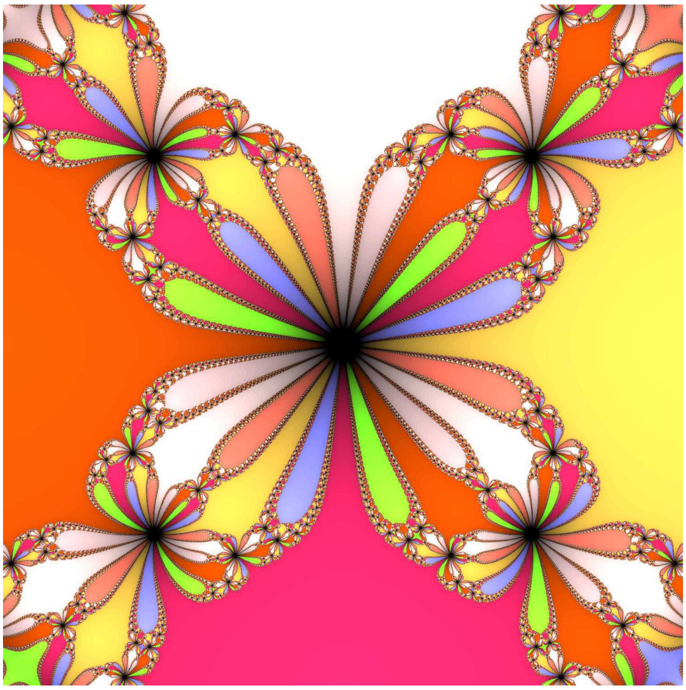
TP n°21 : Méthode de NEWTON



1 La méthode de NEWTON

1.1 Application du cours

1. *Retrouver* la suite $(x_n)_{n\in\mathbb{N}}$ définie par récurrence, de premier terme x_0 donné, qui est au cœur de la méthode de NEWTON. Il faut chercher à *établir* cette suite par vous-mêmes, avec un dessin, *sans regarder le cours*.
2. Implémenter cette méthode dans une fonction `newton(f, fprime, x0, eps)` qui prendra en entrée une fonction `f` de dérivée `fprime`, de première approximation `x0`, de précision `eps` et qui renverra le premier terme x_k de la suite tel que $|x_k - x_{k-1}| \leq \epsilon$.
3. Déterminer une approximation de $\sqrt{2}$ à 10^{-15} près avec cette méthode.
4. Modifier la fonction pour qu'elle renvoie de plus le nombre d'itérations effectuées. Combien d'itérations sont nécessaires pour trouver l'approximation de $\sqrt{2}$ précédente.
5. Comparer avec le nombre d'itérations nécessaire pour la méthode par dichotomie.

1.2 Convergence vers différentes racines

On considère l'équation $f(x) = x^3 - 2x^2 + 1 = 0$.

6. Déterminer la dérivée et les racines de f.
7. Implémenter la méthode de NEWTON, avec comme critère d'arrêt $|f(x)| \leq \epsilon$ et comme point de départ un paramètre $x0$.
8. Résoudre l'équation précédente pour $x_0 = -0.5$, $x_0 = 0.5$ et $x_0 = 1.5$. Qu'en pensez-vous ?
9. Résoudre l'équation précédente pour $x_0 = 0$. Qu'en pensez-vous ?

10. Pour x_0 qui varie entre -1 et 3 avec un pas de 0.1, afficher le couple formé par x_0 et la racine trouvée par la méthode de NEWTON. Commenter.
11. Résoudre l'équation précédente pour x_0 qui varie entre -5 et 10 avec un pas de 0.001 et représenter graphiquement la valeur de la solution trouvée en fonction de x_0. Commenter.

2 Fractals de NEWTON

Commencer par aller lire rapidement l'article suivant :

http://images.math.cnrs.fr/La-methode-de-Newton-et-son.html

2.1 Gestion des images

On peut générer des images en utilisant le module PIL dédié :

PYTHON
```python
from PIL import Image
```

Pour créer une image : `img = Image.new('RGB', (largeur, hauteur))` chaque pixel sera représenté par un triplet (R, G, B) d'entiers codés sur un octet (entre 0 et 255). Le triplet $(0, 0, 0)$ représente le noir, et le triplet $(255, 255, 255)$ le blanc.

Pour accéder aux pixels on peut faire : `pixels = img.load()`. Ensuite, il suffit de demander `pixels[i,j]` pour lire ou modifier un pixel (qui correspond à un tuple).

On peut ensuite appeler `img.show()` pour visualiser l'image ou `img.save(nom_fichier)` pour la sauvegarder.

2.2 Un fractal de NEWTON

On se propose d'appliquer la méthode de Newton à la fonction polynomiale définie par

$$P(z) = (z-1)\left(z - a + \frac{1}{2}\right)\left(z + a + \frac{1}{2}\right)$$

avec une variable z complexe et où $a = -0{,}00508 + 0{,}33136i$.

Ce polynôme admet trois racines complexes : 1, $a - \frac{1}{2}$ et $-a - \frac{1}{2}$. On donne $P'(z) = 3z^2 - a^2 - \frac{3}{4}$.

> On peut manipuler des nombres complexes en python :
> - le complexe i s'écrivant `1j`, donc $1 + 2i$ s'écrit `1+2j` et $x + iy$ devient `x + 1j * y`. On peut aussi utiliser `complex(x, y)`
> - Le module s'obtient avec `abs`. On dispose aussi des fonctions `real` et `imag`.

