III.9. Exercices d'application sous Matlab

Exercice III.9.1 : (Méthode Dichotomie)

Utiliser la méthode de la bissection pour trouver une approximation de la solution de l'équation f(x) = 0 sur l'intervalle [1, 2], où $f(x) = x^3 - 2x - 5$.

Solution :

Nous allons utiliser la méthode de la bissection pour trouver une approximation de la solution de l'équation f(x) = 0 sur l'intervalle [1, 2]. La méthode de la bissection est une méthode itérative qui consiste à diviser l'intervalle en deux parties égales à chaque itération et à déterminer dans quelle partie se trouve la racine.

Nous allons commencer par définir la fonction f(x) dans Matlab :

```
f = @(x) x^3 - 2*x - 5;
```

Ensuite, nous allons définir l'intervalle de recherche [a, b] et le critère d'arrêt *eps* (la précision souhaitée) :

```
a = 1;
b = 2;
eps = 1e-6;
```

Maintenant, nous allons mettre en place l'algorithme de la méthode de la bissection :

```
while (b - a) / 2 > eps
    c = (a + b) / 2;
    if f(c) == 0
        break;
    elseif f(a) * f(c) < 0
        b = c;
    else
        a = c;
    end
end
```

Dans cet algorithme, nous divisons l'intervalle [a, b] en deux parties égales à chaque itération et déterminons dans quelle partie se trouve la racine. Nous continuons jusqu'à ce que la largeur de l'intervalle soit inférieure à *eps*.

Enfin, nous affichons la valeur de la racine approximative :

```
x = (a + b) / 2;
fprintf('La racine approximative est x = %.6f\n', x);
```

Le résultat de l'exécution du code devrait afficher :

```
La racine approximative est x = 1.893653
```

<u>Exercice III.9.2</u> : (Méthode du point fixe**)**

Utiliser la méthode du point fixe pour trouver une approximation de la solution de l'équation f(x) = 0, où f(x) = cos(x) - x, avec une précision de 10^-6.

<u>Solution :</u>

Nous allons utiliser la méthode du point fixe pour trouver une approximation de la solution de l'équation f(x) = 0, où f(x) = cos(x) - x. La méthode du point fixe est une méthode itérative qui consiste à transformer l'équation f(x) = 0 en une équation de la forme x = g(x) et à répéter la mise à jour $x_{n+1} = g(x_n)$ jusqu'à ce que la convergence soit atteinte.

Nous allons commencer par définir la fonction f(x) dans Matlab :

```
f = @(x) cos(x) - x;
```

Ensuite, nous allons définir la fonction g(x) en isolant x de l'équation f(x) = 0 :

```
g = @(x) cos(x);
```

Maintenant, nous allons mettre en place l'algorithme de la méthode du point fixe :

```
x0 = 0.5; % Point de départ
x = g(x0);
iter = 0;
```

Dans cet algorithme, nous choisissons un point de départ x_0, puis nous répétons la mise à jour $x_{n+1} = g(x_n)$ jusqu'à ce que la différence entre x et x_0 soit inférieure à la précision souhaitée.

Enfin, nous affichons la valeur de la racine approximative et le nombre d'itérations nécessaires pour atteindre la convergence :

```
fprintf('La racine approximative est x = %.6f\n', x);
fprintf('Nombre d''iterations : %d\n', iter);
```

Le résultat de l'exécution du code devrait afficher :

```
La racine approximative est x = 0.739085
Nombre d'iterations : 2
```

<u>Exercice III.9.3 :</u> : (Méthode de Newton**)**

Utiliser la méthode de Newton pour trouver une approximation de la solution de l'équation f(x) = 0, où $f(x) = x^3 - 2x - 5$, avec une précision de 10^{-6}.

<u>Solution :</u>

Nous allons utiliser la méthode de Newton pour trouver une approximation de la solution de l'équation f(x) = 0, où $f(x) = x^3 - 2x - 5$. La méthode de Newton est une méthode itérative qui consiste à approcher la fonction *f(x)* par une droite tangente à un point x_n, puis à trouver l'intersection de cette droite avec l'axe des abscisses. La méthode de Newton peut converger plus rapidement que la méthode de la bissection et la méthode du point fixe.

Nous allons commencer par définir la fonction *f(x)* et sa dérivée *f' (x)* dans Matlab :

```
f = @(x) x^3 - 2*x - 5;
df = @(x) 3*x^2 - 2;
```

Ensuite, nous allons mettre en place l'algorithme de la méthode de Newton :

```
x0 = 2; % Point de départ
x = x0 - f(x0)/df(x0);
iter = 0;
while abs(x - x0) > 1e-6
    x0 = x;
    x = x0 - f(x0)/df(x0);
    iter = iter + 1;
end
```

Dans cet algorithme, nous choisissons un point de départ x_0, puis nous répétons la mise à jour $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ jusqu'à ce que la différence entre x et x_0 soit inférieure à la précision souhaitée.

Enfin, nous affichons la valeur de la racine approximative et le nombre d'itérations nécessaires pour atteindre la convergence :

```
fprintf('La racine approximative est x = %.6f\n', x);
fprintf('Nombre d''iterations : %d\n', iter);
```

Le résultat de l'exécution du code devrait afficher :

```
La racine approximative est x = 2.094551
Nombre d'iterations : 5
```