Compte rendu de calcul scientifique 1

Courtois Thibault, Imperatore Valentin

2022-2023

Professeur :

Monsieur M. ERSOY, Maître de Conférences, Seatech

Table des matières

1 Introduction

L'objectif de ce TP est d'étudier différentes méthodes qui permettent de résoude les équations types f(x)=0 sur [a,b] avec une fonction continue et à valeur réelle sur [a,b]. Nous allons étudier deux méthodes itératives en utilisant le langugage de programmation Fortran77 : la méthode du point fixe et la méthode de Newton.

2 Fonction étudiée

Dans ce TP nous allons étudié la fonction $f(x) = x - e^{-x}$ sur l'intervalle [0,1].
Cette fonction est continue et dérivable sur l'intervalle d'étude et sa dérivée est strictement croissante.
On peut donc appliquer le théorème des valeurs intermédiaires à la fonction f sur l'intervalle [0,1] : il existe un unique $\alpha \in [0,1]$ tel que $f(\alpha) = 0$.

A l'aide de Fortran et de gnuplot, on peut tracer la fonction et trouver graphiquement la valeur de α à $10^{-}2$ près.

3 Méthode du point fixe

La méthode du point fixe est basé sur la transformation suivante : $f(x) = 0 \Longleftrightarrow x = \Phi(x)$ (la fonction Φ n'est pas unique).

Les fonctions Φ doivent respecter les propriétés stipulé par le théorème du point fixe :

- l'intervalle d'étude doit être stable par la fonction Φ.

- elle doit être contractante au voisinage de [a,b].

Dans le cas de la fonction $f(x) = x - e^{-x}$ sur [0,1], trois fonctions conviennent :
$\Phi_1(x) = e^{-x}$
$\Phi_2(x) = x - \frac{2}{3+e^{-1}}(x - e^{-x})$
$\Phi_3(x) = -ln(x)$

La méthode du point fixe consiste à construire la suite :

$$\begin{cases} x_0 \in [a,b] \\ x_{n+1} = \Phi(x_n) \end{cases}$$

L'algorithme suivant permet de calculer les termes de la suite $x(n)$ et de déterminer le point fixe à une précision donné :

Initialisation :
$x \leftarrow x_0$
$k \leftarrow 0$
eps $\leftarrow$ precision
while $|f(x)| < eps$ ***et*** $k<kmax$ **do**
| $x \leftarrow \Phi(x)$ $k = k + 1$
end

On va coder cet algorithme en fortran pour les trois fonctions Φ décrite précédemments et regarder les 9 premières valeurs de la suite pour chaque fonction.

```
n = 100
dx = 1.0d0/n
do i=1,N+1
   x(i) = (i-1)*dx
   y(i) = x(i)-dexp(-x(i))
enddo

open(1,file='exp')
do i=1,N+1
  write(1,*) x(i),y(i)
enddo
close(1)

x1(1) = 0.5
do i=2,9
   x1(i) = dexp(-x1(i-1))
   print*, x1(i)
enddo

end
```

```
n = 100
dx = 1.0d0/n
do i=1,N+1
   x(i) = (i-1)*dx
   y(i) = x(i)-dexp(-x(i))
enddo

open(1,file='exp')
do i=1,N+1
  write(1,*) x(i),y(i)
enddo
close(1)

x1(1) = 0.5
do i=2,9
   x1(i) = -log(x1(i-1))
   print*, x1(i)
enddo

end
```

FIGURE 1 – Code fortran pour Φ_2 et Φ_3

```
0.60653065971263342
0.54523921189260505
0.57970309487806826
0.56006462793890188
0.57117214897721513
0.56486294698032347
0.56843804757006622
0.56640945274692078
```

```
0.53919043956270862
0.55538808812696416
0.56217966708438027
0.56504386865291556
0.56625468698052306
0.56676706684727596
0.56698398194726751
0.56707582907758480
```

```
 0.69314718055994529
 0.36651292058166435
  1.0037215043023100
-3.7145966378050060E-003
                     NaN
                     NaN
                     NaN
                     NaN
```

FIGURE 2 – Résultats pour Φ_1, Φ_2 et Φ_3

On observe que Φ_1 et Φ_2 convergent vers α (le point fixe) mais que la troisème fonction ne converge absolument pas. En fait cela est normale car elle ne respecte pas le critère de stabilité de l'intervalle d'étude du théorème du point fixe.

On va maintenant coder ma méthode du point fixe pour trouver la valeur de α pour une précision donnée. Ici la précision sera de 10^{-16}. Pour les fonctions Φ_1 et Φ_2.

On utilise les programmes suivant :

```
c Méthode du point fixe
        a=0.5
        kmax = 1000
        k=0
        eps = 1e-16
        DO while(dabs(a-dexp(-a))>eps.AND.(k<kmax))
           a = dexp(-a)
           k = k+1
        end do
        print*, a, k

        end
```

```
c Méthode du point fixe
        C = 2/3+dexp(-1.0d0)
        a=0.5
        kmax = 1000
        k=0
        eps = 1e-16
        DO while(dabs(c*(a-dexp(-a)))>eps.AND.(k<kmax))
           a = a-c*(a-dexp(-a))
           k = k+1
        end do
        print*, a, k

        end
```

FIGURE 3 – Code fortran pour Φ_1 et Φ_2

On obtient les résultats suivant :

```
 0.56714329040978384          62
```

(a)
Φ_1

```
0.56714329040978373          39
```

(b)
Φ_2

On observe que la vitesse de convergence n'est pas la même entre les deux fonctions (pour Φ_2 la convergence est beaucoup plus rapide).

