Système d’équations non linéaires

A. Ramadane, Ph.D.

Les phénomèmes non linéaires sont très fréquents dans les applications.

Le problème consiste à trouver le ou les vecteurs $x = (x_1, x_2, x_3, \ldots, x_n)$ vérifiant les n équations non linéaires suivantes :

$$\left\{ \begin{array}{lcl} f_1(x_1, x_2, x_3, \ldots, x_n) & = & 0 \\ f_2(x_1, x_2, x_3, \ldots, x_n) & = & 0 \\ f_3(x_1, x_2, x_3, \ldots, x_n) & = & 0 \\ \vdots & = & \vdots \\ f_n(x_1, x_2, x_3, \ldots, x_n) & = & 0 \end{array} \right.$$

Forme compacte d'un système non linéaire

$$F(x) = 0$$

où $x = (x_1, x_2, x_3, \ldots, x_n)$ et $F = (f_1, f_2, f_3, \ldots, f_n)$

Résidu

$$R(x) = -F(x)$$

Matrice jacobienne

$$J(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

1. Etant donné une approximation initiale $x^{(0)}$
2. Résoudre le système linéaire

$$J(x^{(k)})\,\delta x = -F(x^{(k)})$$

3. Mettre à jour la solution

$$x^{(k+1)} = x^{(k)} + \delta x$$

4. Si $\frac{\|\delta x\|}{\|x^{(k+1)}\|} = \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k+1)}\|} < \epsilon_1$ et/ou $\|F(x^{(k+1)})\| < \epsilon_2$, la convergence est atteinte.

Remarques

- l'étape 1 de l'algorithme s'écrit aussi

$$x^{k+1} = x^k + J(x^k)^{-1} R(x^k)$$

  soulignant le lien avec la méthode de Newton en dimension 1.
- L'algorithme de Newton n'est pas applicable si $\det J(x^0) = 0$
- On peut montrer que si l'algorithme converge alors en général

$$\|x - x^{n+1}\| \approx C \|x - x^n\|^2 \Leftrightarrow \|e^{n+1}\| \approx C \|e^n\|^2$$

  Donc en général la convergence est quadratique.
- Si la matrice jacobienne est singulière au point solution x ($\det J(x) = 0$) alors on perd la convergence quadratique. C'est l'analogue du cas en dimension 1 : on perd la convergence quadratique si la racine est de multiplicité supérieure à 1 ($f'(r) = 0$).

Exemple

Considérons le système non linéaire

$$\begin{cases} 4xy - 2x &= 0 \\ x^2 - xy^2 &= 0 \end{cases} \tag{1}$$

$$x^0 = (1/2, -1), \quad R(x^0) = \begin{pmatrix} -3 \\ -1/4 \end{pmatrix}, \quad J(x^0) = \begin{pmatrix} -6 & 2 \\ 0 & 1 \end{pmatrix}$$

$$J(x^0)\delta = -R(x^0)$$

$$\delta = \begin{pmatrix} -5/12 \\ 1/4 \end{pmatrix}$$

$$x^1 = x^0 + \delta = \begin{pmatrix} 1/12 \\ -3/4 \end{pmatrix}$$

Soit le système d'équations non-linéaires:

$$4x^2 + y^2 = 4;$$

$$x - y^2 = -1.$$

(a) Déterminer graphiquement le nombre de solutions de ce système.

(b) Écrire le système linéaire à résoudre pour trouver une approximation de (x, y) de façon itérative. Ne pas résoudre le système.

(c) Expliquer pourquoi il serait contre-indiqué d'utiliser le point $(-1, 0)$ comme approximation initiale.

(d) Est-ce qu'on pourrait contourner le problème en démarrant avec l'approximation initiale $(-1+\varepsilon, 0+\varepsilon)$, où ε est la précision machine? Pour vous aider à répondre à cette question, trouver une borne supérieure de l'erreur relative $\frac{\|\vec{e}\|_\infty}{\|\vec{x}\|_\infty}$.

(a) 3 solutions.

(b) Le vecteur correction $\vec{\delta}$ est solution du système linéaire

$$\begin{pmatrix} 8x^0 & 2y^0 \\ 1 & -2y^0 \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} = -\begin{pmatrix} 4(x^0)^2 + (y^0)^2 - 4 \\ x^0 - (y^0)^2 + 1 \end{pmatrix}.$$

On a ensuite $x \simeq x^0 + \delta_x$ et $y \simeq y^0 + \delta_y$.

(c) La matrice jacobienne évaluée en la solution $(-1\,,\,0)$ n'est pas pas inversible.

(d) On a $\|A\|_\infty = 8 - 6\epsilon$ et $\|A^{-1}\|_\infty = \frac{4{,}5-4\epsilon}{\epsilon(7-8\epsilon)}$. Avec$\epsilon = 2^{-23}$, on a $\mathrm{Cond}_\infty A \simeq 2,3 \times 10^{16}$, la matrice est mal conditionnée.

