DIRO
IFT 2425

# EXAMEN INTRA (TP N° 5 & 6)

**Bissection/Point Fixe**
**Monstres Numériques & Fractales du Point Fixe et de Newton**

*Max Mignotte*

DIRO, Département d'Informatique et de Recherche Opérationelle.
http : //www.iro.umontreal.ca/∼mignotte/IFT2425/
*E-mail : mignotte@iro.umontreal.ca*

## I. Méthodes Dichotomique & du point fixe

Le principal intérêt de la loi de distribution de Weibull réside dans sa flexibilité. Ainsi, pour certaines valeurs d'un de ses paramètres (le paramètre de forme $c$), on retrouve la loi exponentielle ($c = 1$) et la loi de Rayleigh ($c = 2$). Cette propriété explique l'utilisation qu'en font les statisticiens pour modéliser des échantillons de variables aléatoires exponentielles dont les critères de dépendances ne sont pas pleinement vérifiés.

En traitement d'image, cette flexibilité est aussi particulièrement intéressante pour modéliser la distribution du bruit de speckle dans une image SONAR [1], ou des niveaux de gris d'une images RADAR de la surface de la mer, ou encore pour modéliser la norme du gradient des niveaux de gris associé aux contours d'une image [2]. La loi de distribution de Weibull est définie par :

$$\mathcal{W}(y; c, \alpha) = \frac{c}{\alpha}\left(\frac{y}{\alpha}\right)^{c-1} \cdot \exp\left(-\frac{y^c}{\alpha^c}\right)$$

avec $y > 0$, $\alpha > 0$ et $c > 0$. $\alpha$ et $c$ sont respectivement un paramètre d'*échelle* et de *forme* strictement positifs. Nous avons représenté en Fig. 1 le tracé de cette loi pour différentes valeurs du facteur de forme $c$.

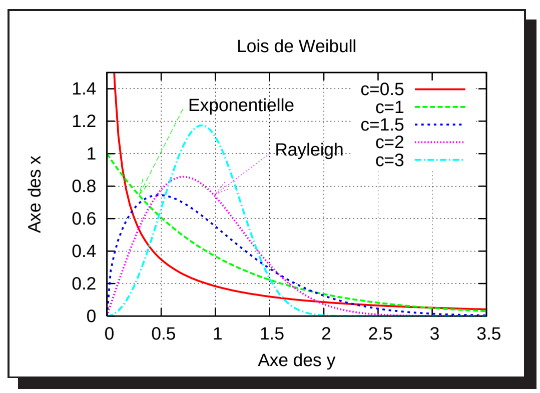


FIGURE 1 – Lois de Weibull pour différentes valeurs du facteur de forme $c$ (dans cet exemple, $\alpha = 1$)

La méthode du Maximum de Vraisemblance (MV) permet d'obtenir une estimation du vecteur de paramètres $\Phi_y = (\alpha, c)$ pour un échantillon $y$ donné (c'est-à-dire pour un ensemble de $N$ valeurs aléatoires $y_i$ suivant la distribution de Weibull). Si on suppose l'indépendance des différentes variables aléatoires $y_i$ de l'échantillon $y$, la fonction de vraisemblance $\mathcal{L}$ s'écrit :

$$\mathcal{L}(\Phi_y) = P(y/\Phi_y) = \left(\frac{c}{\alpha^c}\right)^N \prod_{i=1}^{N} \left[ y_i^{(c-1)} \exp\left(-\frac{y_i^c}{\alpha^c}\right)\right]$$

Pour déterminer $\hat{\alpha}_{\text{MV}}$ et $\hat{c}_{\text{MV}}$ (valeurs qui maximisent la fonction de vraisemblance), on doit résoudre simultanément les deux équations suivantes $\partial \ln \mathcal{L}(\Phi_y)/\partial\alpha = 0$ et $\partial \ln \mathcal{L}(\Phi_y)/\partial c = 0$, conduisant au système d'équations :

$$
\begin{aligned}
-N + \frac{1}{\hat{\alpha}^{\hat{c}}} \sum_{i=1}^{N} y_i^{\hat{c}} &= 0 \\
\frac{N}{\hat{c}} + \sum_{i=1}^{N} \ln y_i - N \ln \hat{\alpha} - \sum_{i=1}^{N} \big(\frac{y_i}{\hat{\alpha}}\big)^{\hat{c}} \ln\big(\frac{y_i}{\hat{\alpha}}\big) &= 0
\end{aligned}
$$