On cherche à savoir, à partir d'un complexe z_0 (correspondant à un pixel donné de l'image), vers quelle racine va tendre la suite de la méthode de Newton.

Pour cela, on va itérer la méthode un certain nombre n fixé de fois, on va vérifier si la distance du nombre complexe obtenu z_n à une des racines α est suffisamment petite : $|z_n - \alpha| \leq \varepsilon$ avec ε fixé.

On tracera alors le point d'une couleur différente si on s'approche de l'une des trois racines, ou bien d'aucune.

On pourra aussi appliquer la fonction par exemple aux polynômes de la forme $P(z) = z^n - 1$.

12. Écrire une fonction `type_point(z0, P, dP, racines, n, epsilon)` prenant en argument

- un complexe `z0`,
- des fonctions `P` et `dP`,
- une liste `racines`,
- un entier `n`,
- un flottant `epsilon`

et renvoyant un entier égal

- au numéro de la racine α telle que $|z_n - \alpha| < \varepsilon$ où z_n est obtenu par la méthode de Newton à partir de $z0$ (et donc n itérations).
- à -1 s'il n'y en a pas.

13. Procéder au tracé dans une fonction

```
trace(P, dP, racines, mini=-1, maxi=1, n=15, epsilon=1e-8, N=200,
                    nom='newton')
```

selon le principe suivant :

- On peut choisir des couleurs aléatoires pour chacune des racines avec

```
couleurs = [tuple(np.random.randint(50, 200, 3)) for i in
                    range(len(racines))]
```

- On crée une image vierge de type RGB de taille (N, N) (tous les pixels sont noirs par défaut) dans une variable `img`.
- On récupère le contrôle des pixels dans une variable `pixel`.
- On crée des subdivisions régulières `x` et `y` de N points de $[mini, maxi]$ pour nos abscisses et nos ordonnées avec `linspace`.
- Pour chaque coordonnées (i, j) de pixel, on cherche le type du point $x_i + \mathrm{i}y_j$ où x_i et y_j sont obtenues à partir des subdivisions : `x[i]` et `y[j]`.
- On remplace le pixel de coordonnées (i, j) par la couleur correspondant au type de point si on n'a pas obtenu -1.
- On trace/sauvegarde (dans le fichier `'\{\}.png'.format(nom)`) l'image.

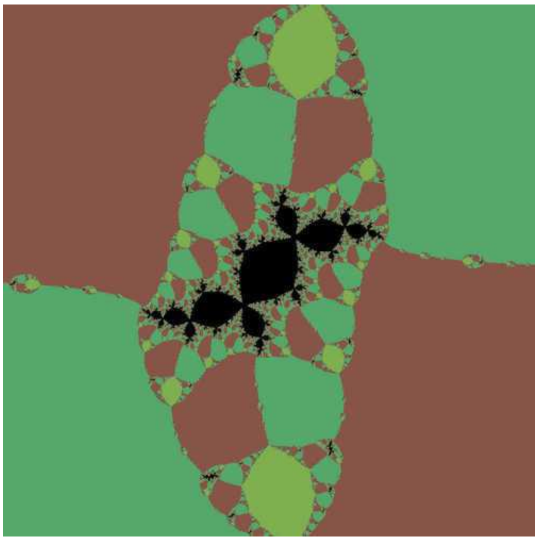
Déjà fini ?

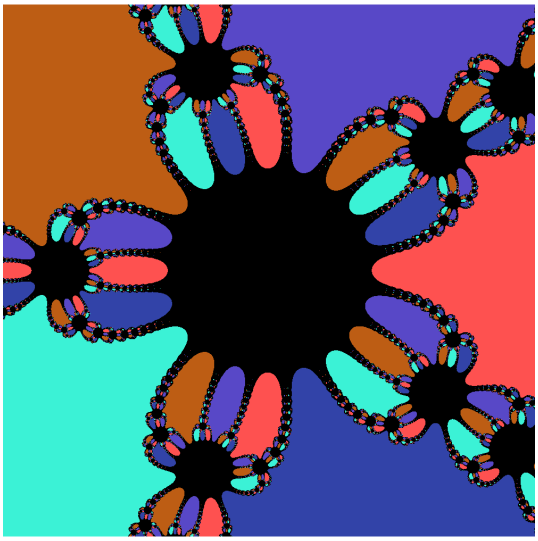
14. Reprendre le fractal de Newton et garder pour chaque point le nombre d'itérations nécessaires avant d'arriver à une racine à ε près.

- Dégrader la couleur du point en fonction de ce nombre.
- Essayer une version monochrome. (Pas besoin de connaître les racines !)
- Essayer de passer en négatif.

Pour $P(z) = (z-1)\left(z-a+\frac{1}{2}\right)\left(z+a+\frac{1}{2}\right)$.

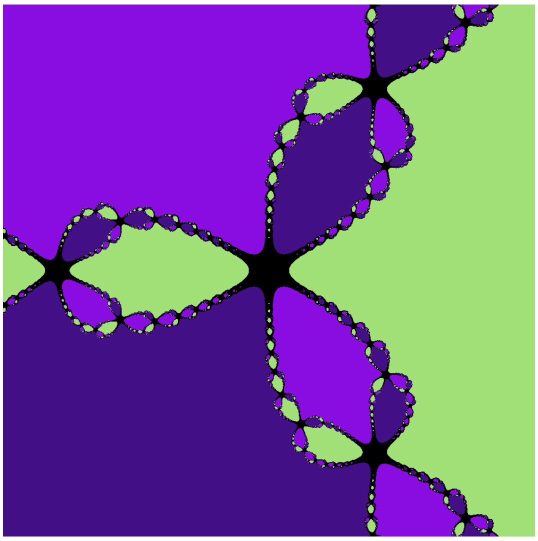
Pour $P(z) = z^5 - 1$.

