DIRO
IFT 2425

DEVOIR N° 1

Méthode de Newton & Fractales

Max Mignotte

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

I. Méthode de Newton

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 $c > 0$. α 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.

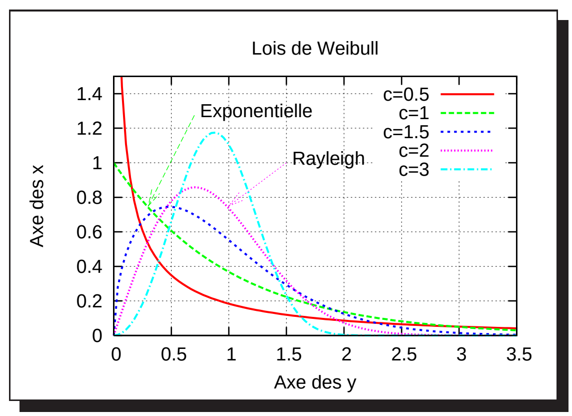

FIG. 1 – Lois de Weibull pour différentes valeurs du facteur de forme c (dans cet exemple, $\alpha = 1$)

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 :

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

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

$$f(c_{\mathrm{MV}}; y_i) = \frac{\sum_{i=1}^{N}\left(y_i^{c_{\mathrm{MV}}}\ln y_i\right)}{\sum_{i=1}^{N} y_i^{c_{\mathrm{MV}}}} - \frac{1}{c_{\mathrm{MV}}} - \frac{1}{N}\sum_{i=1}^{N}\ln y_i = 0 \qquad (1)$$

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].

TAB. 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

1– Implémenter[1] la relation itérative de Newton $c_{\mathrm{MV}}^{[n+1]} = g(c_{\mathrm{MV}}^{[n]}; y_i)$ qui converge vers la solution de l'équation $f(c_{\mathrm{MV}}; y_i) = 0$ lorsque l'estimée initiale (premier terme de la suite) est $c_{\mathrm{MV}}^{[0]} = 0.5$, avec comme critère d'arrêt, une tolérance sur $\hat{c}_{\mathrm{MV}}$ et une tolérance sur f de 10^{-6}. Utiliser pour la dérivée de cette relation itérative, une approximation du style : $f'(x) \approx [f(x+\epsilon) - f(x)]/\epsilon$ (avec par exemple $\epsilon \approx 10^{-5}$).

II. Fractales de Newton et Autres

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, et 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 :

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

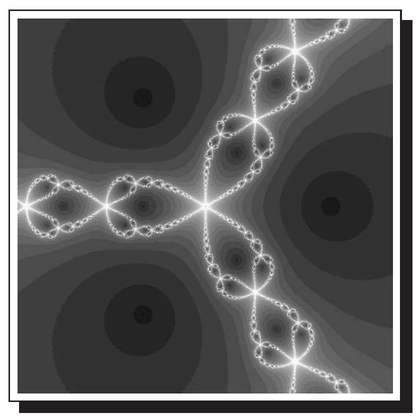

FIG. 2 – Fractale de Newton pour $f(z) = z^3 - 1$ où chaque pixel est d'autant plus blanc que la série converge lentement.

En fonction de la valeur de départ z_0, cette suite itérative convergera (si il y a convergence) plus ou moins vite. Ainsi, on peut retrouver le graphe de la Fig. 2 en exprimant tout d'abord la suite (1) 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 ($y_{[n+1]} = f_i(y_{[n]})$). Pour chaque cellule ou pixel de coordonnée i, j (ligne, colonne) d'un tableau ou image 2D de dimension 512×512 (longueur$\times$largeur), on itère ces deux suites (l'une après l'autre) en considérant comme valeur de départ $z_{[0]} = x_{[0]} + j\, y_{[0]}$ avec :

[1] En toute rigueur (i.e., théoriquement et non plus expérimentalement comme dans ce TP), on rappelle que l'on devrait montrer tout d'abord (cf. cours), que la fonction $g(x)$ choisie dans la relation itérative $c^{[n+1]} = g(c^{[n]})$ est telle que $|g'(x)| < 1, \ \forall x \in J$, un intervalle comprenant la racine et le premier élément de la suite $c^{[0]}$ ou bien que la racine et $c^{[0]}$ ne sont pas séparé par un *extrema* de la fonction f, i.e., $f'(x) \neq 0, \ \forall x \in J$.

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

de telle façon que $x_{[0]}$ et $y_{[0]}$ varient (pour chaque cellule de ce tableau 2D) de -1.5 à 1.5 et que pour la cellule centrale de ce tableau 2D, on considère donc la valeur de départ $x_0 = 0, y_0 = 0$, c'est-à-dire $z_0 = 0$. On remplie finalement ce tableau 2D en $[i][j]$ en un niveaux de gris (compris entre $[0 - 255]$) d'autant plus blanc (ou proche de 255) que la série converge lentement vers une solution et l'affichage du résultat de ce tableau 2D sous la forme d'une image nous donne finalement la Fig. 2.

