

TP3

Fabien Navarro

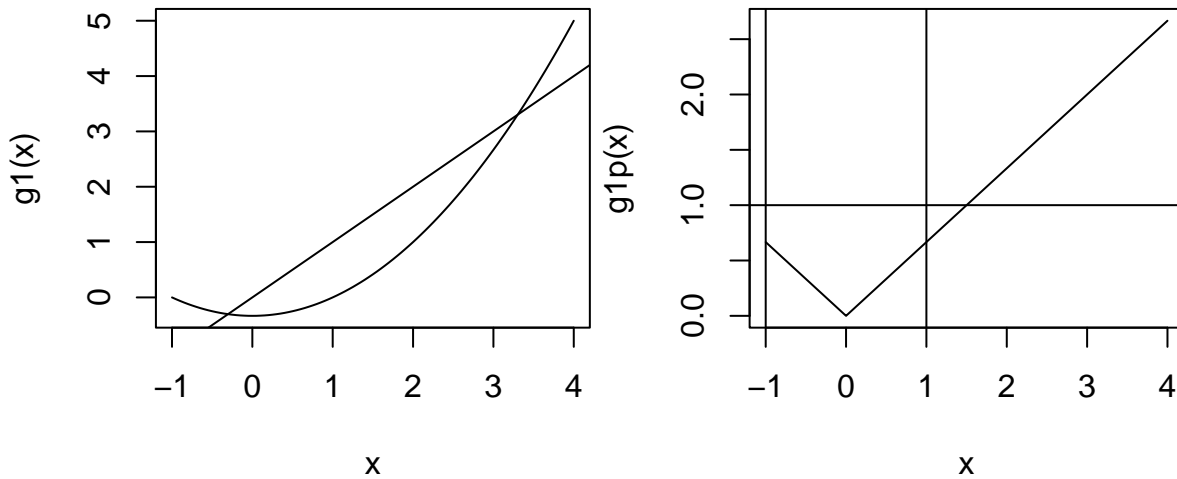
2024-03-07

Exercice 1. Calculer les points fixes x^* des fonctions suivantes :

1. $g(x) = (x^2 - 1)/3, x \in [-1, 1]$.
2. $g(x) = (x^2 - 1)/3, x \in [3, 4]$.

Dans chaque cas étudier $|g'(x^*)|$ et décider si la suite des itérés successifs converge ou non.

```
g1 <- function(x){(x^2-1)/3}
g1p <- function(x){abs(2*x/3)}
curve(g1, from = -1, to = 4)
abline(0,1)
curve(g1p, from = -1, to = 4)
abline(v=1)
abline(v=-1)
abline(h=1)
```



On calcule les points fixes de $g : g(x) = x$

$$g(x) = x \Leftrightarrow (x^2 - 1)/3 - x = 0 \Leftrightarrow x^* = (3 - \sqrt{13})/2, \quad x^* = (3 + \sqrt{13})/2.$$

Donc, on $\{(3 - \sqrt{13})/2, (3 + \sqrt{13})/2\}$ l'ensemble des points fixes de g .

Déterminons le signe de $g'(x)$ pour $x = (3 - \sqrt{13})/2$ et $(3 + \sqrt{13})/2$:

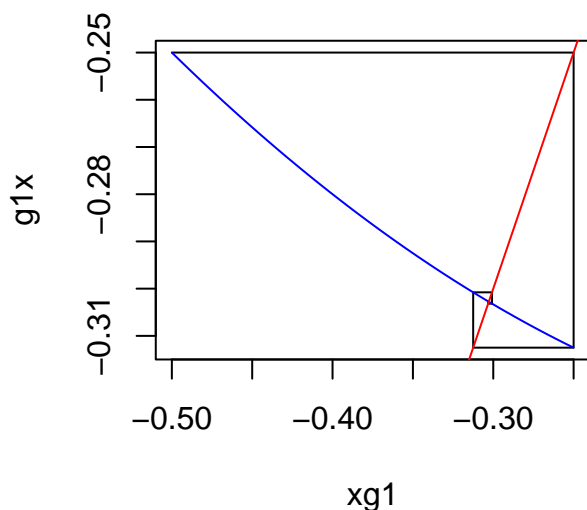
$$g'(x) = \frac{2x}{3}.$$

On a donc $g'((3 + \sqrt{13})/2) = 1 + \sqrt{13}/3 > 1$, point répulsif et $g'((3 - \sqrt{13})/2) = 1 - \sqrt{13}/3 < 1$ point attractif.

Les valeurs maximales et minimales de $g(x)$ pour $x \in [-1, 1]$ et $x \in [3, 4]$ peuvent être atteintes soit aux bornes de l'intervalle soit lorsque la dérivée est égale à 0.

```
x0 = -0.5
resg1 <- pfixe2(x0=x0,
               eps=1e-6,
               fun=g1,
               Nmax=100)

g1x <- rep(resg1$x, each=2)
xg1 <- c(x0, head(g1x,-1))
plot(xg1,g1x,type="s")
curve(g1,add=T, col="blue")
abline(0,1,col="red")
```



Exercice 2.

1. (a) Pour avoir $|x_n - x^*| \leq \varepsilon$, il suffit d'avoir $k^n |x_1 - x_0| / (1 - k) \leq \varepsilon$.
i.e,

$$\begin{aligned}
 k^n &\leq \frac{\varepsilon(1-k)}{|x_1 - x_0|} \\
 \Leftrightarrow n \log(k) &\leq \log\left(\frac{\varepsilon(1-k)}{|x_1 - x_0|}\right) \quad (k < 1) \\
 \Leftrightarrow n &\geq \log_k\left(\frac{\varepsilon(1-k)}{|x_1 - x_0|}\right)
 \end{aligned}$$

Il suffit de prendre $N_0 = E\left(\log_k\left(\frac{\varepsilon(1-k)}{|x_1 - x_0|}\right)\right) + 1$.

- (b) Taylor à l'ordre 1 :

$$e_{n+1} = x_{n+1} - x^* = g(x_n) - g(x^*) = (x_n - x^*)g'(x^*) + o(x_n - x^*) = e_n g'(x^*) + o(e_n).$$

Si on néglige $o(e_n)$ devant e_n , alors on peut écrire que

$$e_{n+1} \approx g'(x^*)e_n.$$

(c) Vu 1.b. : $e_{n+1} \approx [g'(x^*)]^2 e_{n-1}$. Posons,

$$H(n) : \quad e_n = [g'(x^*)]^n e_0.$$

Or,

$$\begin{aligned} e_{n+1} &\approx g'(x^*)e_n \\ \Rightarrow e_{n+1} &\approx g'(x^*)^{n+1}e_0, \quad i.e., H(n+1) \end{aligned}$$

Conclusion : on a bien $e_n = [g'(x^*)]^n e_0$.

Donc si,

$$|g'(x^*)| < 1, \quad \lim_{n \rightarrow \infty} (g'(x^*))^n = 0,$$

et $e_n \rightarrow 0$ quand $n \rightarrow \infty$.

Et donc,

$$\lim_{n \rightarrow \infty} x_n = x^*.$$

(d) On suppose que $-1 < g'(x^*) < 0$. Puisque $|g'(x^*)| < 1$ alors x_n converge vers x^* , or $e_{n+1} \approx g'(x^*)e_n$. Donc e_n et e_{n+1} sont de signes opposés.

