MT05, Projet n°2: Transformation du plan

Baptiste Toussaint

baptiste.toussaint@utt.fr

27 février 2024

Table des matières

1 Introduction et objectifs du projet

L'objectif de ce projet est d'évaluer les racines d'une fonction f, continue, par le biai de deux méthodes numériques : la méthode par dichotomie et la méthode de Newton.

2 Méthode par dichotomie

2.1 Explication

— Soit une fonction f, continue sur l'intervalle $[a, b]$
— $a < b$
— $f(a)f(b) < 0$

D'après le théorème des valeurs intermédiaires, f s'annule au moins une fois sur $]a, b[$. Ce résultat se démontre facilement :
Si $f(a)f(b) < 0$, soit $f(a) < 0$, soit $f(b) < 0$ mais pas les deux. Commme f est continue (donc sans discontinuité), on pourra tracer sa courbe "sans lever le crayon". La courbe passera forcément par 0 si l'on veut relier les points a et b.

La méthode par dichotomie revient à prendre le point milieu de l'intervalle $[a, b]$: c, d'observer la position de $f(c)$ et la comparer avec 0. Si $f(c) < 0$ alors la racine de f se trouve dans l'intervalle $[a, c]$, sinon dans $[c, b]$, et ainsi de suite.

De manière plus générale on va prendre $c = \frac{a+b}{2}$ et observer le signe de $f(a)f(c)$:
— Si $f(a)f(c) > 0$, alors $f(a)$ et $f(c)$ ont le même signe (différent de celui de $f(b)$, par définition), donc le changement de signe s'effectue dans l'intervalle $[c, b]$
— Si $f(a)f(c) < 0$, alors $f(a)$ et $f(c)$ n'ont pas le même signe, donc le changement de signe s'est effectué dans l'intervalle $[a, c]$
— La valeur de $f(c)$ est très proche de 0 ou égal à 0, alors on a trouvé la racine de f.

2.2 Exemple : $f(x) = x^3 - 10x + 2$

2.2.1 Existance des racines

Tableau de variation de la fonction f.

x	$-\infty$		$-\frac{\sqrt{30}}{3}$		$\frac{\sqrt{30}}{3}$		$+\infty$
$f'(x)$		$+$	0	$-$	0	$+$	
$f(x)$	$-\infty$	$\nearrow$	≈ 14.172	$\searrow$	≈ -10.172	$\nearrow$	$+\infty$

On peut définir trois intervalles sur $\mathbb{R}$: $]-\infty; -\frac{\sqrt{30}}{3}]$, $[-\frac{\sqrt{30}}{3}; \frac{\sqrt{30}}{3}]$, $[\frac{\sqrt{30}}{3}; +\infty[$. f change de signe sur chacun de ces intervalles, et f étant continue, on peut appliquer le théorème des valeurs intermédiaires, donc il existe une racine t_i de f sur chacun de ces intervalles tel que : $t_1 < t_2 < t_3$.

2.2.2 Détermination des racines par dichotomie

```
17 ### Algorithme de dichotomie Méthode boucle for
18 printf(" Dichotomie version boucle for !!!! \n")
19 racines = [];
20 bornes = [-3.5 -2 -1 1 2 3.5];
21 c_n = []; % Pour la convergence de la suite cn
22 for j = 1:2:5
23   %printf("\n première racine")
24   a=bornes(j);
25   b=bornes(j+1);
26   for i=1:200
27     %printf("\nIteration:%d\n", i)
28     %a
29     c=(a+b)/2;
30     c_n = [c_n c];
31     %b
32     z=f(a)*f(c);
33     if (z==0)
34       r1=c;
35       %printf("Une racine en r1")
36     elseif (z<0)
37       b=c;
38       %printf("z est négatif donc b=c. b= %f\n", b)
39     else
40       a=c;
41       %printf("z est positif donc a=c. a= %f\n", a)
42     endif
43   endfor
44   racines = [racines c];
45 endfor
46
47 printf("Les racines sont: \n")
48 racines
```

Cette implémentation va rechercher les trois racines de f. La valeur de i (ligne 26) de la boucle for définira la précision de la recherche. Ici l'algorithme réalisera 200 itérations avant de rendre son résultat.