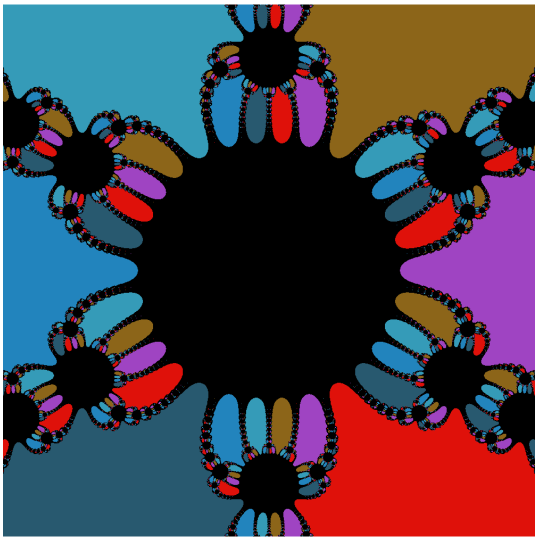




Pour $P(z) = z^3 - 1$.

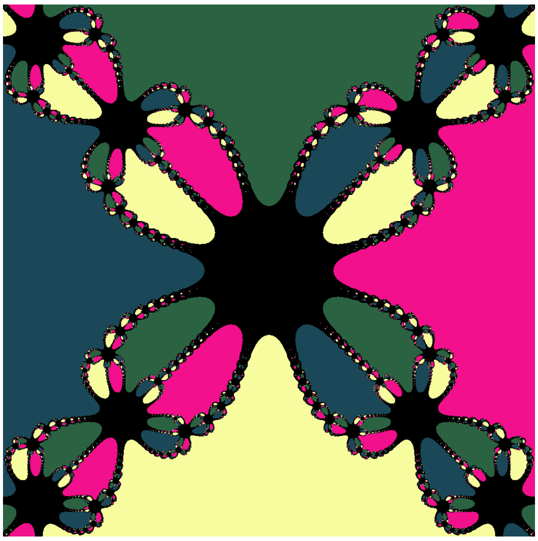
Pour $P(z) = z^6 - 1$.

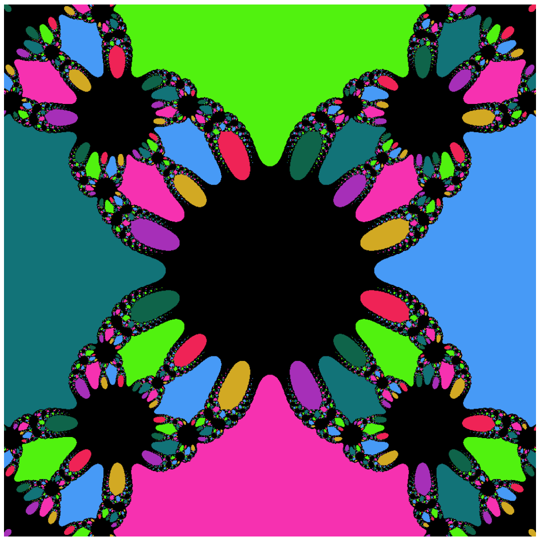




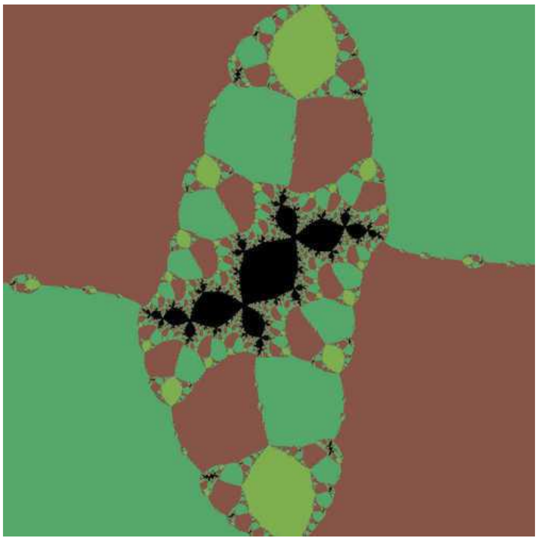
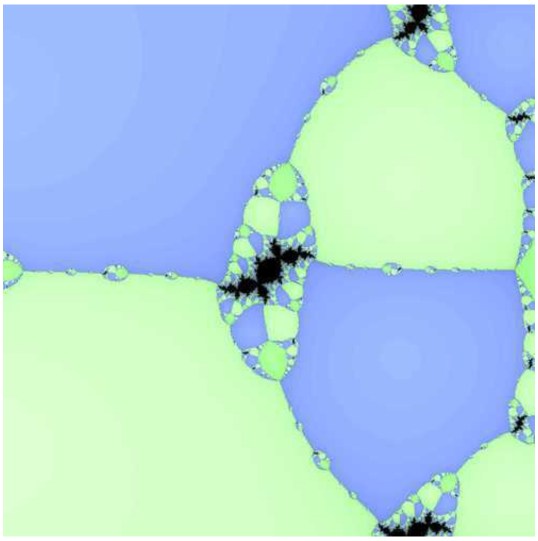
Pour $P(z) = z^4 - 1$.

