Analyse Numérique

Corrigés des Travaux Pratiques 2013 – 2014
Séances 8

2.a) La fonction vectorielle $F(\mathbf{x})$ s'annule si et seulement si une de ses normes $\|F(\mathbf{x})\|$ s'annule. On peut donc visualiser $\|F(x,y)\|$ en utilisant la fonction `mesh` et voir où cette fonction croise le plan $z = 0$.

2.b,c) On peut procéder comme suit

```
function tp7ex2
   % la fonction F(x)
   %   : argument est un vecteur (x,y)
   %   : retourne un vecteur 2x1
   F = @(x)( [x(1).^2-x(2)-1; ...
                 (x(1)-2).^2+(x(2)-0.5).^2-1] );
   % la matrice Jacobienne Fp(x)
   %   : argument est un vecteur (x,y)
   %   : retourne une matrice 2x2
   Fp = @(x)( [2.*x(1),   -1; ...
               2.*(x(1)-2), 2.*x(2)-1] );
   % la fonction ||F(x,y)||
   %   : argument est un vecteur (x,y)
   %   : retourne un scalaire
   nF = @(x)( norm( F(x) ) );
   % représenter nF via mesh
   X = 0:0.1:2;
   Y = 0:0.1:2;
   for i = 1 : length(X)
       for j = 1 : length(Y)
           Z(i,j) = nF( [X(i);Y(j)] );
       end
   end
   mesh(Y, X, Z );
   % les deux points sont aux alentoures
   % de (1.5,1.5), et (1,0)
   maxit = 10;
   res = 1e-12;
   [x it r] = newton(F, Fp, [1.5;1.5], res, maxit)
   [x it r] = newton(F, Fp, [1; 0], res, maxit)
```

Notons que le code de l'instruction `newton` utilisée ici est le même que celui de la séance précédente ; vous le trouverez également plus bas dans ce corrigé. Un tel recyclage est possible grâce, entre autres, à une subtilité dans l'encodage de l'itération

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})},$$

qui est codée comme

```
x = x - fp(x)\f(x)
```

au lieu d'utiliser une variante plus «habituelle»

```
x = x - f(x)/fp(x)
```

Les deux instructions sont équivalentes dans le cas d'une seule équation non linéaire (car `x`, `f(x)` et `fp(x)` sont des scalaires), mais c'est bien la première variante qui correspond à l'itération vectorielle

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - F'(\mathbf{x}^{(k)})^{-1}F(\mathbf{x}^{(k)})$$

En effet, dans ce cas `x`, `f(x)` sont des vecteurs, `fp(x)` est une matrice (la matrice Jacobienne) et `fp(x)\f(x)` correspond à la résolution d'un système linéaire.

```
function [x it r] = newton(f, fp, x0, res, maxit)
% X = newton(F,FP,X0,RES,MAXIT)
%    détermine un zéro de la fonction F(X) d'une ou de
%    plusieurs variables avec X0 comme point de départ
% arguments:
%    F - la fonction dont on détermine un zéro
%    FP - la dérivée de F
%    X0 - approximation initiale du zéro
%    RES - la méthode s'arrête si |F(X)| < RES
%    MAXIT - le nombre maximal d'itérations
% sortie:
%    X - l'approximation calculée du zéro
%    IT - nombre d'itérations utilisées
%    R - valeur de |F(X)|
x = x0;
for it = 1:maxit
    r = norm(f(x));
    if(r < res)
        it--;
        return;
    elseif(fp(x) == 0)
        error(['La méthode a obtenu une dérivée nulle en x = ',
           num2str(x)]);
    end
    x = x-fp(x)\f(x);
end
```