CHAPITRE 2
Problèmes non linéaires

1. Rappels de cours

1.1. Objectif

Soit $f : \mathbb{R}_n \to \mathbb{R}_n$. Trouver $x^* \in \mathbb{R}_n$ tel que $f(\mathrm{x}*) = 0$.

Toutes les méthodes que nous exposerons dans cette sous-section construisent une suite x_n qui vérifie :

$$\lim_{n\to\infty} (x_n) = x^* \text{ où } f(\mathrm{x}*) = 0.$$

1.2. Méthodes numériques

1.2.1. Le cas où $n = 1$

On suppose que $f : \mathbb{R} \to \mathbb{R}$ est continue et qu'il existe x^* tel que $f(\mathrm{x}*) = 0$, et un intervalle $[a, b]$ tel que $f(a). f(b) < 0$.

1.2.1.1. Dichotomie ou « bissection »

<u>Idée</u> : Construire deux suite (a_n) et (b_n) telles que :

$$f(a_n). f(b_n) < 0 \text{ et } \lim_{n\to\infty} (a_n - b_n) = 0.$$

<u>Algorithme de Dichotomie</u> : Soient (a_0) et (b_0) telles que $f(a_0). f(b_0) < 0$, $x_0 = \frac{1}{2}(a_0 + b_0)$

```
Tant que |f(x_n)| > ε
    Si f(a_n).f(x_n) > 0
        a_{n+1} = x_n
    Sinon
        b_{n+1} = x_n
    Fin
Fin tant que
```

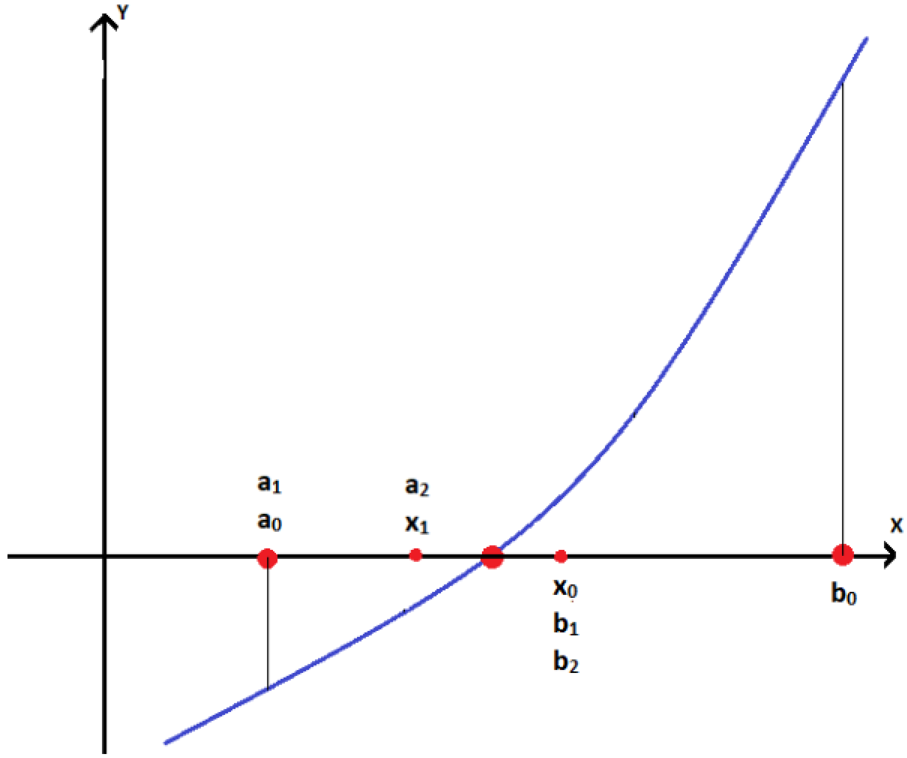


Illustration de la méthode de Dichotomie

<u>**Convergence de la méthode**</u> :

Comme on a $(b_{n+1} - a_{n+1}) = \frac{1}{2}(b_n - a_n)$ alors $(b_{n+1} - a_{n+1}) = \frac{1}{2^n}(b_0 - a_0)$.

Et comme $|x_n - x^*| \leq \frac{1}{2}(b_n - a_n)$ on a $\forall n \in \mathbb{N}$, $|x_n - x^*| \leq \frac{1}{2^n}(b_0 - a_0)$.