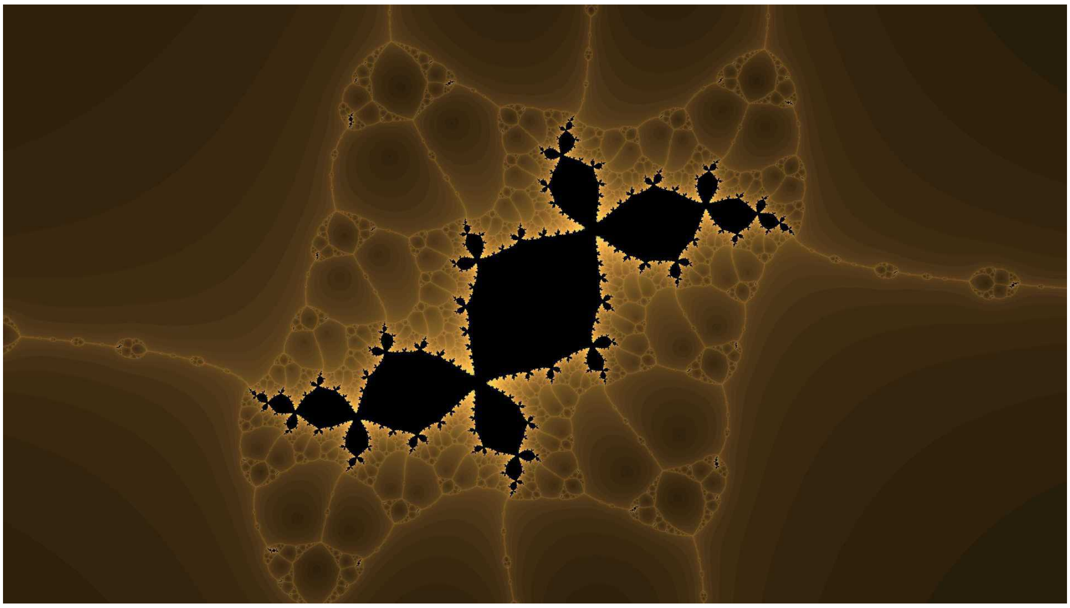
Pour $P(z) = x^8 + 15x^4 - 16$ dont les racines sont $\pm 1, \pm \mathrm{i}, \sqrt{2}(\pm 1 \pm \mathrm{i})$.



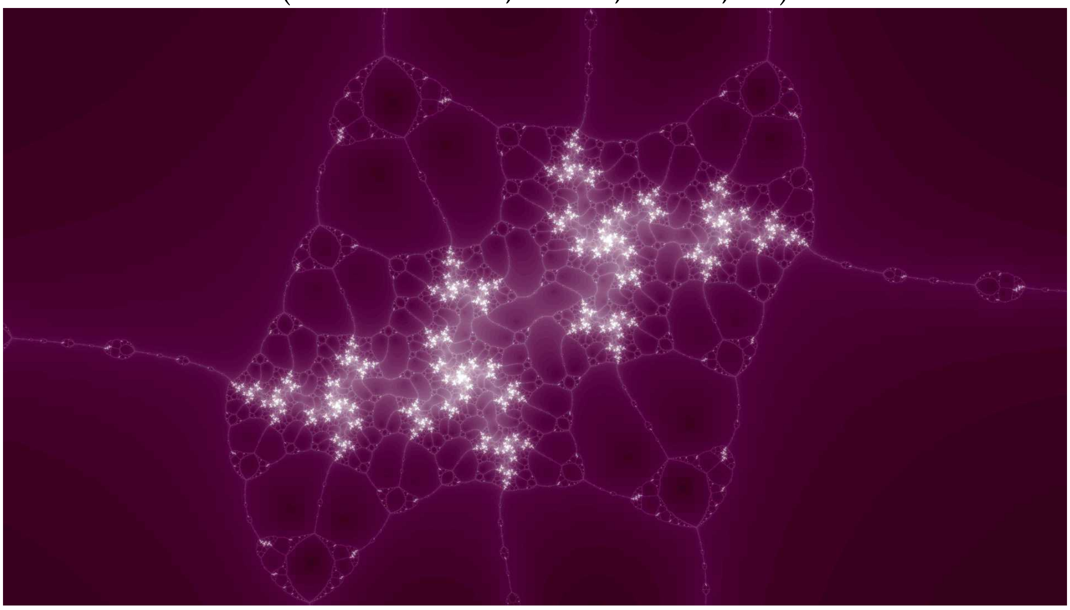


Pour $P(z) = (z-1)\left(z-a+\frac{1}{2}\right)\left(z+a+\frac{1}{2}\right)$.

