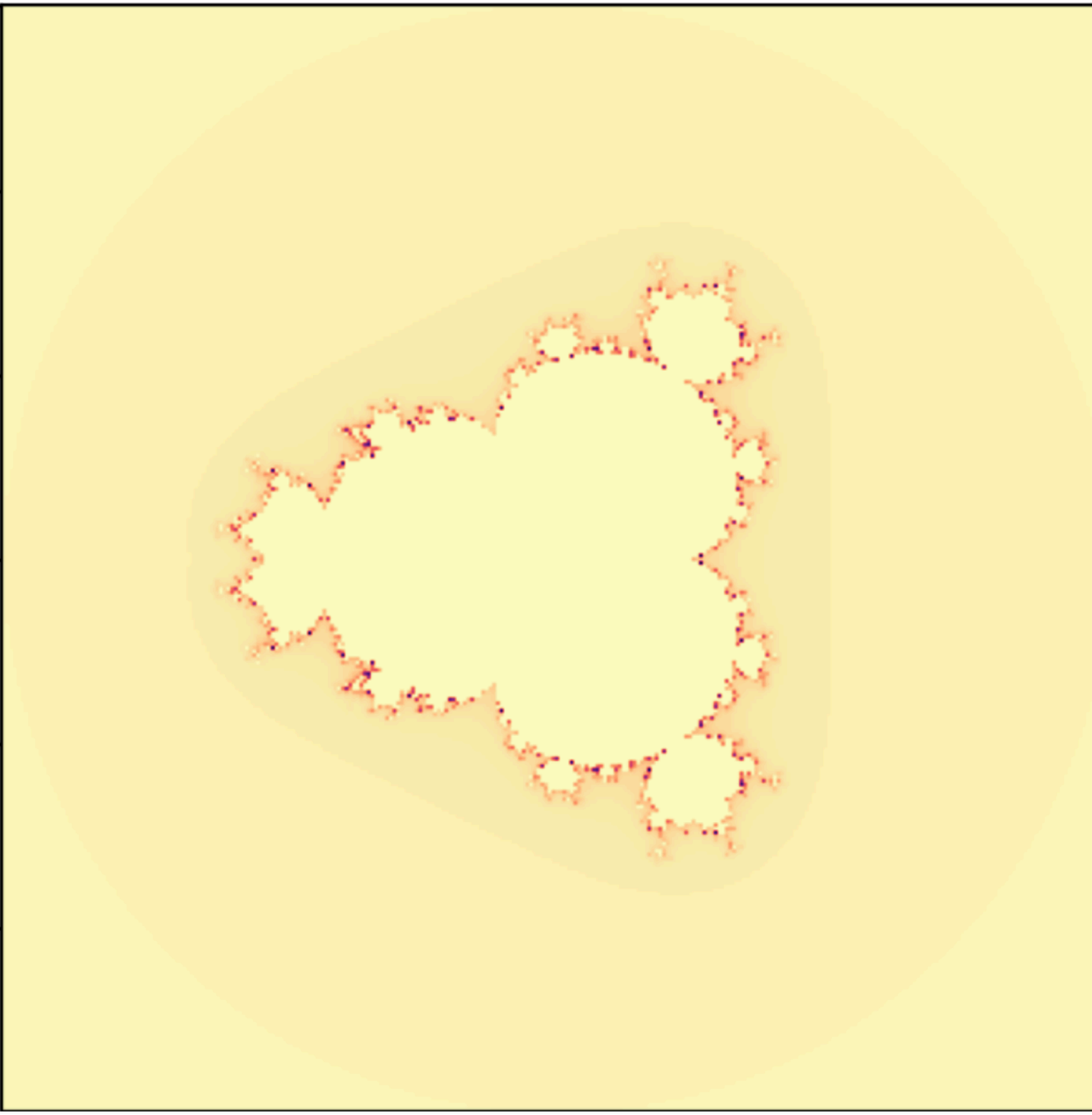
Nombres Complexe

Application : Multibrot , resize Julia et Cercle Julia

1er étudiant : GHERBI Kamel-Mahdi

2ème étudiant : YOUNES Loay



Fonctionalités du logiciel

- Il permet de générer des animations et des photos correspondant au thèmes proposés à patir des données saisies de l'utilisateur, soit Multibrot soit resize Julia soit Cercle Julia, et il peut soit afficher une animation ou une photo soit les télécharger en .gif / .PNG respectivement.