Après simplification, on trouve une forme explicite pour $\alpha_{\text{MV}}$ : $\alpha_{\text{MV}} = \big(\frac{1}{N}\sum_{i=1}^{N} y_i^{c_{\text{MV}}}\big)^{\frac{1}{c_{\text{MV}}}}$. Par contre on ne peut pas obtenir de forme explicite pour l'estimateur $c_{\text{MV}}$. En effet $c_{\text{MV}}$ est solution de l'équation :

$$
f(c_{\text{MV}}; y_i) = \frac{\sum_{i=1}^{N} \big(y_i^{c_{\text{MV}}} \ln y_i\big)}{\sum_{i=1}^{N} y_i^{c_{\text{MV}}}} - \frac{1}{c_{\text{MV}}} - \frac{1}{N} \sum_{i=1}^{N} \ln y_i = 0 \tag{1}
$$

qui n'a pas de solution analytique explicite et qui ne peut se résoudre que numériquement (par approximations successives) par l'une des méthodes numériques vues en cours [1,2].

Table 1 – Valeurs des $N = 10$ points de données $y_i$

| $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ | $y_8$ | $y_9$ | $y_{10}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0.11 | 0.24 | 0.27 | 0.52 | 1.13 | 1.54 | 1.71 | 1.84 | 1.92 | 2.01 |

## I.1 Méthode Dichotomique

Puisque l'estimateur du MV est unique, nous savons donc que l'Équation (1) admet forcément une solution. De plus, comme on sait que $c_{\text{MV}} > 0$, il suffira donc de démarrer la méthode de la *bissection* avec un intervalle dont la borne inférieure sera une valeur presque nulle ($c_{\text{MV}}^{\text{min}} = 0.50$) et la borne supérieure une valeur (que l'on sait être) trop grande pour $c_{\text{MV}}$ ($c_{\text{MV}}^{\text{max}} = 5.0$, par exemple).

**I.1.a–** Donner le nombre d'itérations qui serait nécessaire à la méthode de la bissection pour arriver, avec cet intervalle, à une estimation de la racine avec 5 cse après la virgule (sans estimer explicitement cette racine par la méthode de la bissection). Afficher avec un PRINTF à l'écran un raisonnement succinct et la réponse (le nombre d'itérations que vous prévoyez d'être nécessaire pour la méthode de bissection) comme préambule à la question suivante.

**I.1.b–** Programmer le méthode de bissection pour la recherche de la racine $\hat{c}_{\text{MV}}$ de l'Équation (1) en utilisant des nombres flottants, l'intervalle $[c_{\text{MV}}^{\text{min}}, \ldots, c_{\text{MV}}^{\text{max}}]$ et avec comme critère d'arrêt, une tolérance sur $\hat{c}_{\text{MV}}$ de $10^{-6}$.

## I.2 Méthode du Point Fixe

Isolez maintenant le $\hat{c}_{\text{MV}}$ qui se trouve au milieu de l'Équation (1) ($f(.) = \ldots - \frac{1}{c_{\text{MV}}} - \ldots = 0$) pour transformer $f(c_{\text{MV}}; y_i) = 0$ en une relation du type $c_{\text{MV}} = g(c_{\text{MV}}; y_i)$ puis en une suite itérative de la forme $c_{\text{MV}}^{[n+1]} = g(c_{\text{MV}}^{[n]}; y_i)$.

**I.2.a–** Programmer la méthode du point fixe lorsque l'estimée initiale (premier terme de la suite) est $c_{\text{MV}}^{[0]} = 2.50$, avec comme critère d'arrêt, une tolérance sur $\hat{c}_{\text{MV}}$ de $10^{-6}$.

