

Analyse I

Exercices Pratiques

Licence 1 Informatique

Guillaume Metzler

Institut de Communication (ICOM)

Université de Lyon, Université Lumière Lyon 2

Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Abstract

Cette fiche propose d'étudier quelques sujets liés au thématique et qui font appel à l'outil informatique (ici la mise en place de quelques programmes en Python) pour la résolution de problèmes ou encore l'illustration de certains résultats que l'on serait amené à établir.

L'objectif est de mettre en évidence des notions de vitesse de convergence des algorithmes, de mettre en place des algorithmes de résolution de problème mathématique ou encore d'utiliser l'outil informatique pour générer des objets obtenus à l'aide de la mathématique.

Nous ne pourrons pas aborder ces sujets pendant les séances de cours mais ils sont conçus pour être travaillés en autonomie et ne devrait pas poser de problème en fin de première année de Licence Informatique.

Contents

1 Estimation de π	3
1.1 Principe de la méthode de <i>Monte Carlo</i>	3
1.2 Estimation de π	5
1.3 Etude de la vitesse de convergence	6
2 Estimation de la valeur d'une intégrale	8
2.1 Méthodes des rectangles	9
2.1.1 Méthodes des rectangles décentrés	9
2.1.2 Méthodes des rectangles centrés	12
2.2 Méthode des trapèzes	15
2.3 Méthode de Simpson	18
3 Approche par dichotomie	20
4 Optimisation d'une fonction	24
4.1 Cas d'une fonction convexe	24
4.2 Cas d'une fonction non convexe	30
4.3 Une solution à la non convexité	30
5 Informatique, art et suites complexes	35
5.1 Suite logistique	35
5.2 Un pas vers le chaos	38
5.3 Ensembles de Julia	42
5.4 Ensemble de Mandelbrot	45

1 Estimation de π

L'objectif dans cette section est de développer un code permettant de déterminer, de façon approchée la valeur du nombre $\pi \simeq 3.1415\dots$

Pour faire cela, on emploiera une méthode dite de *Monte Carlo*, une méthode couramment employée en Statistiques pour estimer certaines quantités à l'aide de données et par le biais de simulations.

1.1 Principe de la méthode de *Monte Carlo*

La méthode de *Monte Carlo* est utilisée pour estimer des quantités qui peuvent s'écrire comme une moyenne de variables aléatoires.

Par exemple, lorsque l'on cherche à déterminer si une pièce est truquée ou non, on va lancer N fois notre pièce, les lancers sont supposés indépendants, et on va compter le nombre de "face" obtenu.

Si on note p la probabilité d'obtenir et face et X_i la variable aléatoire "obtenir face au i -ème lancer", alors une estimation de p est donnée par

$$p \simeq \frac{1}{n} \sum_{i=1}^n X_i.$$

Il s'agit donc de *valeur moyenne prise par la variable aléatoire X* .

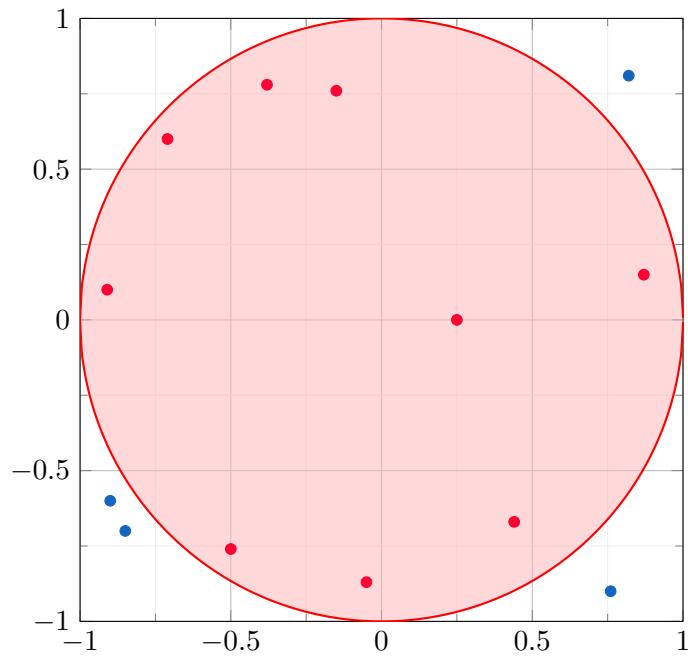
On peut également retrouver cela pour faire du calcul d'intégrale, notamment pour calculer une surface.

Vous verrez qu'une *espérance mathématique*, *i.e.* que l'on appelle aussi classiquement une moyenne, peut s'écrire sous la forme d'une intégrale. Imaginons que je cherche à calculer

$$\mathbb{E}[f(X)] = \int f(x) d\mu(x),$$

où X est une variable aléatoire qui suit une loi uniforme dans $[0, 1]$ et μ est la mesure (une sorte de fonction) associée à cette loi. Pour estimer la valeur moyenne de $f(X)$, *i.e.* son *espérance*, on va tirer un très grand nombre de valeur de X selon une loi uniforme dans $[0, 1]$, on détermine leur image par f et on en fait la moyenne.

C'est un processus semblable que l'on va utiliser pour estimer la valeur de π en fonction de la surface du cercle et du carré représentée ci-dessous.



La garantie de convergence se base sur un résultat en probabilité que l'on appelle **loi forte des grands nombres**. Elle stipule que “la moyenne d'une série de valeurs prise par une variable aléatoire converge vers la valeur moyenne de cette variable aléatoire”. La formulation exacte est la suivante.

Théorème 1.1: Loi forte des grands nombres

Soit $(X_n)_{n \in \mathbb{N}}$ une suite de variables aléatoires *indépendantes et identiquement distribuées*. On suppose aussi que ces variables admettent une espérance (*i.e.* elles admettent une valeur moyenne finie).

Alors

$$\frac{1}{n} \sum_{i=1}^n X_i = \frac{X_1 + X_2 + \dots + X_n}{n} \xrightarrow[n \rightarrow +\infty]{p.s.^a} \mathbb{E}[X].$$

^aLa convergence est dite *presque sûre*. On ne s'attardera pas sur ce mode de convergence qui est le plus fort en théorie des probabilités

Regardons plus précisément comment nous allons déduire la valeur de π à l'aide de ce résultat et d'un peu de géométrie.

1.2 Estimation de π

Principe Considérons un cercle de rayon $r > 0$ centré en l'origine de notre repère. Ce cercle peut alors être contenu dans un carré, centré en l'origine de notre repère également, dont la longueur des côtés est égale à $2r$.

Le rapport de entre la surface du cercle et la surface du carré est égale

$$\frac{\pi r^2}{(2r)^2} = \frac{\pi r^2}{4r^2} = \frac{\pi}{4}.$$

Ainsi, si l'on tire n points aléatoirement et de façon uniforme dans ce carré (*i.e.* nous le point peut se retrouver n'importe où dans le carré avec la probabilité), alors, en moyenne, un nombre de $n \frac{\pi}{4}$ de ces points se retrouveront dans à l'intérieur du cercle.

Ainsi, si on suppose que l'on tire n points aléatoirement dans le carré, que l'on note m le nombre de points qui se trouvent dans le cercle, alors nous avons, en moyenne

$$m \simeq \frac{n\pi}{4} \iff \pi = \frac{4m}{n}.$$

Implémentation

Le programme que vous cherchez à développer devra tirer n points aléatoirement dans un carré de côté r . Vous devrez ensuite vérifier si le point en question se trouve dans le cercle ou non. Pour cela, on rappelle qu'un point de coordonnées (x, y) se trouve dans un cercle de rayon r centré en l'origine du repère, si et seulement si $x^2 + y^2 < r^2$. Votre programme devra enfin compter le nombre de points m qui se trouvent à l'intérieur du cercle afin d'obtenir l'approximation de π que l'on affichera.

Votre code comportera les étapes suivantes :

1. On demandera à l'utilisateur une valeur de n .
2. Il faudra tirer les coordonnées des points dans un carré.
3. Il faudra vérifier que le point appartient ou non au cercle.
4. On devra retourner une estimation l'estimation de π .

Affichage On pourra maintenant chercher à visualiser le résultat obtenu, *i.e.* chercher à voir le nombre de points à l'intérieur et à l'extérieur du cercle.

Vous développerez un code qui permettra de visualiser les points qui se trouvent à l'intérieur du cercle (ils seront colorés en rouge) et ceux à l'extérieur (seront colorés en bleu). Toutes les 1000 itérations, on actualisera le graphique et on mettra à jour l'estimation de la valeur de π .

1.3 Etude de la vitesse de convergence

On va maintenant chercher à étudier à quel point notre estimation converge rapidement vers la vraie valeur, *i.e.* est-ce qu'il nous faut un grand nombre de points pour obtenir une estimation très précise de la valeur de π .

Si la **Loi forte des grands nombres** nous renseigne sur la convergence vers la valeur que l'on cherche à estimer, nous n'avons aucune informations sur la vitesse à laquelle on s'approche de cette valeur, *i.e.* de combien de points ais-je besoin pour avoir une “une bonne estimation” ?

La réponse est apportée par un autre résultat très important en théorie des probabilités¹ que l'on appelle le *Théorème Central Limite*.

Théorème 1.2: Théorème Central Limite

Soit $(X_n)_{n \in \mathbb{N}}$ une suite de variables aléatoires *indépendantes et identiquement distribuées* à valeurs dans \mathbb{R} . On suppose que nos variables aléatoires admettent une variance finie notée σ^2 .

Alors :

$$\frac{\sqrt{n}}{\sigma} \left(\frac{X_1 + X_2 + \dots + X_n}{n} - \mathbb{E}[X] \right) \xrightarrow[n \rightarrow +\infty]{\text{loi }} \mathcal{N}(0, 1)$$

où $\mathcal{N}(0, 1)$ fait référence la loi normale centrée et réduite, *i.e.* de moyenne $\mu = 0$ et de variance $\sigma^2 = 1$.

¹Il s'agit d'une forme de convergence que l'on pourrait qualifier de plus faible par rapport à la convergence presque sûre

Ce théorème nous dit que si nous cherchons à approcher l'espérance $\mathbb{E}[X]$ d'une variable aléatoire X par une méthode de Monte-Carlo, ou pourra dire que l'erreur d'estimation

$$\mathbb{E}[X] - \frac{X_1 + X_2 + \dots + X_n}{n}$$

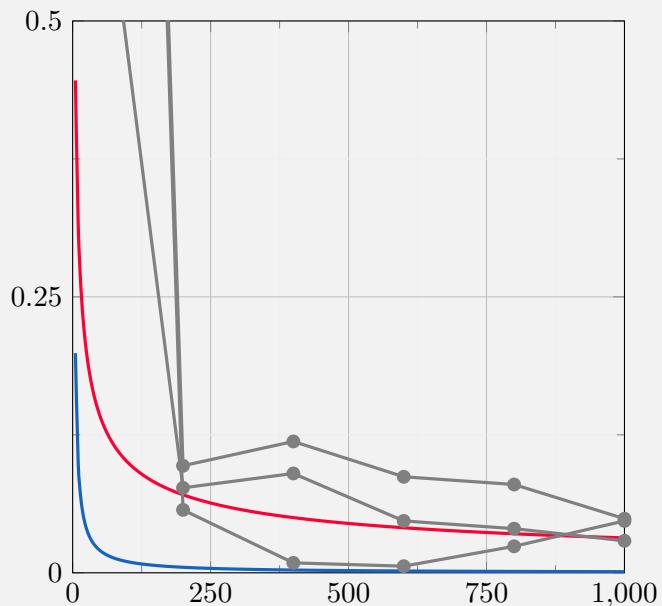
¹Ce résultat sera très souvent utilisé à partir de la troisième année de Licence dans les cours de Statistiques Inférentielles et de Modèles Linéaire.

est de l'ordre de $\frac{\sigma}{n}$.

