

## TP2 : Résolution des systèmes d'équations non linéaires Méthode de Newton-Raphson

**But de TP :** Durant ce TP, nous allons mettre en œuvre l'algorithme de méthode de résolution des équations non linéaires étudiées pendant le cours : La méthode de Newton-Raphson.

Partie I: Méthode de Newton-Raphson

*Cas 1 : système non linéaire*

On se propose d'appliquer cette méthode pour la recherche des racines de la fonction non linéaire suivante.

$$f(x) = e^x - 2\cos(x)$$

Créer un fichier M-File contenant le programme ci-dessous puis l'exécuter.

```
x=-1:0.1:1;
f=exp(x)-2*cos(x);
figure(1);
plot(x,f); grid on;
title('Fonction : f(x)=exp(x)-2*cos(x)');
```

- 1- Ce programme vous permet de : .....
- 2- D'après la courbe obtenue, quelle est la valeur initiale  $x_0$  pour avoir une convergence rapide.  $f(x_0) \approx 0$ .  $x_0 = \dots\dots\dots$

Pour chercher la solution de  $(x)$ , on peut rajouter au programme précédent quelques lignes suivantes :

```
Clear all
clc;
x(1)=input('Donner la valeur initiale x(1): \n');
e=1e-10;
n=5000;
for i=2:n
f=exp(x(i-1))-2*cos(x(i-1));
diff=exp(x(i-1))+2*sin(x(i-1));
x(i)=x(i-1)-f/diff;
if abs(x(i)-x(i-1))<=e
xp=x(i);
fprintf('xp=%f\n',x(i));
break;
end
end
j=1:i;
figure(2);
plot(j,x(j),'*r',j,x(j));
xlabel('Nombre d\'itérations');
title('Convergence de la solution : Méth. de Newt.-Raph.');
```

Après exécution de ce programme :

- 1- donner les valeurs successives  $x(i)$ .
- 2- Après combien d'itérations y va-t-il converger ?
- 3- Donner la solution rapprochée :
- 4- Trouver la solution en utilisant la commande Matlab " *fzero* ".