**I.2.b–** En utilisant la racine estimée précédemment et la fonction $g(.)$, calculer la constante asymptotique de cette méthode itérative du point fixe et son ordre de convergence et vérifier numériquement qu'avec celle-ci, il est normal que l'on soit arriver au résultat estimé (à la tolérance voulue) après le nombre d'itérations donnée par cette méthode. Afficher avec un PRINTF à l'écran un raisonnement succinct qui résume ce que vous avez fait numériquement et qui affiche à l'écran ces deux réponses (convergence et borne supérieure du nombre d'itérations) à la suite du programme de la question I.2.a.

**I.2.c–** Implémenter la méthode itérative suivante (en utilisant toujours des FLOAT (en FLOAT mais avec une tolérance sur $\hat{c}_{\text{MV}}$ et sur $f$ de $10^{-6}$ et $c_{\text{MV}}^{[0]} = 2.50$) qui demande à chaque itération, deux étapes de la méthode du point fixe :

**Point Fixe Accéléré Par Aitken**

**Initialisation :** $c^{[0]} = 0.1$
**while** *critère d'arrêt non rempli* **do**
- **1.** Calculer : $c^{[1]} = g(c^{[0]}; y_i)$
- **2.** Calculer : $c^{[2]} = g(c^{[1]}; y_i)$
- **3.** Calculer : $c^{[3]} = c^{[0]} - \frac{(c^{[1]} - c^{[0]})^2}{(c^{[2]} - 2\, c^{[1]} + c^{[0]})}$
- **4.** $c^{[0]} \leftarrow c^{[3]}$

**I.2.d–** Programmer la méthode de l'interpolation linéaire lorsque l'intervalle de départ est $[c_{\text{MV}}^{\min} = 0.50 \quad c_{\text{MV}}^{\max} = 5.0]$ avec comme critère d'arrêt, une tolérance sur $\hat{c}_{\text{MV}}$ ou une tolérance sur $f$ de $10^{-6}$.

# II. Monstre Numérique & Fractales du Point Fixe et de Newton

## II.1 Monstre Numérique

Le terme *Monstre Numérique* vient de l'excellent article d'Essex *et al.* [3] répertoriant quelques célèbres exemples de problèmes d'erreures numériques existants en notation flottantes. Un exemple bien connu est la fonction :

$$y(x) = (x^2 - 2x + 1) - (x - 1)^2$$

Une arithmétique idéale donnerait bien sur $y(x) = 0$ quelque soit la valeur de $x$. Mais un calcul en notation flotante fait apparaître la soustraction de deux nombres approximés quasi-égaux qui peut s'avérer être une catastrophique numérique (cf. cours) et ceci restera vrais tant que les compilateurs ne seront assez intelligent pour faire un ré-arrangement algébrique automatique de cette formule !

En utilisant, un programme et compilateur c++ et en affichant la courbe $y(x)$ lorsque $x$ évolue dans l'intervalle $1.0 \pm 7.0 \times 10E-5$ et $y$ dans l'intervalle $0.0 \pm 0.8 \times 10E-8$, un intéressant graphe (cf. Fig. 2) émerge qui nous montre à quoi ressemble le motif généré par ce monstre numérique ...

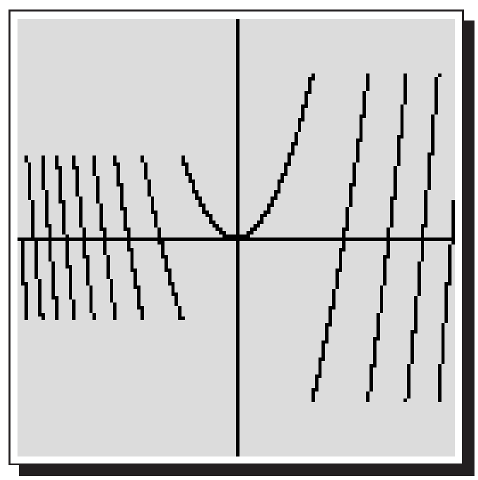

FIGURE 2 – Motif généré par le monstre numérique $(x^2 - 2x + 1) - (x - 1)^2$ sur $x = 1.0 \pm 7.0 \times 10E-5$ et $y = 0.0 \pm 0.8 \times 10E-8$

**II.1.a–** Retrouver ce graphe en affichant intelligemment le résultat $y(x)$ (pour $x$ et $y$ évoluant dans le bon intervalle) dans un tableau deux dimensions qui s'affichera ensuite à l'écran sous forme d'image [3].