Découpage du projet

- Le projet est composé de deux fichiers pour les fractales de Julia et de Mandelbrot, contenant les fonctions de base pour découper le plan. Ensuite, un fichier est dédié à la génération d'images des deux types de fractales, un autre pour la génération d'animations, et enfin un fichier contenant le code de l'interface graphique.

L'Interface Graphique

- Code source de l'interface graphique :
Les variables globales qui stockent les valeurs saisies par l'utilisateur sont définies au début du fichier. Ensuite, la création de la frame et la configuration du chemin sont spécifiées. Des gestionnaires d'événements sont utilisés pour chaque bouton afin de gérer les événements de l'interface graphique.

- Conception de l'interface graphique : Le prototype a été réalisé avec Figma, puis converti avec le logiciel open-source Tkinter Designer. Ensuite, le code a été complété, les variables ont été structurées et les événements ont été ajoutés.

Génération d'animations

- Une fonction d'animation est définie pour chaque animation qui prend en paramètres les données qui déterminent une animation + une valeur qui détermine si la valeur va être sauvegardée ou affichée et soit sauvegarde soit affiche l'animation
- Par choix de structuration , elles contiennent les définissions des fonctions qui génèrent chaque frame

Génération d'image :

- Une fonction de génération d'image est définie pour chaque image qui prend en paramètres les données ainsi que l'entier définissant si l'image sera sauvegardée ou affichée et retournent l'image.

Découpage du plan :

- On a utilisé la même fonction est bornée_average pour les deux fractales car elles fonctionnent de la même façon. Il y a juste une petite différence dans l'incrémentation de puissance qui sera utilisée pour l'animation de la fractale de Mandelbrot et le Z qui sera utilisé pour la fractale de Julia. Donc on a préféré mettre de côté une seule fonction et l'appeler avec des paramètres correspondant à chaque utilisation.

Manuel

L'interface graphique se compose de deux parties. La première partie est dédiée à la fractale de Julia, tandis que la seconde est réservée à la fractale de Mandelbrot.

Chaqu'une des animations est composée de deux partie la partie pour la génération d'animation (Une pour Mandelbrot , deux pour Julia) est en haut et la partie de génération de photos est en bas.

Pour toutes les fenêtres

- Les paramètres sont initialisés par défaut. Ainsi, en cliquant directement sur "Afficher" ou "Télécharger", on obtiendra un affichage initialisé par défaut.
- L'utilisateur peut donner les frames qu'il souhaite dans l'encadré (entre 50 et 1024 p)
- Si l'utilisateur met plus que 1024 p ça remet à la résolution par défaut (360 p)
- Pour les nombre de frames , si l'utilisateur met plus que 200 frames par secondes , ça remet les frames à 50
- Il peut choisir les couleurs entre les 6 couleurs proposées (Seuls ses couleurs existent)
- Il peut choisir les paramètres dans l'ordre qu'il veut , quand il clique les choix sont prises en compte et pas besoin de tapez sur entrer pour envoyer les données au programmes (les données sont transmis directement)
- Pour afficher une animation ou une photo, cliquer sur l'icône d'œil dans la fenêtre , la vidéo ou la photo s'affiche sur le simulateur de Matplotlib.
- Pour télécharger une animation, cliquer sur le symbole de flèche de bas , l'animation est téléchargée en format .gif dans le répertoire courant
- Pour télécharger une photo, cliquer sur le symbole de flèche de bas , l'animation est téléchargée en format .dans le répertoire courant

Génération d'animation de Multibrot

- L'animation affiche jusqu'à la puissance d = 7 dans l'expression zn = zn-1 ^ d + c

- Si l’utilisateur clique sur afficher ou télécharger sans varier des paramètres , des paramètres sont prédéfinis par défaut , ses paramètres sont :

* Nombre de frames = 50
* Résolution = 360 p
* Couleur = spring

Génération de photo Mandelbrot

- Si des paramètres ne sont pas définis par défaut : L’image est affichée ou téléchargée avec :