Attention ! Cette erreur n'est pas exacte, mais elle est probabiliste. Il se peut, avec une certaine probabilité que l'erreur soit plus élevée, cela va dépendre des tirages effectuées. Plus précisément cette erreur est distribuée selon une $\mathcal{N}(0, \frac{\sigma^2}{n})$.

Implémentation

En vous basant sur le code précédent, vous essaierez de mettre en évidence la convergence de l'erreur d'estimation avec une vitesse de $\frac{1}{\sqrt{n}}$ afin d'obtenir un graphique semblable à celui ci-dessous pour n allant jusqu'à 1000.



Votre représentation de l'erreur doit ressembler à l'une des courbes grises.

Dire que la vitesse de convergence est en $\frac{1}{\sqrt{n}}$ signifie que pour diviser votre erreur par 10, il vous faudra multiplier le nombre d'exemples par 100.

2 Estimation de la valeur d'une intégrale

L'objectif de ce travail est de chercher une façon approchée de calculer une intégrale par une méthode non statistique contrairement à ce que nous avons pu faire précédemment.

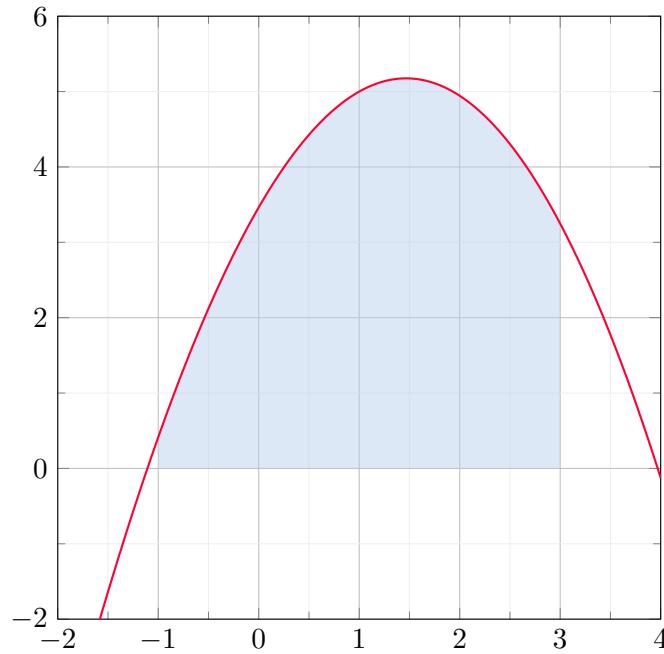
Plus précisément, notre objectif va être de calculer l'intégrale d'une fonction continue $f : [a, b] \rightarrow \mathbb{R}$ à l'aide de trois méthodes différentes :

- la méthode des rectangles
- la méthode des trapèzes
- la méthode de Simpson

Par exemple, on cherchera à évaluer l'intégrale de la fonction de la fonction f définie sur l'intervalle $[-2, 4]$ par

$$f(x) = (x + 2)\sqrt{3 + x} - x^2,$$

dont on cherchera la valeur de l'intégrale sur le segment $[-1, 3]$ comme représentée ci-dessous :



2.1 Méthodes des rectangles

Cette première méthode consiste à approcher la valeur de l'intégrale en subdivisant notre intervalle et en approchant l'aire sous la courbe à l'aide de rectangles.

Nous proposons deux approches mais qui n'ont pas du tout les mêmes propriétés, comme vous le verrez.

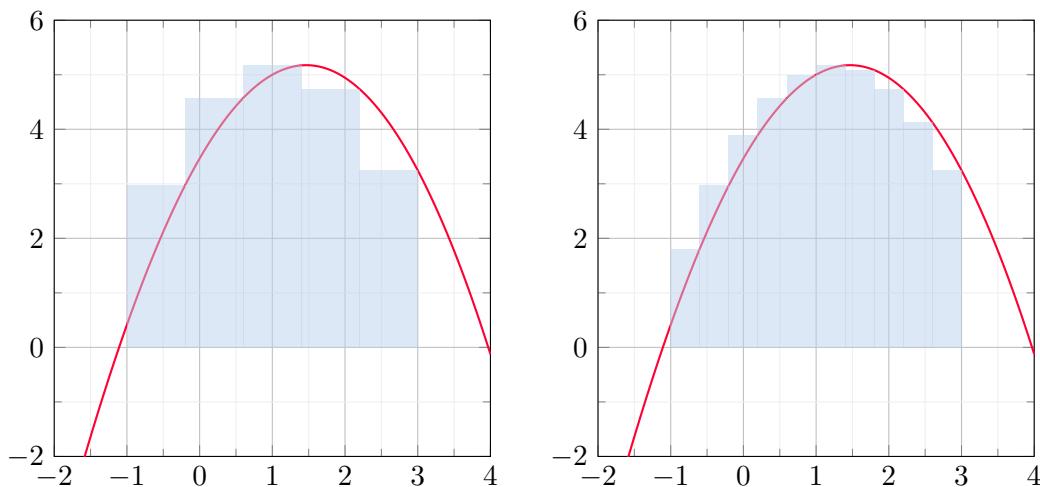
2.1.1 Méthodes des rectangles décentrés

Etant donné un segment $[a, b]$, une subdivision de longueur n est une suite de segments de la forme $[x_k, x_{k+1}]$ tels que $x_{k+1} - x_k = \frac{b - a}{n}$.

Etant donné un segment $[x_k, x_{k+1}]$ le rectangle considéré aura une hauteur h_k égale à :

$$h_k = f(x_{k+1}),$$

i.e. la hauteur du rectangle est égale à l'image de x_{k+1} par la fonction f . Sur notre exemple précédent, une subdivision de taille 5 et 10 nous donnera les approximations suivantes : à gauche pour une subdivision de taille 5 et à droite pour une subdivision de taille 10.



Nous aurions également pu prendre des rectangles de hauteur $f(x_k)$ ce qui change bien évidemment l'approximation effectuée mais pas les propriétés de convergence de cette approximation.

L'aire estimée par cette méthode est alors égale à la somme des aires des différents rectangles, i.e. l'intégrale est approchée par

$$R_n = \sum_{k=1}^n (x_{k+1} - x_k) \times f(x_{k+1}),$$

où $x_1 = a$ et $x_{n+1} = b$. L'aire que l'on souhaite alors estimer est alors obtenue par un passage à la limite.

Etudions maintenant la convergence de cette méthode.

Erreur d'approximation On va d'abord regarder l'erreur effectuée sur chaque segment de la subdivision puis sommer ces erreurs là et effectuer une petite majoration pour obtenir l'erreur d'approximation e_n sur le segment $[a, b]$. On cherchera à démontrer le résultats suivant :

Proposition 2.1: Evaluation de l'erreur

Si f est une fonction de classe C^1 sur $[a, b]$, alors, pour tout entier naturel n non nul, on a

$$\left| R_n - \int_a^b f(t) dt \right| \leq \frac{M(b-a)^2}{2n},$$

où M est une constante telle que pour tout $x \in [a, b]$, $|f'(x)| \leq M$.

Proof. Supposons que $[a, b]$ est découpée en n segments de mêmes longueurs et considérons le k -ème segment, i.e. le segment $[x_k, x_{k+1}]$. Alors l'erreur d'approximation $e_{n,k}$ de l'intégrale sur ce segment est égale à

$$e_{n,k} = (x_{k+1} - x_k) f(x_{k+1}) - \int_{x_k}^{x_{k+1}} f(x) dx.$$

Comme f est de classe C^1 sur $[a, b]$, on peut alors appliquer l'inégalité des accroissements finis. Dans ce cas, il existe une constante $M = \sup_{x \in [a,b]} |f'(x)| > 0$ telle que pour tout $x, x' \in [a, b]$ on ait :

$$f(x) - f(x') \leq M|x - x'|.$$

Ainsi

$$\begin{aligned}
\left| R_n - \int_a^b f(t) dt \right| &= \left| \sum_{k=1}^n e_{n,k} \right|, \\
&= \left| \sum_{k=1}^n (x_{k+1} - x_k) f(x_{k+1}) - \sum_{k=1}^n \int_{x_k}^{x_{k+1}} f(t) dt \right|, \\
&= \left| \sum_{k=1}^n \int_{x_k}^{x_{k+1}} (f(t) - f(x_k)) dt \right|, \\
&\quad \downarrow \text{inégalité triangulaire et inégalité sur l'intégrale} \\
&\leq \sum_{k=1}^n \int_{x_k}^{x_{k+1}} |f(t) - f(x_k)| dt, \\
&\quad \downarrow \text{inégalité des accroissements finis} \\
&\leq \sum_{k=1}^n \int_{x_k}^{x_{k+1}} M |t - x_k| dt, \\
&\leq \sum_{k=1}^n M \frac{(x_{k+1} - x_k)^2}{2} dt, \\
&\quad \downarrow \text{pour tout } k, x_{k+1} - x_k = \frac{b-a}{n} \\
&= \sum_{k=1}^n \frac{M(b-a)^2}{2n^2}, \\
&= \frac{M(b-a)^2}{2n}.
\end{aligned}$$

□

On note alors que cette approche a une vitesse de convergence dite *linéaire*. En effet, lorsque l'on multiplie le nombre de rectangles par 10, l'erreur d'approximation par 10 !

Implémentation

On cherchera à estimer la valeur de l'intégrale de la fonction f définie de $[-2, 4]$ dans \mathbb{R} par

$$f(x) = x^2 - 3x + 2.$$

- Créer une fonction qui dépend de n , entier gérant la taille de la subdivision du segment étudié, et qui estime la valeur de l'intégrale de f par la méthode des rectangles décentrés.

- Représenter l'erreur d'estimation en fonction de n et comparer cette valeur à la majoration de l'erreur d'approximation établie dans la proposition précédente.

La convergence est dite linéaire car si vous multipliez la valeur de n par 10, par exemple, alors l'approximation sera 10 fois plus précise.

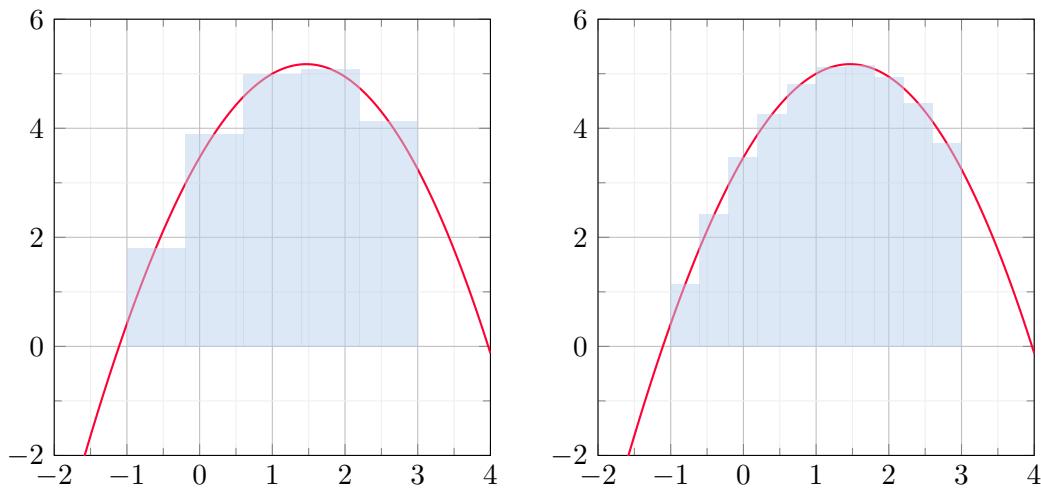
Regardons maintenant une approche où les rectangles sont centrées. Nous verrons que cette approche est plus précise et permet de converger plus rapidement vers la vraie valeur.