## II.2 Fractales de Newton

Une figure fractale ou fractale (néologisme créé par Benoît Mandelbrot en 1974 à partir de la racine latine fractus, qui signifie brisé, irrégulier), est une courbe ou surface de forme irrégulière ou morcelée qui se crée en suivant des règles déterministes ou stochastiques impliquant une homothétie interne. Dans la "théorie de la rugosité" développée par Mandelbrot, une fractale désigne des objets dont la structure est invariante par changement d'échelle.

La fractale de Newton est un ensemble frontière défini dans le plan complexe caractérisé par l'application de la méthode itérative de Newton vue en cours ($x_{[n+1]} = x_{[n]} - f(x_{[n]})/f'(x_{[n]})$) sur une fonction polynômiale $f(z)$ de variable complexe $z$, comme par exemple ; $f(z) = z^3 - 1$. On construit une fractale de Newton sur $f(z)$ en appliquant la méthode itérative de Newton sur cette fonction, c'est à dire, si on considère $f(z) = z^3 - 1$, en itérant la suite itérative :

$$z_{[n+1]} = z_{[n]} - \frac{z_{[n]}^3 - 1}{3\, z_{[n]}^2} \qquad (1)$$

En fonction de la valeur de départ $z_0$, cette suite itérative convergera (si il y a convergence) vers l'une des trois racines possibles $r_k$ de $f(z)$, i.e. ; $r_0 = 1$, $r_1 = -1/2 + j\sqrt{3}/2$ ou $r_2 = -1/2 - j\sqrt{3}/2$ (où $j$ est le nombre imaginaire pur tel que $j^2 = -1$). Si cette suite itérative converge vers la racine $r_k$, alors $z_{[0]} = x_{[0]} + j\, y_{[0]}$ appartient à la région $G_k$ de coordonnée $(x_{[0]}, y_{[0]})$ qui est aussi appelée le bassin d'attraction de la racine $r_k$. Les trois racines possibles de la convergence de la suite itérative (1) divisent donc le plan complexe en trois régions et la Figure obtenue est la Fractale de Newton pour la fonction complexe considérée. Toutefois, pour tout polynôme de degré égal au moins à 2, il existe des points pour lesquels la suite de Newton ne converge pas, c'est le cas de la frontière des bassins d attraction de chaque racine.

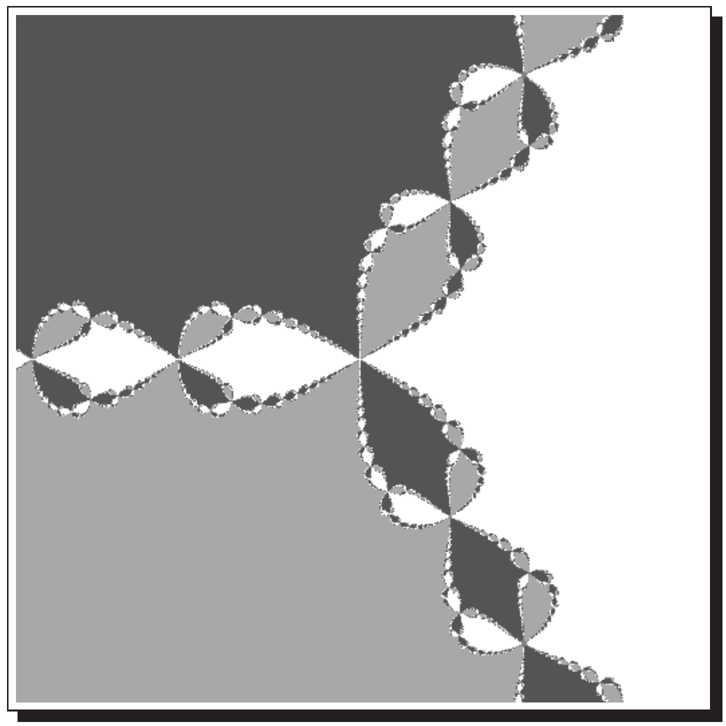

Figure 3 – Image fractale de Newton associée au polynome complexe $f(z) = z^3 - 1$ et les trois bassins d'attraction des racines du polynôme (en blanc, gris clair et gris foncé.