1.2.1.2. Méthode du point fixe

<u>Idée</u> : Trouver $g : \mathbb{R} \rightarrow \mathbb{R}$ telle que :

$$f(x) = 0 \Leftrightarrow x = g(x).$$

<u>Théorème</u> : Soit $g : \mathbb{R} \rightarrow \mathbb{R}$ dérivable et possédant un point fixe x^* vérifiant $|g(x^*)| < 1$. Alors il existe $[a, b]$ tel que si $x_0 \in [a, b]$ alors la suite (x_n) définie par $x_{n+1} = g(x_n)$ converge vers x^*.

1.2.1.3. Méthode de Newton

On suppose que f est deux fois continûment dérivable et on se place au voisinage de x_0, on peut écrire :

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2}(x - x_0)^2 f''(x_0 + \partial(x - x_0)) \ , \ \partial \in]0,1[\ .$$

Idée : Définir x_1 comme la solution de l'équation linéaire :

$$f(x_0) + (x - x_0)f'(x_0) = 0,$$

c'est-à-dire, si $f'(x_0) \neq 0$, $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$, et de construire une suite (x_n).

Algorithme de Newton :

x_0 donné

Tant que $|f(x_n)| > \varepsilon$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \ (\Leftrightarrow \ x_{n+1} = g(x_n))$$

Fin tant que

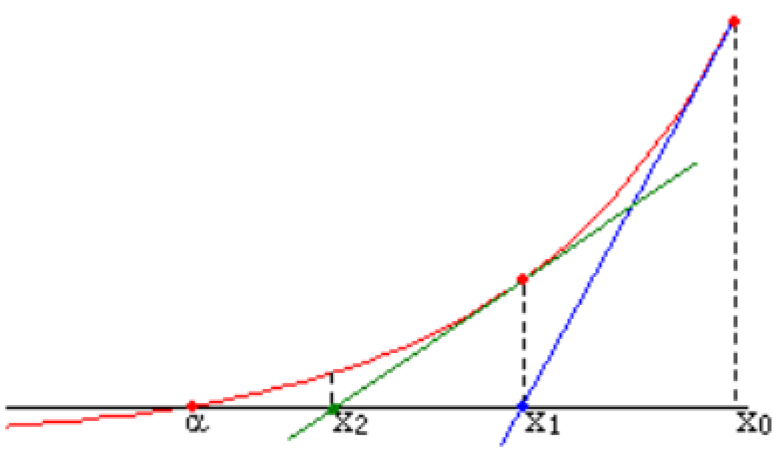


Illustration de la méthode de Newton

Remarque : Comment choisit-on x_0 ?

Mais avant de répondre, voyons que la méthode est une méthode de point fixe très particulière : on a ici

$$g(x) = x - \frac{f(x)}{f'(x)} \text{ et } g'(x) = 1 - \frac{\left(f'(x)\right)^2 - f(x)f'(x)}{\left(f'(x)\right)^2}.$$

D'où $g'(x^*) = 1 - \frac{\left(f'(x*)\right)^2}{\left(f'(x*)\right)^2} = 0$, donc il existe forcément $[a, b]$ tel que $\forall x \in [a, b], |g'(x)| < 1$.

Convergence de la méthode : la convergence est quadratique :

$$\exists n \geq 0 \, , |x_{n+1} - x*| \leq c|x_n - x*|^2.$$

Si $|x_0 - x*| < \frac{1}{c}$ alors le méthode de Newton converge, et quadratiquement.

1.2.2. Le cas où $n > 1$ (La méthode de Newton)

<u>Idée</u> : est la même que pour $n = 1$. On se donne $x_0 \in \mathbb{R}_n$ on a :

$$f(x_0 + h) = f(x_0) + h\,.J_f(x_0) + |h|\mathcal{E}(h)$$

et on en déduit l'approximation affine de f en $x_0, i.e.$ la fonction $T(x)$ définie par :

$$T(x) = f(x_0) + (x - x_0)\,.J_f(x_0).$$

Si la matrice $J_f(x_0)$ est inversible, on peut déduire x_1 tel que $T(x_1) = 0$, soit :

$$f(x_0) + (x_1 - x_0)\,.J_f(x_0) = 0 \text{ , soit } x_1 = x_0 - (\,J_f(x_0))^{-1} f(x_0)$$

