TD 10 - Analyse : méthode de Newton en dimension 2

On cherche à approcher numériquement une solution du sytème d'équations suivant :

$$\begin{cases} x^2 + y^2 = 2 \\ x^2 - y^2 = 1 \end{cases}$$

On réécrit ce système sous la forme : $f(x_1,\, x_2) = 0$ où f est une application de $\mathbb{R}^2$ dans $\mathbb{R}^2$ qui au vecteur $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ associe le vecteur $Y = \begin{pmatrix} f_1(x_1,\, x_2) \\ f_2(x_1,\, x_2) \end{pmatrix} = \begin{pmatrix} x_1^2 + x_2^2 - 2 \\ x_1^2 - x_2^2 - 1 \end{pmatrix}$.

On souhaite appliquer la méthode de Newton et comparer avec le résultat obtenu avec la fonction *fsolve* de Scilab qui permet la résolution de systèmes non linéaires.

Donnons le principe de la méthode de Newton en dimension 2.
Comme pour le cas de la dimension 1, la méthode consiste à regarder la solution de l'équation $f(X) = 0$ comme solution de l'équation $F(X) = X$ avec $F(X) = X - Df(X)^{-1} \circ f(X)$ où $Df(X)$ est la matrice jacobienne de f que l'on suppose inversible.
Dans l'exemple, la matrice jacobienne s'écrit : $\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} 2x_1 & 2x_2 \\ 2x_1 & -2x_2 \end{pmatrix}$; (inversible pour x_1 et x_2 non nuls)
On construit alors une suite de vecteurs (X_k) définie par X_0 et pour tout entier naturel, $X_{k+1} = F(X_k)$.
On arrête le calcul des termes successifs de la suite lorsque $\|X_{k+1} - X_k\| < \epsilon$ où ϵ est la précision voulue.

Travail demandé :
Programmer la méthode de Newton sur l'exemple proposé, et déterminer numériquement une solution du système.

1. Déterminer les valeurs exactes des solutions.
2. Définir deux fonctions qui prennent en argument le vecteur $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ et qui renvoient respectivement $f(X)$ et $Df(X)$.
3. Implémenter la méthode de Newton. On pourra en plus de la condition d'arrêt $\|X_{k+1} - X_k\| < \epsilon$, limiter le nombre d'itérations pour le cas où la méthode ne converge pas.
4. Afficher la solution et comparer le résultat avec la fonction *fsolve* : $X = fsolve\,(X_0,\, f)$.

Script Scilab

```
// méthode de Newton
// avec la fonction f(x1,x2) définie comme suit :
function Y=f(X)
    Y=[X(1)^2+X(2)^2-2;X(1)^2-X(2)^2-1];
endfunction
// ---------------------------------------------
utilisateur=x_mdialog('précision et nombre maximum d''itérations',...
['eps';'n'],['0.001';'10']);
eps=evstr(utilisateur(1));
n=evstr(utilisateur(2));
X0=evstr(x_matrix('vecteur initial X0',[1;1]));
// --------------------------------------------
function Df=Jac(X)
    Df(1,1)=2*X(1);
    Df(1,2)=2*X(2);
    Df(2,1)=2*X(1);
    Df(2,2)=-2*X(2);
endfunction
// ---------------------------------------------
// méthode de Newton
function [X, dist]=Newton(f, Jac, eps, n)
    X=X0;
    k=1;
    ep2=eps^2;
    dist2=eps^2+1;
    while k<=n & dist2>ep2
        Xk=Jac(X)\f(X);
        X=X-Xk;
        dist2=Xk'*Xk;
        k=k+1;
    end
    dist=sqrt(dist2);
endfunction
// -------------------------------------------
[X,dist]=Newton(f,Jac,eps,n);
disp('solution de f(x1,x2)=0 :');
disp(X);
// utilisation de la fonction fsolve
y=fsolve(X0,f);
disp('solution de f(x1,x2)=0 avec fsolve :');
disp(y);
```