La fractale de Newton présente, comme toute fractale, une apparence complexe, malgré une description simple et des auto-similarités visibles à toutes échelles. Elle suggère également que la méthode de Newton peut être très sensible aux conditions initiales et que deux points initiaux infiniment proches peuvent converger vers des racines différentes. Elle montre, enfin, que chaque point de la fractale de Newton est un point-frontière multiple, séparant chacun des $n$ bassins d'attraction. Si deux points infiniment proches convergent vers deux racines distinctes, alors il existe un troisième point, infiniment proche également, qui converge vers la troisième racine [3].

**II.2.a–** Retrouver le graphe de la Fig. 3 en exprimant tout d'abord la suite itérative (1) comme deux suites itératives ; l'une associée à sa partie réelle ($x_{[n+1]} = x_{[n]} - \ldots$) et l'autre à sa partie imaginaire ($y_{[n+1]} = y_{[n]} - \ldots$). Pour chaque cellule d'un tableau $2D$, de dimension $512 \times 512$ (longueur×largeur), faites 20 itérations de ces deux suites (l'une après l'autre) en considérant comme valeur de départ $z_{[0]} = x_{[0]} + j\, y_{[0]}$ et avec :

$$x_{[0]} = 3 \times (j - \text{largeur}/2.0)/(\text{largeur} - 1) \qquad (2)$$
$$y_{[0]} = 3 \times (i - \text{longueur}/2.0)/(\text{longueur} - 1) \qquad (3)$$

avec $''i''$ et $''j''$ correspondant respectivement au numéro de la ligne et de la colonne de la cellule, de telle

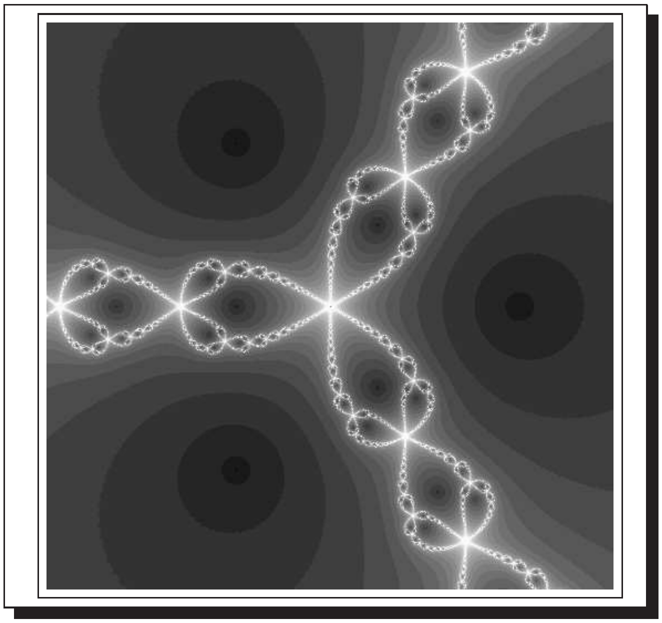

FIGURE 4 – Image fractale de Newton associée au polynome complexe $f(z) = z^3 - 1$ ou chaque pixel de l'image est colorié en fonction de la rapidité de la convergence de la suite.

façon que $x_{[0]}$ et $y_{[0]}$ varient, pour chaque cellule de ce tableau $2D$, de $-1.5$ à $1.5$ et que la cellule centrale de ce tableau $2D$ ait pour coordonné $(0,0)$. Remplir finalement ce tableau $2D$ en $[i][j]$ de respectivement 255, 170 ou 84 (correspondant respectivement à la couleur blanche, grise blanche et grise foncée) si la convergence à lieu respectivement pour la racine $r_0$, $r_1$ ou $r_2$ (attention à la comparaison entre deux flottants !). Afficher et sauver le résultat de ce tableau $2D$ sous forme d'image [2] (qui s'affichera ensuite à l'écran).

