

Transformation complexe $z \mapsto z^2$

La visualisation d'une fonction f à variable réelle x s'opère d'une manière naturelle en construisant dans le plan rapporté à un repère orthonormé $(O; \vec{i}, \vec{j})$ l'ensemble des points de coordonnées $(x; y)$ où $y = f(x)$.

La visualisation d'une fonction f à variable complexe z telle que la fonction carrée : $z \mapsto z^2$ ne peut s'opérer d'une manière aussi simple attendu que la représentation d'un nombre complexe z requiert déjà un plan et que l'image par f du nombre z requiert également un plan. Autrement dit, représenter une transformation complexe nécessite 4 dimensions, 2 pour la variable complexe z et 2 supplémentaires pour son image $f(z)$.

Le problème est que l'œil et le cerveau humain ne sont configurés que pour appréhender un univers tridimensionnel.

Comment représenter la transformation complexe $f: z \mapsto z^2$?

De la même manière qu'un cube (objet tridimensionnel) est visualisable par son projeté dans un plan (dimension 2), la représentation quadridimensionnelle d'une fonction à variable complexe pourra être visualisée via sa projection dans un espace tridimensionnel.

Soit $f: z \mapsto z^2$.

Considérons le nombre complexe $z = x + iy$.

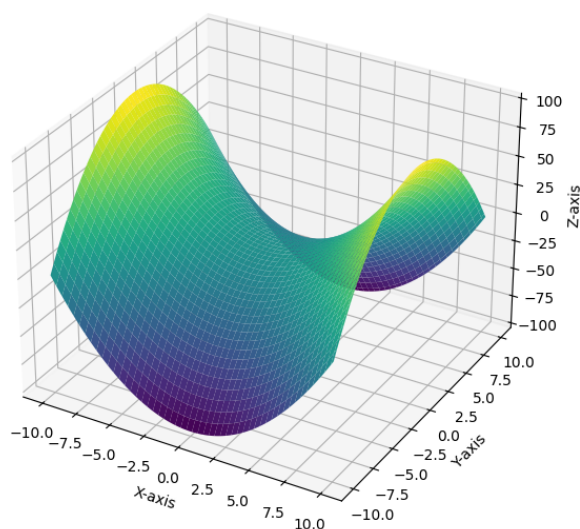
On a : $f(z) = z^2 = (x + iy)^2 = x^2 - y^2 + i(2xy)$

Donc : $Re(f(z)) = x^2 - y^2$ et $Im(f(z)) = 2xy$.

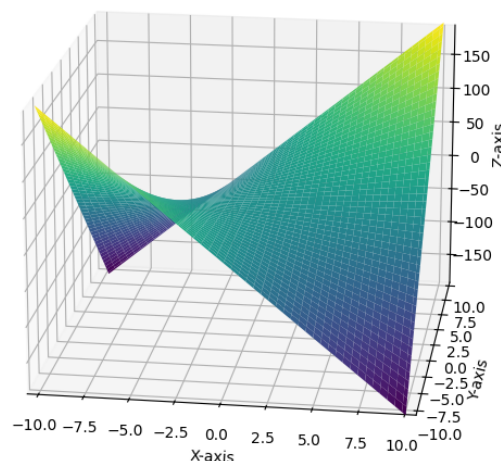
Pour représenter la fonction complexe carrée f , l'idée est de représenter les projections de la représentation quadridimensionnelle de f dans l'espace tridimensionnel $(x; y; Re(f(z)))$, puis dans l'espace tridimensionnel $(x; y; Im(f(z)))$.

A l'aide des bibliothèques python, numpy et matplotlib, il est aisé de programmer les représentations de ces deux projections, telles que ci-dessous :

3D Plot of $z = x^2 - y^2$



3D Plot of $z = 2xy$



Code Python permettant la visualisation de la projection de la représentation de f dans l'espace tridimensionnel $(x; y; Re(f(z)))$.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Define the range for x and y
5 x = np.linspace(-10, 10, 100)
6 y = np.linspace(-10, 10, 100)
7
8 # Create a meshgrid of x and y values
9 X, Y = np.meshgrid(x, y)
10
11 # Compute Z = x^2 - y^2
12 Z = X**2 - Y**2
13 |
14 # Create a 3D plot
15 fig = plt.figure(figsize=(10, 7))
16 ax = fig.add_subplot(111, projection='3d')
17
18 # Plot the surface
19 ax.plot_surface(X, Y, Z, cmap='viridis', alpha=1)
20
21 # Customize the plot
22 ax.set_title('3D Plot of z = x^2 - y^2', fontsize=14)
23 ax.set_xlabel('X-axis')
24 ax.set_ylabel('Y-axis')
25 ax.set_zlabel('Z-axis')
26
27 # Show the plot
28 plt.savefig("plot.png")
29 plt.show()
```