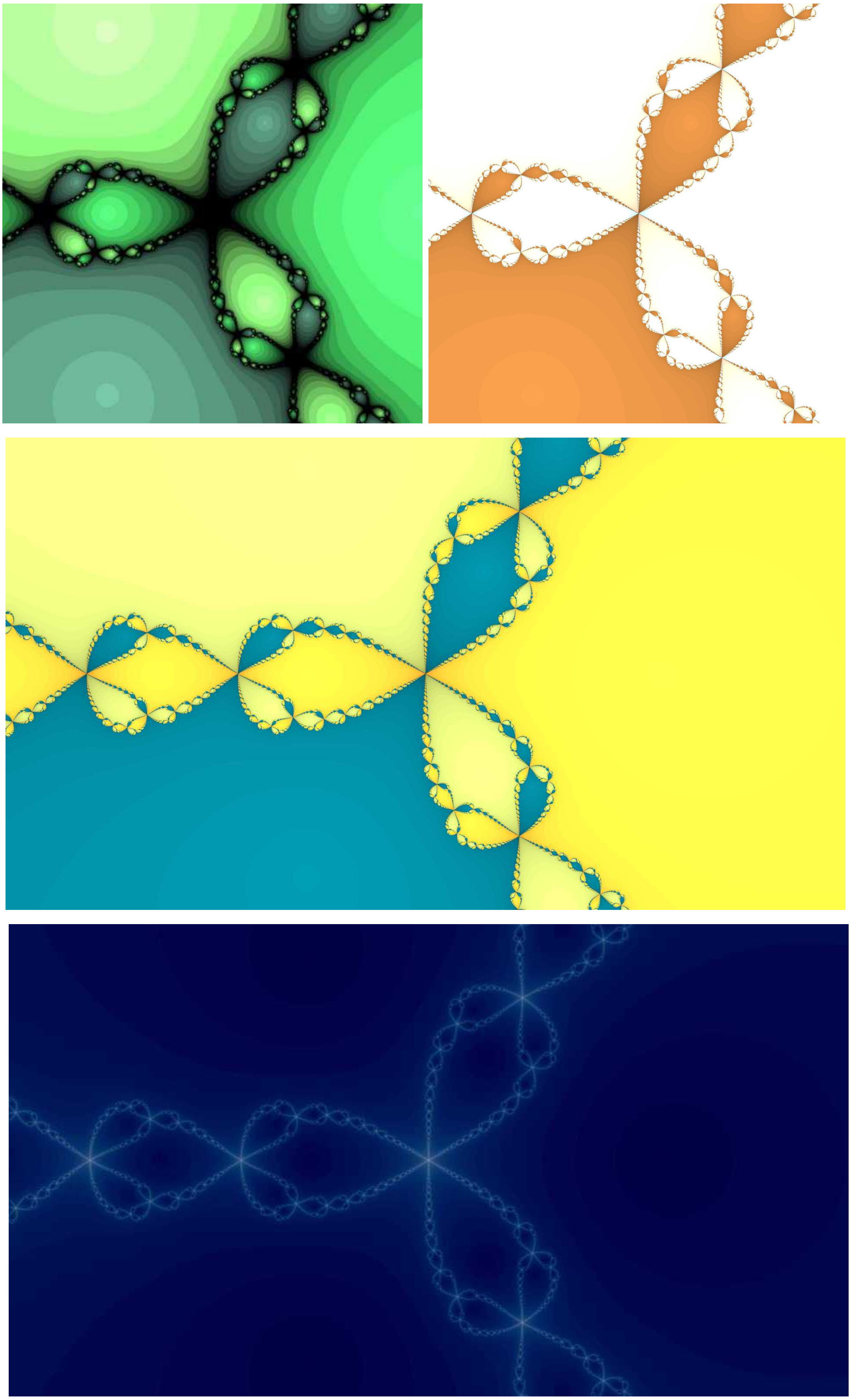





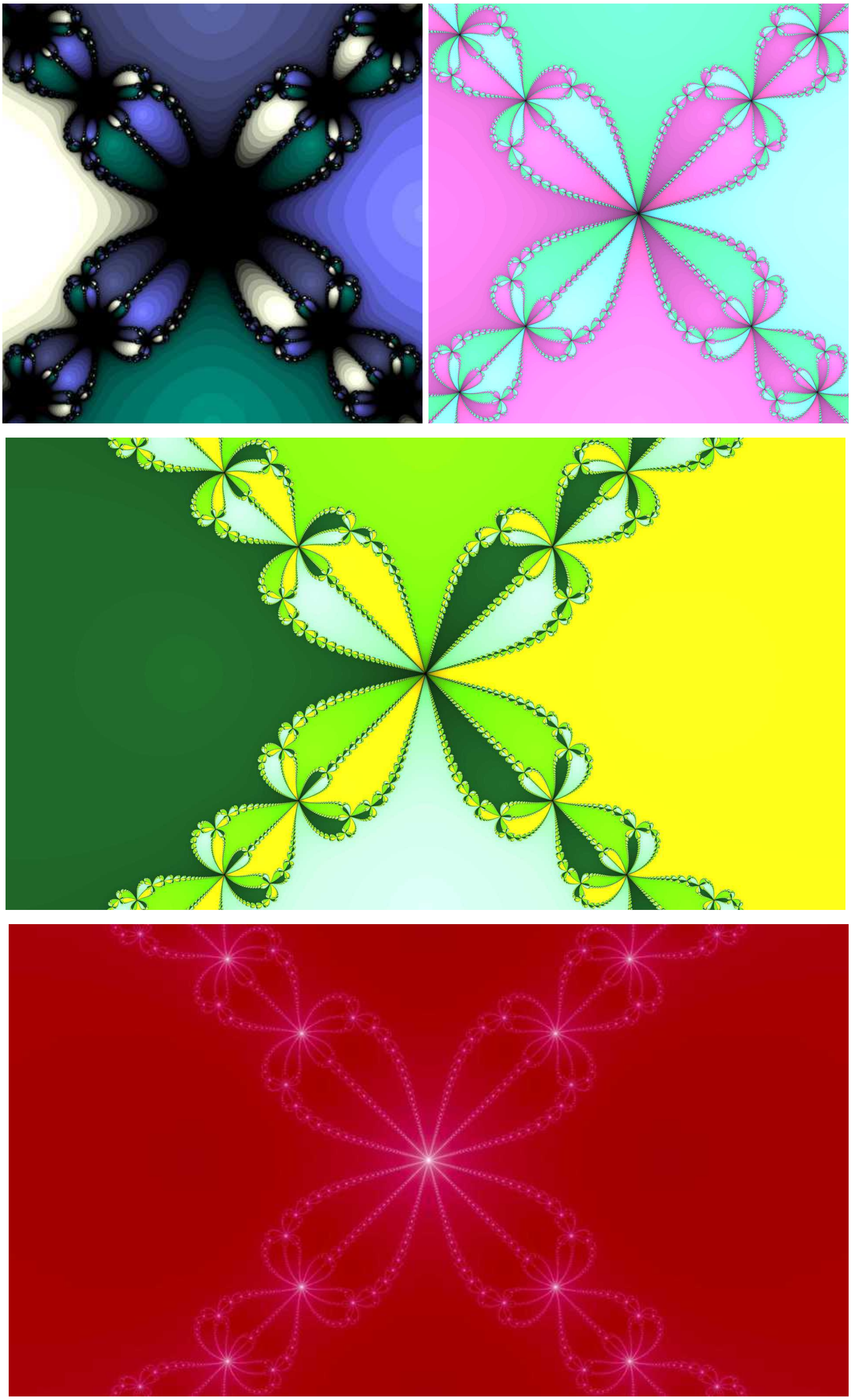
(Pour ce dernier, $a = -0{,}005 + 0{,}33i$.)