Une autre solution est de passer par une boucle while, qui se basera sur une valeur d'erreur acceptable. Ici l'erreur sera de 10^{-8} :

```
### Algorithme de dichotomie Méthode boucle while
printf(" Dichotomie version boucle while !!!! \n")
racines = [];
bornes = [-3.5 -2 -1 1 2 3.5];
c_n = []; % Pour la convergence de la suite cn
for j = 1:2:5
  a=bornes(j);
  b=bornes(j+1);
  c=(a+b)/2;
  c_n = [c_n c];
  while (abs(f(c))>10^(-8))
    z=f(a)*f(c);
    if (z==0)
      r1=c;
      %printf("Une racine en r1")
    elseif (z<0)
      b=c;
      %printf("z est négatif donc b=c. b= %f\n", b)
    else
      a=c;
      %printf("z est positif donc a=c. a= %f\n", a)
    endif
    c=(a+b)/2;
    c_n = [c_n c];
  endwhile
  racines = [racines c];
endfor

printf("Les racines sont: \n")
racines
```

Dans les deux cas les algorithmes proposent des résultats égaux :

```
 Dichotomie version boucle for !!!!
Les racines sont:
racines =

  -3.2579   0.2008   3.0571

 Dichotomie version boucle while !!!!
Les racines sont:
racines =

  -3.2579   0.2008   3.0571
```

Les racines de f sont donc (approximativement) $R_f = \{-3.2579; 0.2008; 3.0571\}$

2.3 Explication de la convergence

2.3.1 Question 1

Soit L_0 la longeur de l'intervalle $[a_0;\ b_0]$. On a alors : $L_0 = \frac{b-a}{2}$. De même on a : $L_1 = \frac{L_0}{2}$.
Plus généralement on peut définir la récurence suivante :

$$L_{k+1} = \frac{L_k}{2}$$

Cette suite est une suite géométrique de raison $\frac{1}{2}$, donc on a :

$$\begin{cases} L_0 = \frac{b+a}{2} \\ \\ L_k = (\frac{b+a}{2}) \cdot (\frac{1}{2})^k \end{cases}$$

2.3.2 Question 2

On a établit que le théorème des valeurs intermédiaires était applicable sur l'intervalle $]a; b[$ pour f, f étant coninue sur cet intervalle (plus précisément, f est un polynôme, donc de classe C^∞ sur $\mathbb{R}$). Comme $f(a)f(b) < 0$ et f monotone sur $]a; b[$, il n'existe une unique racine x^* sur $]a; b[$ (la monotonie garantit la bijectivité de la fonction et donc $f(x^*) = 0$ n'admet qu'une et une seule solution).

A chaque itération de l'algorithme par dichotomie, l'intervalle de recherche est modifié en $]a'; b'[$ comme suit :
- $a' = a$ et $b' = \frac{b+a}{2}$ ou
- $a' = \frac{b+a}{2}$ et $b' = b$
- $f(\frac{b+a}{2}) = 0$, dans ce cas la racine est trouvée

Par définition à chaque itération, soit on trouve exactement la racine, soit elle se trouve dans l'intervalle $]a'; b'[$ nouvellement crée.

Dans les deux cas $\frac{b+a}{2}$ appartient à $]a; b[$, donc $]a; \frac{b+a}{2}[$ comme $]\frac{b+a}{2}; b[$ sont des sous-intervalles de $]a'; b'[$ et en possèdent les mêmes propriétés.

On peut conclure qu'il n'existe une unique racine x^* sur $]a'; b'[$. On pourra répéter ce principe jusqu'à k, les propriétés du théorème des valeurs intermédiaires resteront valide pour $]a_k; b_k[$. De ce fait on peut conclure qu'il existe une unique racine x^* sur $]a_k; b_k[$, pour toutes valeurs de k.

2.3.3 Question 3

On a c_n la suite qui correspond aux valeurs succesives des milieux de $[a; b]$. c_n prend donc des valeurs réelles dans $\mathbb{R}$, aussi, on sait que $x^* \in [a; b]$ et $c_n \in [a; b]$. Par définition on a donc :

$$|c_k - x^*| < |b - a|$$

Une borne supérieure de la suite $|c_k - x^*|$ est donc $|b - a|$.
On peut affiner cette borne. En effet on a :

$$\begin{cases} c_0 = \frac{b_0 - a_0}{2} \\ \\ c_k = \frac{c_{k-1} + \frac{a_{k-1}}{b_{k-1}}}{2} \end{cases}$$