2.1.2 Méthodes des rectangles centrés

Description On considère à nouveau la même subdivision que précédemment. Ensuite, étant donné un segment $[x_k, x_{k+1}]$ le rectangle considéré aura une hauteur h_k égale à

$$h_k = f\left(\frac{x_{k+1} - x_k}{2}\right),$$

i.e. la hauteur du rectangle est égale à l'image de la moitié du segment $[x_k, x_{k+1}]$ par la fonction f . Sur notre exemple précédent, une subdivision de taille 5 et 10 nous donnera les approximations suivantes : à gauche pour une subdivision de taille 5 et à droite pour une subdivision de taille 10.



L'aire sous notre courbe est alors approchée par

$$R_n = \sum_{k=1}^n (x_{k+1} - x_k) \times f\left(\frac{x_{k+1} - x_k}{2}\right),$$

où $x_1 = a$ et $x_{n+1} = b$. L'aire que l'on souhaite alors estimer est alors obtenue par un passage à la limite.

Regardons à nouveau la convergence de cette méthode.

Erreur d'approximation On s'intéresse ici à l'erreur e_n effectuée par cette méthode. On va d'abord regarder l'erreur effectuée sur chaque segment de la subdivision puis sommer ces erreurs là et effectuer une petite majoration. Le résultat est le suivant :

Proposition 2.2: Evaluation de l'erreur

Si f est une fonction de classe C^2 sur $[a, b]$, alors, pour tout entier naturel n non nul, on a

$$\left| R_n - \int_a^b f(t) dt \right| \leq \frac{M'(b-a)^3}{24n^2},$$

où M est une constante telle que pour tout $x \in [a, b]$, $|f''(x)| \leq M$.

Proof. Supposons que $[a, b]$ est découpée en n et considérons le k -ème segment, i.e. le segment $[x_k, x_{k+1}]$. Alors l'erreur d'approximation e_k de l'intégrale sur ce segment est égale à

$$e_{n,k} = (x_{k+1} - x_k) f\left(\frac{x_{k+1} - x_k}{2}\right) - \int_{x_k}^{x_{k+1}} f(x) dx.$$

Dans la suite, pour simplifier les notations, nous poserons $x'_k = (x_{k+1} - x_k)/2$. Si notre fonction f est au moins de classe C^2 , i.e. dérivable et la dérivée est continue, alors la formule de Taylor-Lagrange à l'ordre 2 nous dit pour tout $x \in [a, b]$ il existe un réel $c_0 \in [a, b]$ tel que

$$f(x) = f(x'_k) + (x - x'_k) f'(x'_k) + \frac{(x - x'_k)^2}{2} f''(c_0).$$

Comme f est de classe C^2 sur $[a, b]$, il existe donc $M > 0$ tel que pour tout $x \in [a, b]$, $f''(x) < M$. En intégrant de part et d'autre de l'inégalité sur le segment $[x_k, x_{k+1}]$, on a alors

$$|e_{n,k}| = \left| \int_{x_k}^{x_{k+1}} f(x) - f(x'_k) dx \right|,$$

$$\begin{aligned}
&\downarrow \text{ formule de Taylor-Lagrange} \\
&\leq \left| f'(x_k) \underbrace{\int_{x_k}^{x_{k+1}} (x - x'_k)}_{\downarrow \text{ cette première intégrale est nulle}} \right| + \frac{M}{2} \int_{x_k}^{x_{k+1}} (x - x'_k)^2, \\
&= 0 + \frac{M}{6} [(x - x'_k)^3]_{x=x_k}^{x=x_{k+1}}, \\
&\downarrow x_{k+1} - x_k = \frac{b-a}{n} \text{ donc } x_{k+1} - x'_k = -(x_k - x'_k) \frac{b-a}{2n} \\
&= \frac{M}{3} \left(\frac{b-a}{2n} \right)^3, \\
&= \frac{M}{24n^3} (b-a)^3.
\end{aligned}$$

En sommant alors sur les différents rectangles, on a

$$\left| R_n - \int_a^b f(t) dt \right| = |e_n| \leq \sum_{k=1}^n |e_{n,k}| \leq \sum_{k=1}^n \frac{M(b-a)^3}{24n^3} = \frac{M(b-a)^3}{24n^2}.$$

□

Implémentation

On cherchera à estimer la valeur de l'intégrale de la fonction f définie de $[-2, 4]$ dans \mathbb{R} par

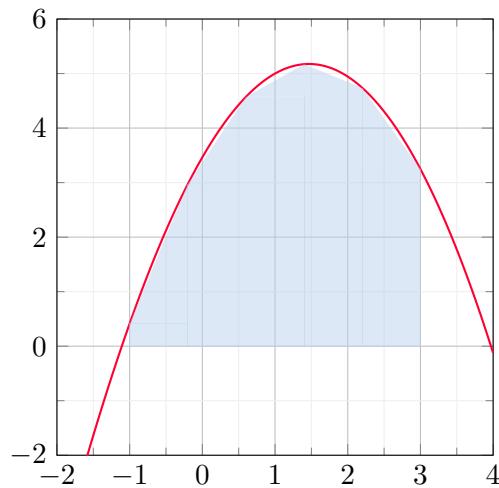
$$f(x) = x^2 - 3x + 2.$$

1. Créer une fonction qui dépend de n , entier gérant la taille de la subdivision du segment étudié, et qui estime la valeur de l'intégrale de f par la méthode des rectangles centrés.
2. Représenter l'erreur d'estimation en fonction de n et comparer cette valeur à la majoration de l'erreur d'approximation établie dans la proposition précédente.

On note alors que cette approche a une vitesse de convergence dite *quadratique* cette fois-ci. En effet, lorsque l'on multiplie le nombre de rectangles par 10, on divise cette fois-ci l'erreur d'approximation par 100 !

2.2 Méthode des trapèzes

Description On reprend exactement le même principe que pour les deux méthodes précédentes. Sur notre exemple, nous obtenons l'approximation suivante avec une subdivision de taille 5 de notre segment.



On remarque cette approximation semble beaucoup mieux approcher l'aire sous la courbe représentative de notre fonction.

L'aire du trapèze t_k défini sur le segment $[x_k, x_{k+1}]$ est donnée par

$$t_k = (x_{k+1} - x_k) \frac{f(x_{k+1}) + f(x_k)}{2}.$$

En sommant, notre intégrale est approchée par T_n définie par

$$T_n = \sum_{k=1}^n (x_{k+1} - x_k) \frac{f(x_k) + f(x_{k+1})}{2} = \frac{b-a}{n} \left(\frac{f(a) + f(b)}{2} + \sum_{k=2}^{n-1} f(x_k) \right).$$

Regardons maintenant l'erreur d'approximation de cette nouvelle approche.

Erreur d'approximation L'erreur de cette approximation par méthodes des trapèzes est également quadratique comme le présente la proposition suivante :

Proposition 2.3: Evaluation de l'erreur

Si f est une fonction de classe C^2 sur $[a, b]$, alors, pour tout entier naturel n non nul, on a

$$\left| T_n - \int_a^b f(t) dt \right| \leq \frac{M''(b-a)^3}{12n^2},$$

où M'' est une constante telle que pour tout $x \in [a, b]$, $|f''(x)| \leq M''$.

Proof. On suppose à nouveau que notre fonction f est de classe C^2 sur le segment $[a, b]$, ainsi, on sait qu'il existe une constante M'' telle que pour tout $x \in [a, b]$ on ait $f''(x) \leq M''$.

On utilisera ensuite le lemme suivant

Lemme 2.1: Evaluation de l'erreur

Soit f est une fonction de classe C^2 sur $[a, b]$. On considère également P un polynôme de degré 1 qui coïncide avec f en $x = a$ et $x = b$.

Alors il existe un entier $\xi \in [a, b]$ tel que

$$f(x) - P(x) = (x-a)(x-b) \frac{f''(\xi)}{2}.$$

Proof. Pour tout réel $x \in [a, b]$, on considère la fonction $q(x) = (x-a)(x-b)$ et la fonction W

$$W(t) = f(t) - P(t) - \frac{q(t)}{q(x)}(f(x) - P(x)).$$

La fonction W s'annule en au moins 3 points donc la dérivée seconde s'annule en au moins 1 point (c'est une conséquence d'une double application du théorème de Rolle). Donc, il existe $\xi \in [a, b]$ tel que pour tout $x \in [a, b]$

$$W''(\xi) = 0 \iff f''(\xi) - \frac{2(f(x) - P(x))}{(x-a)(x-b)} \iff f(x) - P(x) = (x-a)(x-b) \frac{f''(\xi)}{2}.$$

□

L'erreur d'estimation $e_{n,k}$, sur le k -ème trapèze est donnée par

$$e_{n,k} = \left| (x_{k+1} - x_k) \frac{f(x_{k+1}) + f(x_k)}{2} - \int_{x_k}^{x_{k+1}} f(x) dx \right|.$$

On peut alors effectuer les majorations suivantes en utilisant le lemme précédent, et le fait que notre dérivée seconde est bornée par M'' , pour majorer notre erreur $e_{n,k}$

$$\begin{aligned} e_{n,k} &\leq \left| (x_{k+1} - x_k) \frac{f(x_{k+1}) + f(x_k)}{2} - \int_{x_k}^{x_{k+1}} f(x) dx \right|, \\ &\quad \downarrow \text{on applique le lemme précédent} \\ &\leq \int_{x_k}^{x_{k+1}} (x - x_k)(x_{k+1} - x) \frac{f''(\xi)}{2}, \\ &\quad \downarrow \text{continuité de la dérivée seconde sur un segment} \\ &\leq \int_{x_k}^{x_{k+1}} (x - x_k)(x_{k+1} - x) \frac{M''}{2} dx, \\ &= \left(\int_{x_k}^{x_{k+1}} (x_{k+1} - x)(x - x_{k+1}) dx + \int_{x_k}^{x_{k+1}} (x_{k+1} - x)(x_{k+1} - x_k) dx \right) \frac{M''}{2}, \\ &= -\frac{M''(x_{k+1} - x_k)^3}{6} + \frac{M''(x_{k+1} - x_k)^3}{4}, \\ &= \frac{M''}{12}(x_{k+1} - x_k)^3. \end{aligned}$$

Ainsi, l'erreur d'approximation de notre intégrale approchée par T_n s'obtient en sommant sur les différents segments de notre partition. D'où

$$\begin{aligned} \left| T_n - \int_a^b f(t) dt \right| &\leq \sum_{k=1}^n |e_{n,k}|, \\ &\quad \downarrow \text{on utilise ce qui précède} \\ &\leq \sum_{k=1}^n \frac{M''}{12}(x_{k+1} - x_k)^3, \\ &\leq \sum_{k=1}^n (b - a)^3 \frac{M''}{12n^3}, \\ &\leq (b - a)^3 \frac{M''}{12n^2}. \end{aligned}$$

□

Implémentation

On cherchera à estimer la valeur de l'intégrale de la fonction f définie de $[-2, 4]$ dans \mathbb{R} par

$$f(x) = x^2 - 3x + 2.$$

1. Créer une fonction qui dépend de n , entier gérant la taille de la subdivision du segment étudié, et qui estime la valeur de l'intégrale de f par la méthode des rectangles décentrés.
2. Représenter l'erreur d'estimation en fonction de n et comparer cette valeur à la majoration de l'erreur d'approximation établie dans la proposition précédente.
3. Représenter, sur un même graphique, la convergence de des différentes méthodes, *i.e.* l'erreur d'approximation effectuée par les différentes méthodes.