<u>Algorithme de la méthode</u> :

$x_0 \in \mathbb{R}_n$ donné

Tant que $|f(x_n)| > \mathcal{E}$

$x_{n+1} = x_n - (\,J_f(x_n))^{-1} f(x_n)$

Fin

2. Résolution d'une équation à une inconnue

Dans cette partie, on cherche à résoudre l'équation :

$$x^2 = 2 \quad (1)$$

Pour y parvenir, on va tour d'abord choisir un intervalle $[a, b]$ dans lequel se situe x^* une seule solution de (1). En l'occurrence, on choisit $a = 1\ et\ b = 2$: on a donc $x^* = \sqrt{2}$. L'équation (1) peut s'écrire de façon équivalente $f(x) = 0$ ou $g(x) = 0$ avec :

$$f(x) = x^2 - 2\,,\, g(x) = \frac{x+2}{x+1}$$

On cherche à comparer plusieurs méthodes pour trouver x^* :

2.1. Méthode de dichotomie

On choisit a = 1, b = 2, on a :

$$f(1) = 1^2 - 2 = -1 < 0,\ f(2) = 2^2 - 2 = 2 > 0, f(1).f(2) < 0$$

Sur l'intervalle $[1,2]$, on construit 3 suites $(a_n), (b_n), (x_n)$ de la manière suivante :

```
function y=f(x)
        y = x^2 - 2;
endfunction

a = 1;
b = 2;
for k = 1 : 100
    x(k) = (a + b) / 2
    if f(a) * f(x(k)) < 0
        b = x(k);
    else
        a = x(k);
    end
    if abs(f(x(k))) < 1e-10
        break;
    end
end
```

Code source de la méthode Dichotomie

2.2. Méthode de point fixe

Dans cette méthode, on trouve une fonction $g(x)$ telle que :

$$f(x) = 0 \Leftrightarrow x = g(x).$$

Par exemple, on prend $g(x) = \frac{x+2}{x+1}$, on a donc :

$$x = g(x) \Leftrightarrow x^2 + x = x + 2 \Leftrightarrow x^2 - 2 = 0 \Leftrightarrow x = \sqrt{2}\ ou\ x = -\sqrt{2}$$

```
function y=f(x)
	y = x^2 - 2;
endfunction

function y=g(x)
	y = (x+2)/(x+1);
endfunction

x = 1.5;

for k = 1 : 100
	if abs(f(x(k))) < 1e-10
		break;
	end
	x(k+1) = g(x(k));
end
```

Code source de la méthode point fixe

2.3. Méthode de Newton

On prend $x_1 = 1.5$, on contruit une suite (x_n) avec : $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$

```
function y=f(x)
	y = x^2 - 2;
endfunction

function y=fPrime(x)
	y = 2*x;
endfunction

x = 1.5;

for k = 1 : 100
	if abs(f(x(k))) < 1e-10
		break
	end
	x(k+1) = x(k) -   f(x(k)) / fPrime(x(k));
end
```

Code source de la méthode de Newton

2.4. Méthode de la sécante

On prend $x_1 = 1.4$ et $x_2 = 1.5$, on contruit une suite (x_n) avec :

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

```
function y=f(x)
    y = x^2 - 2;
endfunction

x(1) = 1.4;
x(2) = 1.5;
for k = 2 : 100
    if abs(f(x(k))) < 1e-10
        break ;
    end
    tmp = ( f(x(k)) - f(x(k-1)) ) / (x(k) - x(k-1)) ;
    x(k+1) = x(k) - f(x(k)) / tmp;
end
```