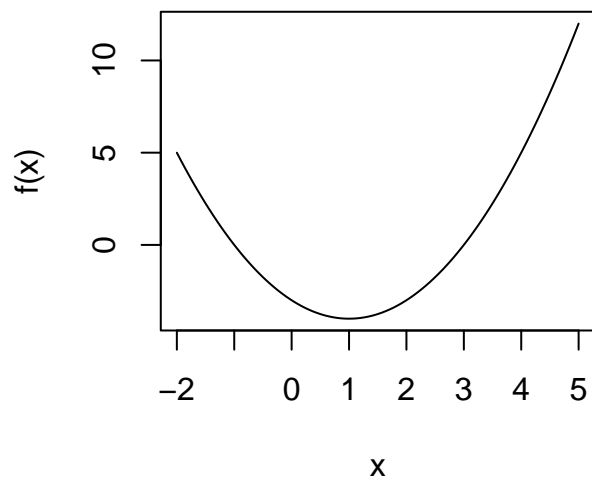
2

```
pfixe <- function(x0, eps, fun, Nmax){  
  # Input :  
  #       - x0 point depart  
  #       - eps precision  
  #       - fun fonction pour laquelle un pt fixe  
  #       - Nmax nb max d'iteration autorise  
  # Output :  
  #       - xnp approx de la solutin x^*  
  #       - iter nb d'iteration associee  
  #       (pour une precision eps)  
  n <- 0  
  xn <- x0  
  xnp1 <- fun(xn)  
  while (abs(xn-xnp1) > eps & n < Nmax){  
    xn <- xnp1  
    xnp1 <- fun(xn)  
    n <- n + 1  
  }  
  return(list("x"=xnp1,  
             "iter"=n+1))  
}
```

```
# variante while/if  
pofixe <- function(x0, eps=1e-5, fun, Nmax=50){  
  n <- 1  
  while(n < Nmax){  
    x <- fun(x0)  
    if ( abs(x - x0) < eps){  
      return(list("x"=x, "iter"=n))  
    }  
    n <- n + 1  
    x0 <- x  
  }  
  return(list("x"=x0, "iter"=n))  
}
```

3.

```
f <- function(x) x^2-2*x-3  
curve(f, -2, 5)
```



```

g1 <- function(x){sqrt(2*x+3)}# CV
g2 <- function(x){3/(x-2)}# CV
g3 <- function(x){(x^2-3)/2} # DV

x0 <- 4
eps <- 1e-8
Nmax <- 100

resg2 <- pfixe(x0=x0,
               eps=eps,
               fun=g2,
               Nmax=Nmax)

resg1 <- pfixe(x0=x0,
               eps=eps,
               fun=g1,
               Nmax=Nmax)

resg3 <- pofixe(x0=x0,
               eps=eps,
               fun=g3,
               Nmax=2)

```

4.

```

pfixe2 <- function(x0, eps, fun, Nmax){
  n <- 0
  xn <- x0
  xnp1 <- fun(xn)
  xnp <- c()
  while (abs(xn-xnp1) > eps & n < Nmax){
    xn <- xnp1
    xnp1 <- fun(xn)
    n <- n + 1
  }
}

```

```

    xnp[n] <- xnp1
  }
  return(list("x"=c(fun(x0),xnp),
              "n"=n+1))
}

```

```

resg1_2 <- pfixe2(x0=x0,
                 eps=eps,
                 fun=g1,
                 Nmax=Nmax)

```

```

resg2_2 <- pfixe2(x0=x0,
                 eps=eps,
                 fun=g2,
                 Nmax=Nmax)

```

```

resg3_2 <- pfixe2(x0=x0,
                 eps=eps,
                 fun=g3,
                 Nmax=5)

```

```

iterpfixe <- function(x0, x1, k, eps){
  floor(log(eps*(1-k)/abs(x1-x0))/log(k)) + 1
}
g1p <- function(x){(2*x-3)^(-1/2)}
g2p <- function(x){-3/(x-2)^2}
g3p <- function(x){x}
k <- 0.4
x1 <- g2(x0)
iterpfixe(x0, x1, k, eps)

