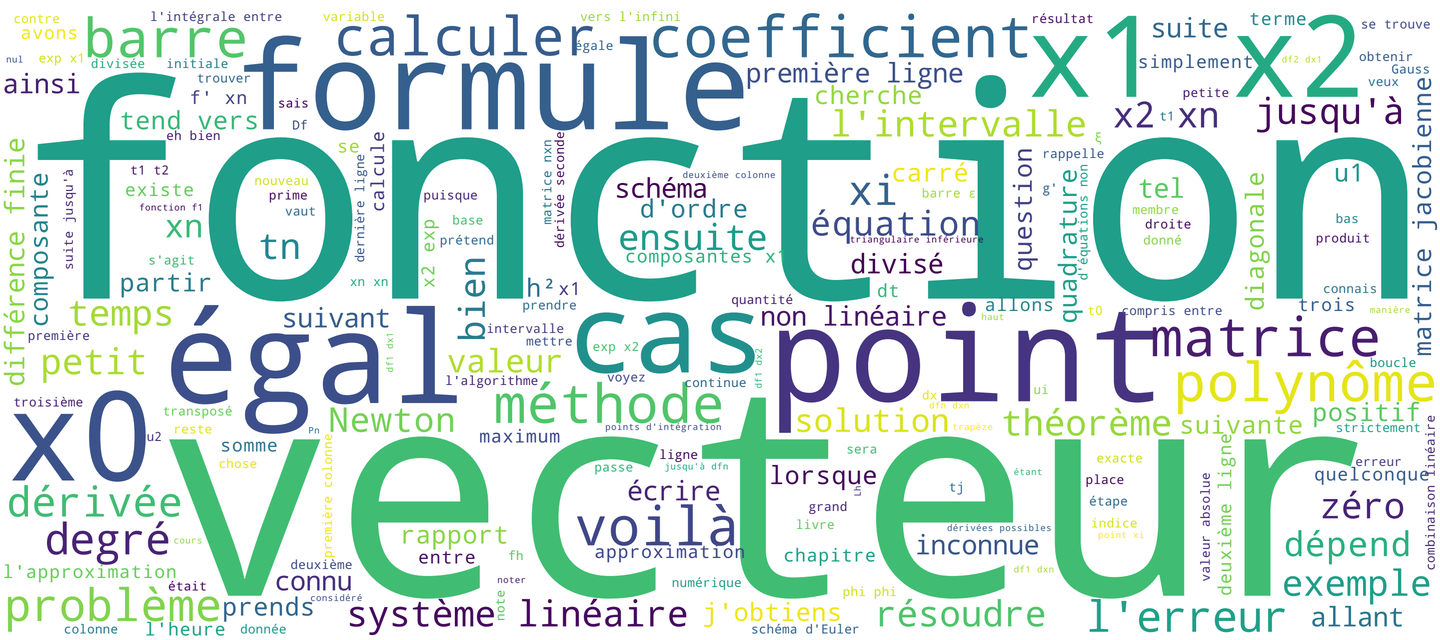
Chapitre 8 : Systèmes d'équations non linéaires

Introduction à l'analyse numérique

Prof. Marco Picasso



Search MOOC

Video

EPFL

Chap 8 - Systèmes d'équations non linéaires

$\vec{x} \in \mathbb{R}^N$ tq $\vec{f}(\vec{x}) = \vec{0}$ $\qquad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$ $\qquad \vec{f} : \mathbb{R}^N \to \mathbb{R}^N,\ \vec{x} \to \vec{f}(\vec{x})$ $\qquad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1\, x_2 \cdots x_N) \\ f_2(x_1\, x_2 \cdots x_N) \\ \vdots \\ f_N(x_1\, x_2 \cdots x_N) \end{pmatrix}$