A chaque itération de l'algorithme par dichotomie, on a $c_k \in [a_k; b_k]$ et $x^* \in [a_k; b_k]$. Donc par définition on a :

$$|c_k - x^*| < L_k$$

2.4 Question 4

Donc $|c_k - x^k|$ est une suite majorée par L_k et $|c_k - x^k| > 0$. Or L_k converge vers 0. Donc $|c_k - x^k|$ converge vers 0 (théorème des gendarmes) :

$$0 \leq |c_k - x^k| < L_k$$

Donc :

$$\begin{gathered} |c_k - x^*| < L_k \\ L_k < c_k - x^* < L_k \\ x^* - L_k < c_k < L_k + x^* \\ \lim_{k \to +\infty}(L_k) = 0 \\ \lim_{k \to +\infty}(x^* - L_k < c_k < L_k + x^*) = x^* < c_k < x^* \end{gathered}$$

Donc c_k converge vers x^*.

2.5 Question 5

On souhaite déterminer le nombre n minimal tel que : $|c_n - x^*| < \epsilon$. Si on prend $\epsilon = L_n$ Soit :

$$\begin{gathered} L_n = \epsilon \\ (\tfrac{b+a}{2}) \cdot (\tfrac{1}{2})^n = \epsilon \\ (\tfrac{1}{2})^n = \tfrac{2\epsilon}{b+a} \\ 2^n = \tfrac{b+a}{2\epsilon} \\ log_2(2^n) = log_2(\tfrac{b+a}{2\epsilon}) \end{gathered}$$

Donc :

$$n = E(log_2(\tfrac{b+a}{2\epsilon})) + 1$$

Avec E la partie entière.

2.6 Question 6

Si on prend $\epsilon = 10^{-10}$ on pour la racine entre 2 et 3.5 a alors :

$$\begin{gathered} n = E(34.5412) + 1 \\ n = 35 \end{gathered}$$

3 Méthode par l'algorithme de Newton

On va ici déterminer les facines de f par la méthode de Newton. Cette méthode est très ancienne est était déjà utlisé par les grecs dans l'antiquité. On considère généralement que la méthode de Newton possède des meilleurs propriétées de convergences face à la dichotomie.

3.1 Racherche des racines

Dans cette partie nous allons travailler à la recherche de racine du polynôme $f(x) = x^3 - 10x + 2$, dans $\mathbb{R}$ et $\mathbb{C}$.

3.1.1 Question 1

a)
Pour la fonction : $f(x) = x^3 - 10x + 2$, nous pouvons appliquer l'algorithme de Newton suivant :

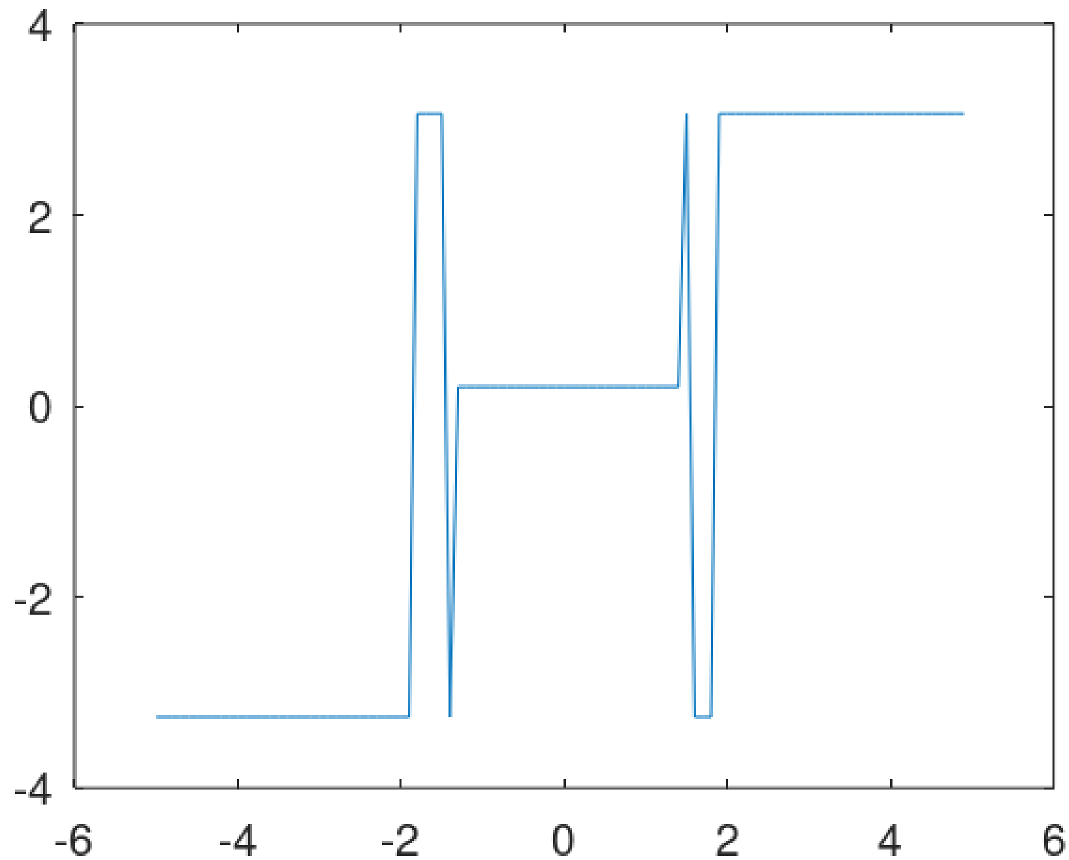
$$\begin{cases} x_0 \\ \\ x_{n+1} = x_n - \frac{x_n^3 - 10x_n + 2}{3x_n^2 - 10} \end{cases}$$