On note alors que cette approche a également une vitesse de convergence quadratique comme la méthode des rectangles centrés.

2.3 Méthode de Simpson

Il s'agit d'une dernière méthode d'approximation que l'on se contente de présenter avec les résultats associés mais sans démonstration, ni implémentation.

Description La méthode de Simpson consiste à approximer une portion de la courbe de f , *i.e.* sur un segment de la subdivision, par un polynôme de degré 2 qui prend les mêmes valeurs que f en les extrémités des segments $[x_k, x_{k+1}]$ et en les milieux de ces derniers que l'on note ξ_k .

Cette approximation se fait avec un polynôme de *Lagrange*.

Ainsi, on peut montrer que la valeur approchée S_n de l'intégrale de f sur le segment $[a, b]$ par la méthode de Simpson est donnée par

$$S_n = \frac{(b-a)}{6n} \sum_{k=1}^n (f(x_k) + f(x_{k+1}) + 4f(\xi_k)).$$

Erreur d'approximation La proposition suivante montre que la vitesse de convergence vers la vraie valeur de l'intégrale est cette fois-ci cubique si l'on suppose que la fonction f de classe C^3 .

Proposition 2.4: Evaluation de l'erreur

i) f est de classe C^3 sur $[a, b]$, alors, pour tout entier naturel n , on a

$$\left| S_n - \int_a^b f(t) dt \right| \leq (b-a)^4 \frac{M}{192n^3},$$

où M est un nombre réel strictement positif tel que $\sup_{x \in [a,b]} |f^{(3)}(x)| \leq M$.

C'est en général cette approche qui est employée par les outils numériques pour estimer la valeur d'une intégrale.

3 Approche par dichotomie

Dans ce petit travail on cherchera à fournir une méthode dite *constructiviste* d'une version du Théorème des Valeurs Intermédiaires dont l'énoncé est le suivant :

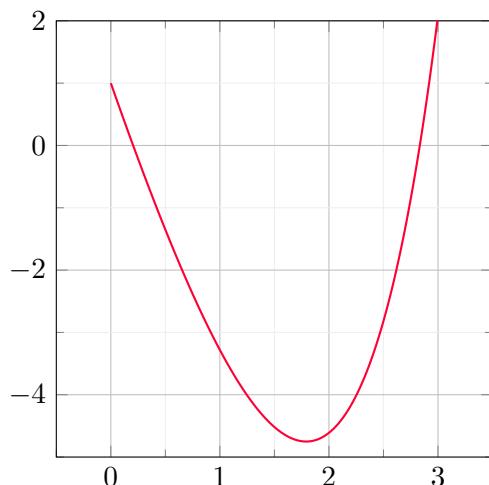
Théorème 3.1: Théorème des Valeurs Intermédiaires

Soit f une fonction continue d'un intervalle I dans \mathbb{R} et soient a et b deux points de I tels que $a < b$. Supposons également que l'on a $f(a)f(b) < 0$.

Alors il existe un réel $c \in [a, b]$ tel que $f(c) = 0$

Il est parfois simple de trouver pour quelles valeurs de x , une fonction f s'annule. C'est par exemple le cas pour certains polynômes de degré 2 ou pour les polynômes de degré 3 pour lesquels on sait trouver les zéros par méthodes des radicaux (*i.e.* on a une expression explicite des racines). ²

Mais cela n'est pas le cas pour toutes les fonctions comme la fonction f définie de \mathbb{R} dans \mathbb{R} par $f(x) = e^x - 6x$ dont la représentation graphique est donnée ci-dessous :



Trouver une expression simple des zéros de cette fonction n'est pas évident même si l'on arrive à montrer que cette fonction s'annule une ou plusieurs fois, en étudiant les variations de la fonction par exemple.

²Il est en fait possible de trouver de façon explicite les racines d'un polynôme de degré 1, 2, 3 ou 4. Pour les polynômes de degré 3 on sera amené à utiliser la **méthode de Cardan** et pour les polynômes de degré 4, on utilisera la **méthode de Ferrari**. En revanche, on peut montrer qu'à partie du degré 5 (voir **Théorie de Galois**, il n'est plus possible de trouver les racines d'un polynôme par méthode des radicaux, *i.e.* en utilisant la fonction racinée carrée.

Exercice

On s'intéresse à la fonction f définie de $[0, 3]$ dans \mathbb{R} par

$$f(x) = e^x - 6x.$$

1. Montrer que f est une fonction continue.
2. Montrer que f satisfait les conditions du théorème des valeurs intermédiaires et qu'elle s'annule au moins une fois sur l'intervalle $[0, 3]$
3. Montrer que f s'annule exactement deux fois sur l'intervalle $[0, 3]$ et en dresser le tableau de variation.

Il serait par contre intéressant de trouver une solution approchée des valeurs de x pour lesquelles la fonction f s'annule.

Pour cela, on va utiliser un raisonnement par dichotomie.

Raisonnement par dichotomie On va chercher à couper, itérativement, notre intervalle de recherche en deux.

On considère deux suites $(a_n)_{n \in \mathbb{N}}$ et $(b_n)_{n \in \mathbb{N}}$ qui vont définir les bornes de notre intervalle de recherche, telles que $a_0 = a$ et $b_0 = b$.

On pose $m_1 = \frac{a_0 + b_0}{2}$ et on évalue $f(m_1)$. On a donc deux possibilités :

1. $f(m_1)$ et $f(a_0)$ sont de signes contraires,
2. $f(m_1)$ et $f(b_0)$ sont de signes contraires.

Dans le premier cas, cela signifie que la valeur c recherchée se trouve dans l'intervalle $[a_0, m_1]$, sinon, deuxième cas, elle se trouve dans l'intervalle $[m_1, b_0]$.

- Si on est dans le cas 1., on posera alors $a_1 = a$ et $b_1 = m_1$
- Si on est dans le cas 2., on posera alors $a_1 = m_1$ et $b_1 = b$

Et on recommence le processus ! On posera ensuite $m_2 = \frac{a_1 + b_1}{2}$ et on évalue $f(m_2)$ et on applique la même règle que précédemment.

On a donc deux possibilités :

1. $f(m_2)$ et $f(a_1)$ sont de signes contraires,

- $f(m_2)$ et $f(b_1)$ sont de signes contraires.

Dans le premier cas, cela signifie que la valeur c recherchée se trouve dans l'intervalle $[a_1, m_2]$, sinon, deuxième cas, elle se trouve dans l'intervalle $[m_2, b_1]$...

On répète alors le processus un grand nombre de fois afin d'arriver aussi proche que souhaitait de la valeur inconnue. L'algorithme de dichotomie est appliqué à chaque fois au sous-intervalle dans lequel on observe un changement de signe. La longueur de l'intervalle étudié est divisée par 2 à chaque itération.

Ainsi, à la première itération du processus, nous sommes au plus à une distance au plus de $\frac{b-a}{2}$ de la valeur recherchée, puis de $\frac{b-a}{4}$ à la deuxième itération, etc. Plus généralement, on montre par récurrence le résultat suivant

Proposition 3.1: Erreur méthode par dichotomie

L'erreur, en valeur absolue de l'approche par dichotomie à l'itération n est au plus égale à $\frac{b-a}{2^n}$.

Ainsi, si on note x_n , la n -ème valeur de x obtenue par le raisonnement et x^* la valeur recherchée (*i.e.* le zéro de notre fonction) on a

$$|x^* - x_n| \leq \frac{b-a}{2^n}.$$

A l'aide de cette proposition, on peut montrer que si l'on cherche à approcher la solution à $\varepsilon > 0$ près, *i.e.* si on souhaite que $|x^* - x_n| < \varepsilon$, alors il faudra itérer le processus

$$n = \left\lceil \log_2 \frac{b-a}{\varepsilon} \right\rceil,$$

où $\lceil x \rceil$ représente le plus petit entier supérieur à x ³.

On va maintenant chercher à appliquer cette méthode

³La démonstration est laissée en guise d'exercice. Sur le plan algorithmique, ce n'est pas la méthode la plus efficace pour recherche un 0 d'une fonction, sa convergence est lente. Il existe d'autres méthodes (comme la méthode de Newton) qui permet de rechercher des zéros de façon plus efficace. En revanche cette méthode a l'avantage d'être applicable à beaucoup plus de situations

Implémentation

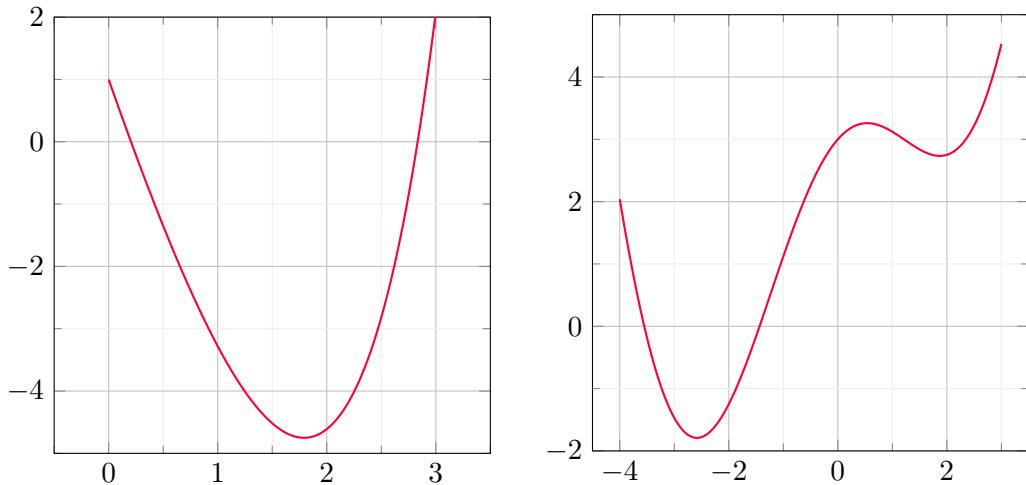
L'objectif est de chercher les zéros de la fonction f précédemment définie, *i.e.* à la fonction f étudiée sur le segment $[0, 3]$ par

$$f(x) = e^x - 6x.$$

1. On commence par se placer sur l'intervalle $[0, \ln(6)]$ et on admettra que la fonction f s'annule en $x = 0.204$ environ.
 - (a) Mettre en oeuvre l'approche par dichotomie pour approcher le zéro de cette fonction f sur l'intervalle d'étude.
 - (b) D'après la Proposition 3.1, combien d'itérations seront nécessaires pour avoir une solution approchée inférieure à 10^{-2} , *i.e.* au centième prêt.
 - (c) Représenter l'erreur d'approximation au cours des itérations et comparer la au résultat de la Proposition 3.1.
2. Répéter le processus sur l'intervalle $[\ln(6), 3]$ où l'on admettra qu'une valeur approchée du zéro est 2.833.

