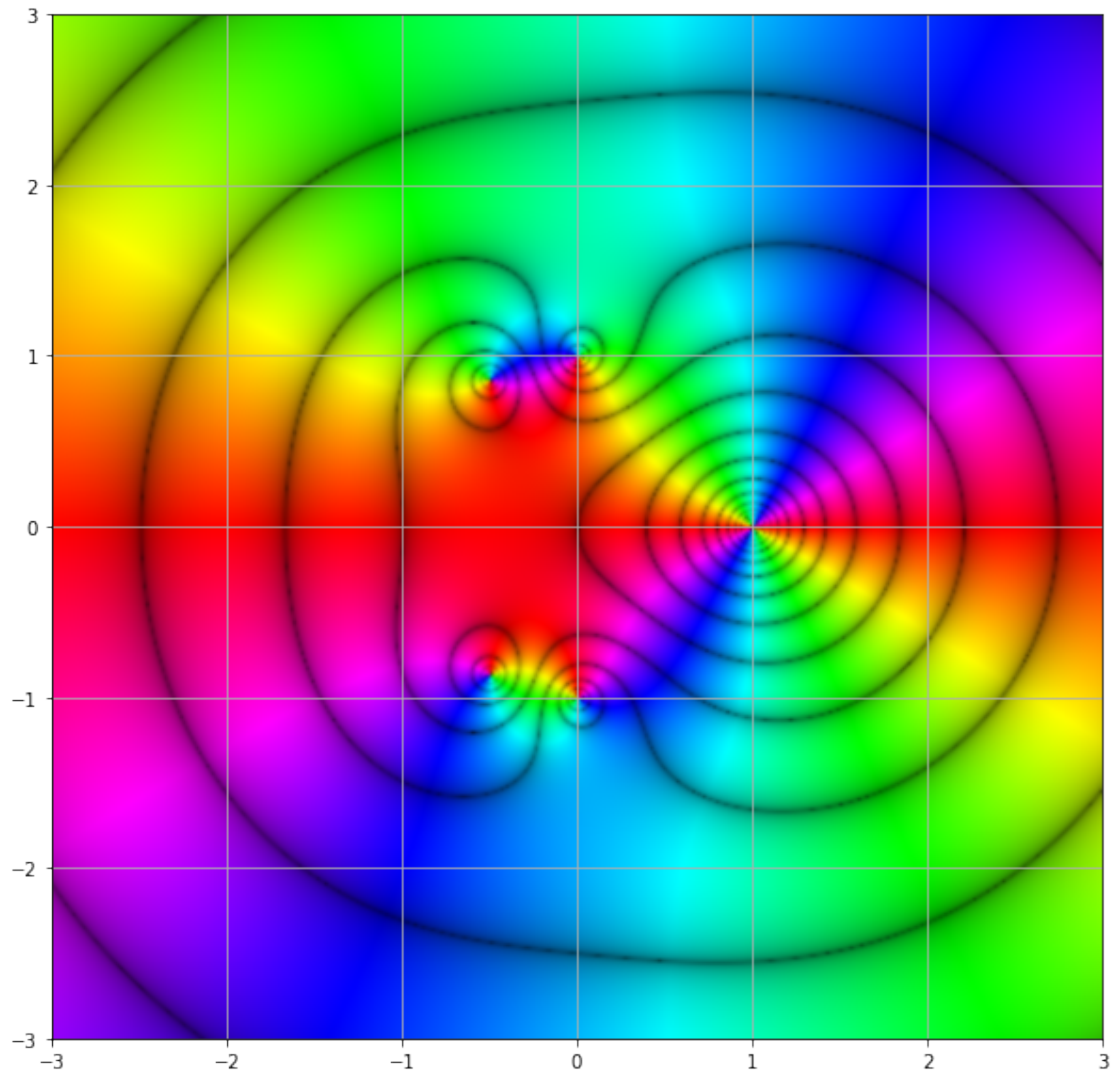
- Pour une racine d'ordre n , n cycles de couleurs en sens normal
- Pour un pôle d'ordre n , n cycles de couleurs en sens inverse

1.3.3 3.3 Une fraction avec quelques pôles et racines

Pour terminer, la fraction $\frac{X^2+1}{(X-1)^2(X^2+X+1)}$.

In [16]: $M = 3$

```
plot_complex(lambda z: (z ** 2 + 1) / ((z - 1) ** 2 * (z ** 2 + z + 1)), extent=[-M, M,
```



Deux racines simples, $-i$ et i . Un pôle double, 1 , et deux pôles simples, j, j^2 . Regardez les sens de rotation, le nombre de cycles et le nombre de cercles concentriques, ils permettent de distinguer les pôles et les racines.

Exercice : Mettez M à 10 et m à 0.05. Se pourrait-il que ...

Proposition : Le nombre de cycles de couleurs sur une courbe qui entoure les racines et les pôles est égal au nombre de racines moins le nombre de pôles, comptés avec leur multiplicité. C'est à dire au degré de la fraction !

Exercice : Comment reconnaît-on une fraction de degré 0 ? De degré < 0 ? De degré > 0 ? Expérimentez ...

1.4 4. Fonctions holomorphes

1.4.1 4.1 C'est quoi ?

Définition : Soit $\Omega \subset \mathbb{C}$ un ensemble *ouvert*. Soit $f : \Omega \rightarrow \mathbb{C}$. On dit que f est **holomorphe** sur Ω lorsque f est dérivable en tout point de Ω .