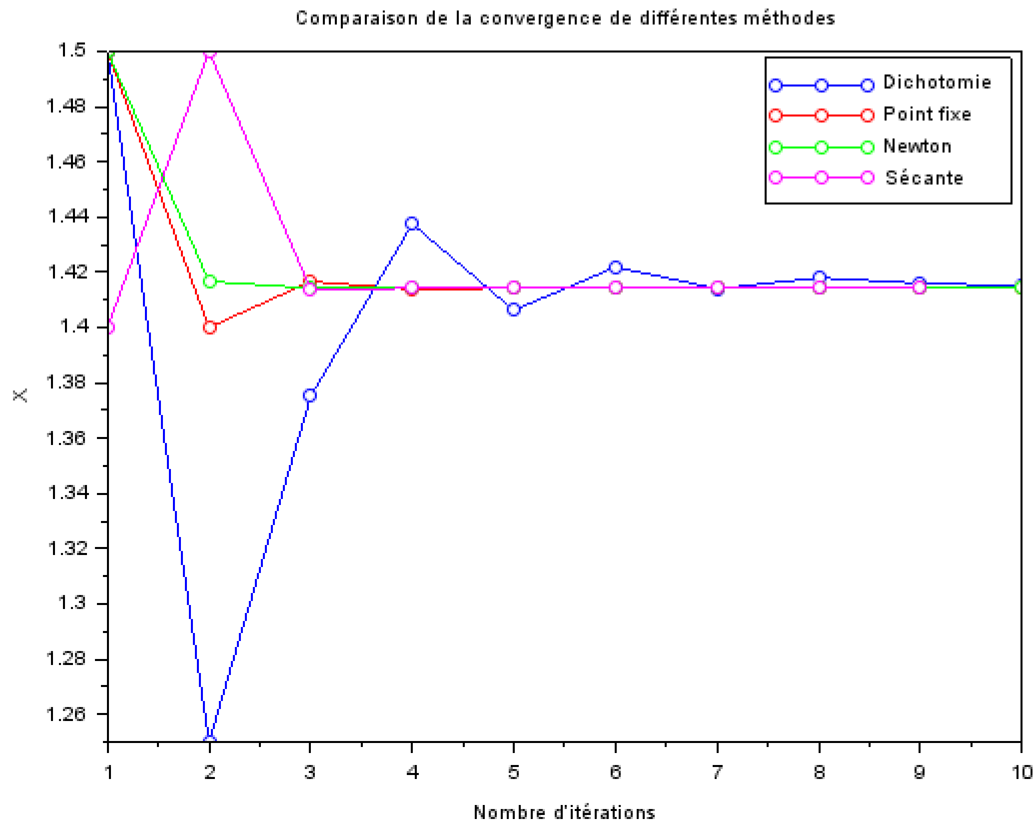
Code source de la méthode de la sécante

2.5. Comparaison de la convergence de différentes méthodes

L'ordre de convergence $p \geq 1$ d'une méthode est défini par :

$$\lim_{k \to \infty} \frac{|x_{k+1} - x*|}{|x_k - x*|} = \alpha > 0$$

En utilisant Scilab, nous trouvons expérimentalement que méthode de dichotomie et de point fixe sont d'ordre 1, méthode de Newton est d'ordre 2 et méthode de la sécante est d'ordre de $\frac{1+\sqrt{5}}{2}$. A l'aide de Scilab, j'ai réalisé un graphique consultable permettant de comparer la convergence des différentes méthodes vers la solution de l'équation. Cette comparaison a été effectuée pour un nombre d'itération $n = 10$



2.6. La macro fsolve

La macro fsolve de Scilab permet de résoudre des systèmes d'équations non-linéaire avec une méthode inspirée de la méthode de Newton. On résout $x^2 = 2$ en utilisant la macro fsolve de la manière suivante :

```
function y=f(x)
   y=x^2-2;
endfunction

function d=fprime(x)
   d=2*x;
endfunction

x0 = 1.5;
x = fsolve(x0,f,fprime);
disp(x);
```

3. Applications

3.1. GPS

Le GPS est un système de positionnement basé sur la connaissance de la distance du récepteur R à trois satellites (situés à des orbites de l'ordre de 28000km).

On suppose que les trois satellites au moment du calcul de distance ont les positions suivantes dans un repère cartésien d'origine le centre de la terre (unité km) :

$$S1 = (-11716.227778, -10118.754628, 21741.083973)$$

$$S2 = (-12082.643974, -20428.242179, 11741.374154)$$

$$S3 = (14373.286650, -10448.439349, 19596.404858)$$

Sachant que les trois distances respectives au récepteur ont été calculées et valent :

$$(d1, d2, d3) = (22163.847742, 21492.777482, 21492.469326)$$

On pose :

$$f(X) = \begin{pmatrix} ||X - S1||^2 - d1 \\ ||X - S2||^2 - d2 \\ ||X - S3||^2 - d3 \end{pmatrix}$$