On considère le système non linéaire suivant :

$$\begin{cases} x^2 + y^2 + z^2 & = & +1; \\ y^2 - z^2 & = & +1; \\ y - z^2 & = & -1. \end{cases}$$

(a) Écrire le système linéaire à résoudre à chaque itération lorsqu'on utilise la *méthode de Newton.*

(b) i. Montrer que ce système d'équations algébriques a $\vec{x} = (0 \quad -1 \quad 0)^\mathrm{T}$ comme solution. Donner une interprétation géométrique à cette solution.

ii. Nous avons utilisé la fonction `sysnl` de la bibliothèque numérique du cours pour résoudre ce sytème d'équations, en prenant comme approximation initale $\vec{x}_{(0)} = (1 \ \ 1 \ \ 1)^\mathrm{T}$. Nous observons les résultats numériques suivants :

ii. Nous avons utilisé la fonction `sysnl` de la bibliothèque numérique du cours pour résoudre ce sytème d'équations, en prenant comme approximation initale $\vec{x}_{(0)} = (1 \ \ 1 \ \ 1)^T$. Nous observons les résultats numériques suivants :

```
Arguments initiaux :
-------------------
Nombre maximal d'iterations : nmax =       15
Critere d'arret :             epsilon =  1.000000E-06
Differences finies :          h =  1.000000E-03

Iter.                    x_i                        ||R(x_i)||

  0    1.000000E+00   1.000000E+00   1.000000E+00   2.4495E+00
  1   -3.500000E+00   3.000000E+00   2.500000E+00   2.6653E+01
  2   -8.357143E-01   2.200000E+00   1.890000E+00   8.1235E+00
  3    3.185964E+00   2.011765E+00   1.741763E+00   1.6231E+01
  4    6.513373E-01   2.000046E+00   1.732091E+00   6.4246E+00
  5   -4.280240E+00   2.000000E+00   1.732051E+00   2.4320E+01
  6   -1.439225E+00   2.000000E+00   1.732051E+00   8.0714E+00
  7    1.364843E+00   2.000000E+00   1.732051E+00   7.8628E+00
  8   -1.515633E+00   2.000000E+00   1.732051E+00   8.2971E+00
  9    1.221555E+00   2.000000E+00   1.732051E+00   7.4922E+00
 10   -1.845108E+00   2.000000E+00   1.732051E+00   9.4044E+00
 11    7.033674E-01   2.000000E+00   1.732051E+00   6.4947E+00
 12   -3.913513E+00   2.000000E+00   1.732051E+00   2.1316E+01
 13   -1.190181E+00   2.000000E+00   1.732051E+00   7.4165E+00
 14    1.925533E+00   2.000000E+00   1.732051E+00   9.7077E+00
 15   -5.952434E-01   2.000000E+00   1.732051E+00   6.3543E+00

  Il n'y a pas de convergence apres 15 iterations.
```

Expliquer pourquoi on observe ce comportement. Est-ce qu'on pourrait obtenir la convergence de l'algorithme en changeant de condition initiale? Pourquoi?

iii. Est-ce qu'on pourrait améliorer la performance de la méthode de Newton en utilisant plutôt `h = 1.000000E-12`? Pourquoi?

(c) Un programmeur/numéricien inexpérimenté a programmé son implémentation de la méthode de Newton en utilisant l'évaluation numérique des composantes de la matrice Jacobienne. Voici une partie de son programme MATLAB :

```
for i =1: nbeq
  acc = zeros(nbeq,1);
  acc(i) = h;
  jac(:,i) = (feval(f,x0 + acc) - feval(f,x0))./h;
end
```

Identifier la méthode numérique utilisée et évaluer le choix qui a été fait. Proposer une alternative qui sera « meilleure » et réécrire la commande MATLAB qui doit être modifiée.

(a) La matrice jacobienne et le vecteur résidu s'écrivent:

$$J(x,y,z) = \begin{pmatrix} 2x & 2y & 2z \\ 0 & 2y & -2z \\ 0 & 1 & -2z \end{pmatrix} \quad \vec{R}(x,y,z) = \begin{pmatrix} x^2+y^2+z^2-1 \\ y^2-z^2-1 \\ y-z^2+1 \end{pmatrix}$$

On doit résoudre $\quad J(x^0,y^0,z^0)\vec{\delta} = -\vec{R}(x^0,y^0,z^0)$.

(b) Le point $[0 \quad -1 \quad 0]^T$ est le point d'intersection des 3 coniques.

(c) i. L'approximation initiale $\vec{x}^0 = [1 \quad 1 \quad 1]^T$ n'est pas dans le bassin d'attraction de la solution.

ii. le choix de $h = 10^{-12}$, pourrait entrainer l'élimination de chiffres significatifs par la soustraction de nombres voisins lors du calcul de la matrice jacobienne.

(d) La formule aux différences utilisée est d'ordre 1, la formule aux différences d'ordre 2 est plus appropriée. Le programme modifié est le suivant:

```
for i =1: nbeq
  acc = zeros(nbeq,1);
  acc(i) = h;
  jac(:,i) = (feval(f,x0 + acc) - feval(f,x0 - acc) )./(2*h);
end
```