4 Optimisation d'une fonction

Dans cette partie, on va chercher à déterminer, de façon algorithmique comment on peut trouver le minimum d'une fonction f dans différents cas. Nous commencerons par étudier le cas où la fonction est convexe puis on s'attachera à étudier le cas où la fonction est non convexe.



Comme nous avons pu le voir en cours, une fonction convexe voit son extremum être un minimum global de la fonction. En revanche, une fonction non convexe (et non concave) peut admettre plusieurs extrema et de natures différentes, plusieurs minima et des maxima.

Notre objectif va être de développer des moyens algorithmiques de trouver ces

4.1 Cas d'une fonction convexe

On rappelle qu'une fonction f est dite convexe sur \mathbb{R} si pour tout $x, y \in \mathbb{R}$ et pour tout $\alpha \in [0, 1]$, on a

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y).$$

Nous avons vu en cours que si une fonction est convexe, elle admet donc un extremum qui est alors un minimum global de la fonction, et c'est ce minimum que l'on va chercher à déterminer.

Si la plupart du temps, le minimum d'une fonction est simple à déterminer de façon analytique comme c'est le cas pour la fonction f définie par $f(x) = 3x^2 - 5x + 2$. Pour

certaines fonctions, trouver le minimum globale de façon analytique peut se révéler plus compliquée comme avec les fonctions g et h respectivement définies par

$$g(x) = x^2 + e^x \quad \text{et} \quad h(x) = 4 + x^2 + \frac{1}{x^2 + 1}.$$

Exercice

Pour les différentes fonctions précédemment définies, *i.e.* les fonctions f, g et h

1. Etudier la fonction et dresser son tableau de variations.
2. Montrer que la fonction est convexe.
3. Quand cela est possible, déterminer la valeur de x pour laquelle la fonction atteint son minimum.

On se propose maintenant de trouver un moyen algorithmique de trouver ce minimum. On comprend facilement que pour le trouver, il suffit de parcourir l'ensemble des valeurs possibles de x jusqu'à trouver celle pour laquelle f est minimale ... mais il y en a une infinité ! On va donc chercher à utiliser les propriétés de la fonction f de sorte que l'on puisse se rapprocher du minimum à coup sûr, à chaque itération de notre algorithme.

Principe Supposons que l'on se donne une valeur de x , on peut donc évaluer la valeur de la fonction f en ce point. On va donc étudier comment perturber la valeur de x , en retranchant $t\delta_x$ par exemple (où $\delta_x > 0$ et t est une constante positive proche de 0).

On peut donc faire un développement limité à l'ordre 1 de la fonction f en $f(x-t\delta_x)$, ce qui nous donne

$$f(x - t\delta_x) = f(x) - tf'(x) \times \delta_x + t\varepsilon(t), \quad \text{où} \quad \lim_{t \rightarrow 0} \varepsilon(t) = 0.$$

D'après cette relation (mais cela sera encore plus évident lorsque l'on travaillera avec des fonctions à plusieurs variables), pour minimiser $f(x - t\delta_x)$ par rapport à δ_x , je dois minimiser $-tf'(x)\delta$ par rapport à δ_x , ce qui se fait en prenant δ_x du **même signe** (**dans le même sens**) que la **dérivée**, *i.e.* $\delta_x = f'(x)$. Ma nouvelle valeur de x devient alors $x - tf'(x)$ et je peux alors recommencer le processus.

On serait donc amener à construire une suite $(x_k)_{k \in \mathbb{N}}$ des valeurs x qui, on l'espère, convergera vers le minimum de la fonction f . Cette suite $(x_k)_{k \in \mathbb{N}}$ serait alors définie par :

$$x_{k+1} = x_k - t f'(x_k).$$

Quel est alors le rôle du paramètre $t > 0$? En fait ce paramètre va nous servir à définir à quel point je me déplacer dans le sens opposé du gradient, c'est ce que l'on appelle le **pas de descente**. Plus ce pas est petit, plus on reste proche de la valeur courante de x , plus il est grand, plus on s'éloigne de la valeur courante de x .

Quel est alors un bon choix de cette valeur de $t > 0$? On ne peut pas choisir cette valeur de t n'importe comment au risque de diverger ou encore de ne converger que très lentement vers la solution, comme cela est illustré en Figure 1. On verra un petit peu plus tard comment bien choisir cette valeur de t en fonction des propriétés de la fonction f (*i.e.* du caractère lipschitzien de la dérivée $f'(x)$ et du minimum de la dérivée seconde $f''(x)$).

Implémentation

On va chercher à mettre en évidence le phénomène observé en Figure 1. Pour cela, on va considérer la fonction f définie par

$$f(x) = 3x^2 - 6x + 2.$$

On initialisera notre algorithme à $x_0 = 3$ et calculera les 100 premières valeurs de la suite $(x_k)_{k \in \mathbb{N}}$ définie par

$$x_{k+1} = x_k - t f'(x_k).$$

1. Déterminer la valeur minimale de f et la valeur en laquelle f atteint ce minimum. On notera x^* cette valeur.
2. Pour les valeurs de t suivantes, on représentera graphiquement les différentes valeurs prises par la suite $(x_k)_{k \in \mathbb{N}}$ et on dira si la suite semble converger ou non vers x^* .
 - (a) $t = 0.34$,
 - (b) $t = 0.33$,
 - (c) $t = 0.01$.

Une fois que l'on a choisi un bon pas de descente, la prochaine question que l'on peut légitimement se poser concerne le nombre d'itérations: **quand dois-je m'arrêter ?**. On se pose donc la question du critère d'arrêt. On suppose que l'algorithme a convergé lorsque la valeur de la dérivée de la fonction étudiée est proche de 0, *i.e.* lorsque

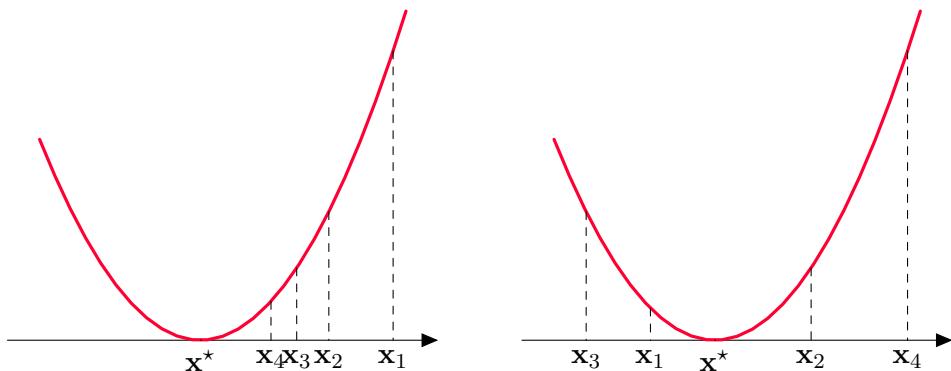


Figure 1: Illustration de l'importance du choix du pas d'apprentissage ρ . Une faible valeur implique une convergence très faible (à gauche). Une valeur trop grande entraîne la divergence de l'algorithme (droite).

$$|f'(x)| \leq \eta, \quad \text{où} \quad \eta > 0.$$

Question

Expliquer pourquoi le fait de choisir $|f'(x_k)| \leq \eta$, où $\eta > 0$ est un critère d'arrêt raisonnable pour notre algorithme.

Implémentation

Implémenter la méthode de descente de gradient pour déterminer le minimum des fonctions g et h . On essaiera différentes valeurs du pas d'apprentissage t et [Desmos](#) afin de s'assurer de la convergence de la procédure.

Choix du pas d'apprentissage Il existe plusieurs façons de choisir son pas d'apprentissage t , il peut être constant ou encore variable au cours des itérations. Quand les fonctions à optimiser sont très complexes, le plus souvent **on fera décroître le pas d'apprentissage $t = t_k$ vers 0 au cours des itérations**⁴.

Regardons maintenant comment bien choisir notre pas d'apprentissage et supposons que la fonction f que l'on cherche à minimiser est deux fois continuement dérivable, *i.e.* de classe C^2 .

⁴C'est loin d'être la seule solution et nous verrons en deuxième année d'autres techniques, *i.e.* d'autres [algorithmes de descente de gradient](#) de sélection du pas d'apprentissage en fonction du problème étudié

Proposition 4.1: Convergence de l'algorithme de descente de gradient

Soit I un intervalle de \mathbb{R} et soit f une fonction de classe C^2 telle que

1. sa dérivée f' soit lipschitzienne, *i.e.* il existe $L > 0$ telle que

$$\forall x, y \in I, \quad |f'(x) - f'(y)| \leq L|x - y|.$$

2. il existe $\alpha > 0$ telle que pour tout $x, f''(x) \geq \alpha$, ou, de façon équivalente, pour tout $x, y \in I$, on a

$$(f'(x) - f'(y))(x - y) \geq \alpha(x - y)^2$$

La fonction f est alors dite α -elliptique.

Alors la méthode de descente de gradient converge pour tout $0 < t < \frac{2\alpha}{L^2}$.

On va tenter de démontrer ce résultat qui n'est pas très compliqué et qui repose sur des définitions et propriétés vues en cours.

Exercice

On considère une fonction f de classe C^2 sur un intervalle I de \mathbb{R} qui vérifie les hypothèses de la Proposition 4.1. On considère que la fonction f atteint son minimum en x^* .

1. En utilisant la définition de x_{k+1} , montrer que

$$x_{k+1} - x^* = x_k - x^* - t(f'(x_k) - f'(x^*)).$$

2. Calculer $(x_{k+1} - x^*)^2$ à l'aide de la question précédente.

3. En utilisant les hypothèses 1. et 2. de la Proposition 4.1, montrer que

$$(x_{k+1} - x^*)^2 \leq (1 + t^2 L^2 - 2\alpha t)(x_k - x^*)^2.$$

4. Dans la suite, on posera $\gamma^2 = 1 + t^2 L^2 - 2\alpha t$. Montrer, par récurrence, que l'on a

$$(x_{k+1} - x^*)^2 \leq (\gamma^2)^{k+1} (x_0 - x^*)^2$$

5. A quelle condition sur γ , la quantité $(\gamma^2)^{k+1}$ tend vers 0 ?

6. En déduire une condition sur t dépendante de α et L telle que la suite $(x_k - x^*)_{k \in \mathbb{N}}$ tende vers 0, i.e. telle que la suite $(x_k)_{k \in \mathbb{N}}$ tende vers x^* .

Quelques applications Maintenant que l'on a établi un critère de convergence de notre algorithme, on peut essayer d'appliquer cela à différentes fonctions dont on cherche le minimum à l'aide de l'algorithme de descente de gradient.