```

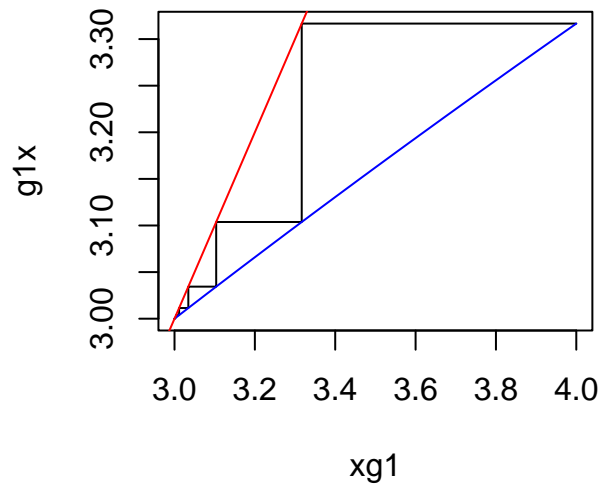
Analyse de la convergence (1. (e-f)).

L'équation $x^2 - 2x + 3 = 0$ possède deux racines réelles $x_1 = -1$ et $x_2 = 3$. On a $g'_1(x) = (2x - 3)^{-1/2}$ et $g'_1(x_1) = 1 \geq 1$ point fixe douteux, $g'_1(x_2) = 1/3$, point fixe attractif (cas $0 < g'(x^*) < 1$). De plus, $g'_2(x) = -3/(x - 2)^2$, $g'_2(x_1) = -1/3$ point fixe attractif (cas $-1 < g'(x^*) < 0$) et $g'_2(x_2) = -3$ point fixe répulsif. Pour g_3 , on a $g'(x) = x$ et x_1 et x_2 sont des points fixes répulsifs, pas de convergence (cas $|g'(x^*)| > 1$ et $|g'(x^*)| = 1$ douteux).

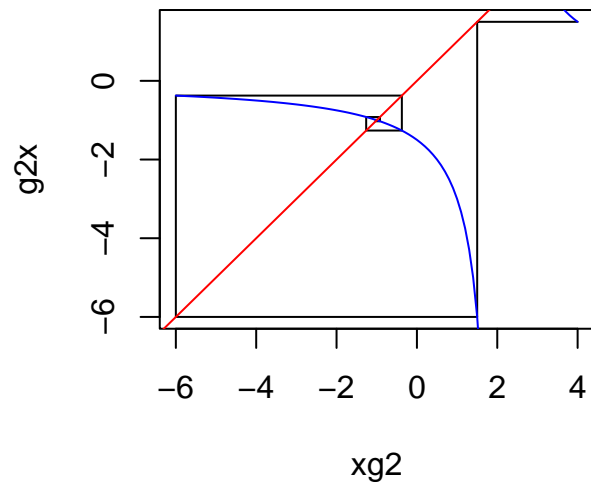
```

g1x <- rep(resg1_2$x, each=2)
xg1 <- c(x0, head(g1x,-1))
plot(xg1,g1x,type="s")
curve(g1,add=T, col="blue")
abline(0,1,col="red")

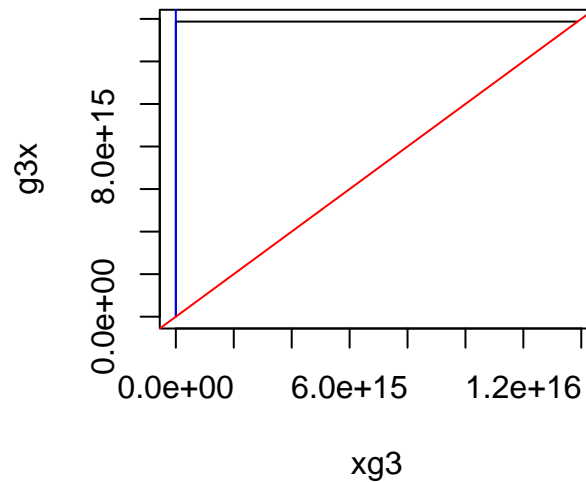
```



```
g2x <- rep(resg2_2$x, each=2)
xg2 <- c(x0, head(g2x, -1))
plot(xg2, g2x, type="s")
curve(g2, add=T, col="blue")
abline(0, 1, col="red")
```

