

Problèmes

Parallelisation



By Professeur
Cédric Bilat
Cedric.Bilat@he-arc.ch

Fractale Newton

Contexte

Problème

Soit $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ définit par

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}$$

avec la notation

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 \quad (*)$$

On s'intéresse au zéros de f , ie au point $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$ tel que

$$f(x) = 0$$

Autrement dit-on s'intéresse aux solutions du système **non linéaire** de 2 équations à 2 inconnues suivants :

$$(S) \begin{cases} f_1(x_1, x_2) = 0 & I \\ f_2(x_1, x_2) = 0 & II \end{cases}$$

Algorithme

L'algorithme de Newton solutionne se type de problème quel que soit la dimension de $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Cet algorithme est itératif. Notons

$$x^0, x^1, x^2, \dots, x^i, x^{i+1}, \dots$$

cette suite d'itérée.

Observation

Remarquez que *l'indice d'itération* apparait en haut, alors que *l'indice de composante* apparait en bas dans (*). L'itéré x^i est en fait dans le cas des fonctions 2D.

$$x^i = \begin{pmatrix} x_1^i \\ x_2^i \end{pmatrix} \in \mathbb{R}^2$$

Start point

x^0 suffisamment proche de la solution x^* , sinon l'algorithme de *Newton* pourrait diverger.

Itération

On passe de x^i à x^{i+1} avec

$$x^{i+1} = x^i - J_f^{-1}(x^i) f(x^i)$$

où $J_f^{-1}(x^i)$ est l'inverse de la **jacobienne** de f en x^i

Critère arrêt

(A1) Arrêter l'algorithme lorsque les itérés x^i ne bougent plus suffisamment :

$$\frac{\|x^i - x^{i+1}\|^2}{\|x^{i+1}\|^2} < \varepsilon_x$$

(A2) Arrêter lorsque l'itéré x^i est une solution suffisamment bonne du système d'équations (S) :

$$\frac{\|f(x^i) - f(x^{i+1})\|^2}{\|f(x^{i+1})\|^2} < \varepsilon_f$$

Rappel

Mathématique

(R1) Inverse d'une matrice 2x2

Soit

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

On a

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

avec

$$\det(A) = |A| = ad - bc$$

(R2) Jacobienne d'une fonction

Soit $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ et $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$. On a

$$J_f(x) = \begin{pmatrix} \partial_1 f_1 & \partial_2 f_1 \\ \partial_1 f_2 & \partial_2 f_2 \end{pmatrix}$$

avec

$$\partial_1 f = \partial_1 f(x) = \frac{\partial f(x)}{\partial x_1} = \frac{\partial f(x_1, x_2)}{\partial x_1}$$

Applications

Contexte

Soit $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ définit par

$$f(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix} = \begin{pmatrix} x_1^3 - 3x_1x_2^2 - 1 \\ x_2^3 - 3x_1^2x_2 \end{pmatrix}$$

Avec la notation

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$$

On s'intéresse au zéros de f , ie au point $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$ tel que $f(x) = 0$. Autrement dit-on s'intéresse aux solutions du système **non linéaire** de 2 équations à 2 inconnues suivants :

$$(S) \begin{cases} \overbrace{x_1^3 - 3x_1x_2^2 - 1}^{f_1(x)} = 0 & I \\ \underbrace{x_2^3 - 3x_1^2x_2}_{f_2(x)} = 0 & II \end{cases}$$

Ce système possède trois solutions que l'on note :

$$x_a^* = (1 \ 0)^t$$

$$x_b^* = (-1/2 \ \sqrt{3}/2)^t$$

$$x_c^* = (-1/2 \ -\sqrt{3}/2)^t$$

On laisse au lecteur vérifier que c'est bien juste !

TODO

Appliquer l'algorithme de Newton en utilisant tous les points de départs possibles dans

$$[-2, 2] \times [-2, 2]$$

Si pour le point de départ choisit l'algorithme converge vers

- x_a^* , colorier le en **noir**
- x_b^* , colorier le en **gris**
- x_c^* , colorier le en **blanc**

Indications

- (I1) En associant une couleur a tous les points dans

$$[-2, 2] \times [-2, 2]$$

selon la règle ci-dessus, vous devriez voir apparaître une image de type **fractale** :

le *fractal de Newton* !

- (I2) Le critère d'arrêt est à votre bon sens ! Il faut peut-être prendre les critères originaux de l'algorithme de Newton et les combiner avec les idées ci-dessous propres à la nature du problème qui nous intéresse ici !

Idées :

- (S1) Arrêter lorsque l'itéré x^i est suffisamment proche d'une des solutions x_a^*, x_b^*, x_c^* , ie lorsque l'on se trouve dans un disque de rayon epsilon autour de l'une de ces solutions. Posons x^* l'une de ces solution. Arrêter lorsque :

$$\frac{\|x^i - x^*\|^2}{\|x^*\|^2} < \varepsilon_{x^*}$$

- (S2) Arrêter lorsque les itérés x^i restent suffisamment proche, suffisamment longtemps de l'une des solutions x^* . En effet, les itérés x^i peuvent passer près d'une des solutions x^* , mais converger finalement vers une autre solution x^* . « Un satellite passe près d'une planète, mais atterrit finalement sur une autre ». Pour éviter ceci :

$$\frac{\|x^s - x^*\|^2}{\|x^*\|^2} < \varepsilon_{x^*} \quad \forall s \in [i_1, i_2] = E \quad \text{avec} \quad |E| = M$$

- (S3) Mélanger les idées précédentes !

Animations

On s'inspire ici des idées d'animation de Mandelbrot et Julia.

- On se fixe un paramètre N .
- On effectue N itérations exactement.
- On regarde parmi x_a^* x_b^* x_c^* quel est le point le plus proche du dernier élément x^N de notre suite. On colorie en conséquence.

L'animation s'effectue ensuite en faisant varier le paramètre N représentant la longueur de la suite, que l'on utilise.

Note : On peut imaginer des animations sur d'autres principes. On peut jouer par exemple aussi avec les paramètres suivants :

- \mathcal{E}_x
- \mathcal{E}_f
- \mathcal{E}_{x^*}
- M

Qui aura la plus belle animation ?!

Contraintes

(C1) On doit pouvoir zoomer sur le fractale

Implémentation

Conseils

(C1) Utiliser la version fonctionnelle de l'API Image fournie.

Validation

Effectuer les variations suivantes :

- Taille de l'image (rectangulaire, pas carrée !)
- dg et db

Pour taille de l'image prenez par exemple

```
int dw = 16 * 80;  
int dh = 16 * 60;
```

Pour les contraintes à satisfaire sur dg et db , utilisez

```
Devices ::printAll() ;
```

Speedup

Mesurer les coefficients de *speedup* des différentes implémentations.

Pour canevas, utiliser le document

speedup_simple.xls.

Au besoin adapter ce canevas.

End