- la fonction f définie de \mathbb{R} dans \mathbb{R} par

$$f(x) = 5x^2 - 6x + 2.$$

- la fonction g définie de $[-2, 2]$ dans \mathbb{R} par

$$g(x) = x^2 + \arctan(x).$$

- La fonction h définie de $[-\pi, \pi]$ dans \mathbb{R} par

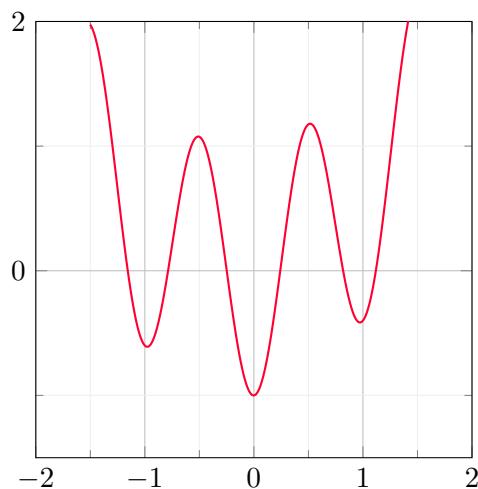
$$h(x) = x^2 + \cos(x)$$

- La fonction k définie de $[-1, -3]$ dans \mathbb{R} par

$$k(x) = \sin(x) + \cos(x)$$

Malheureusement, toutes les fonctions étudiées ne sont pas forcément convexes comme cela peut être le cas de la fonction u définie de \mathbb{R} dans \mathbb{R} par

$$u(x) = \cos(2\pi x) + \frac{x^2}{2} + 0.1x.$$



4.2 Cas d'une fonction non convexe

L'hypothèse de convexité n'est cette fois-ci pas satisfaite, on va par exemple considérer la fonction u précédemment définie.

Implémentation

Soit u la fonction définie de \mathbb{R} dans \mathbb{R} définie par

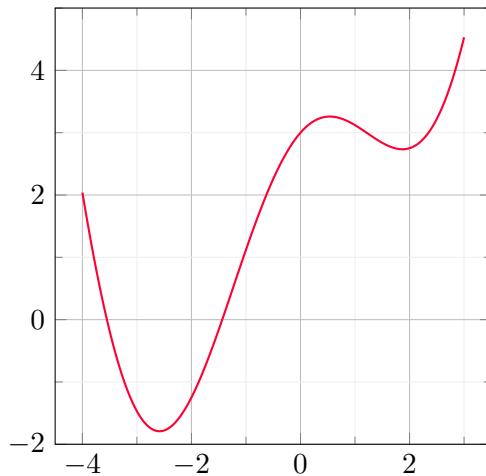
$$u(x) = \cos(2\pi x) + \frac{x^2}{2} + 0.1x$$

1. Montrer que la fonction u n'est pas convexe.
2. Déterminer son minimum global, noté x^* .
3. Déterminer la dérivée de la fonction u et une constante de lipschitz de u' .
4. Essayer de minimiser la fonction u à l'aide de l'algorithme de descente de gradient en prenant un pas d'apprentissage de l'ordre de 0.005 et en testant différentes initialisations (on fixera un nombre d'itérations égal à 1000). Qu'observez-vous ?

Etudions maintenant une solution qui permet de palier au problème précédemment rencontré.

4.3 Une solution à la non convexité

On se propose de présenter une solution pour trouver le minimum global d'une fonction non convexe, comme celle illustrée ci-dessous



On s'attachera uniquement à présenter la méthode mais on ne cherchera pas à expliquer ou à justifier le fonctionnement de cette méthode⁵. Cette dernière permet de garantir que l'on convergera vers le minimum global mais ... pas à coup sûr ! Uniquement avec une certaine probabilité, ce que nous montrerons via des simulations

Recuit simulé Il s'agit d'un algorithme qui se base sur des lois de physique qui décrivent le refroidissement d'un système⁶. Plus précisément, on cherche à étudier la probabilité p qu'un système physique se trouve dans un état avec une certaine énergie E et qui dépend de la température du système

$$p(x) = p_0 e^{\left(-\frac{E(x)}{k_B T} \right)},$$

où $p_0 > 0$, k_B est appelée la constante de Boltzmann.

On va s'inspirer de cette loi de la physique et essayer de voir comment utiliser cette notion de température pour étudier l'énergie de notre système, qui n'est rien d'autre que la fonction que l'on cherche à optimiser. Dit autrement, la probabilité que l'on va chercher à étudier est la suivante

$$p(x) = e^{\left(-\frac{f(x)}{T} \right)}.$$

Mais comment lier cette probabilité à la minimisation de notre fonction f ?

⁵En effet, les outils utilisés sont dits *probabilistes* et nécessitent de travailler avec des processus aléatoires.

⁶C'est une notion qui trouve ses origines dans la métallurgie dans lequel le système de *recuit* est employé.

Metropolis Hastings Pour une température T donnée, l'algorithme de Metropolis-Hastings⁷ consiste à générer une suite d'états (ici de valeurs de x) qui converge vers un ensemble d'états obéissant à la loi de probabilité de Boltzmann (ou une autre loi) mais laissons ce point là de côté et regardons son fonctionnement en pratique.

On commence par une valeur x choisie aléatoirement dans l'intervalle $]a, b[$ dans lequel on cherche le minimum de notre fonction. La suite d'états (x) est générée par l'algorithme suivant :

- on fixe une variable Δx qui permettra de faire varier l'état de notre système, *i.e.* perturber la valeur de x dans la recherche du minimum de la fonction f . On définira un nouvel état x' par la relation

$$x' = x \pm \Delta x \quad (1)$$

- si notre nouvel état est favorable, *i.e.* si $f(x') < f(x)$, alors on accepte ce nouvel état (en effet, on se rapproche du minimum). Ce nouvel état est plus *probable* au sens de Boltzmann.
- si notre nouvel état est moins favorable, *i.e.* si $f(x) < f(x')$, alors on accepte ce nouvel état (en effet, on se rapproche du minimum) avec probabilité égale au rapport des probabilités des deux états, *i.e.* égale à

$$p(x) = e^{\left(-\frac{f(x') - f(x)}{T}\right)} \quad (2)$$

Algorithme du recuit simulé L'idée de cet algorithme est *simple* et se détache de la notion de gradient précédemment présentée. On va plutôt parcourir notre espace de recherche de façon *aléatoire* et on acceptera notre solution (notre nouvelle valeur de x) si elle permet de bien diminuer la valeur de f . Mais ce seul argument ne suffit pas car on prend le risque d'être enfermé dans un *minimum local*, c'est pourquoi on autorise aussi notre algorithme à accepter des valeurs de x moins favorables avec une certainement probabilité.

On peut alors se demander le rôle du paramètre T dit de température. On remarque que ce dernier permet de contrôler la probabilité d'accepter un état moins favorable et donc de potentiellement sortir d'un minimum local. L'idée sera ensuite de faire baisser

⁷La présentation est inspirée de celle effectuée [ici](#).

Algorithm 1: Algorithme de recuit simulé

Inputs : une fonction f à minimiser, l'intervalle $[a, b]$ de recherche

Output: la valeur de x

```
1: on pose  $\Delta x = 0.01$ 
2: liste_T =  $10^{2:-2}$  (une liste de taille 200)
3:  $x$  tiré aléatoirement dans  $[a, b]$ 
4: for  $T$  dans liste_T do
5:   for  $i = 1, \dots, m$  do
6:     tirer aléatoirement la valeur signe égale à 1 ou  $-1$ 
7:      $x' = x + signe \times \Delta x$  (Equation 2).
8:     if  $f(x') < f(x)$  then
9:        $x \leftarrow x'$ 
10:    else
11:      calculer  $p$  à l'aide de l'Equation 2
12:       $x \leftarrow x'$  avec une probabilité  $p$ .
13:    end if
14:  end for
15: end for
```

cette température de sorte à ce que la valeur de x se stabilise (on peut voir que p tend vers 0 lorsque T tend vers 0) au fur et à mesure des itérations de notre algorithme présenté en Algorithme 1.

Implémentation

Illustrons maintenant cette procédure sur la fonction f définie par

$$f(x) = \frac{x^2}{2} + x + 3 \cos(x).$$

dont on cherchera le minimum sur l'intervalle $[-4, 3]$.

On rappelle que l'on fixera max_iter à la valeur 1000, on fera varier le paramètre de température T de 100 à 10^{-2} et Δx à 0.01.

1. Créer une fonction *RecuitSimule* qui prendra comme paramètres : la fonction à minimiser, les bornes de l'intervalle de recherche et qui met en oeuvre l'algorithme de recuit simulé précédemment décrit.
2. Pour chaque valeur de T , on affichera la valeur de x obtenue sur un graphe. Qu'est-ce que vous observez ?
3. On admettra que le minimum de cette fonction est localisé autour du point $x^* = -2.585$. Exécuter la procédure de recuit simulé 100 fois et évaluer

le nombre de fois où l'algorithme converge vers le minimum global. On admettra que l'algorithme converge lorsque la différence (en valeur absolue) entre la valeur retournée par la fonction recuit simulé et x^* est inférieure à 0.02. (La procédure peut être un peu longue)

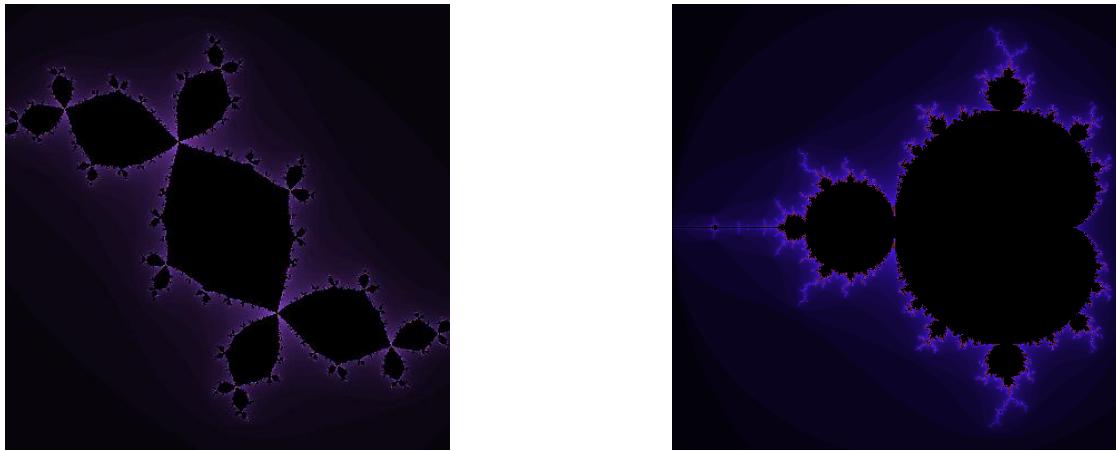


Figure 2: Représentation de deux fractales très célèbres de l'étude de la dynamique dite *holomorphe* (étude des suites de nombres complexes, le terme holomorphe faisant référence à des fonctions définies dans le champ des nombres complexes). A gauche, on retrouve le *Lapin de Douady*, du nom du célèbre mathématicien français **Adrien Douady (1935-2006)** et à droite, l'ensemble de *Mandelbrot* du nom du mathématicien Benoît Mandelbrot (1924-2010).

5 Informatique, art et suites complexes

Dans cette dernière séance de mathématiques sous Python, on va s'intéresser à s'intéresser à des objets issus d'une branche appelée "*la théorie du chaos*".

Le but n'est pas d'étudier d'un point de vue mathématique ces objets mais de s'en servir comme prétexte pour réaliser des simulations sur Python et obtenir des figures "artistiques" comme celle obtenue en Figure 2 représentant les très célèbres *Lapin de Douady* ou encore *Ensemble de Mandelbrot*.

5.1 Suite logistique

Regardons ce que l'on appelle le chaos à travers la définition d'une suite définie par récurrence de la forme $x_{n+1} = f(x_n)$. Plus précisément, on va s'intéresser à une équation très connue et fondamentale dans l'étude de la dynamique des populations, que l'on appelle *Application Logistique de May* qui définit une suite $(x_n)_{n \in \mathbb{N}}$ par

$$x_{n+1} = \mu x_n(1 - x_n), \quad x_0 \in [0, 1].$$

On la retrouve aussi sous une forme dite différentielle, à savoir

$$x'(t) = \mu x(t)(1 - x(t)), \quad x(0) \in [0, 1],$$

où μ représente un facteur d'expansion de la population.