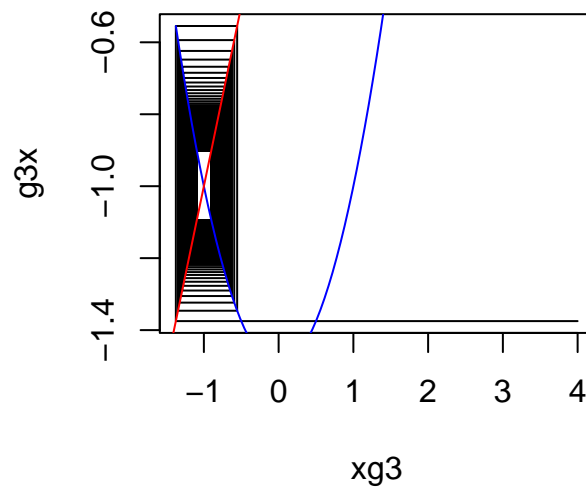


```
g3x <- rep(resg3_2$x, each=2)
xg3 <- c(x0, head(g3x, -1))
plot(xg3, g3x, type="s")
curve(g3, add=T, col="blue")
abline(0, 1, col="red")
```

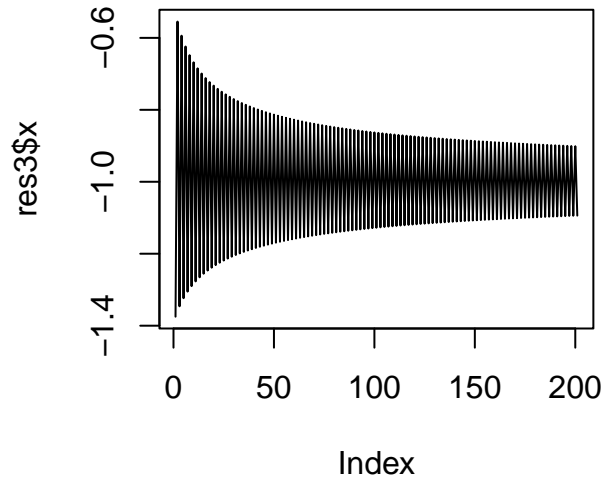


Dans le cas douteux, par exemple pour g_3 et $x^* = -1$ si on choisit une initialisation différente (de $x_0 = 4$), plus proche de $x^* = -1$ (à gauche ou à droite), on peut observer que la suite va osciller autour de x^* . (cela ne signifie pas qu'il n'existe pas de cas douteux pour lesquels une analyse plus approfondie permettrait de conclure, e.g. $g(x) = \sin x, x \in [0, \pi/2], x^* = 0, |g'(x^*)| = 1$).

```
res3 <- pfixe2(x0=-0.5,
  eps=eps,
  fun=g3,
  Nmax=200)
g3x <- rep(res3$x, each=2)
xg3 <- c(x0, head(g3x,-1))
plot(xg3,g3x,type="s")
curve(g3, add=T, col="blue")
abline(0,1,col="red")
```



```
plot(res3$x,type="l")
```