* Resolution = 360 p
* Couleur = spring

Génération d’animation cercle Julia

- Pour choisir cette animation , soit l’utilisateur doit écrire 1 sur l’encadré correspondant à circulaire ou par résolution soit elle est choisie par défaut.
- Deux encadrés se composant d’un encadré pour la partie réelle et un autre encadré pour la partie imaginaire pour le nombre complexe à partir duquel l’utilisateur souhaite commencer l’animation
- Si aucun type d'animation n'est choisi pour la partie fractale de Julia ou l’une des paries du nombre complexe n’est pas entre -2 et 2 , l'animation circulaire de Julia sera sélectionnée par défaut :
- * nombre complexe de début = 0.2 - 0.255 i.
- * résolution 360 p
- * nombre de frames = 50

  * couleur = spring

Génération de photo cercle Julia

- Deux encadrés se composant d’un encadré pour la partie réelle et un autre encadré pour la partie imaginaire pour le nombre complexe à pour lequel l’utilisateur souhaite voir une photo ou une photo à qui est appliquée une translation
- Si des paramètres ne sont pas modifiés les paramètres prédéfinis sont :

  * Nombre complexe de début = 0.2 - 0.255 i
  * résolution = 360 p
  * couleur = spring

Génération d'animation resize Pour choisir cette animation , l'utilisateur doit écrire 2 sur l'encadré correspondant à circulaire ou par résolution (L'animation étant originallement une animation qui varie la résolution mais elle faisait un truc imprévu et on voulait la modifier - exprimé dans la section difficultés et espérence -)

- Comme les paramètres et les valeurs par défaut de cercle Julia , la res max

Bilan du projet

Difficultés

1. Des animations qui sont parfois très lentes à générer en gif.
(Ça prend plus de 20 minutes)

Il faut faire un compromis , soit générer avec une plus grande résolution et un nombre plus petit de frames soit le contraire

2. La fonction FuncAnimation quand elle fait appel à la fonction d'animation avec d'autres données que le numéro du frame.

Sol : Définir la fonction d'animation dans une fonction plus grande qui contient les paramètres

3. La compréhension de Julia était un peu difficile , spécialement que Julia correspond à un choix de nombre complexe

Sol : Commencer par la fonction est_Bornee_Julia et supposer d'avoir choisi le nombre complexe en écrivant la fonction puis rendre le code plus abstrait.

4. Ne pas arriver à coder une fonction zoom pour Mandelbrot , la fonction buguait toujours.

Sol : Au lieu de penser à cette idée , on a trouvé une idée plus facile , l'animation de Multibrot (on ne varie qu'une seule valeur).

5. La gestion des erreurs qui peuvent confronter l'utilisateur (Plus notamment quand il ne fait pas entrer des paramètres ou quand il entre une résolution qui est au delà de 1024)

Sol : Traiter le cas quand l'utilisateur tape plus que 1024 l'animation est générée par défaut à l'animation 360 p . Pour ne pas prendre une animation qui est trop lourde.

Sol : Les variables et les paramètres sont initialisés par défault , et utiliser des forçage de types comme : le nombre de frames soit obligatoirement un int (sinon la valeur de nb de frames qui est initialisée par défaut est injectée dans l'animation)