Exemple: $N=2$ $\quad (x_1, x_2)$ tq $\begin{cases} 2x_1 - x_2 + e^{x_1} = 0 \\ -x_1 + 2x_2 + e^{x_2} = 0 \end{cases}$ $\qquad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 - x_2 + e^{x_1} \\ -x_1 + 2x_2 + e^{x_2} \end{pmatrix}$

Newton ?: $N=1$ $\qquad x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$ $\qquad f'(x)$

Je veux maintenant étendre la méthode de Newton au cas des systèmes d'équations non linéaires. Le problème que je souhaite résoudre maintenant est le suivant : je cherche x dans R^n, donc un vecteur à n composantes x1, x2,..., xn tel que f vecteur de x vecteur est égal à 0. Ici x c'est le vecteur de composantes x1, x2,..., xn et f c'est une fonction de R^n dans R^n qui à x retourne f vecteur de x vecteur et f(x) c'est, la première composante c'est une application f1 qui dépend de x, c'est-à-dire x1, x2,... xn, f2 qui dépend de x1, x2,..., xn jusqu'à la dernière composante qui est une fonction fn qui dépend de x1, x2,..., xn. Exemple dans le cas n=2 : je cherche (x1, x2) tel que 2.x1 - x2 + exp(x1) = 0 et -x1+2.x2 + exp(x2) = 0 Dans ce cas-là le vecteur x c'est bien évidemment le vecteur de composantes (x1, x2) et f(x) c'est f1 qui dépend de x1 et de x2 et f2 qui dépend de x1 et de x2. f1 c'est la première ligne de l'équation c'est 2.x1 - x2 + exp(x1) première équation, et f2 c'est -x1 + 2.x2 + exp(x2) Comment écrire la méthode de Newton ? Réécrivons la méthode dans le cas n=1, une équation à une inconnue. On avait écrit xn+1 = xn - f(xn)/f'(xn) Ici on avait une fonction f d'une variable et on calcule la dérivée f'.

Notes

Summary

0m 01s

Chap 8 - Systèmes d'équations non linéaires

$\vec{x} \in \mathbb{R}^N$ tq $\vec{f}(\vec{x}) = \vec{0}$ $\quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$ $\quad \vec{f} : \mathbb{R}^N \to \mathbb{R}^N, \ \vec{x} \to \vec{f}(\vec{x})$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1, x_2 \cdots x_N) \\ f_2(x_1, x_2 \cdots x_N) \\ \vdots \\ f_N(x_1, x_2 \cdots x_N) \end{pmatrix}$

Exemple : $N=2$ (x_1, x_2) tq $\begin{cases} 2x_1 - x_2 + e^{x_1} = 0 \\ -x_1 + 2x_2 + e^{x_2} = 0 \end{cases}$ $\quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 - x_2 + e^{x_1} \\ -x_1 + 2x_2 + e^{x_2} \end{pmatrix}$

Newton ? : $N=1$ $\quad x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$ $\quad f'(x) \longrightarrow Df(\vec{x})$ matrice $N \times N$ jacobienne

$$Df(\vec{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}) & \frac{\partial f_1}{\partial x_2}(\vec{x}) & \cdots & \frac{\partial f_1}{\partial x_N}(\vec{x}) \\ \frac{\partial f_2}{\partial x_1}(\vec{x}) & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{pmatrix}$$

Ex : $N=2$

Alors maintenant le problème c'est que l'on a affaire à une fonction f qui est une fonction f1 de n variables, f2 de n variables,..., fn de n variables. Je peux calculer par exemple df1/dx1, df1/dx2, df1/dxn, df2/dx1 et ainsi de suite jusqu'à dfn/dxn. Il y a donc n fois n dérivées que je peux calculer à partir de cette fonction f(x). Je vais mettre ces n fois n dérivées dans ce que je vais noter Df(x) qui est la matrice nxn, n lignes n colonnes, que j'appelle la matrice jacobienne. Df en un point x quelconque c'est donc la matrice qui contient toutes les dérivées possibles, la première c'est df1/dx1 évaluée en (x1, x2,..., xn) c'est-à-dire x vecteur, ensuite df1/dx2 évaluée en x jusqu'à df1/dxn évaluée en x. J'ai donc ici calculé toutes les dérivées possibles à partir de cette fonction f1 qui dépendait de (x1, x2,..., xn). Je fais la même chose pour la deuxième ligne : df2/dx1 évaluée en x, df2/dx2, jusqu'à df2/dxn. Et je fais la même chose jusqu'à la dernière ligne donc je vais avoir dfn/dx1, dfn/dx2, jusqu'à dfn/dxn qui est la dérivée de fn par rapport à la dernière variable. J'ai bien ici une matrice nxn. Dans le cas de l'exemple ici, n=2, dans le cas du petit exemple que j'ai considéré ici.