5. $g(x) = \sqrt{1+x}$

```
g5 <- function(x){sqrt(1+x)}
x0 <- 1.5
resg5 <- pfixe2(x0=x0,
               eps=eps,
               fun=g5,
               Nmax=Nmax)

abs(diff(resg5$x))
#> [1] 2.545347e-02 7.902138e-03 2.445397e-03 7.560037e-04 2.336500e-04
#> [6] 7.220486e-05 2.231282e-05 6.895068e-06 2.130696e-06 6.584215e-07
#> [11] 2.034635e-07 6.287367e-08 1.942903e-08 6.003901e-09

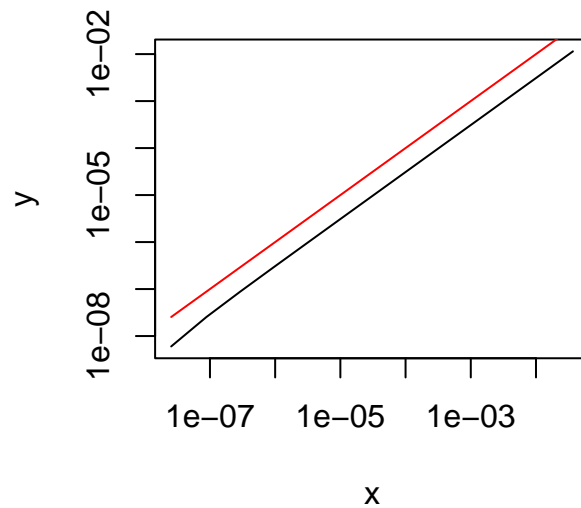
en <- resg5$x-resg5$x[length(resg5$x)]
abs(en[2:length(en)]/en[1:(length(en)-1)])
#> [1] 0.3101134 0.3093552 0.3091210 0.3090476 0.3090215 0.3090023 0.3089605
#> [8] 0.3088313 0.3084149 0.3070645 0.3026582 0.2880072 0.2360680 0.0000000
```

Le calcul numérique de l'erreur peut être utilisé pour vérifier les estimations théoriques, mais aussi pour comparer la vitesse de convergence de différentes méthodes pour approcher une racine x^* de f . Comme on ne connaît (en général) pas x^* , on calculera en pratique l'erreur $|e_n| = |x_n - x_a^*|$ pour $n \leq n_{it} \leq N_{max}$ et $x_a^* = x^*$ si x^* est connu et x_a^* une valeur approchée de x^* aussi précise que possible (e.g., la dernière valeur de la suite calculée pour une précision donnée).

Ordre p si $|e_n| \approx C|e_{n-1}|^p$, ou encore $\log(|e_n|) \approx \log(C) + p \log(|e_{n-1}|)$. Les points $(\log(|e_{n-1}|), \log(|e_n|))$ sont sur une droite de pente p . Pour déterminer l'ordre on trace donc les vecteurs $x = \log(|e_{n-1}|)$ et $y = \log(|e_n|)$, la courbe que l'on obtient est une droite de pente p , on peut ajouter la droite d'équation $y = px$.

```
x <- abs(en[1:(length(en)-2)])
y <- abs(en[2:(length(en)-1)])
plot(x, y,
     log="xy",
     type="l")
```

```
p = 1
lines(x,
      p*x,col="red",type="l")
```

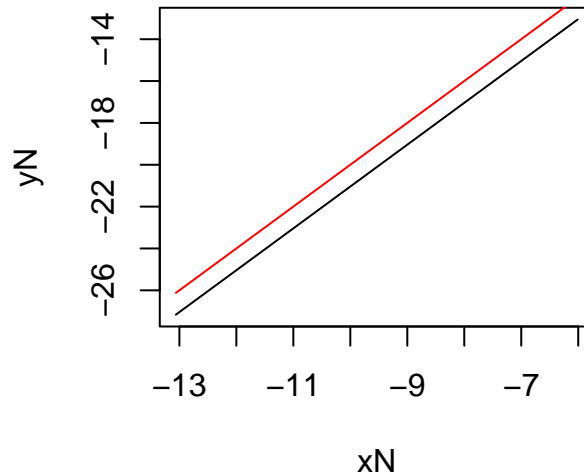


La méthode est d'ordre 1, vitesse : $C \approx 0.3$.

