

TP1 - M43 Analyse numérique

Gloria Faccanoni

Ce sujet, de 3 pages, comporte 3 exercices et 1 DM.

On s'intéresse dans ce TP aux diverses méthodes vues en cours et/ou en TD pour le calcul du zéro d'une fonction réelle de variable réelle.

1 Exercice

1. Programmer la méthode de la dichotomie.
2. Chercher le zéro de la fonction $f: x \mapsto x^2 - x - 2$ sur l'intervalle $[0; 3.5]$ avec une tolérance de 10^{-10} sur l'erreur absolue.

2 Exercice

1. Programmer la méthode de Newton (on utilisera une dérivée approchée).
2. Chercher le zéro de la fonction $f: x \mapsto x^2 - x - 2$ sur l'intervalle $[0; 3.5]$ avec une tolérance de 10^{-10} sur l'erreur absolue. Déterminer expérimentalement les sous-intervalles pour lesquels la méthode de Newton converge vers $\ell = 2$.

3 Exercice

1. Programmer la méthode de point fixe.
2. Analyser (sur le papier !) la convergence de la méthode de point fixe $x^{k+1} = \phi(x^k)$ pour le calcul des zéros de la fonction $f: x \mapsto x^2 - x - 2$ quand on utilise les fonctions d'itération suivantes

$$\begin{aligned}\phi_1: x &\mapsto x^2 - 2, \\ \phi_2: x &\mapsto \sqrt{2+x}, \\ \phi_3: x &\mapsto -\sqrt{2+x}, \\ \phi_4: x &\mapsto 1 + \frac{2}{x}, \quad x \neq 0.\end{aligned}$$

On vérifiera notamment que la méthode ne converge pas avec ϕ_1 , elle converge seulement vers 2 avec ϕ_2 et ϕ_4 et elle converge seulement vers -1 avec ϕ_3 .

3. Vérifier ces résultats expérimentalement.

4 DM

Envoyer à l'adresse gloria.faccanoni@univ-tln.fr un seul fichier **NOM-PRENOM.c** permettant de rechercher les zéros de la fonction $f: x \mapsto x^2 - x - 2$ sur l'intervalle $[0; 3.5]$ avec une tolérance de 10^{-10} sur l'erreur absolue avec la **méthode de Lagrange**. Déterminer expérimentalement les sous-intervalles pour lesquels la méthode converge vers $\ell = 2$.

Fichier 1: math.c

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/*
 *
 * gcc -o math -lm math.c && ./math
 *
 */
int main(int argc, char** argv) {

    double toll = 1.0e-10;

    double f(double x) {
        return x * x - x - 2.0;
    }

    // Dichotomie
    double a = 0.0;
    double b = 3.5;
    if (a > b) {
        double temp;
        temp = a;
        a = b;
        b = temp;
    }
    while ((b - a) > toll) {
        if (f((a + b) / 2.0) * f(a) > 0)
            a = (a + b) / 2.0;
        else
            b = (a + b) / 2.0;
    }
    printf("\nZero avec dichotomie: %t%lf\n", (a + b) / 2.0);

    // Newton
    double x, old_x;

    double derivee(double (*f)(double), double x, double toll) {
        return ((f(x + toll / 2.0) - f(x - toll / 2.0)) / toll);
    }

    x = 1.0;
    do {
        old_x = x;
        x -= f(x) / derivee(f, x, toll);
    } while (fabs(x - old_x) > toll);
    printf("Zero avec Newton: %t%lf\n", x);

    // Pt Fixe
    x = 1.0;
    double phi(double x) {
        return sqrt(2.0 + x);
        //return -sqrt(2.0 + x);
    };
    do {
        old_x = x;
        x = phi(x);
    } while (fabs(x - old_x) > toll);
    printf("Zero avec point fixe: %t%lf\n", x);

    // Lagrange ou secante
```

```
double c;
while ((b - a) > toll) {
    c=a-(b-a)/(f(b)-f(a))*f(a);
    if (f(c) * f(a) > 0)
        a = c;
    else
        b = c;
}
printf("Zero avec Lagrange: \t%f\n", a-(b-a)/(f(b)-f(a))*f(a));

// Corde
double pente=(b-a)/(f(b)-f(a));
do{
    old_x=x;
    x-=pente*f(x);
} while (fabs(x - old_x) > toll);
printf("Zero avec corde: \t%f\n\n", x);

return (EXIT_SUCCESS);
}
```