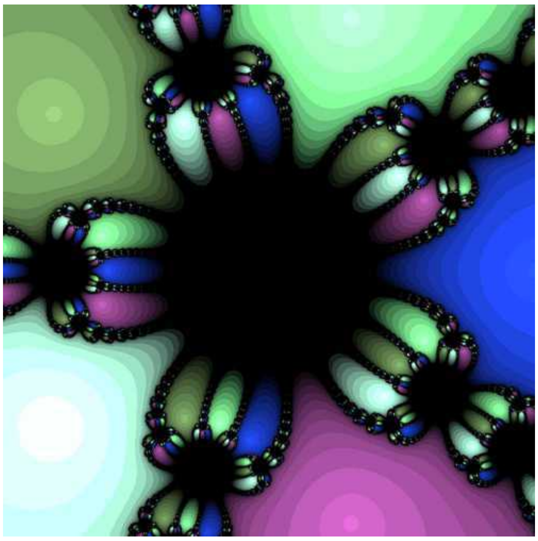
Pour $P(z) = z^3 - 1$.

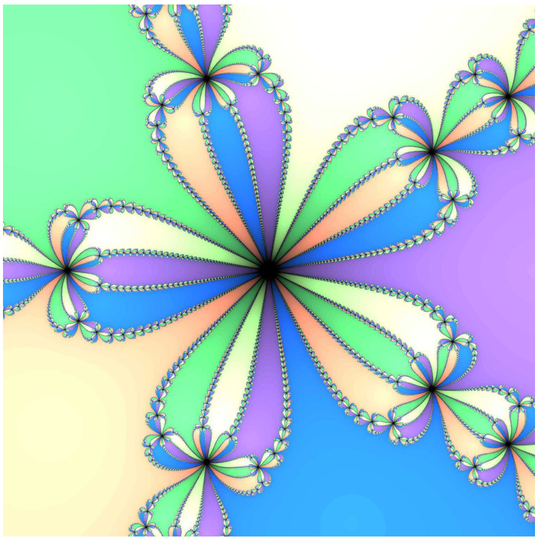


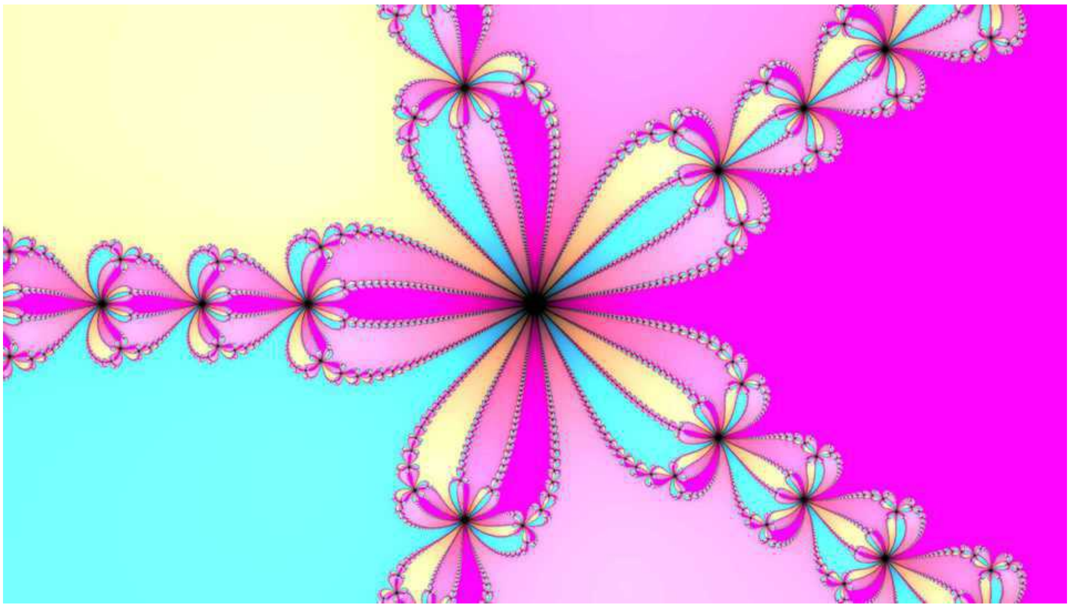
Pour $P(z) = z^4 - 1$.