6. $f(x) = x^2 - 2$, $f'(x) = 2x$, $g(x) = x - \frac{x^2-2}{2x}$

```
gN <- function(x){x-(x^2-2)/(2*x)}
resN <- pfixe2(x0=x0,
              eps=eps,
              fun=gN,
              Nmax=Nmax)

enN <- resN$x-sqrt(2)
xN <- log(abs(enN[1:(length(enN)-2)]))
yN <- log(abs(enN[2:(length(enN)-1)]))
plot(xN, yN,
     type="l")
p = 2
lines(xN,
      p*xN,col="red",type="l")
```



```
abs(enN[2:(length(enN)-1)]) / abs(enN[1:(length(enN)-2)]^2)
#> [1] 0.3529412 0.3535224
```

La méthode est d'ordre 2, la vitesse $C \approx 0.35$. La vitesse pourrait également être estimée par une regression linéaire, tout comme l'ordre.

```
exp(coef(lm(yN~xN))["(Intercept)"])
#> (Intercept)
#> 0.3524466
coef(lm(yN~xN))[2]
#> xN
#> 1.999767
```

Exercice 3.

1. Newton 2d

```
Newton2d <- function(x0, eps=.Machine$double.eps^(1/2), fun, J, Nmax){
  vecnorm <- function(x) { sqrt(sum(x^2)) }
  n <- 0
  err <- eps + 1
  x <- x0
  while (err > eps && n < Nmax){
    n <- n + 1
    Fx <- fun(x)
    Jx <- J(x)
    z <- solve(Jx, Fx)
    x <- x - z
    err <- vecnorm(z)
  }
  Fx <- vecnorm(fun(x))
  if (n > Nmax && err > eps) {
    cat("pas de convergence")
  }
  return(list(zero=x, fnorm=Fx, niter=n))
}
```

```

}

F1 <- function(x) c(3*x[1]^2 - x[2]^2, 3*x[1]*x[2]-x[1]^3-1)
JF1 <- function(x)
  matrix( c(6*x[1], -2*x[2], 3*x[2]-3*x[1]^2, 3*x[1]), 2, 2,
          byrow = T)

X0 <- c(1,1)
eps <- 1e-8
Nmax <- 100
Newton2d(x0 = X0,
  eps = eps,
  fun = F1,
  J = JF1,
  Nmax = Nmax)
#> $zero
#> [1] 0.4594763 0.7958364
#>
#> $fnorm
#> [1] 0
#>
#> $niter
#> [1] 6

```

Exercice 4.

1. \Rightarrow On suppose que x^* est racine de multiplicité m . On va montrer que

$$f^{(k)}(x^*) = 0,$$

pour $0 \leq k \leq m-1$, $f^{(m)}(x^*) \neq 0$.

Formule de Leibniz : si f et g sont n fois dérivables, alors (fg) l'est aussi.

$$(fg)^{(n)} = \sum_{p=0}^n C_n^p f^{(p)} g^{(n-p)}.$$

x^* racine d'ordre m , par définition, on a :

$$f(x) = (x - x^*)^m h(x)$$

avec $h(x) \neq 0$.

$$\begin{aligned}
 f^{(k)}(x) &= ((x - x^*)^m h(x))^{(k)} \\
 &= \sum_{j=0}^k C_j^k ((x - x^*)^m)^{(j)} \cdot h(x)^{(k-j)}.
 \end{aligned}$$

— pour $0 \leq k \leq m-1$

$$\begin{aligned} ((x-x^*)^m)^{(k)} &= C(x-x^*)^{m-k} \\ ((x-x^*)^m)^{(k)} \Big|_{x=x^*} &= 0 \Rightarrow f^{(k)}(x^*) = 0. \end{aligned}$$

— pour $k = m$

$$f^{(m)}(x) = h(x)$$

et pour $x = x^*$, $f^{(m)}(x^*) = h(x^*) \neq 0$.