Espérences d'amélioration

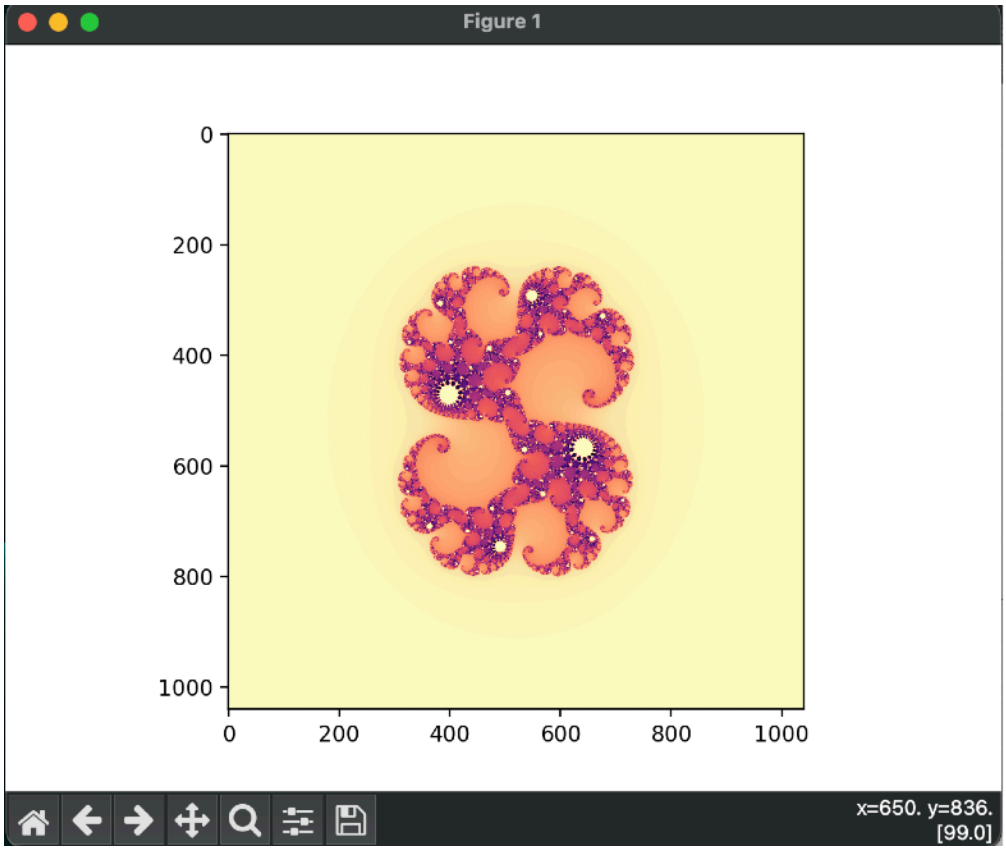
1. Prendre plus de temps pour coder une animation zoom pour les deux Julia et Mandelbrot (Une animation qui montre tous les détails à un nombre complexe donné)

2. Apprendre plus de retouches aux algorithmes proposée pour les rendre plus efficaces

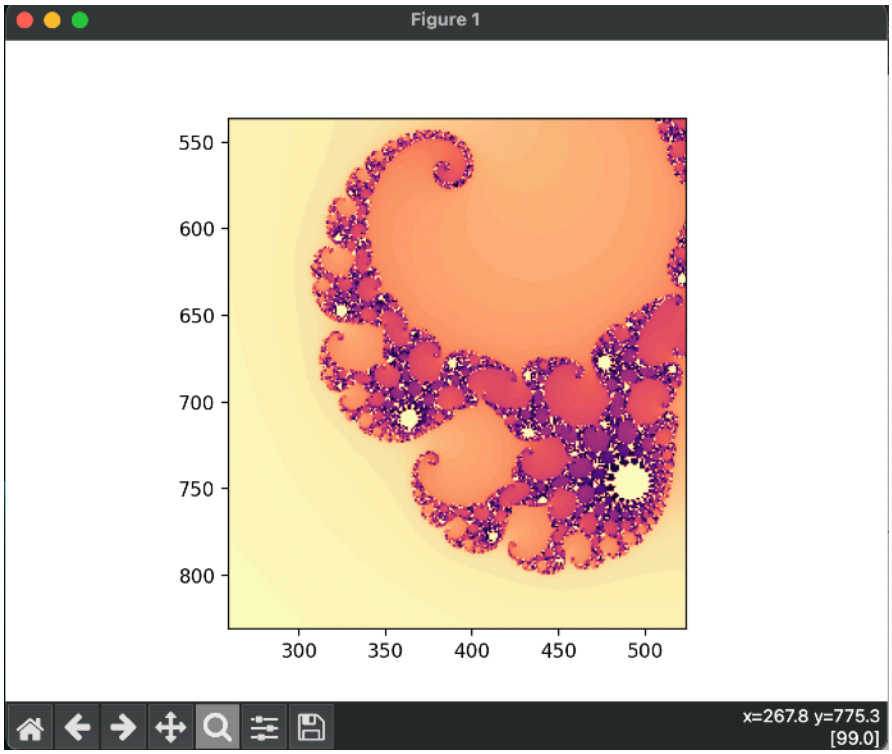
3. L'animation qui varie la résolution on voulait voir pourquoi elle faisait un truc imprévu , mais vu la complexité de notre interface et de la structuration du projet , on avait peur d'introduire des bugs à notre appli qui marché
On souhaite vraiment de prendre le temps de plannifier les animations juste avant de les coder afin de se concentrer uniquement sur le code (et de ne pas introduire des modifications imprévues)

