Méthode de Newton

Comme il a été montré précédemment, la méthode de dichotomie exploite uniquement le signe de la fonction f aux extrémités des sous-intervalles. Lorsque cette fonction est différentiable, on peut établir une méthode plus efficiente en exploitant les valeurs de la fonction f et de ses dérivées. Le but dans cette section, est la programmation, sous Matlab, de la méthode itérative de Newton. Afin d'appréhender cette dernière, soit la figure ci-dessous :

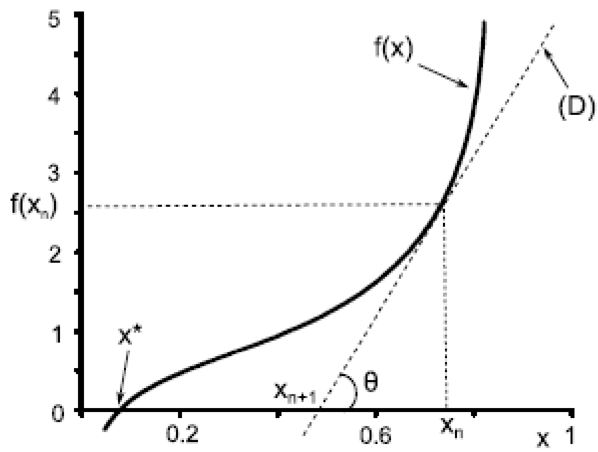

Figure 1 : Principe de la méthode de Newton.

Géométriquement, la solution approchée x_{n+1} n'est autre que le point d'intersection de l'axe des abscisses et la tangente, au point $(x_n, (x_n))$, d'équation D : $y = f'(x_n) \times (x_n - x_{n+1}) + f(x_n)$. Notons que x^* est la véritable racine de l'équation $(x) = 0$, dont on cherche à approcher. À partir de la figure ci-dessus, on a :

$$\tan\theta = \frac{f(x_n)}{x_n - x_{n+1}}$$

Or on sait que :

$$f'(x_n) = \lim_{(x_n - x_{n+1}) \to 0} \frac{f(x_n) - 0}{x_n - x_{n+1}} = \tan\theta$$

À partir des deux Eqs. on obtient ainsi le schéma numérique de la méthode de Newton, soit :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Tenant compte de toutes les méthodes vues jusqu'à présent, on constate que la méthode de Newton nécessite à chaque itération l'évaluation de deux fonctions, à savoir f et de sa dérivée. Néanmoins, cet effort est compensé par une vitesse de convergence accrue, puisque cette méthode est d'ordre deux. Cet accroissement de la vitesse de convergence est conditionné par le choix de la valeur initiale qui doit être la proche possible du zéro recherché.

Exercice :

Nous allons résoudre l'équation : $(x) = x+ex(x)+1$. Nous choisissons $x_0 = -1/2$ comme valeur initiale. Écrire un code matlab, portant sur l'implémentation de la méthode de Newton, en suivant les étapes suivantes :

1. Faire un test si $f'(x) = 0$ =⇒ arrêt du programme.
2. Le critère d'arrêt est : $|xn+1 - xn| < \varepsilon$, xn étant la solution approchée et ε, la tolérance considérée.
3. Afficher la solution approchée xn.
4. Afficher le nombre d'itérations conduisant à la solution approchée.
5. Afficher sur le même graphe, la fonction (x), la solution approchée xn et la droite tangente au point $(xn, (xn))$.

Appliquez le même algorithme pour résoudre l'équation : $(x) = 8x^3 - 12x^2 + 1$

TP N°3 Méthode Numérique

Script Matlab

```
clear all ; close all ; clc ;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



Nmax = 100; x = -1/2; it = 0; tol = 1e-04;
verif = tol + 1/2;

 while (it < Nmax & verif >= tol)

 fx   = inline('x +exp(x) + 1');
 dfx = inline('1 + exp(x)');
 fx = feval(fx, x);
 dfx = feval(dfx,x);

if dfx ~= 0
xn = x - (fx/dfx);
verif = abs(fx/dfx);
x = xn;
elseif dfx == 0
    disp('PAS DE SOLUTION, LA DERIVEE EST NULLE')
end

if (it == Nmax & verif > tol)
  disp('LA METHODE DE NEWTON NE CONVERGE PAS POUR LA TOLERANCE
   CONSIDEREE')
end

it = it +1;
 end

 disp(strcat('CONVERGENCE, LA SOLUTION APPROCHEE EST xn = ', num2str(
   xn)))
 disp(strcat('LE NOMBRE D''ITERATION EST = ', num2str(it)))

 xi = linspace(-12/2,6/2,1000);
 fx   = inline('xi +exp(xi) + 1');
 fxi = feval(fx, xi);
 dfx = inline('1 + exp(x)'); droite = dfx(xn)*(xi - xn) + fx(xn);
```

```
figure('color',[1 1 1]) ; plot(xi,fxi,'LineWidth',2)
hold on ; plot(xi,droite,'r','LineWidth',1)
    hold on
    plot(x,fx(x),'kx','MarkerSize',12,'LineWidth',2)
hold on
plot(x,fx(x),'ko','MarkerSize',12,'LineWidth',2)

xlabel('x','fontweight','b','fontsize',12,'LineWidth',1)
ylabel('f(x)','fontweight','b','fontsize',12,'LineWidth',1)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

text('Interpreter','latex','String','$f(x) = x + exp(x) + 1 $',...
'Position',[-10/2 20/2],'FontSize',15)

text(xn,2*xn,['x_n = ',num2str(xn)],'Position',[-3 2],...
'BackgroundColor',[1 1 1]);
```

Notons que la méthode de Newton converge de façon quadratique uniquement dans le cas où la racine recherchée de f est simple. Dans le cas contraire, elle converge de façon linéaire. Par ailleurs, cette méthode, peut être utilisée afin de résoudre des systèmes d'équations non linéaires.

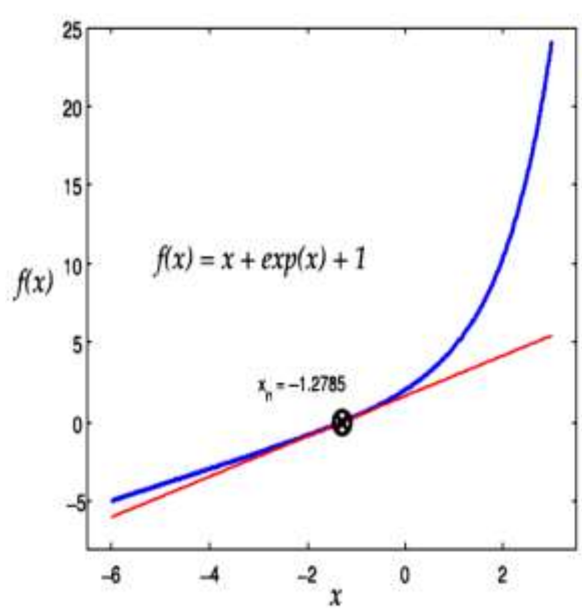

Figure 2: Racine de la fonction f obtenue par la méthode de Newton.