**II.2.b–** Retrouver le graphe de la Fig. 4 en donnant cette fois ci à chaque cellule du tableau $2D$ un niveaux de gris (compris entre $[0-255]$ d'autant plus blanc (ou proche de 255) si la série converge lentement vers une solution [2].

## II.3 Fractales du point fixe

Une autre fractale très intéressante est donnée par la suite itérative de point fixe (défini dans le plan complexe) suivante :

$$z_{[n+1]} = \Big(|\mathrm{Re}(z_{[n]})| + j\,|\mathrm{Im}(z_{[n]})|\Big)^2 + c \qquad (4)$$

dans laquelle $j$ désigne le nombre imaginaire pur ($j^2 = -1$), $\mathrm{Re}(z_{[n]})$ représente la partie réelle de $z_{[n]} = x_{[n]} + jy_{[n]}$, c'est-à-dire ; $\mathrm{Re}(z_{[n]}) = x_{[n]}$, $\mathrm{Im}(z_{[n]})$ représente la partie imaginaire de $z_{[n]} = x_{[n]} + jy_{[n]}$, c'est-à-dire ; $\mathrm{Im}(z_{[n]}) = y_{[n]}$, $|.|$ désigne la valeur absolue, $z_{[0]} = 0$ (*i.e.*, $x_{[0]} = 0$ et $y_{[0]} = 0$) et $c = c_x + j\,c_y$ est un nombre complexe pour lequel $c_x$ et $c_y$ varient (pour chaque cellule ou pixel d'un tableau 2D ou d'une image) de $-2$ à $1.5$. Après 1000 itérations de cette suite, on remplie finalement ce tableau 2D (ou cette image) en $[i][j]$ avec une valeur de niveaux de gris d'autant plus blanche que la série converge rapidement vers l'infinie (ou converge lentement ver une solution). l'affichage du résultat de ce tableau 2D sous la forme d'une image nous donne finalement la Fig. 5.

**II.2.c–** Pour cette question, on vous demande de retrouver l'image de la Fig. 6 en exprimant, tout d'abord, la suite (4) comme deux suites itératives ; l'une associée à sa partie réelle ($x_{[n+1]} = f_r(x_{[n]})$) et l'autre à sa partie imaginaire ($y_{[n+1]} = f_i(y_{[n]})$) et d'itérer ces deux suites pour 1000 itérations avec $x_{[0]} = 0$

et $y_{[0]} = 0$ et $c_x$ et $c_y$ ($c = c_x + j\,c_y$) prenant respectivement un nombre (SIZE $\times$ SIZE) de valeurs discrètes comprises entre $x_{\min} = -1.8$ et $x_{\max} = -1.7$ (pour $c_x$ et representant les colonnes de ce tableau) et $y_{\min} = -0.07$ et $y_{\max} = 0.005$ (pour $c_y$ et representant les lignes de ce tableau).
Remplir chaque cellule de ce tableau 2D (de cette image) en $[i][j]$ avec une valeur de niveaux de gris d'autant plus blanche que la série converge rapidement vers l'infinie, on calculera, pour chaque itération de la suite, le module au carré de $z_{[n+1]}$, c'est-à dire $|z_{[n+1]}|^2 = (x^2_{[n+1]} + y^2_{[n+1]})$ et on notera $k$ le nombre d'itérations nécessaires pour lequel $|z_{[n+1]}|^2| > 4$ (on suppose qu'il y a divergence dès que cela arrive). La cellule $[i][j]$ de ce tableau est ensuite remplie par le niveau de gris 255-ln[k+1]) (*i.e.*, TAB[i][j]=255-(ln(k+1))) (ln est le logarithme népérien).
On represente ensuite le tableau ou image 2D de dimension SIZE$\times$SIZE (longueur$\times$largeur) avec SIZE$= 512$ sous la forme d'une image [3].