4. Afficher des messages d'erreurs lors des erreurs au lieu d'afficher les fractales initialisés par défaut pour informer l'utilisateur qu'il y avait une erreur.

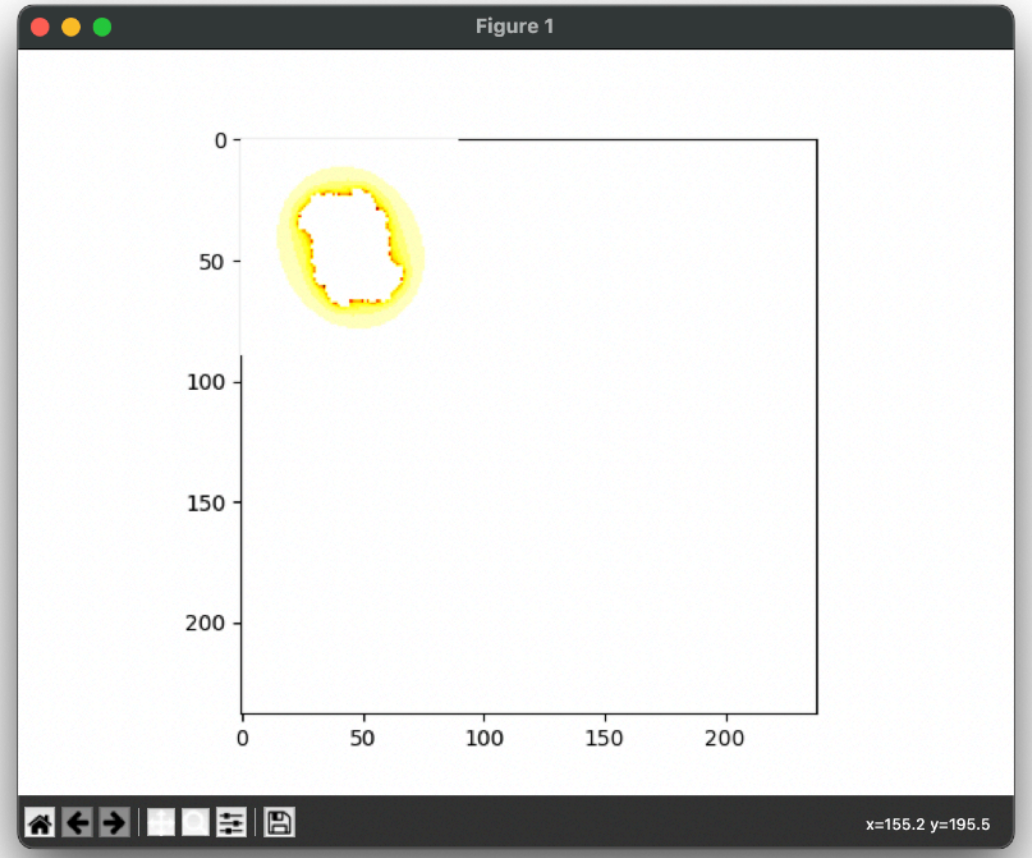
Annexe



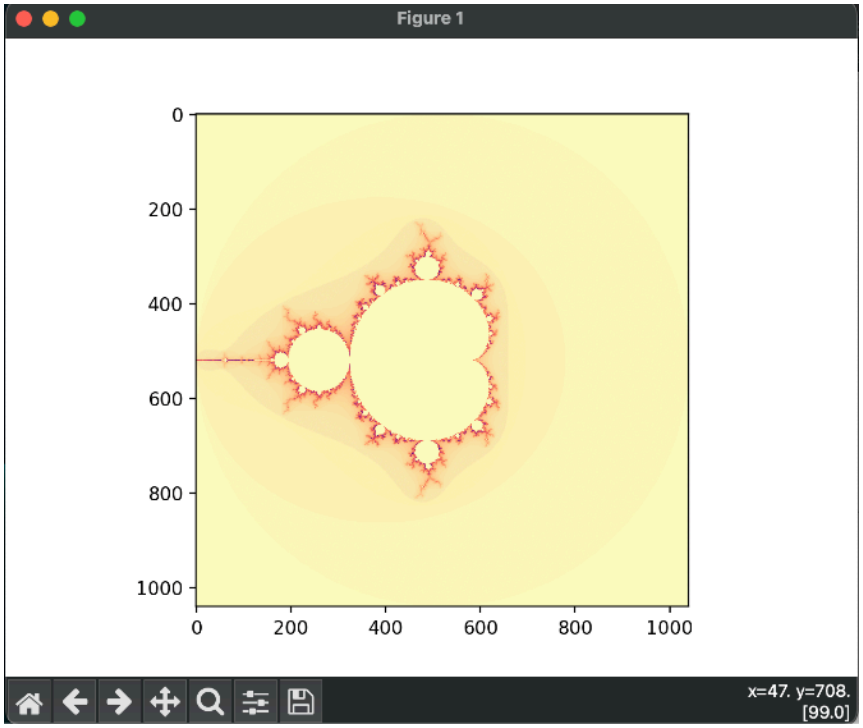
La photo du cercle Julia avec la résolution à 1024 , partie réelle 0.28 et partie imaginaire 0.008
Et le degré est à zéro , couleur magma



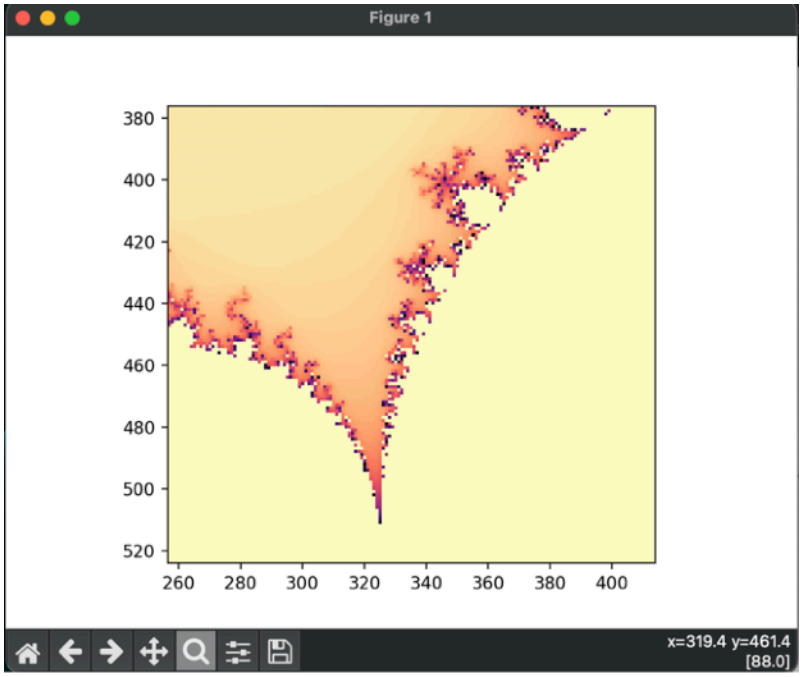
Même photo agrandie une fois de plus



Une photo de l’animation resize
Avec les paramètres par défaut
Nombre complexe : 0,285 + 0,013 i
Couleur : Spring
Resolution : 360 p