4 Méthode de Newton

Cette méthode permet d'approximer le zéro d'une fonctrion f à partir de son développement de Taylor à l'odre 1.

$f(x) \approx f(x_0) + f'(x_0)(x - x_0) \iff 0 \approx f(x_0) + f'(x_0)(x - x_0)$
On peut alors construire une suite qui tant vers le zéro de la fonction f :

$x_0 \in [a, b]$
$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \iff x_{n+1} = \Phi(x_n)$ avec $\Phi(x_n) = x - \frac{f(x)}{f'(x)}$ On va maintenant implémenter cette méthode à l'aide du code fortran suivant :

```
c Méthode de Newton
      C = 2/3+dexp(-1.0d0)
      a=0.5
      kmax = 1000
      k=0
      eps = 1e-16
      DO while(dabs(a-dexp(-a))>eps.AND.(k<kmax))
         a = a - (a-dexp(-a))/(1+dexp(-a))
         k = k+1
      end do
      print*, a, k
```

```
0.56714329040978384          5
```

FIGURE 4 – Code fortran pour l'algo de Newton et résulats

On observe que 5 itérations suffisent pour obtenir un résultat d'une précision de 10^{-16}

5 Vitesse de convergence

Au vu des résulats précédents, on peut se demander pourquoi les méthodes ont des vitesses de convergences si éloignées.

5.1 Méthode du point fixe

On note $e_n = |x_{n+1} - x_n|$ l'erreur à l'itération n de la méthode.

Alors $e_n = |\phi(x_n) - \phi(x_{n-1})| \leq K|x_n - x_{n-1}|$ (propriété d'une fonction contractante).
Par récurrence on obtient :

$|x_{n+1} - x_n| \leq K^n |x_1 - x_0|$.

Donc l'erreur e_n tend vers 0 pour K<1 et la vitesse de convergence de la méthode est lié à l'odre de e_n. Si $e_{n+1} \approx \phi'(x)e_n$, la méthode du point fixe est à au moins une convergence linéaire. La rapidité de convergence dépend bien sur du coefficient K qui varie suivant la fonction ϕ choisi.

5.2 Méthode de Newton

On note x* la limite de la suite $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
Cette écriture est équivalente à $\Longleftrightarrow x_{n+1} = \Phi(x_n)$ avec $\Phi(x_n) = x - \frac{f(x)}{f'(x)}$.
On a donc $x^* = \phi(x^*)$ donc x* est un point fixe de ϕ.
On s'intéresse ici à la quantitée $x_n - x^*$ pour étudier la vitesse de convergence de la suite.
$x_n - x^* = \phi(x_{n-1}) - x^*$
Or $\phi(x) - x^* = x - \frac{f(x)}{f'(x)} - x^* = \frac{xf'(x)-f(x)-x^*f'(x)}{f'(x)} = -\frac{f(x)+f'(x)(x^*-x)}{f'(x)}$.

On reconnait le début d'une développement de Taylor de f en x* au numérateur.

$f(x^*) = 0 = f(x) + f'(x)(x^* - x) + \frac{1}{2}f''(c)(x^* - x)^2$ avec c une constante entre x et x*.
On peut donc remplacer le numérateur :

$\phi(x) - x^* = \frac{1}{2}\frac{f"(c)(x^*-x)^2}{f'(x)}$ donc $|\phi(x) - x^*| \leq \frac{1}{2}\frac{|f"(c)|}{|f'(x)|}|x^* - x|^2$

On pose M = sup$|f"|$ et m = inf$|f"|$ sur l'intervalle d'étude I.
Alors $\frac{1}{2}\frac{|f"(c)|}{|f'(x)|} \leq \frac{1}{2}\frac{M}{m} = K$

D'où $|\phi(x) - x^*| \leq K|x^* - x|^2 \Longleftrightarrow |K(\phi(x) - x^*)| \leq |K(x - x^*)|^2$

On peut maintenant appliquer cette inégalité à la suite x_n :
$|K(x_n - x^*)| \leq |K(x_{n-1} - x^*)|^2$

et $|K(x_n - x^*)| \leq |K(x_{n-2} - x^*)|^{2x2}$

Donc par récurrence on aboutit au résultat suivant : $|K(x_n - x^*)| \leq |K(x_0 - x^*)|^{2^n}$

Donc si $|K(x_0 - x^*)| < 1$, **la méthode converge quadratiquement**.

6 Conclusion

La méthode du point fixe à une vitesse de convergence linéaire qui dépend de la fonction ϕ choisi. Cela explique les différences de rapidité de convergence pour les foncions ϕ_1, ϕ_2 et ϕ_3.

La méthode de Newton à une vitesse de convergence quadratique, elle est donc beaucoup plus rapide que la méthode du point fixe. D'où les différences de nombre d'itérations observés.

Cependant, la méthode du point fixe ne nécessite pas de calculer la dérivé de la fonction.Elle reste donc plus simple à mettre en place.