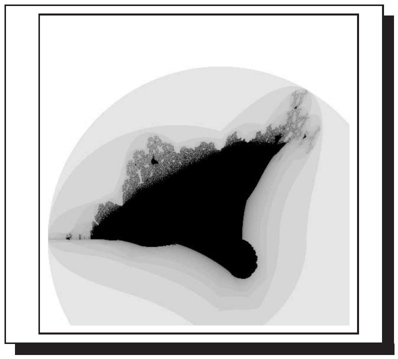

FIGURE 5 – Image fractale donnée par la suite itérative de point fixe de l'Équation (4) pour une image de taille $128 \times 128$ et $[x_{\min}\!=\!-2\ ,\ x_{\max}\!=\!1.5]$, $[y_{\min}\!=\!-2.5\ ,\ y_{\max}\!=\!1]$.

---

1.

## Remise & Rapport

Pour la remise électronique, utilisez le programme *remise* (*man remise* pour plus de détails) pour remettre votre code dans le répertoire TP5. N'oubliez pas d'inscrire votre nom, courrier électronique en commentaire en haut du fichier .c remis. Les noms des programmes à remettre devront avoir le format suivant *TP5-IFT2425-<Numéro de la question>.c* (par exemple : *TP5-IFT2425-I1b.c*). Les programmes devront se compiler et s'executer sur Linux et sur les ordinateurs Linux du DIRO.

2.

## Conseils Pratiques -1-

Aider vous du programme que je vous donne sur ma page web. Pour obtenir une image, l'idée est de remplir un tableau $2D$ d'un dégradé de niveaux de gris (correspondant au niveaux de gris de chaque pixel de cette image) comme une feuille de papier dessin ou vous tracerez la Figure dans l'intervalle considéré. N'oublier pas d'utiliser avant l'affichage la fonction RECAL() pour faire en sorte que ces niveaux de gris soient toujours compris entre 0 et 255 pour une image au format pgm (0 indiquant le noir, 255 le blanc et 128 le gris moyen).
Plus précisément, en initialisant la variable FLAG_GRAPH=1 et remplisser le tableau $2D$ nommé Graph2D dans le programme (initialement remplie d'un dégradé de niveaux de gris (du blanc au noir)), vous tracerez la Figure dans l'intervalle considéré. Chaque cellule de ce tableau $2D$ de dimension $512 \times 512$ correspond à une coordonnée $(x, y)$ et à une valeur complexe initiale $z = x + jy$ avec $x$ et $y$ variant de $-1.5$ à $1.5$. Le centre de l'image correspond à la coordonnée (0.0).

3.

## Conseils Pratiques -2-

Utiliser le programme que je vous donne sur ma page web comme programme de départ. Pour une image de taille $512 \times 512$ et $x_{\min} = -1$ et $x_{\max} = 1$, $y_{\min} = -1$ et $y_{\max} = 1$, la cellule [0][0] de ce tableau est donc associée au coord. ($x_{\min} = -1, y_{\min} = -1$). La cellule [256][256] de ce tableau est donc associée au coord. ($x_{\min} = 0, y_{\min} = 0$). La cellule [257][257] de ce tableau est donc associée au coord. ($x_{\min} = 0.00390625, y_{\min} = 0.00390625$). La cellule [511][511] de ce tableau est donc associée au coord. ($x_{\min} = 1, y_{\min} = 1$). Utiliser ensuite la fonction RECAL donné dans le programme pour recaler les niveaux de gris de cette image entre 0 et 255.