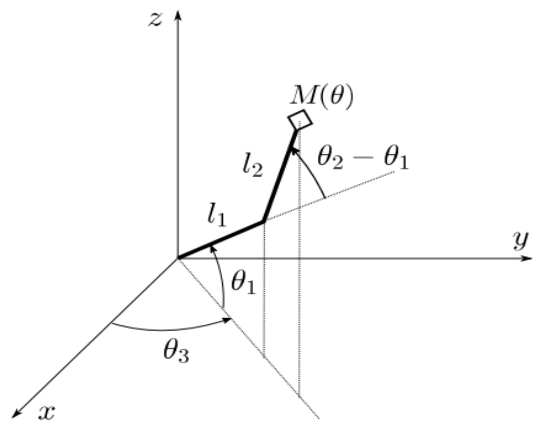
Pour X^* (la position du récepteur R dans le repère associé au centre de la Terre), on a bien : $f(X^*) = 0$.

```
function out=f(R)
   out = [norm(R-S1)^2; norm(R-S2)^2; norm(R-S3)^2] - d.^2;
endfunction

S1 = [-11716.227778;-10118.54628;21741.083973];
S2 = [-12082.643974;-20428.242179;11741.374154];
S3 = [14373.286650;-10448.439349;19596.404858];
d  = [22163.847742;21492.777482;21492.469326];

R0 = zeros(3,1);
[R,v,info] = fsolve(R0,f)
```

Code source de GPS

Le récepteur R se trouve expérimentalement à 6369.2864 kms du centre de la Terre.

3.2. Cinématique inverse



On considère sur la figure au-dessus un bras robot articulé dans l'espace d'origine O, avec deux segments dans un même plan contenant l'axe (Oz) et pouvant tourner autour de cet axe. Le premier segment de longueur $l_1$faisant un angle θ_1 avec $(0, x, y)$ et un deuxieme segment de longueur l_2 faisant un angle θ_2−θ_1 avec le premier segment. Le plan contenant le bras fait un angle θ3 avec (O, y, z). Les coordonnées de l'extrémité du bras sont données par :

$$M(\theta) = (cos\theta_3(l_1 cos\theta_1 + l_2 cos\theta_2), sin\theta_3(l_1 sin\theta_1 + l_2 sin\theta_2), l_1 sin\theta_1 + l_2 sin\theta_2).$$

On cherche à déterminant θ tel que $M(\theta) = A$ où A est un point du plan, on a donc les relations suivantes :

$$cos\theta_3(l_1 cos\theta_1 + l_2 cos\theta_2) - x_A = 0$$

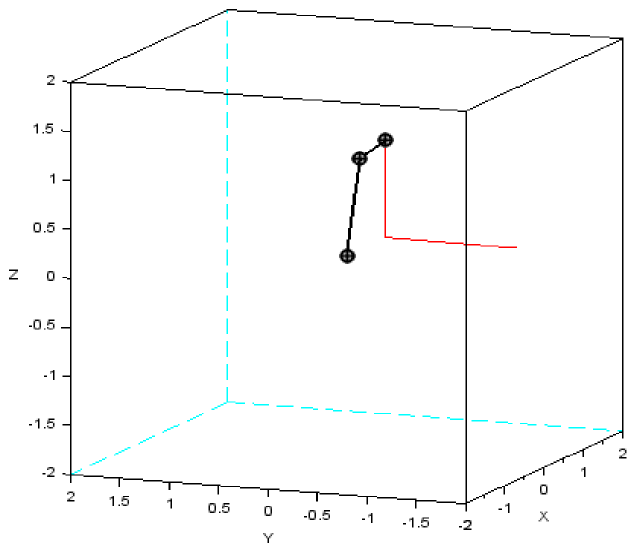
$$sin\theta_3(l_1 sin\theta_1 + l_2 sin\theta_2) - y_A = 0$$

$$l_1 sin\theta_1 + l_2 sin\theta_2 - z_A = 0$$

On peut ensuite résoudre la fonction Scilab fsolve :

```
function out=f(t)
    M = [ cos(t(3)) * (l(1) * cos(t(1)) + l(2) * cos(t(2)))
          sin(t(3)) * (l(1) * sin(t(1)) + l(2) * sin(t(2)))
          l(1) * sin(t(1)) + l(2) * sin(t(2)) ];
    out = M - A;
endfunction
```