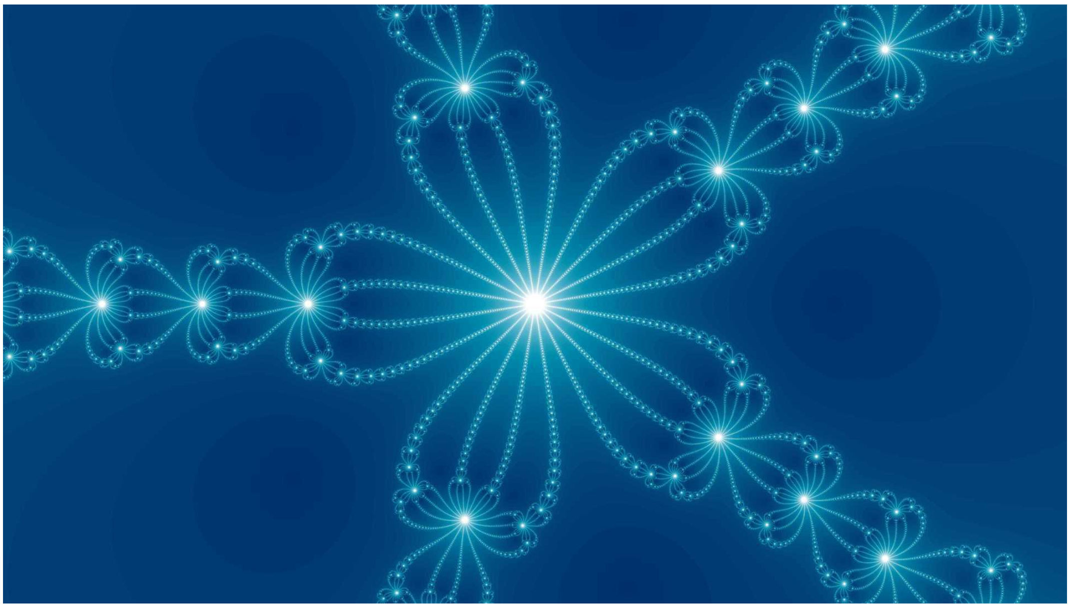


Pour $P(z) = z^5 - 1$.