Dans les deux cas on suppose que le paramètre $\mu \in [0, 4]$ et que notre population est comprise entre 0 et 1. Le paramètre $\mu x(t)$ représente la faculté de la population à se reproduire : plus il y a d'individus, plus la reproduction est élevée. Ce terme est ensuite multiplié par $(1 - x(t))$ qui freine la reproduction selon les capacités, l'espace disponibles, ici fixé à 1.

On va d'abord chercher à déterminer si notre suite $(x_n)_{n \in \mathbb{N}}$, définie par $x_{n+1} = f(x_n)$ où $f(x) = \mu x(1 - x)$ converge ou non vers un point fixe mais aussi à déterminer la valeur de ce point fixe.

Proposition 5.1: Nature d'un point fixe

Soit $f : [a, b] \rightarrow \mathbb{R}$ de classe C^1 . Si une suite récurrente $x_{n+1} = f(x_n)$ converge vers un point ℓ c'est un point fixe de f (*i.e.*, on a $f(\ell) = \ell$).

Le point fixe ℓ est dit **attractif** si $|f'(\ell)| < 1$, *i.e.* il existe un intervalle J contenant ℓ , stable par f tel que pour tout $x_0 \in J$, la suite récurrente $x_{n+1} = f(x_n)$ associée converge vers ℓ et on a une inégalité $|x_n - \ell| \leq k^n |x_0 - \ell|$ avec $0 < k < 1$.

Le point fixe est dit **répulsif** si $|f'(\ell)| > 1$, une suite récurrente $x_{n+1} = f(x_n)$ ne peut converger vers ℓ que si elle est constante et égale à ℓ à partir d'un certain rang.

Proof. La démonstration repose sur l'inégalité des accroissements finis. Mais on ne cherchera pas à démontrer ce résultat ici. \square

Exercices

On commence par étudier la fonction f et on suppose que le paramètre μ appartient à l'intervalle $[0, 4]$

1. Montrer que pour tout $x \in [0, 1]$, $f(x) \in [0, 1]$.
2. Montrer que la fonction f est concave.
3. Déterminer deux points fixes de f ainsi que leur nature (attractif ou répulsif) à l'aide de la proposition précédente.

On va maintenant analyser informatiquement la convergence de notre suite $(x_n)_{n \in \mathbb{N}}$ pour différentes valeurs de μ .

Implémentation

Soit $\mu \in [0, 4]$ et considérons la suite $(x_n)_{n \in \mathbb{N}}$ définie par

$$x_{n+1} = \mu x_n(1 - x_n), \quad .$$

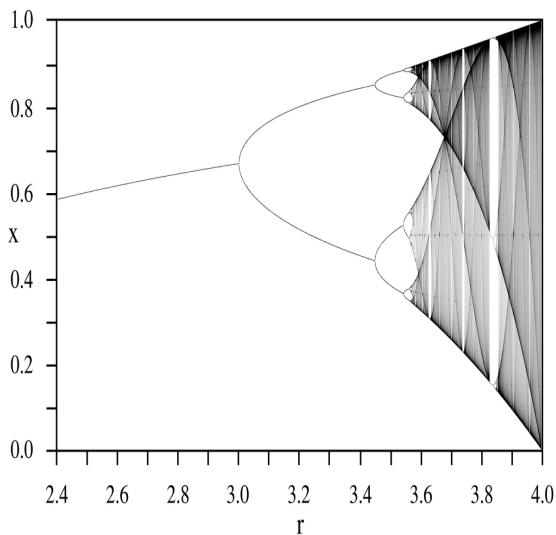
1. Représenter les 100 premières valeurs prises par la suite $(x_n)_{n \in \mathbb{N}}$ dans les cas suivants et décrivez les graphiques obtenus :
 - (a) lorsque $0 \leq \mu < 1$
 - (b) lorsque $1 < \mu < 2$
 - (c) lorsque $2 < \mu < 3$
 - (d) lorsque $3 < \mu < 3.5$
 - (e) lorsque $3.5 < \mu < 3.6$
 - (f) lorsque $3.8 < \mu < 4$

On prendra le soin d'écrire une fonction qui dépendra de deux paramètres n et μ qui représentent respectivement le nombre d'itérations de nombre d'algorithme et la valeur de notre coefficient d'expansion de la population.

2. Vos observations sont elles cohérentes avec l'étude précédemment effectuée ?
3. Dans le cas où $3 < \mu < 3.5$, comment devrions-nous choisir x_0 de sorte à ce que la suite soit constante ?

L'implémentation précédente a montré que le comportement de la suite était grandement différent en fonction de la valeur de μ . Plus précisément, on remarque que plus la valeur de μ augmente, plus la périodicité des solutions de notre suite augmente jusqu'à devenir *chaotique*.

Diagramme de bifurcation On va chercher à représenter les différentes valeurs entre lesquelles notre suite $(x_n)_{n \in \mathbb{N}}$ oscillent en fonction de la valeur de μ . Autrement dit, on va chercher à représenter le diagramme de bifurcation de la suite $(x_n)_{n \in \mathbb{N}}$, comme ci-dessous :



Implémentation

En employant la fonction précédente et en la modifiant, reproduire le diagramme de bifurcation ci-dessus.

Pour obtenir ce diagramme, il faut représenter les solutions périodiques de votre suite $(x_n)_{n \in \mathbb{N}}$ en fonction de la valeur de μ .

5.2 Un pas vers le chaos

La notion de *Chaos* est née de la physique ou des lois de la nature et trouve son origine chez les grands physiciens tels que Leibniz, Newton ou encore Laplace.

Mais on peut être en droit de se demander ce qu'est le *Chaos*. Si l'on peut s'intéresser à différentes natures de Chaos, on s'occupe ici uniquement du Chaos dit **déterministe**, par opposition au Chaos dit **stochastique**. Cela signifie qu'ils sont régis par une loi bien déterminée, qui doit permettre, en théorie, de décrire leur évolution à partir d'un état initial connu. Mais la réalité est bien plus que complexe que cela comme l'a montré **Henri Poincaré (1854-1912)** et notre système peut présenter des comportements différents selon la condition initiale. Un point que l'on illustre souvent avec cette célèbre phrase de **Edward Lorenz (1917-2008)** : *Le battement d'ailes d'un papillon au Brésil peut-il provoquer une tornade au Texas ?*. C'est d'ailleurs cette personne qui a montré que les phénomènes météorologiques peuvent comporter une variété importante de phénomènes quasi-périodiques mais au comportement instable car très sensibles aux paramètres a priori non contrôlables.



Figure 3: Deux fractales que l'on peut retrouver dans la nature : le choux romanesco à gauche et une fougère à droite.

Mais revenons-en à notre Chaos déterministe. Vous en avez eu un aperçu lors de la suite logistique précédemment étudiée, ce sont en fait des systèmes physiques ou dynamiques pour lesquels le comportement est imprévisible. Dans l'exemple de notre suite logistique, nous avons vu qu'en modifiant la valeur de notre paramètre régissant la capacité de reproduction de notre population μ , nous n'étions plus en mesure de prévoir le comportement de notre suite et les différentes valeurs que peut avoir cette dernière (faites tendre le paramètre μ vers 4 et tenter de voir si la suite présente ou non des périodicités).

Mais ce n'est évidemment pas la seule nature de chaos et on lui associe souvent d'autres objets que l'on retrouve très souvent dans la nature et dont quelques illustrations sont données en Figure 3, comme le *Choux Romanesco* ou une simple *Fougère*.

Bien sûr, on ne cherchera pas étudier ces objets particuliers, mais on peut remarquer une chose concernant, la régularité des motifs qui les composent ! En effet, si vous zoomez sur ces objets là, vous aurez le sentiment que ces motifs sont toujours identiques, peu importe l'échelle de l'observation ! C'est ce que l'on appelle des **fractales**⁸.

On va présenter deux fractales qui seront un peu prétextes pour faire de la programmation avec Python et une librairie graphique simple : *qtido*⁹. Bien évidemment, cela demandera aussi de la réflexion pour représenter ces objets !

Le Tapis de Serpinski Le Tapis de Serpinski, représenté en Figure 4, est un fractal (de dimension 1.9) que le lecteur aura déjà certainement rencontré au cours de différentes

⁸Ces objets sont étudiés majoritairement pour leur beauté, quand on les regarde d'un air naïf, mais ils possèdent énormément de propriétés mathématiques, dont l'une des plus surprenantes qui est que ce sont des objets de dimension **non entière** mais cette notion de dimension se abordé plus tard.

⁹Vous trouverez des informations relatives à cette librairie [ici](#)

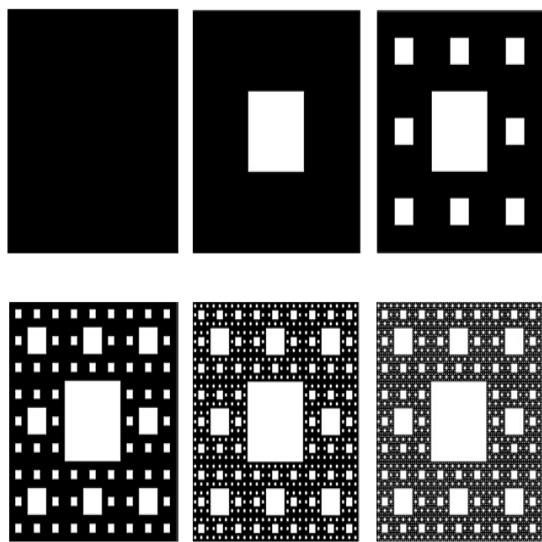


Figure 4: Illustration des différentes itérations de la construction du Tapis de Serpinski.

lectures. Son processus de construction est itératif comme on peut le voir sur cette même figure et dont la construction est la suivante.

On part d'un simple carré noir que l'on peut voir comme notre initialisation. Pour obtenir notre premier carré blanc, on va découper notre carré noir en 9 carrés égaux (3 lignes et 3 colonnes) et on va colorer notre carré central. Pour obtenir la deuxième image on effectue la même opération sur les 8 carrés noir précédents qui n'ont pas été colorés en blanc, *i.e.* chacun est à nouveau découper en 9 carrés réguliers et on colore le carré central. Ce processus est itéré pour obtenir les images successives de notre Tapis de Serpinski.

Implémentation

On cherchera à donner une représentation du Tapis de Serpinski pour les premières itérations, *i.e.* reproduire notre précédente illustration à l'aide de la bibliothèque *qtido* et notamment de sa fonction *rectangle*.

Le plus simple est de commencer par faire un petit dessin sur feuille afin de déterminer les coordonnées des différents carrés. Ne tentez pas tout de suite de faire une fonction pour représenter une itération donnée mais essayer de construire chaque étape afin de comprendre le fonctionnement.