On prendre $l_1 = 1, l_2 = \frac{2}{3}$. Lorsque le point $A(t)$ est défini par une courbe, par exemple lorsqu'il décrit les arêtes d'un cube $A(t) = (0, 1, t)$ avec $t \in [0, 1]$, on utilise la fonction Scilab au-dessus pour représenter les positions successives du bras robot :



Les positions successives du bras robot

4. La fractale de Newton

On cherche donc à résoudre dans $\mathbb{C}$ l'équation :

$$z^3 - 1 = 0.$$

Les solutions de cette équation sont bien évidemment :

$$z_1 = 1 \; ; \; z_2 = -\frac{1}{2} + \frac{\sqrt{3}}{2} i \; ; \; z_3 = -\frac{1}{2} - \frac{\sqrt{3}}{2} i \, .$$

En appliquant la méthode de Newton, la suite complexe récurrente (z_n) définie par z_0 et

$$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} = z_n - \frac{{z_n}^3 - 1}{3{z_n}^2} = \frac{2{z_n}^3 + 1}{3{z_n}^2} = \frac{2}{3} z_n + \frac{1}{3{z_n}^2}$$

permet d'approcher les solutions de l'équation. Lorsque l'on choisit un z_0 donné, la suite (z_n) peut converger vers l'une des trois solutions ou ne pas converger du tout. Dans ce dernier cas, z_0 appartient à la fractale de Newton. On considére une discrétisation à N^2 points du rectangle [−1, 1] × [−1, 1], c'est à dire deux matrices X et Y où chaque point i, j du rectangle a pour coordonnées (x_{ij}, y_{ij}) et on considère les nombres complexes correspondants dans la matrice Z où :

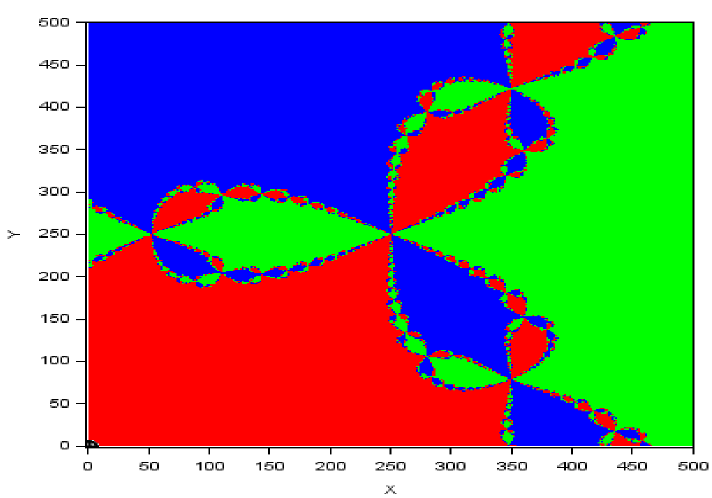
$$\forall i, j \in \{1; \ldots; N\},\; z_{ij} = x_{ij} + i * y_{ij}$$

Le but est de construire une matrice M que l'on représentera ensuite avec Matplot. Le terme m_{ij} est défini ainsi : le nombre complexe z_{ij} est utilisé comme condition initiale z_0 pour la méthode de Newton.

Si le méthode converge (après un nombre d'itérations fixé) à la précision machine vers l'une des racines on attribute le nombre 1,2 ou 3 à m_{ij}.

```
N = 500;
Eps = 1e-15;
plot([-1 1] , [-1 1] , ".");
[X,Y] = meshgrid(linspace(-1,1,N));
Z = X + %i * Y;
s1 = 1;
s2 = (-1 + %i * sqrt(3)) / 2;
s3 = (-1 - %i * sqrt(3)) / 2;
for  i = 1 : N
     Z = 2/3 * Z + (1/3) * Z.^(-2);
     C = ones(N,N) * color("white");
     C(abs(Z-s1) < eps) = color("green");
     C(abs(Z-s2) < eps) = color("red");
     C(abs(Z-s3) < eps) = color("blue");
     Matplot(C,rect = [-1 -1 1 1]);
end
```

Code source de la fractale de Newton



Résultat de la fractale de Newton