b)
Pour rechercher les racines nous allons effectuer une recherche à partir d'un maillage de l'intervale $[-5;5]$. L'intervale est découpé en 100 valeurs par pas de 0.1, et chacune de ces 100 valeurs sera un x_0 dont on étudiera la convergence. On obtient alors le code suivant :

```
### Maillage
f=@(x)(x^3-10*x+2);
fprime=@(x)(3*x^2-10);
subdiv = -5:0.1:5; # La subdivision de l'intervale
# Pour chaque valeur de subdiv on va observer la
# convergence
racines = [];
N = 50 # précision de recherche
for j = 1:100
  x = subdiv(j); # x_0
  for k=1:N
    x = x-(f(x)/fprime(x));
  endfor
  racines = [racines, x];
endfor

printf("Les racines pour chaque point du maillage sont: ")
racines
plot(racines)
```

On observe alors qu'il ressort 3 racines, identiques a celles trouvée par méthodes de dichotomie, et leurs répartition est comme suit :



On observe que pour un x_0 proche de -5, la méthode converge vers -3.2579, pour x_0 proche de 0, la méthode converge vers 0.2008, et pour x_0 proche de 5, la méthode converge vers 3.0571. Il existe cependant des zones dégénéré, vers -1 et 1, qui correspondent aux extremum locaux de la fonction, là où sa dérivé est presque paralèlle aux abscisses. Dans ce cas la méthode de Newton va créer un x_{n+1} très loin de la racine recherchée et finira par converger sur l'une des deux racine latérales : -3.2579, et 3.0571.

c)

Par définition la méthode de Newton n'est pas définie pour $f'(x) = 0$ (dénominateur nul). Les racines exactes de la dérivée sont : $-\sqrt{\frac{10}{3}}$, et $\sqrt{\frac{10}{3}}$.

Donc si x_n est égale à l'une de ces racines, alors la méthode ne sera pas fonctionnelle.

Cependant on peut ajouter que même sans commencer avec $x_0 = \pm\sqrt{\frac{10}{3}}$, il suffit prendre un x_0 dont la suite x_n passera par $\pm\sqrt{\frac{10}{3}}$ pour que cette méthode ne fonctionne pas.

3.1.2 Question 2

On a la fonction $g(x) = \sqrt{|x|}$. Pour cette fonction la méthode ne pourra pas converger car la fonction n'admet pas de dérivée continue sur $\mathbb{R}$.

Démonstration :

$$\text{On pose } g(x) = \sqrt{h(x)} \text{ avec } h(x) = |x|.$$
$$\text{Donc } g'(x) = \frac{1}{2\sqrt{h(x)}} \cdot h'(x),$$
$$\text{Donc : } g'(x) = \frac{1}{2\sqrt{|x|}} \cdot \pm 1.$$

$$g'(x) = \begin{cases} \frac{1}{2\sqrt{x}}, \text{ si } x > 0 \\ \\ -\frac{1}{2\sqrt{-x}}, \text{ si } x < 0 \end{cases}$$

Or on a :

$$\begin{cases} lim_{0\to 0^+} \frac{1}{2\sqrt{x}} = +\infty \\ \\ lim_{0\to 0^-} - \frac{1}{2\sqrt{-x}} = -\infty \end{cases}$$

Donc g n'est pas de classe C^2 sur $\mathbb{R}$.