Notes

Summary

2m 47s

Chap 8 - Systèmes d'équations non linéaires

$\vec{x} \in \mathbb{R}^N$ tq $\vec{f}(\vec{x}) = \vec{0}$ $\quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$ $\quad \vec{f} : \mathbb{R}^N \to \mathbb{R}^N,\ \vec{x} \to \vec{f}(\vec{x})$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1\, x_2 \cdots x_N) \\ f_2(x_1\, x_2 \cdots x_N) \\ \vdots \\ f_N(x_1\, x_2 \cdots x_N) \end{pmatrix}$

Exemple: $N=2$ (x_1, x_2) tq $\begin{cases} 2x_1 - x_2 + e^{x_1} = 0 \\ -x_1 + 2x_2 + e^{x_2} = 0 \end{cases}$ $\quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 - x_2 + e^{x_1} \\ -x_1 + 2x_2 + e^{x_2} \end{pmatrix}$

Newton ?: $N=1$ $\quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ $\quad f'(x) \longrightarrow D\vec{f}(\vec{x})$ matrice $N \times N$ jacobienne

$$D\vec{f}(\vec{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}) & \frac{\partial f_1}{\partial x_2}(\vec{x}) & \cdots & \frac{\partial f_1}{\partial x_N}(\vec{x}) \\ \frac{\partial f_2}{\partial x_1}(\vec{x}) & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{pmatrix} \qquad \text{Ex : } N=2 \quad D\vec{f}(\vec{x}) = \begin{pmatrix} 2+e^{x_1} & -1 \\ -1 & 2+e^{x_2} \end{pmatrix}$$

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

N qque $\quad D\vec{f}(\vec{x}^n)(\vec{x}^n - \vec{x}^{n+1}) = \vec{f}(\vec{x}^n)$ $\quad \vec{x}^n = \begin{pmatrix} x_1^n \\ x_2^n \\ \vdots \\ x_N^n \end{pmatrix}$

J'ai f(x), voilà son expression : (f1(x1,x2),f2(x1,x2)) et donc je peux calculer les dérivées, Df(x), j'ai une matrice 2x2, je dois dériver la première ligne par rapport à x1, je trouve 2 + exp(x1), quand je dérive la première ligne par rapport à x2, je trouve -1 et sur la deuxième ligne j'ai : -1 et 2 + exp(x2) Je sais maintenant calculer la matrice jacobienne de la fonction f évaluée au point x. Revenons à la méthode de Newton maintenant. J'avais dit dans le cas n=1, je vais écrire cette méthode de Newton de manière un peu différente, je vais écrire que f'(xn).(xn-xn+1) = f(xn) une écriture équivalente à cette écriture ici. Dans le cas n quelconque, n équations, n inconnues, à la place de f'(xn) j'ai la matrice jacobienne évaluée en x^n, ici le vecteur x^n c'est le vecteur de composantes (x1)^n, (x2)^n,...,(xN)^n Evidemment je vais partir de x^0, donc (x1)^0, (x2)^0,..., (xn)^0 Revenons à l'écriture, j'ai ici la dérivée fois l'incrément. Notez ici que j'ai mis l'indice itération en haut alors que tout à l'heure il était en bas tout simplement parce que l'indice du bas est réservé aux composantes. Donc Df(x^n).(x^n - x^(n+1)) égal f(x^n). Vous avec ici une matrice, ici un vecteur, et ici un vecteur.