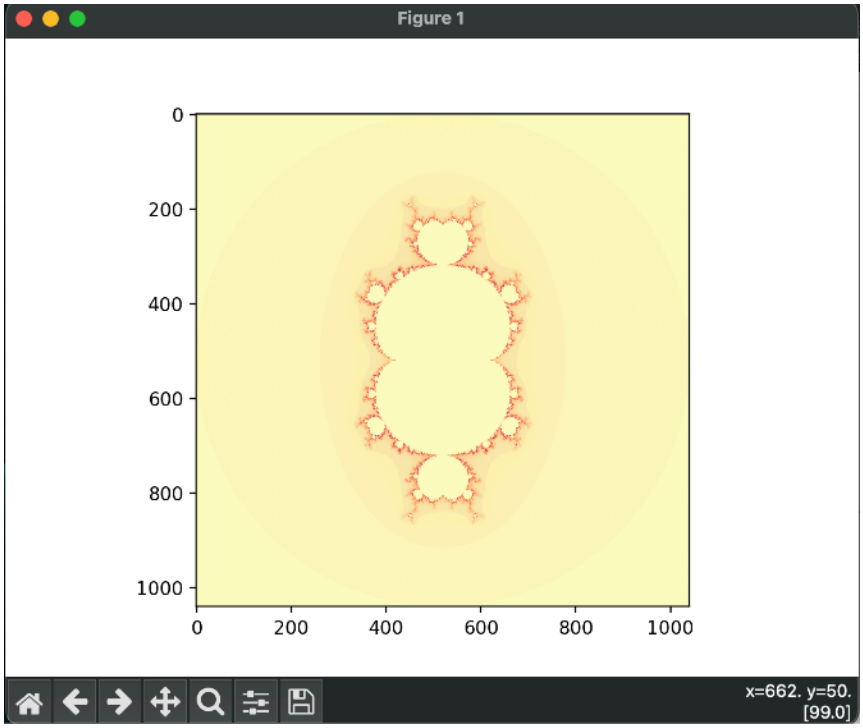


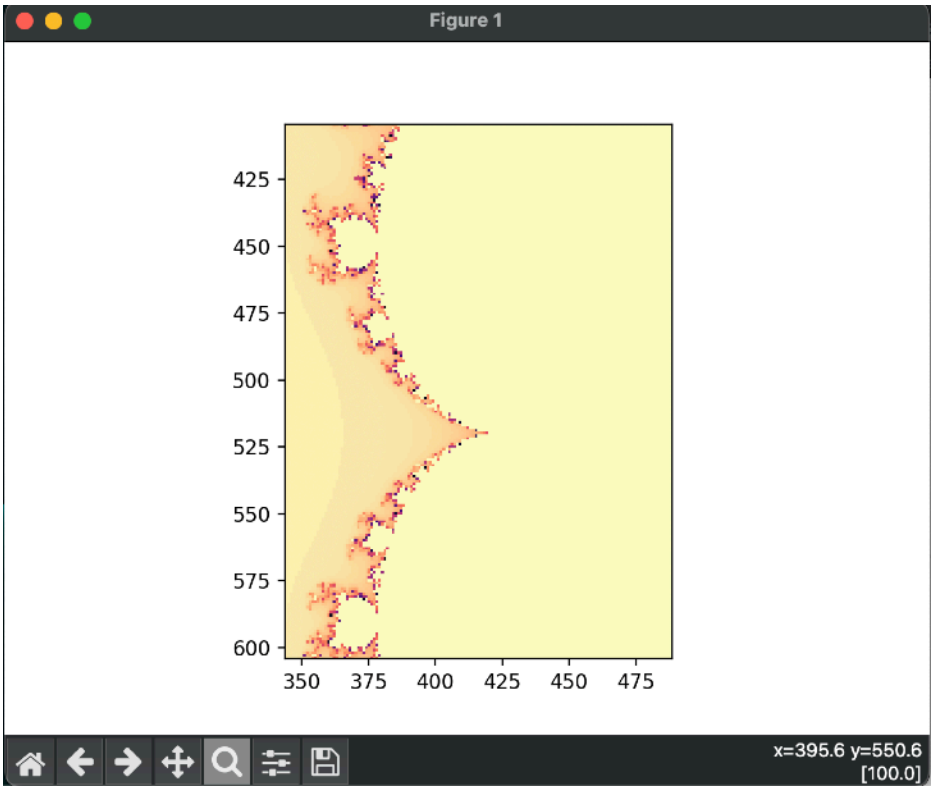
La photo du Mandelbrot avec la résolution à 1024, couleur est magma



Même photo mais agrandie dans la sea Valley

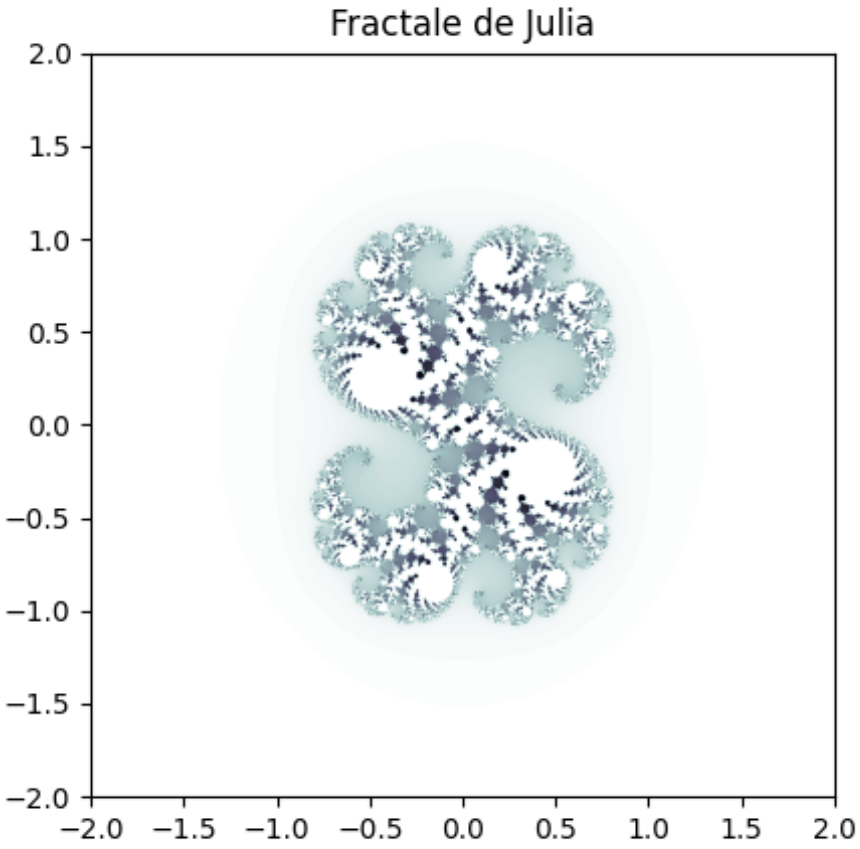
Cette photo est prise en s'arrêtant dans la vidéo de l'animation Multibrot (1024 p , couleur = "magma", nb de frames = 50)

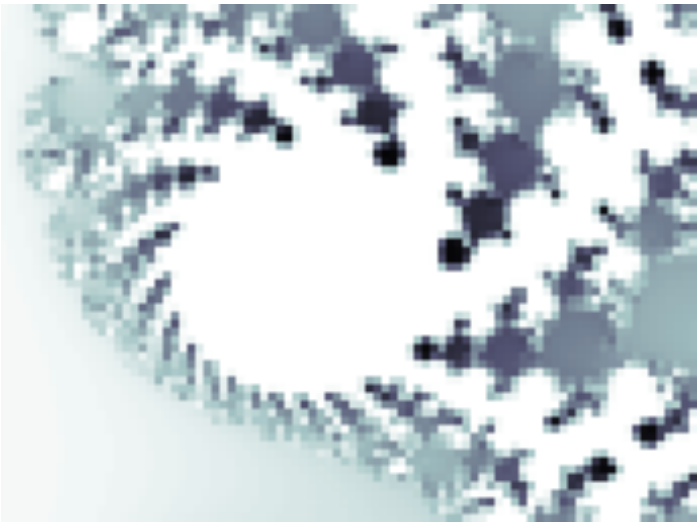




Même photo , agrandie une fois de plus

La fractale Julia correspondant au nombre complexe 0,285 + 0,013 i avec la couleur bone et la résolution 1024 p





L’animation agrandie une fois de plus.