\Leftarrow On suppose que $f^{(k)}(x^*) = 0$, pour $0 \leq k \leq m-1$, $f^{(m)}(x^*) \neq 0$. Montrons que x^* est racine d'ordre m .

$f \in C^\infty(I)$, on fait un DL de Taylor-Young au voisinage de x^* :

$$f(x) = f(x^*) + (x-x^*)f'(x^*) + \dots + \frac{(x-x^*)^{m-1}}{(m-1)!}f^{(m-1)}(x^*) + \frac{(x-x^*)^m}{m!}f^{(m)}(x^*) + \frac{(x-x^*)^{m+1}}{(m+1)!}f^{(m+1)}(x^*).$$

Or par hypothèse, nous avons que $f^{(k)}(x^*) = 0$, pour $0 \leq k \leq m-1$, donc

$$\begin{aligned} f(x) &= \frac{(x-x^*)^m}{m!}f^{(m)}(x^*) + \frac{(x-x^*)^{m+1}}{(m+1)!}f^{(m+1)}(x^*) \\ &= (x-x^*)^m \left[\frac{f^{(m)}(x^*)}{m!} + \frac{(x-x^*)}{(m+1)!}f^{(m+1)}(x^*) \right] \\ &= (x-x^*)^m h(x) \end{aligned}$$

Par suite,

$$h(x^*) = \frac{f^{(m)}(x^*)}{m!}$$

Par hypothèse $f^{(m)} \neq 0 \Rightarrow h(x^*) \neq 0$.

2. $f(x) = x \sin x$, $f(0) = 0$ (0 racine de f).

$$f'(x) = \sin x + x \cos x, \quad f'(0) = 0, \quad (0 \text{ racine double}).$$

$$f''(x) = 2 \cos x - x \sin x, \quad f''(0) = 2 \neq 0.$$

Ok d'après 1. (avec $f(x) = x$ et $g(x) = \sin x$).

3. Montrons que dans le cas d'une racine de multiplicité $m > 1$, la méthode de Newton appliquée à cette fonction ne converge pas de façon quadratique.

Newton :

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

Posons

$$g(x) = x - \frac{f(x)}{f'(x)}$$

$$\begin{aligned}
e_{n+1} &= g(x^* + e_n) - g(x^*) \\
&= g(x^*) + e_n g'(x^*) + \frac{e_n^2}{2} g''(x^*) + \dots - g(x^*)
\end{aligned}$$

et posons

$$f(x) = (x - x^*)h(x).$$

On a

$$g'(x) = \frac{f(x)f''(x)}{(f'(x))^2}$$

et

$$\begin{aligned}
f'(x) &= ((x - x^*)h(x))' = m(x - x^*)^{m-1}h(x) + (x - x^*)^m h'(x) \\
f''(x) &= m(m-1)(x - x^*)^{m-2}h(x) + 2m(x - x^*)^{m-1}h'(x) + (x - x^*)^m h''(x) \\
g'(x) &= \frac{(x - x^*)^{2m-2}h(x) [m(m-1)h(x) + 2m(x - x^*)^{m-1}h'(x) + (x - x^*)^2 h''(x)]}{(x - x^*)^{2m-2}(mh(x) + (x - x^*)h'(x))^2}.
\end{aligned}$$

Pour $x = x^*$,

$$g'(x^*) = \frac{m(m-1)h^2(x^*)}{m^2 h^2(x^*)} = \frac{m-1}{m} = 1 - \frac{1}{m}.$$

Si $m > 1$, on a $g'(x) \neq 0$, la convergence ne peut pas être quadratique.

4. On pose

$$g(x) = x - m \frac{f(x)}{f'(x)}.$$

On a

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

Pour $x = x^*$,

$$g'(x^*) = 1 - m + m \frac{m-1}{m} = 0.$$