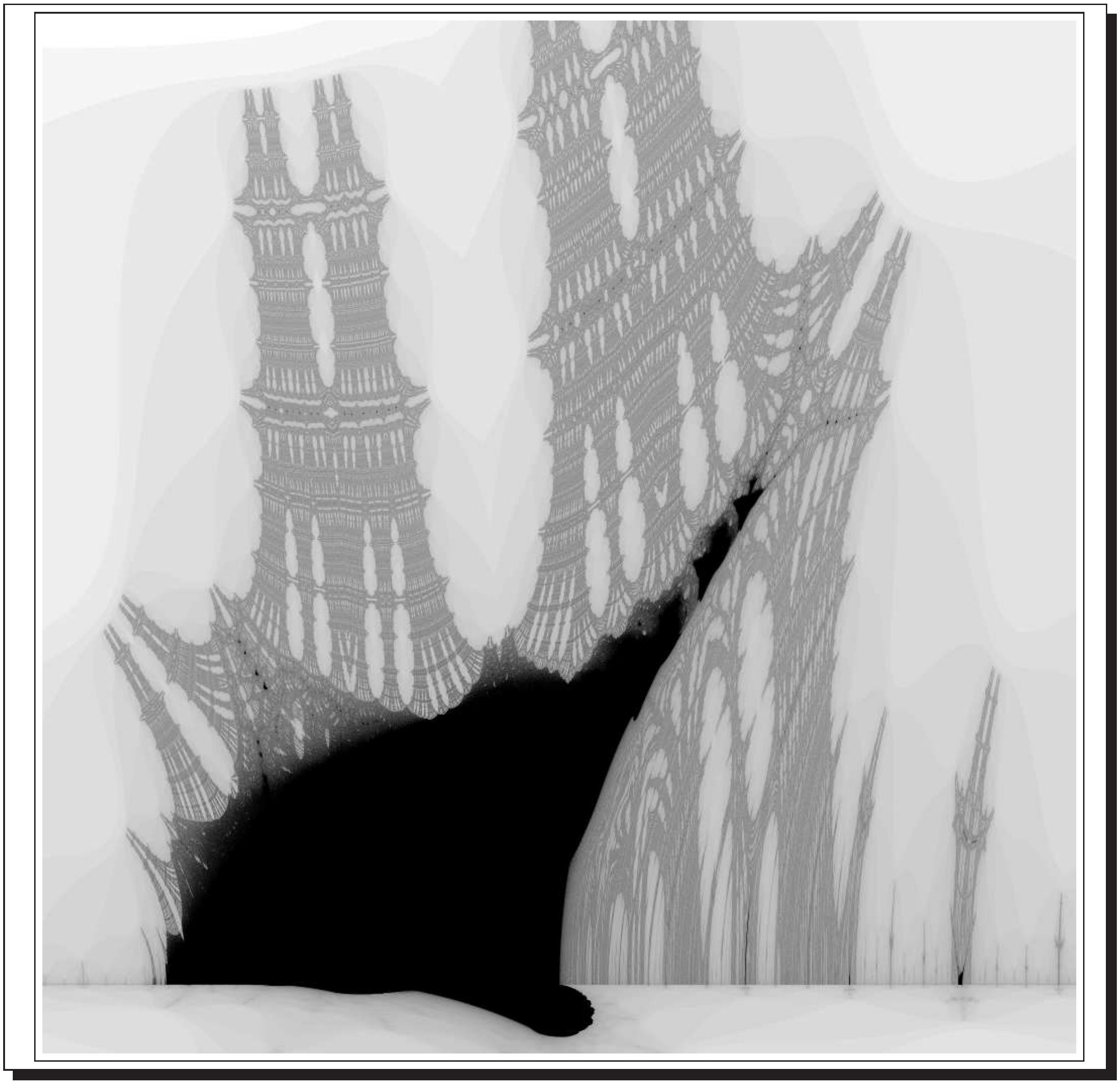

FIGURE 6 – Image fractale donnée par la suite itérative de point fixe de l'Équation (4) pour une image de taille $1024 \times 1024$ et $[x_{\min} = -1.8\ ,\ x_{\max} = -1.7]$, $[y_{\min} = -0.07\ ,\ y_{\max} = 0.005]$. .
... Ne seriez vous pas tenter de naviguer avec ce bateau loin du coronavirus ? ....

## Réferences


[1] M. Mignotte, C. Collet, P. Pérez, and P. Bouthemy. Three-class Markovian segmentation of high resolution sonar images. *Computer Vision and Image Understanding*, 76(3) :191–204, 1999.

[2] F. Destrempes and M. Mignotte. A statistical model for contours in images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5) :626–638, 2004.

[3] C. Essex and M. Davison and C. Schulsky. Numerical Monsters. *SIGSAM Journal*, 34(4) :16-32, December 2000.

[4] Wikipedia