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 et/ou à des vitesses de convergence très différentes [3].

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

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

dans laquelle j désigne le nombre imaginaire pur ($j^2 = -1$), $\text{Re}(z_{[n]})$ représente la partie réelle de $z_{[n]} = x_{[n]} + jy_{[n]}$, c'est-à-dire ; $\text{Re}(z_{[n]}) = x_{[n]}$, $\text{Im}(z_{[n]})$ représente la partie imaginaire de $z_{[n]} = x_{[n]} + jy_{[n]}$, c'est-à-dire ; $\text{Im}(z_{[n]}) = y_{[n]}$, $|.|$ 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 2. 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. 3.

2– Pour cette question, on vous demande de retrouver l'image de la Fig. 4 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).

Pour remplir chaque cellule de 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, 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))). 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[2].

[2]

Conseils Pratiques

Utiliser le programme que je vous donne sur ma page web comme programme de départ. Pour une image de taille 512×512 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.

[3]

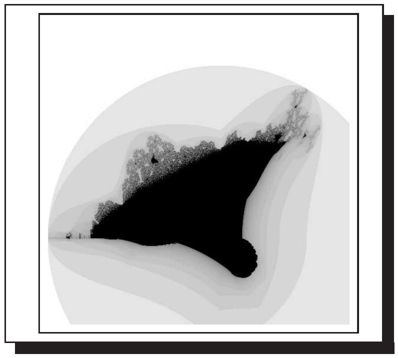

Fig. 3 – Image fractale donnée par la suite itérative de point fixe de l'Équation (4) pour une image de taille 128×128 et $[x_{\min} = -2\ ,\ x_{\max} = 1.5]$, $[y_{\min} = -2.5\ ,\ y_{\max} = 1]$.

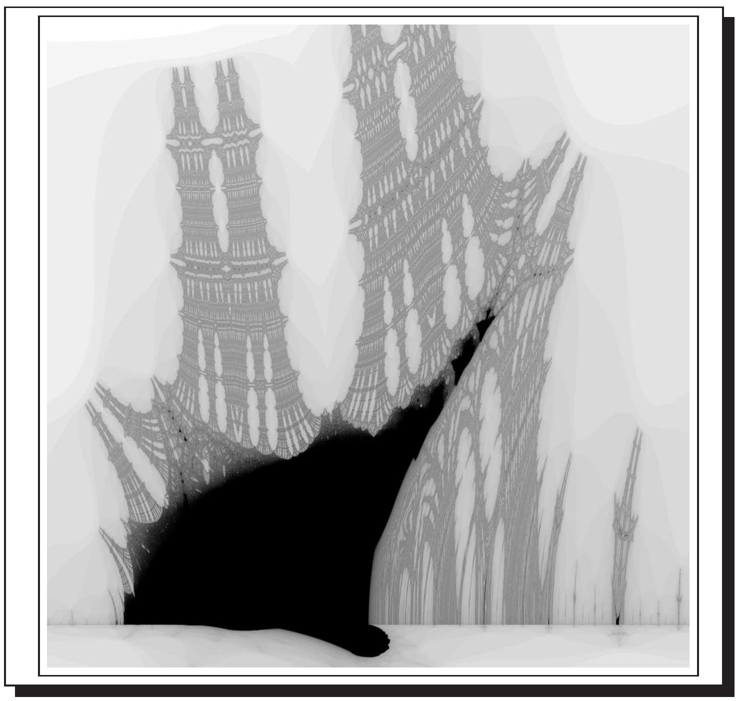

Fig. 4 – Image fractale donnée par la suite itérative de point fixe de l'Équation (4) pour une image de taille 1024×1024 et $[x_{\min} = -1.8\ ,\ x_{\max} = -1.7]$, $[y_{\min} = -0.07\ ,\ y_{\max} = 0.005]$.

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] Wikipédia.

Remise & Rapport

Vous devez rendre électroniquement le(s) programme(s) fait en C avant la date de remise spécifiée dans le fichier *barême* situé sur la page web du cours. Pour la remise électronique, utilisez le programme *remise* (*man remise* pour plus de détails) pour remettre votre code dans le répertoire TP<Numéro du Tp>. N'oubliez pas d'inscrire vos noms, courrier électronique en commentaire en haut du fichier .c remis. Les noms des programmes à remettre devront avoir le format suivant *Tp<Numero du Tp>-IFT2425-<Numéro de la question>.c*. Les programmes devront se compiler et s'executer sur Linux tel qu'indiqué dans le barême.