Notes

Summary

4m 57s

Chap 8 - Systèmes d'équations non linéaires

$\vec{x} \in \mathbb{R}^N$ tq $\vec{f}(\vec{x}) = \vec{0}$ $\quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$ $\quad \vec{f} : \mathbb{R}^N \to \mathbb{R}^N$, $\vec{x} \to \vec{f}(\vec{x})$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1, x_2 \cdots x_N) \\ f_2(x_1, x_2 \cdots x_N) \\ \vdots \\ f_N(x_1, x_2 \cdots x_N) \end{pmatrix}$

Exemple: $N=2$ (x_1, x_2) tq $\begin{cases} 2x_1 - x_2 + e^{x_1} = 0 \\ -x_1 + 2x_2 + e^{x_2} = 0 \end{cases}$ $\quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ $\quad \vec{f}(\vec{x}) = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \begin{pmatrix} 2x_1 - x_2 + e^{x_1} \\ -x_1 + 2x_2 + e^{x_2} \end{pmatrix}$

Newton ?: $N=1$ $\quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ $\quad f'(x) \longrightarrow Df(\vec{x})$ matrice $N \times N$ jacobienne

$$Df(\vec{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1}(\vec{x}) & \frac{\partial f_1}{\partial x_2}(\vec{x}) & \cdots & \frac{\partial f_1}{\partial x_N}(\vec{x}) \\ \frac{\partial f_2}{\partial x_1}(\vec{x}) & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_N} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_N}{\partial x_1} & \frac{\partial f_N}{\partial x_2} & \cdots & \frac{\partial f_N}{\partial x_N} \end{pmatrix}$$

Ex : $N=2$ $\quad Df(\vec{x}) = \begin{pmatrix} 2+e^{x_1} & -1 \\ -1 & 2+e^{x_2} \end{pmatrix}$

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

N qque $\quad \underbrace{Df(\vec{x}^n)}_{\text{connu}} (\underbrace{\vec{x}^n}_{\text{connu}} - \underbrace{\vec{x}^{n+1}}_{\text{inconnue}}) = \vec{f}(\vec{x}^n)$ $\quad \vec{x}^n = \begin{pmatrix} x_1^n \\ x_2^n \\ \vdots \\ x_N^n \end{pmatrix}$

Donc matrice nxn fois vecteur n composantes égal vecteur à n composantes. Ensuite vous avez, si x^n est connu, la matrice jacobienne est calculable donc ceci est connu, x^n est connu par contre x^(n+1) est l'inconnue et, à nouveau, f(x^n) est connue si x^n est connu. Donc il s'agit ici de résoudre un système linéaire à n équations et n inconnues. A chaque étape de la méthode de Newton, on va résoudre un système linéaire de n équations à n inconnues ce qui nous permettra d'obtenir x^(n+1). L'algorithme est le suivant : je pars d'un x^0 donné qui est l'approximation de x tel que f(x) = 0, mon système d'équations non linéaires. Ensuite, pour n=0,1,2..., je vais calculer le second membre du système linéaire que je vais noter b, b c'est f(x^n) x^n étant connu, je peux calculer f(x^n), c'est mon second membre, je vais calculer la matrice du système linéaire que je note A, A c'est Df(x^n) qui est calculable dès l'instant où je connais x^n, je vais résoudre le système linéaire Ay=b et ensuite je vais poser que x^(n+1), j'ai que y c'est x^n - x^(n+1) donc x^(n+1) c'est x^n - y. Et éventuellement je vais mettre un test d'arrêt donc si y est plus petit qu'une certaine valeur, par exemple 10^(-8), je sors de la boucle, je m'arrête j'ai trouvé une très bonne approximation de mon système non linéaire.

Notes

Summary

7m 08s