3.1.3 Question 3

On utilise aussi la méthode de Newton pour les fonctions holomorphes.

Soit le polynôme sur $\mathbb{C}$: $f(z) = z^3 - 10z + 2$. On peut déterminer les racines de ce polynôme par la méthode de Newton.

On va créer un maillage de $\mathbb{C}$ en subdivisant l'axe des réel et des imaginaire en 100 valeurs entre -5 et 5.

On obtiendra alors 10 000 valeurs de x_0 à tester pour la convergence.

On obtiendra alors un vecteur de 10 000 valeurs complexes représentant les racines de la fonction.

On retrouve alors les racines identiques à la fonction réelle correspondante :

- -3.2579 + 0i
- 0.2008 + 0i
- 3.0571 + 0i

3.1.4 Question 4

On a $P(z) = z^7 - 2z^3 + 5$. Les racines de P sont les z_* tel que : $P(z_*) = z_*^7 - 2z_*^3 + 5 = 0$

On a donc :

$$z_*^7 - 2z_*^3 + 5 = 0$$
$$z_*^7 = 2z_*^3 - 5$$
$$|z_*^7| = |2z_*^3 - 5|$$
$$|z_*^7| \leq 2|z_*|^3 + 5$$

On pose $X = |z_*|$ ce qui donne :

$$X^7 \leq 2X^3 - 5$$

Si on prend $X = |z_*| = 2$ on a alors une incohérence :

$$2^7 \leq 2^4 - 5 \; 128 \leq 11$$

Donc on peut affirmer que $|z_*|$, les racines de P sont tel que : $|z_*| \in [0; 2[$.