Le Flocon de Koch Le Flocon de Koch, représenté en Figure 5, est un fractal (de dimension 1.26 environ), il est très proche de l'image que l'on peut se faire d'un flocon de

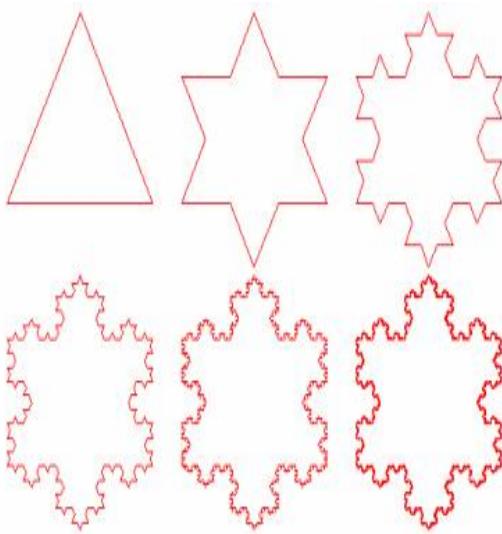


Figure 5: Illustration des premières étapes de construction du Flocon de Koch

neige. Son processus de construction est itératif comme on peut le voir sur cette même figure et dont la construction est la suivante.

On part d'un simple triangle équilatérale que l'on peut voir comme notre initialisation. Pour obtenir notre première itération, on découpe chaque côté de notre triangle en trois longueurs égales, on retire la partie centrale pour remettre un triangle équilatérale non les dimensions sont égales à la longueur du segment retiré. On répète ensuite ce même processus sur les tous les côtés nouvellement obtenus et le processus est itéré afin d'obtenir les différentes images de notre Flocon de Koch.

Implémentation

On cherchera à donner une représentation du Flocon de Koch pour les premières itérations, *i.e.* reproduire notre précédente illustration à l'aide de la bibliothèque *qtido* et notamment de sa fonction *triangle*. Cela est beaucoup plus compliqué que précédemment !

Le plus simple est de commencer par faire un petit dessin sur feuille afin de déterminer les coordonnées des différents triangles. Vous pouvez déjà tenté de reproduire la première ligne.

Dans la prochaine section, on cherchera à représenter des fractales à partir de la définition d'une simple suite, à travers l'étude des ensemble des Julia.

5.3 Ensembles de Julia

On enchaîne cette section avec une partie plus artistique quant à l'utilisation des mathématiques pour la réalisation de figures semblables à celles présentées en Figure 2.

Pour cela, on va à nouveau considérer des suites définies par récurrence de la forme

$$z_{n+1} = f(z_n).$$

A la seule différence que les valeurs de la suite $(z_n)_{n \in \mathbb{N}}$ ne sont plus **réelles mais complexes**. Avant d'en dire plus à ce sujet, on va faire quelques rappels sur les nombres complexes pour la suite de ce travail.

Rappel sur les nombres complexes .

Un nombre complexe $z \in \mathbb{C}$ est un nombre que l'on peut écrire sous la forme

$$z = a + ib,$$

où a, b sont des nombres réels et i est un nombre vérifiant $i^2 = -1$, i.e. il s'agit d'une racine carré de -1 . a est appelé partie réelle de z et b est la partie imaginaire de z .

Etant donné un même nombre complexe z , on appelle **module de z** ¹⁰, que l'on note $|z|$, la valeur

$$|z| = \sqrt{a^2 + b^2}.$$

On finit par rappeler comment effectuer la somme et le produit de nombres complexes. soit $z = a + ib$ et $z' = a' + ib'$ deux nombres complexes alors

$$z + z' = (a + a') + i(b + b').$$

On somme donc les parties réelles d'un côté et les parties imaginaires de l'autre.

De même

$$zz' = (a + ib)(a' + ib') = aa' - bb' + i(ab' + b'a).$$

En particulier, on a :

$$z^2 = a^2 - b^2 + 2iab.$$

¹⁰Vous trouverez aussi la définition de norme euclidienne d'un vecteur lors de votre cours d'Algèbre Linéaire l'année prochaine

Ensembles de Julia .

Ces ensembles trouvent naissance au cours du 20^{me} siècle et ont été nommés ainsi en l'honneur des mathématiciens français **Gaston Julia (1893-1978)** et **Pierre Fatou (1878-1929)** et ils sont complémentaires à l'ensemble de Fatou (que l'on étudie pas ici). En quelques mots, l'ensemble de Fatou est un ensemble de points du champ complexe en lesquels un faible changement de la condition de départ entraîne un faible changement de la suite des itérations, *i.e.* on peut considérer que la suite des itérations reste stable. l'ensemble de Julia est quant à lui, essentiellement caractérisé par le fait qu'une petite perturbation au départ se répercute en un changement radical de cette suite (on retrouve notre notion de chaos).

En dynamique holomorphe, l'ensemble de Julia et l'ensemble de Fatou sont deux ensembles complémentaires l'un de l'autre, définis à partir du comportement d'une fonction dans le champ complexe par composition itérée avec elle-même.

Revenons à l'étude de notre suite $z_{n+1} = f(z_n)$ et considérons la fonction f définie par $f(z) = z^2 + c$, où $c \in \mathbb{C}$. Il s'agit donc d'un polynôme de degré 2. Ce que l'on va chercher à faire, c'est essayer de représenter l'ensemble de Julia associé à cette suite pour une valeur de $c \in \mathbb{C}$ donnée. Plus précisément, on va chercher à représenter ce que l'on appelle l'ensemble de **Julia rempli** qui se définit comme **l'ensemble des valeurs z_0 pour lesquels la suite des itérées par la fonction f converge ou conserve des valeurs bornées en modules**. Il est parfois plus simple de voir l'ensemble de Julia rempli comme l'ensemble des points pour lesquels la suite des valeurs restent bornées en module.

Exercice

Soit $c \in \mathbb{C}$ dont le module est plus petit que 2 et soit $(z_n)_{n \in \mathbb{N}}$ la suite de nombres complexes définie par la relation de récurrence :

$$z_{n+1} = f(z_n) = z_n^2 + c, \quad z_0 \in \mathbb{C}.$$

1. On cherche à déterminer la valeur de c telle que le point 0 soit un périodique de période 3 par l'application f , *i.e.* que

$$f(f(f(0))) = 0.$$

Donner l'équation que doit vérifier c pour que l'égalité précédente soit vérifiée.

2. Pour obtenir l'ensemble de Julia rempli, on a dit qu'il fallait que les valeurs de la suite ne doivent pas devenir infiniment grandes comme décrit précédemment.

Montrer que si pour un entier n , $|z_n| > 2$ alors pour tout $m > n$ on aura $|z_m| > 2$ et la suite prend des valeurs infiniment grandes en module^a.

^aOn utilisera l'inégalité triangulaire inversée qui nous dit que $|a + b| \geq ||a| - |b||$.

Mais comment construire un tel ensemble en pratique à partir d'une valeur de c donnée ? Regardons ensemble les différentes étapes :

- On se place dans le plan en deux dimensions compris dans l'intervalle $[-2, 2]$ sur l'axe des abscisses et $[-2, 2]$ sur l'axe des ordonnées.
- On va créer une grille suffisamment fine sur ce plan et chaque noeud va représenter un point, donc un nombre complexe, et, dans notre cas, une valeur de z_0 .
- Pour chaque élément de la grille, c'est à dire pour les différentes possibles de z_0 , on va calculer les images successives z_n de z_0

$$z_n = \underbrace{f \circ f \circ f \circ \dots \circ f}_{n \text{ fois}}(z_0),$$

jusqu'à ce que l'on atteigne le nombre maximal d'itérations ou que l'un des itérés ait un module plus grand que 2.

- Dans le cas où le module des images de z_0 reste plus petit que 2, on dit que le point appartient à l'ensemble de Julien rempli, sinon il n'y appartient pas.

Il ne reste plus qu'à mettre cela en pratique !

Implémentation

Pour l'implémentation et notamment la représentation de tels ensembles, on se servira du script python intitulé *julia.py*.

On prendra garde à ne pas modifier la structure du code pour l'affichage et on ajoutera son propre code dans les endroits prévus à cet effet.

Vous pourrez ensuite tester votre code avec différentes valeurs de c dont le module est plus petit que 2. On pourra par exemple prendre la valeur de $c = -0.123 + 0.745i$. Pour cette valeur de c on pourra observer la dynamique qui s'installe en calculant les images successives de 0 afin de voir dans quel partie de l'ensemble se trouve les images. N'hésitez pas à tester d'autres valeurs de c comme $c = -1$ ou autres ...

5.4 Ensemble de Mandelbrot

L'ensemble de Mandelbrot, représenté à droite de la Figure 2, est un ensemble très lié aux ensembles de Julia précédemment définis.

On considère toujours la suite $(z_n)_{n \in \mathbb{N}}$ de nombres complexes définie par

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0.$$

Contrairement à l'ensemble de Julia, on va cette fois-ci fixer la valeur de z_0 et c'est la valeur de c que l'on fera varier. Formellement, un point c appartient à l'ensemble de Mandelbrot si la suite $(z_n)_{n \in \mathbb{N}}$ précédemment définie reste bornée.

En ce sens, nous pourrions alors utiliser la même procédure que pour la représentation des ensembles de Julia.

Implémentation

En employant une procédure analogue à celle effectuée pour représenter l'ensemble de Julia, représenter l'ensemble de Mandelbrot.

Pour aller plus loin .

Il est possible de faire un lien *topologique* entre ces deux ensembles. Par exemple, essayer de comparer l'ensemble de Julia associé à une valeur de c qui appartient à l'ensemble de Mandelbrot et à une valeur de c qui n'appartient pas à l'ensemble de Mandelbrot.

Enfin, si vous souhaitez continuer de jouer avec ces ensembles, vous pourriez plus précisément regarder la forme de l'ensemble de Julia selon le *bulbe* de l'ensemble de Mandelbrot dans lequel vous vous trouvez.

Ces objets ont aussi des propriétés arithmétiques et ... ils s'étendent à d'autres polynômes, pas forcément ceux de degrés 2, mais aussi de degrés supérieurs ou encore à des fonctions rationnelles (voir Figure 6) !

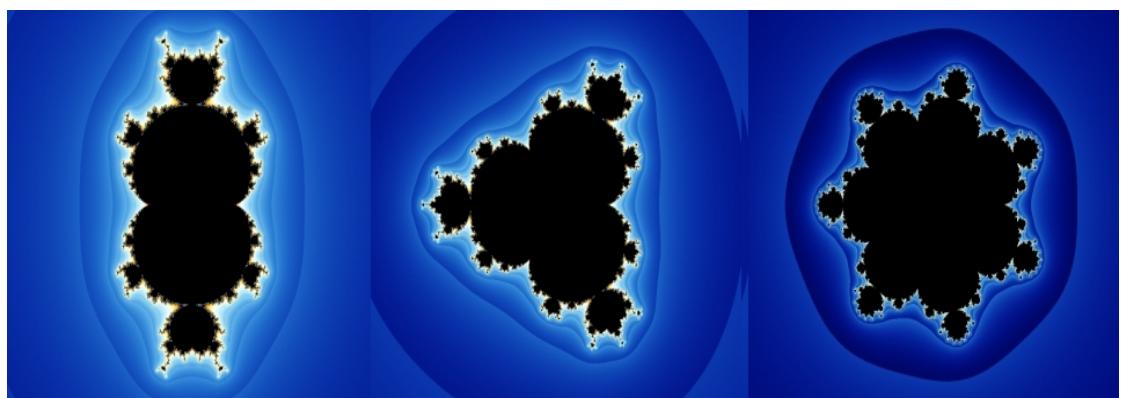


Figure 6: Représentation de l'ensemble de Mandelbrot associé à la fonction $f = z^p + c$ où $p = 3, 4$ et 8 respectivement. Image extraite du site [Maths CNRS](#).