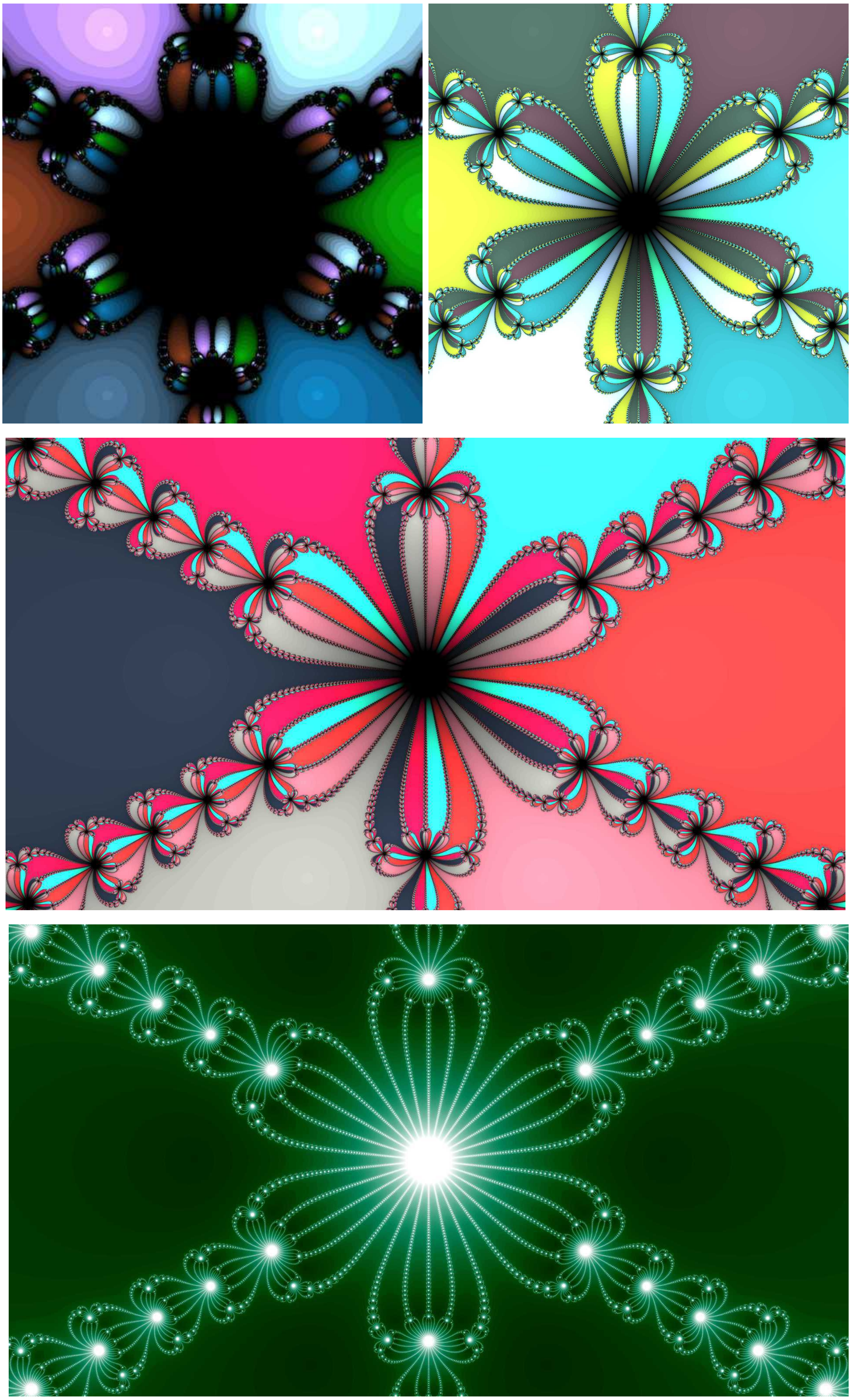




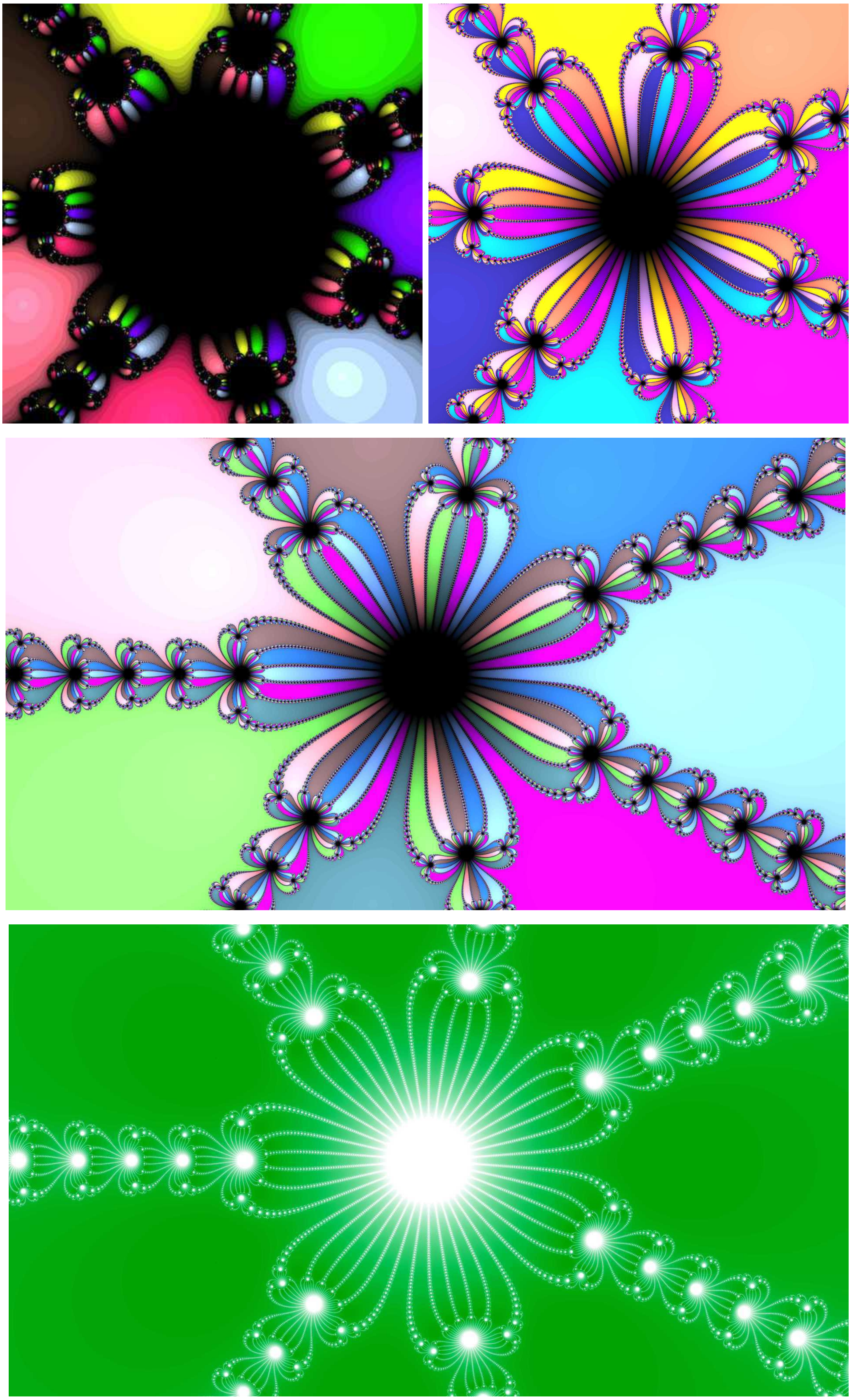




Pour $P(z) = z^6 - 1$.



Pour $P(z) = z^7 - 1$.



Pour $P(z) = z^8 + 15z^4 - 16$ dont les racines sont ± 1, $\pm \mathrm{i}$, $\sqrt{2}(\pm 1 \pm \mathrm{i})$.