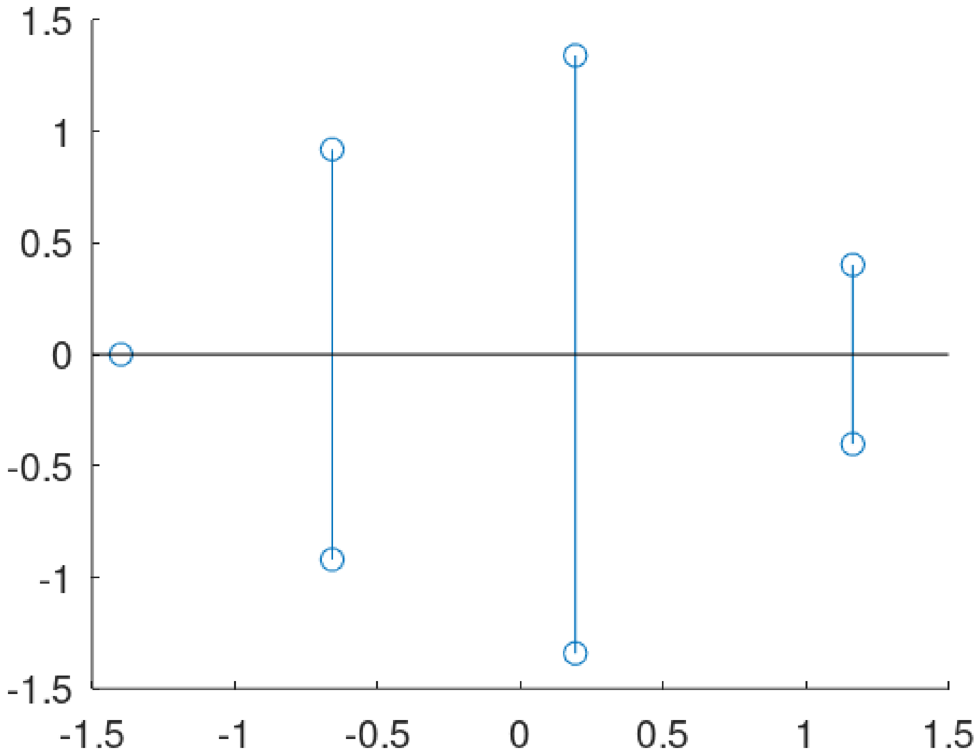
3.1.5 Question 5

Pour retrouver numériquement ces racines ont utilise le code suivant :

```
P=@(z)(z^7-2*z^3+5)
Pprime=@(z)(7*z^6-6*z^2)

N=100; # précision
racines = [];
for X = -2:0.1:1.9
  for Y = -2:0.1:1.9
    z = X+1i*Y;
    for i=1:N
      z = z - (P(z)/Pprime(z));
    endfor
    z = round(real(z)*1000)/1000+1i*(round(imag(z)*1000)/1000);
    if ! (any(racines == z))
      racines = [racines, z];
    endif
  endfor
endfor
racines;
racines_traitee = racines(1:7);
figure
  hold on
      stem(real(racines), imag(racines))
```

Cet algorithme nous donne 9 racines, dont deux qui semblent être des racines dégénérées due à des valeurs de z_0 pour lesquelles l'algorithme ne fonctionne pas.



racines

	1	2	3	4	5	6	7	8	9
1	-0.659 - 0.92i	1.165 - 0.402i	0.193 - 1.341i	-0.659 + 0.92i	-1.399 + 0i	0.193 + 1.341i	1.165 + 0.402i	NaN + 0i	81.912 + 0i
2									
3									
4									
5									
6									
7									
8									

racines_traitee

	1	2	3	4	5	6	7	8	9	10
1	-0.659 - 0.92i	1.165 - 0.402i	0.193 - 1.341i	-0.659 + 0.92i	-1.399 + 0i	0.193 + 1.341i	1.165 + 0.402i			
2										

Les racines sont donc environ égales à :
- 1 : $-0.659 - 0.92i$
- 2 : $1.165 - 0.402i$
- 3 : $0.193 - 1.341i$
- 4 : $-0.659 + 0.92i$
- 5 : $-1.399 + 0i$
- 6 : $0.193 + 1.341i$
- 7 : $1.165 + 0.402i$

3.2 Convergence de l'algorithme de Newton

Pour évaluer les performances de convergences de l'algorithme de Newton il faut remplir plusieurs prérequis :
— f holomorphe, ou à valeurs réelles deux fois continûment dérivable
— x^* est racine de f
— $f(x^*) = 0$
— $f'(x^*) \neq 0$
Pour rappel, analytiquement, les fonctions holomorphes correspondent à des fonctions ne faisant pas intervenir $\bar{z}$. Géométriquement, les fonctions holomorphes conservent les angles.

On souhaite prouver qu'avec l'algorithme de Newton, il existe un voisinage V de x^* tel que $x_0 \in V$ et $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ converge vers x_*.

3.2.1 Question 1

On a $x_{n+1} = g(x_n)$, avec $g(z) = (id - \frac{f}{f'})(z)$, donc $g(z) = z - \frac{f(z)}{f'(z)}$.

3.2.2 Question 2

$g(x^*) = x^* - \frac{f(x^*)}{f'(x^*)}$. Donc on a $g(x^*) = x^*$, car $f(x^*) = 0$. Les racines de f sont donc les points fixes de la transformation g.

3.2.3 Question 3

$g'(z) = 1 - \frac{(f'(z))^2 - f''(z)f(z)}{(f'(z))^2}$, donc $g'(x^*) = 1 - \frac{(f'(x^*))^2 - f''(x^*)f(x^*)}{(f'(x^*))^2} = 1 - \frac{(f'(x^*))^2}{(f'(x^*))^2} = 1 - 1 = 0.$

3.2.4 Question 4

On a établit que $g(x_n) = x_{n+1}$ est convergente vers x^*.

On peut se retourner vers la définiton de la convergence :

$$(\exists \epsilon > 0)(\exists N \in \Omega)\ (\forall n \in \Omega)(\exists l \in \Omega)$$
$$n \geq N \Rightarrow |g(z) - l| < \epsilon$$

Or ici on a :

$$lim_{z \to x^*}(g'(z)) = lim_{z \to x^*}(1 - \frac{(f'(z))^2 - f''(z)f(z)}{(f'(z))^2})$$
$$= 0.$$

Donc on peut dire que $g'(z)$ converge vers x^*, donc :

$$(\exists \epsilon > 0)(\exists Z \in \Omega)\ (\forall z \in \Omega)$$
$$z \geq Z \Rightarrow |g'(z) - 0| < \epsilon$$
$$z \geq Z \Rightarrow |g'(z)| < \epsilon < 1$$