Dérivable *au sens complexe*, bien entendu. C'est à dire que pour tout $z \in \Omega$, la quantité $\frac{f(z+h)-f(z)}{h}$ a une limite lorsque $h \rightarrow 0$, $h \in \mathbb{C}$. La définition de limite est la même que pour les fonctions de variable réelle, sauf qu'ici il est question de module et pas de valeur absolue. Et la limite de nos taux d'accroissements est évidemment appelée la dérivée de f en z et est notée $f'(z)$. La condition de dérivabilité au sens complexe est une condition très restrictive, beaucoup plus que la dérivabilité pour les fonctions de variable réelle. Par exemple, on peut montrer qu'une fonction holomorphe est automatiquement **indéfiniment dérivable**. On peut toujours rêver d'un tel résultat pour les fonctions de variable réelle !

La théorie des fonctions holomorphes est une vaste et riche théorie, remplie de résultats très profonds.

1.4.2 4.2 Qui est holomorphe ?

Si vous ouvrez votre cours de maths sur la dérivation, vous verrez sans peine que les théorèmes sur les opérations sur les fonctions dérivables s'étendent aux fonctions $\mathbb{C} \rightarrow \mathbb{C}$: somme, produit, quotient (là où le dénominateur ne s'annule pas), composée. Après tout, la démonstration de ces théorèmes ne repose que sur les propriétés de la valeur absolue, qui sont les mêmes que celles du module.

Notation : Nous noterons $\mathcal{H}(\Omega)$ l'ensemble des fonctions holomorphes sur l'ouvert Ω . Le paragraphe précédent nous dit que cet ensemble est stable pour l'addition, le produit des fonctions, etc.

Proposition : Les fonctions constantes, les polynômes, sont holomorphes sur $\mathbb{C}$ et leur dérivée est formellement la même que celle obtenue pour les fonctions à valeurs réelles.

Démonstration : Je passe le cas trivial des fonctions constantes. Si nous montrons que $f : z \mapsto z$ est holomorphe sur $\mathbb{C}$, ce sera fini, par stabilité de $\mathcal{H}(\mathbb{C})$ par addition et multiplication. Soit donc $z \in \mathbb{C}$. Soit $h \in \mathbb{C}^*$. On a

$$\frac{f(z+h)-f(z)}{h} = \frac{(z+h)-z}{h} = 1$$

Cette quantité constante tend évidemment vers 1 lorsque $h \rightarrow 0$. Ainsi f est holomorphe sur $\mathbb{C}$ et pour tout $z \in \mathbb{C}$ on a $f'(z) = 1$.

Corollaire : Une fraction rationnelle est holomorphe sur $\mathbb{C}$ privé de l'ensemble des pôles de la fraction.

Démonstration : C'est le quotient de deux polynômes, qui, eux, sont holomorphes.

Il existe bien d'autres fonctions holomorphes. Nous en verrons des tas d'exemples dans ce notebook.

1.4.3 4.3 Qui n'est pas holomorphe ?

Il existe des fonctions en apparence très régulières qui ne sont pas holomorphes. Voici l'exemple à ne pas manquer.

Proposition : La fonction $f : z \mapsto \bar{z}$ n'est holomorphe sur aucun ouvert de $\mathbb{C}$.

Démonstration : $\frac{\overline{z+h}-\bar{z}}{h} = \frac{\bar{h}}{h}$. Lorsque h tend vers 0 en étant réel, cette quantité tend vers 1. Mais si h tend vers 0 en étant imaginaire pur, cette quantité tend vers -1 . Pas de limite, donc, lorsque h tend vers 0.

Idée en l'air : Les "fonctions du genre $z \mapsto f(z) = g(\bar{z})$ " ne sont pas holomorphes. Évidemment, tel quel cela n'a pas de sens. Par exemple $f(z) = \frac{\bar{z}}{z}$ est holomorphe sur $\mathbb{C}^*$:-). Je ne préciserai pas plus avant.

Proposition : Soit $\Omega \subset \mathbb{C} \rightarrow \mathbb{R}$ holomorphe, où Ω est un ouvert *connexe*. Alors f est constante.

Démonstration : Ce résultat est assez surprenant. Cela veut dire qu'une fonction holomorphe est soit triviale (constante) soit "compliquée".

Soit $f : \Omega \subset \mathbb{C} \rightarrow \mathbb{R}$ une fonction holomorphe à valeurs réelles. Soit $z \in \Omega$. Pour h réel, le quotient $\frac{f(z+h)-f(z)}{h}$ est réel. Si cette quantité a une limite lorsque $h \rightarrow 0$, alors elle doit être réelle : $f'(z) \in \mathbb{R}$. Mais pour h imaginaire pur, le quotient $\frac{f(z+h)-f(z)}{h}$ est imaginaire pur. Si cette quantité a une limite lorsque $h \rightarrow 0$, alors elle doit être imaginaire pure : $f'(z) \in i\mathbb{R}$. Ainsi, $f'(z) = 0$. Conclusion $f' = 0$ sur Ω . Nous admettrons ici que la connexité de Ω implique que f est constante.

Conséquence : Une fonction holomorphe ne peut pas être à valeurs uniquement réelles (à moins qu'elle ne soit constante). On peut de même montrer qu'elle ne peut pas être à valeurs uniquement imaginaires pures. Plus généralement, un théorème appelé le **théorème de l'image ouverte** nous dit que si $f \in \mathcal{H}(\Omega)$ alors f est ouverte, c'est à dire que l'image de tout ouvert par f est encore un ouvert.

1.4.4 4.4 Qu'allons nous faire ?

Nous allons dans les deux prochains notebooks explorer des prolongements holomorphes à $\mathbb{C}$ (ou presque) des fonctions usuelles : exponentielle, logarithmes, racines carrées, fonctions trigonométriques. Histoire d'alléger, nous admettrons certains résultats mathématiques.

In []: