

# Chapitre 4

## Exercices et problèmes

### 4.1 Fractales de Mira

On considère la double suite, définie par le  $(x_0, y_0)$ , par la récurrence suivante

$$\begin{aligned}x_{n+1} &= by_n + ax_n + (1-a)\frac{2x_n^2}{1+x_n^2} \\y_{n+1} &= -x_n + ax_{n+1} + (1-a)\frac{2x_{n+1}^2}{1+x_{n+1}^2}\end{aligned}$$

ainsi que deux paramètres  $a$  et  $b$ .

- Réaliser un programme qui calcule et écrit en colonne dans un fichier les valeurs des couples successifs  $(x_n, y_n)$  de cette suite pour des valeurs de  $x_0, y_0, a$ , et  $b$  fixées.
- Tracer l'ensemble de ces couples dans un graphe  $(x, y)$  avec Gnuplot. De jolies combinaisons sont par exemple  $(x_0 = 12, y_0 = 0, a = 0.7, b = 0.9998)$  et celles proposées sur la page : <http://math-world.wolfram.com/MiraFractal.html>.

### 4.2 Suite de Collatz (Syracuse, Ulam, $3x+1$ )

On considère la suite **entière** définie par

$$\begin{aligned}x_{n+1} &= (3x_n + 1)/2 && \text{si } x_n \text{ est impair} \\x_{n+1} &= x_n/2 && \text{si } x_n \text{ est pair}\end{aligned}$$

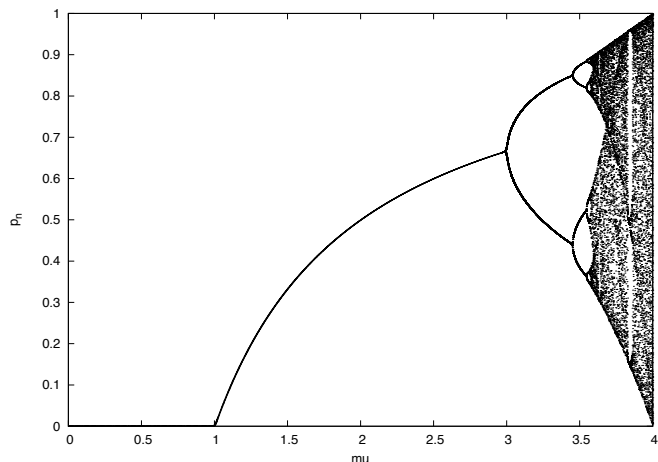
La conjecture de Collatz (qui n'a pas encore été démontrée aujourd'hui) stipule que cette suite finit toujours par atteindre la valeur 1, quelque soit l'entier de départ  $x_0$ .

- Calculer les premières valeurs de cette suite pour un entier  $x_0$  donné (par ex  $x_0 = 27$ ), les stocker dans un fichier et les tracer en fonction de leur indice avec gnuplot.
- Tester la conjecture de Collatz pour différentes valeurs de  $x_0$ .
- Modifier le programme précédent afin de calculer systématiquement les premières valeurs de la suite pour de nombreuses valeurs de  $x_0$  (on pourra par exemple les échelonner en puissance de 2).
- Mesurer et tracer en fonction de  $x_0$  le *temps de vol* et l'*altitude maximale*, c'est à dire respectivement le nombre d'itérations nécessaires pour atteindre 1 et la valeur maximale atteinte.

### 4.3 Évolution des populations et suite logistique de Feigenbaum

La suite logistique de Feigenbaum sert à décrire l'évolution du nombre d'individus dans une population soumise à des ressources finies. Qualitativement, la population tend à s'accroître tant qu'elle a assez de nourriture à sa disposition, c'est à dire tant qu'elle n'est pas trop nombreuse. En revanche, lorsqu'elle possède trop d'individus, la quantité de nourriture par individu diminue, ce qui tend à faire décroître la population.

On peut modéliser cette évolution par la suite non linéaire suivante :



$$p_{n+1} = \mu p_n (1 - p_n)$$

où :

- $\mu > 0$  est le taux de croissance de la population (taux de natalité).
- $p_n$  représente le nombre d'individus dans la population à un temps  $t_n$ , normalisé à sa valeur maximale quand  $\mu \leq 4$  (c'est à dire  $0 \leq p_i \leq 1$ ).
- $p_0$  correspond à la population de départ ( $0 \leq p_0 \leq 1$ )

Travail :

- Pour une population initiale  $p_0$  et un taux de croissance  $\mu$  donnés, calculer les 100 premières valeurs de  $p_n$ . Enregistrer ces valeurs dans un fichier et les tracer en fonction du temps avec gnuplot.
- Tester différentes valeurs de  $\mu$ . On pourra par exemple comparer les cas  $0 \leq \mu \leq 1$ ,  $1 \leq \mu \leq 3$ ,  $3 \leq \mu \leq 3.57$ ,  $3.57 \leq \mu \leq 4$  et  $4 \leq \mu$ .
- On s'intéresse maintenant au comportement asymptotique de la suite logistique quand  $n \rightarrow +\infty$ , pour différentes valeurs de  $\mu$ . Pour au moins 1000 valeurs différentes de  $\mu$  comprises entre 0 et 4, calculer les valeurs prises par  $p_n$  jusqu'au temps correspondant à  $n = 300$ . On considérera qu'un régime stationnaire est atteint dès que  $n \geq 100$ . Stocker dans un même fichier de deux colonnes tous les couples de points  $(\mu, p_i)$  calculés pour  $100 \leq i \leq 300$ . Les représenter sous gnuplot avec la syntaxe suivante : `plot "fichier.dat" with dots notitle .`

## 4.4 Exemples de fractales

### 4.4.1 Notes préliminaires

#### Aspects mathématiques

Beaucoup de fractales sont décrites grâce à des suites complexes :  $z_{n+1} = f(z_n)$ , où  $f : \mathbb{C} \rightarrow \mathbb{C}$  est une fonction du plan complexe. Différentes fonctions  $f$  définissent différentes fractales. Pour être plus précis, on s'intéresse à la **convergence** de ces suites pour construire les fractales associées.

Souvent, on distingue en fait deux types d'ensembles :

- *L'ensemble de Julia* : Dans ce cas, on s'intéresse à la convergence de la suite en fonction du point de départ :  $z_0 = x_0 + iy_0$ . Dans ce cas, la fractale est représentée par la fonction  $F(z_0)$  qui vaut  $F = 0$  si la suite converge et  $F > 0$  sinon.
- *L'ensemble de Mandelbrot* : Dans ce cas, la suite est paramétrisée par une constante complexe  $c = a + ib$ . Et on s'intéresse à la convergence de la suite pour  $z_0 = 0$  et en fonction des valeurs de  $c$ . Dans ce cas, la fractale est représentée par la fonction  $F(c)$  qui vaut  $F = 0$  si la suite converge et  $F > 0$  sinon.

Dans les deux cas, le plus simple est de prendre  $F = 1$  lorsque la suite diverge, ce qui produit des images en noir et blanc. On peut aussi attribuer une valeur qui caractérise la vitesse de divergence afin d'obtenir des images en couleurs.

### Aspects numériques

- Par défaut, le type *complexe* n'existe pas en C. Pour une fonction  $f(z)$  donnée, il faut donc définir deux suites couplées : une pour la partie réelle  $x_{n+1} = f_{\text{Re}}(x_n, y_n)$  et une pour la partie imaginaire  $y_{n+1} = f_{\text{Im}}(x_n, y_n)$ .

- L'idée est ensuite de paver le plan complexe de  $z_0$  (pour l'ensemble de Mandelbrot, ou de  $c$  pour l'ensemble de Julia) avec un certain nombre de valeurs réelles et de valeurs imaginaires, entre des bornes qu'on se fixera. Pour limiter le temps de calcul, vous testerez votre programme avec quelques dizaines de valeurs, puis vous vous limiterez au maximum à environ 1000 valeurs dans chaque dimension.

- Numériquement, on pourra considérer que la suite diverge si  $z_n$  dépasse en norme une valeur  $z_{\text{max}}$  (qu'on se fixera). A l'inverse, on considérera que la suite ne diverge pas si au bout d'un nombre d'itérations maximal  $n_{\text{max}}$  (qu'on se fixera aussi), la suite n'a pas dépassé  $z_{\text{max}}$ .

- Il faudra stocker, dans un fichier de sortie, les valeurs de  $F$  en fonction de  $x_0$  et  $y_0$  (ou  $a$  et  $b$ ). On remplira en ligne les résultats pour différentes parties réelles  $x_0$  (ou  $a$ ), et en colonne ceux pour différentes parties imaginaires  $y_0$  (ou  $b$ ). La première ligne sera réservée pour mettre les valeurs de  $x_0$  (ou  $a$ ), et la première colonne les valeurs de  $y_0$  (ou  $b$ ) :

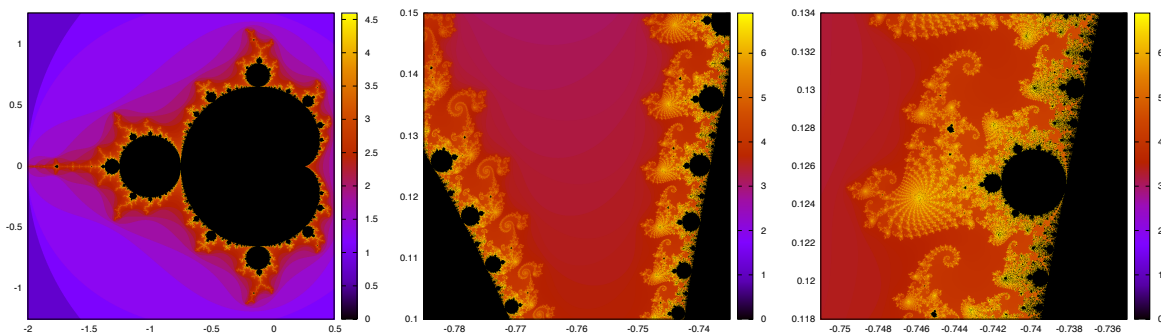
n	x01	x02	x03	...	X0n
y01	F11	F12	F13	...	F1n
y02	F21	F22	F23	...	F2n
y03	F31	F32	F33	...	F3n
...	...	...	...	...	...
y0m	Fm1	Fm2	Fm3	...	Fnm

- Enfin, pour obtenir des images en couleurs, il est plus marquant de caractériser la vitesse de divergence par le nombre d'itérations nécessaires pour atteindre  $z_{\text{max}}$ . On représentera alors en 2D la fonction  $F(z_0)$  (ou  $F(c)$ ) par un code de couleurs. Avec GNUPLOT, utilisez la syntaxe suivante :

```
set autoscale fix
set size square
plot "fichier.dat" nonuniform matrix using 1:2:(log($3)) with image notitle.
```

### 4.4.2 Exercices

#### 1- Ensemble de Mandelbrot pour : $f(z) = z^2 + c$



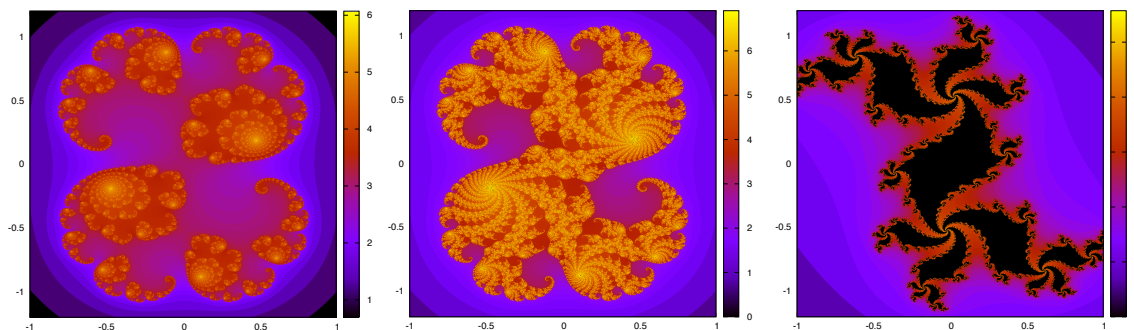
Une fois séparée en partie entière et parties réelle et imaginaire cette fonction est équivalente à la double suite définie par  $x_0 = y_0 = 0$  et :

$$\begin{aligned} x_{n+1} &= x_n^2 - y_n^2 + a \\ y_{n+1} &= 2x_n y_n + b \end{aligned}$$

Pour cette suite, il suffit<sup>1</sup> de prendre  $z_{\max} = 2$  et  $n_{\max} = 1000$ .

- Tracer la fractale de Mandelbrot pour  $-2 < a < 0.5$  et  $-1.25 < b < 1.25$ .
- Zoomer sur des parties intéressantes.

#### 2- Ensemble de Julia pour : $f(z) = z^2 + c$



Pour cette même suite, et une valeur fixée de  $c$ , il s'agit ici d'étudier la convergence en fonction de la valeur de  $z_0$ . On pourra garder  $z_{\max} = 2$  et  $n_{\max} = 1000$ .

- Tracer l'ensemble de Julia pour  $c = 0.285 + 0.010i$ ,  $c = 0.285 + 0.013i$  et/ou  $c = 0.3 + 0.5i$ , sur le plan  $-1. < x_0 < 1.$  et  $-1.2 < y_0 < 1.2$ .

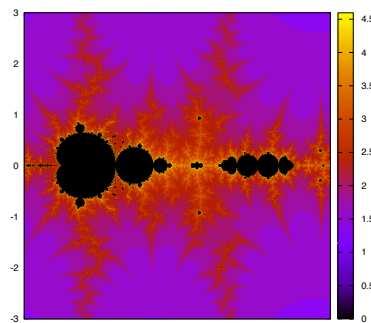
1. on peut montrer que la suite diverge toujours si on dépasse cette valeur

**3- Ensemble de Mandelbrot pour :  $f(z) = \cos(z) + c$**

Une fois séparée en partie entière et parties réelle et imaginaire cette fonction est équivalente à la double suite définie par  $x_0 = y_0 = 0$  et :

$$\begin{aligned} x_{n+1} &= 2 \cos x_n \cosh y_n + a \\ y_{n+1} &= -\sin x_n \sinh y_n + b \end{aligned}$$

Pour cet exercice, on pourra prendre  $z_{\max} = 10^7$  et  $n_{\max} = 100$ .

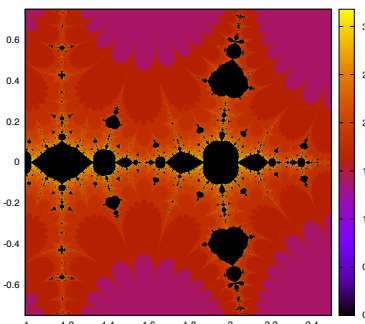


- Tracer l'ensemble correspondant à cette fonction pour  $-3 < x_0 < 3$  et  $-3 < b < 3$ .

**4- Ensemble de Julia pour  $f(z) = z + [1 - (1 + 2z) \cos(\pi z)]/4$**

On peut montrer que cette fonction généralise la suite de Collatz des valeurs entières au plan complexe. Une fois séparée en partie entière et parties réelle et imaginaire cette fonction est équivalente à la double suite définie par  $x_0 = y_0 = 0$  et :

$$\begin{aligned} x_{n+1} &= x_n + \{1 - (1 + 2x_n) \cosh(\pi y_n) \cos(\pi x_n) \\ &\quad - 2y_n \sinh(\pi y_n) \sin(\pi x_n)\} / 4 \\ y_{n+1} &= y_n + \{(1 + 2x_n) \sinh(\pi y_n) \sin(\pi x_n) \\ &\quad - 2y_n \cosh(\pi y_n) \cos(\pi x_n)\} / 4 \end{aligned}$$



- Tracer l'ensemble de Julia pour  $1 < x_0 < 2.5$  et  $-0.75 < y_0 < 0.75$ .

**5- Fractales de Newton**

Soit une fonction du plan complexe  $g(z)$ . Une méthode itérative pour déterminer les zéros de  $g$  est la méthode de *Newton*. Elle consiste à appliquer itérativement la fonction  $f(z) = z - g(z)/g'(z)$ . Selon la valeur de  $z_0$ , cette suite converge vers un des zéros de  $g$  ou diverge. On peut donc lui associer un ensemble similaire à celui de Mandelbrot. Mais cette fois-ci, on définit la convergence quand deux valeurs de  $z$  deviennent plus proche qu'un petit paramètre  $\epsilon = 10^{-10}$  par exemple. Et quand elle converge, on attribue un nombre qui caractérise la vitesse de convergence (par exemple le nombre d'itérations nécessaire à vérifier le critère défini par  $\epsilon$ ).

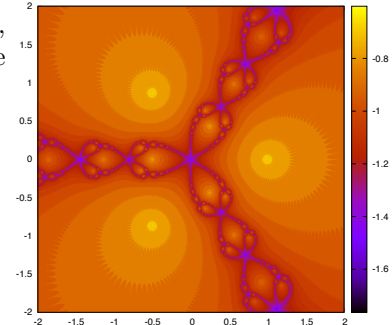
**Newton pour  $g(z) = z^3 - 1$**

Les zéros de  $g$  sont :  $1, (1 + i\sqrt{3})/2$  et  $(1 - i\sqrt{3})/2$ . Pour ce cas,  $f(z) = z - (z^3 - 1)/(3z^2)$  que l'on peut associer à la double suite suivante :

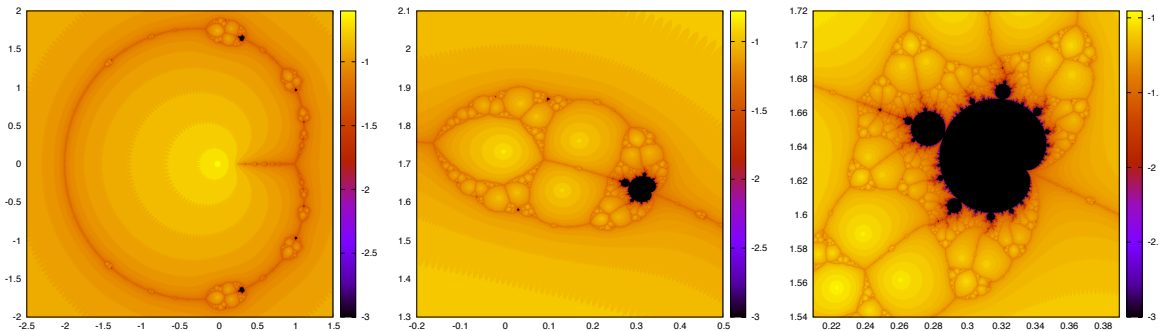
$$\begin{aligned} x_{n+1} &= \frac{2}{3} \left( x_n + \frac{x_n^2 - y_n^2}{(x_n^2 + y_n^2)^2} \right) \\ y_{n+1} &= \frac{2}{3} \left( y_n - \frac{2x_n y_n}{(x_n^2 + y_n^2)^2} \right) \end{aligned}$$

Ici, on pourra prendre  $n_{\max} = 10000$  par exemple.

- Tracer l'ensemble correspondant à cette fonction pour  $-2 < x_0 < 2$  et  $-2 < y_0 < 2$ .
- Zoomer sur des zones particulières



**Newton pour  $g(z) = z^3 + (c - 1)z + c$**



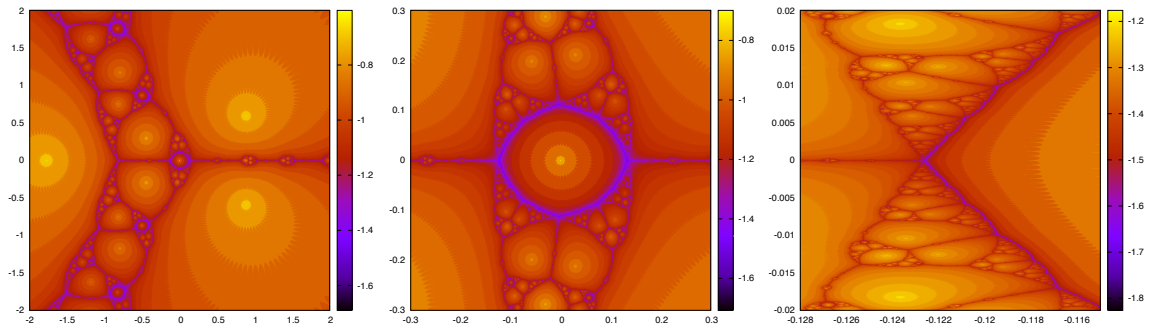
Il s'agit ici d'étudier la suite définie par  $f(z) = z - (z^3 + (c - 1)z + c)/(3z^2 + c - 1)$  pour  $z_0 = 0$  et différentes valeurs de  $c$  (ensemble de Mandelbrot). Cette suite est équivalente à la double suite définie par  $x_0 = y_0 = 0$  et :

$$\begin{aligned} x_{n+1} &= x_n - \frac{A_x B_x + A_y B_y}{B^2} & A_x &= x_n^3 - 3x_n y_n^2 - b y_n + (a - 1)x_n - a \\ y_{n+1} &= y_n - \frac{A_y B_x - A_x B_y}{B^2} & A_y &= -y_n^3 + 3x_n^2 y_n + b x_n + (a - 1)y_n - b \\ & & B_x &= 3(x_n^2 - y_n^2) + a - 1 \\ & & B_y &= 6x_n y_n + b \\ & & B^2 &= B_x^2 + B_y^2 \end{aligned}$$

On pourra prendre  $n_{\max} = 100$ .

- Tracer cette ensemble pour  $-2 < a < 2$  et  $-2 < b < 2$ .
- Zoomer sur des parties intéressantes.

Newton pour  $g(z) = z^3 - 2z + 2$



Il s'agit ici d'étudier la suite  $f(z) = 2(z^3 - 1)/(3z^2 - 2)$  pour différents  $z_0$  (ensemble de Julia). Cette suite est équivalente à la double suite définie par :

$$x_{n+1} = x_n - \frac{A_x B_x + A_y B_y}{B^2}$$

$$y_{n+1} = y_n - \frac{A_y B_x - A_x B_y}{B^2}$$

$$A_x = x_n^3 - 3x_n y_n^2 - 2x_n + 2$$

$$A_y = -y_n^3 + 3x_n^2 y_n - 2y_n$$

$$B_x = 3(x_n^2 - y_n^2) - 2$$

$$B_y = 6x_n y_n$$

$$B^2 = B_x^2 + B_y^2$$

- Tracer l'ensemble correspondant à cette fonction pour  $-2 < x_0 < 2$  et  $-2 < y_0 < 2$ .
- Zoomer sur des zones particulières