3.2.5 Question 5

On a :

$$\begin{gathered}g(x) = x - \frac{f(x)}{f'(x)} \\ g(y) = y - \frac{f(y)}{f'(y)} \\ g(x) - g(y) = x - \frac{f(x)}{f'(x)} - y + \frac{f(y)}{f'(y)} = x - y + \frac{f(y)}{f'(y)} - \frac{f(x)}{f'(x)} \\ |g(x) - g(y)| = |x - y + \frac{f(y)}{f'(y)} - \frac{f(x)}{f'(x)}| \\ |g(x) - g(y)| \leq |x - y| + |\frac{f(y)}{f'(y)} - \frac{f(x)}{f'(x)}| \\ |g(x) - g(y)| \leq \epsilon|x - y|\end{gathered}$$

Avec : $\epsilon = 1 + \frac{|\frac{f(y)}{f'(y)} - \frac{f(x)}{f'(x)}|}{|x-y|}$

3.2.6 Question 6

$$\begin{gathered}|g(z) - x^*| = |g(z) - g(x^*)| \\ |g(z) - x^*| \leq \epsilon|z - x^*| \\ |g(z) - x^*| \leq |z - x^*| + |\frac{f(x^*)}{f'(x^*)} - \frac{f(z)}{f'(z)}| \\ |g(z) - x^*| \leq |z - x^*| + | - \frac{f(z)}{f'(z)}|\end{gathered}$$

On peut poser $\alpha = |z - x^*| + |\frac{f(z)}{f'(z)}|$, et on a alors :

$$|g(z) - x^*| \leq \alpha$$

3.2.7 Question 7

On prend $x_0 \in V$, or on a établit que :
— $x_{n+1} = g(x_n)$
— L'image de z par g appartient à V
Donc Par définiton, si $x_n \in V$ alors $x_{n+1} \in V$.

La propriétée est initialisé pour x_0, et est héréditaire, donc par récurence on a $\forall n \in \mathbb{N}, x_n \in V$.

3.2.8 Question 8

On a :

$$\forall x \in V, \forall y \in V,\ |g(x) - g(y)| < \epsilon|x - y|$$

Donc on a :

$$\begin{gathered}|g(x_{n+1}) - g(x*)| < \epsilon|x_{n+1} - x^*| \\ \epsilon|x_{n+1} - x^*| = \epsilon|g(x_n) - g(x^*)| \\ \epsilon|g(x_n) - g(x^*)| < \epsilon^2|x_n - x^*| \\ |g(x_{n+1}) - g(x^*)| < \epsilon^2|x_n - x^*|\end{gathered}$$

Donc on a :

$$
\begin{gathered}
|g(x_n) - g(x*)| < \epsilon^1 |x_n - x^*| \\
\epsilon |x_n - x^*| = \epsilon^1 |g(x_{n-1}) - g(x^*)| \\
\epsilon |g(x_{n-1}) - g(x^*)| < \epsilon^2 |x_{n-1} - x^*| \\
|g(x_n) - g(x^*)| < \epsilon^2 |x_{n-1} - x^*| \\
\dots \\
|g(x_n) - g(x^*)| < \epsilon^n |x_0 - x^*|
\end{gathered}
$$

3.2.9 Question 9

L'algorithme de Newton va créer une suite de x_n à partir d'une valeur x_0. Les valeurs de x_{n+1} sont donné par la relation $g(x_n) = x_{n+1}$.

En faisant tourner l'algorithme pour n valeurs. On obtiendra alors une valeur de x_n qui devra être le plus proche de x^* la racine de la fonction f.

On a montré que :

$$
\begin{gathered}
|g(x_n) - g(x^*)| < \epsilon^n |x_0 - x^*| \\
|g(x_n) - x^*| < \epsilon^n |x_0 - x^*|
\end{gathered}
$$

Donc pour un x_0 donné, après n étapes de l'algorithme, l'écart relatif entre x_0 et la valeur atteinte $g(x_n) = x_{n+1}$ est bornée par l'écart entre x_0 et x^*, multiplié par un facteur ϵ à la puissance n. Et ce, pour tout n appartenant à $\mathbb{N}$.

On a donc montré que la suite des x_n converge vers la valeur x^*, racine de f. On peut conclure que l'algorithme de Newton converge. On peut préciser que cet algorithme converge avec une vitesse remarquable en ϵ^n, ce qui est bien meilleur que la méthode par dichotomies.