Atelier 8

Objectifs de formation
— Résoudre les équations algébriques de type f(x) = 0
— Passer une fonction en paramètre d'une autre fonction
— Différencier la méthode de Newton et la méthode Regula Falsi

1. : **180 mn donnant lieu à un rendu**
   — Implémenter la fonction
   ```
   int bissection(double *z,
                  double (*f)(double),
                  double a,
                  double b,
                  double eps);}
   ```
   qui cherche une approximation à un zéro z de la fonction f, donnée sous forme de pointeur de fonction `f`, dans l'intervalle $[a; b]$. L'algorithme procède par dichotomie. Il s'arrête dès que la différence entre deux itérés successifs est plus petite en valeur absolue que ε. La valeur de ε est donnée sous la forme de l'argument `eps`. Si le signe de f en a et en b est le même, la fonction considère que f n'a pas de zéro dans l'intervalle et renvoie une valeur nulle pour signaler cet échec. Sinon, la fonction met le zéro z dans la variable pointée par `z` et renvoie une valeur non-nulle.

   Tester la fonction sur les fonctions
   — $f(x) = \exp^x - 4$,
   — $g(x) = \sin(x - 3)$,
   — $h(x) = 162x^3 - 18x - 3$,
   — $p(x) = 2x^2 - x^3 - 2x^4 + x^5$.
   sur les intervalles
   — f sur $[1; 1.5]$,
   — g sur $[-0.5; 0.5]$,
   — h sur $\left[-\frac{1}{3}; 1\right]$,
   — p sur $[-1.5; 0.5]$,
   On utilisera la valeur $\varepsilon = 10^{-8}$ pour les tests.
   — Implémenter la fonction
   ```
   int newton(double *z,
              double (*f)(double),
              double (*fprime)(double),
              double a,
   ```

```
            double b,
            double eps);
```

qui cherche une approximation à un zéro z de la fonction f, donnée sous forme de pointeur de fonction `f`, dans l'intervalle $[a; b]$. L'algorithme est basé sur la méthode de Newton. Il se sert de la dérivée de la fonction f, donnée sous la forme du pointeur de fonction `fprime`. Il s'arrête dès que la différence entre deux itérés successifs est plus petite en valeur absolue que ε. La valeur de ε est donnée sous la forme de l'argument `eps`. L'itération de Newton est amorcée avec $x_0 = \frac{a+b}{2}$. Si à un moment de l'iteration, la dérivée de la fonction devient nulle,l'algorithme s'arrête sur un cas d'échec, en renvoyant une valeur nulle. Sinon, la fonction met le zéro z dans la variable pointée par `z` et renvoie une valeur non-nulle.

Tester la fonction sur les fonctions
- $f(x) = \exp^x - 4$,
- $g(x) = \sin(x - 3)$,
- $h(x) = 162x^3 - 18x - 3$,
- $p(x) = 2x^2 - x^3 - 2x^4 + x^5$.

sur les intervalles
- f sur $[1; 1.5]$,
- g sur $[-0.5; 0.5]$,
- h sur $\left[-\frac{1}{3}; 1\right]$,
- p sur $[-1.5; 0.5]$,

On utilisera la valeur $\varepsilon = 10^{-8}$ pour les tests.

— Implémenter la fonction

```
int findzero(double *z,
             double (*f)(double),
             double (*fprime)(double),
             double a,
             double b,
             double eps);
```

qui cherche une approximation à un zéro z de la fonction f, donnée sous forme de pointeur de fonction `f`, dans l'intervalle $[a; b]$. L'algorithme est basé sur une combinaison de l'algorithme de bissection et de l'itération de Newton. Il se sert de la dérivée de la fonction f, donnée sous la forme du pointeur de fonction `fprime`. L'algorithme démarre avec l'algorithme de bissection et bascule sur l'algorithme de Newton dès que la différence relative entre deux itérés successifs est plus petite en valeur absolue que 0.1. Puis il s'arrête dès que la différence relative entre deux itérés successifs est plus petite en valeur absolue que ε. La valeur de ε est donnée sous la forme de l'argument `eps`.

Si `findZero` finit par trouver un zéro z, il le met dans la variable pointée par `z` et renvoie une valeur non-nulle. En cas d'échec, il renvoie la valeur nulle.

Tester la fonction sur les fonctions
— $f(x) = \exp^x - 4$,
— $g(x) = \sin(x - 3)$,
— $h(x) = 162x^3 - 18x - 3$,
— $p(x) = 2x^2 - x^3 - 2x^4 + x^5$.
sur les intervalles
— f sur $[1; 1.5]$,
— g sur $[-0.5; 0.5]$,
— h sur $\left[-\frac{1}{3}; 1\right]$,
— p sur $[-1.5; 0.5]$,
On utilisera la valeur $\varepsilon = 10^{-8}$ pour les tests.

2. **60 mn**
On veut comparer la méthode de Newton et la méthode Regula-Falsi

— Implémenter la fonction
```
int regula_falsi(double *z,
                 double (*f)(double),
                 double x0,
                 double x1,
                 double eps);
```
qui cherche une approximation à un zéro z de la fonction f, donnée sous forme de pointeur de fonction `f`. L'algorithme est basé sur la méthode Regula-Falsi. Il s'arrête dès que la différence entre deux itérés successifs est plus petite en valeur absolue que ε. La valeur de ε est donnée sous la forme de l'argument `eps`. L'itération Regula-Falsi est amorcée avec x_0 et x_1 passés par les arguments `x0` et `x1`. Si à un moment du calcul, $f(x_{n+1}) - f(x_n) = 0$, l'algorithme s'arrête sur un cas d'échec en renvoyant une valeur nulle. Sinon, la fonction met le zéro z dans la variable pointée par `z` et renvoie une valeur non-nulle.

Tester la fonction avec $f(x) = \exp^x - 4$, $x_0 = 0$, $x_1 = 1$ et $\varepsilon = 10^{-14}$.

— Comparer la méthode de newton de newton et la méthode Regula-Falsi sur la fonction

$$f(x) = x^3 + \frac{29}{21}.x^2 - \frac{67}{147}.x + \frac{5}{147}$$

avec $x_0 = 1$, $x_1 = 2$ et $\varepsilon = 10^{-14}$ puis avec $x_0 = -3$, $x_1 = -2$ et $\varepsilon = 10^{-14}$.

Expliquer les différences.