Algorithmique numérique (LU3IN028)

Cours n°4 : Résolution d'équations non linéaires

Stef Graillat

Sorbonne Université

Résumé du cours précédent

Algorithme d'optimisation en dimension $n \geq 2$

- Algorithme de recherche directe (recherche par motif et Nelder-Meade)
- Algorithme de gradient
- Algorithme de Newton
- Algorithme de Quasi-Newton
- Algorithme du gradient conjugué non linéaire
- Algorithme de moindres carrés non linéaires

Objectifs

1. Méthode de Newton en dimension $n \geq 1$
2. Méthode d'homotopie

Bibliographie

- Scientific Computing, An Introductory Survey, Michael .T. Heath, McGraw-Hill, 2002
- Scientific Computing with Case Studies, Dianne P. O'Leary, SIAM, 2009
- Numerical Methods, D. Faires et R. Burden, 3e édition, Brooks/Cole, 2002

Méthode de Newton

Équations non linéaires

- Étant donnée une fonction f, on cherche une valeur x pour laquelle

$$f(x) = 0$$

- Une solution x est une racine de l'équation ou un zéro de la fonction f.

Équations non linéaires (suite)

Deux cas importants :

- Une seule équation avec une seule inconnue, où $f : \mathbb{R} \to \mathbb{R}$
  Une solution est un scalaire x vérifiant $f(x) = 0$

- Un système de n équations à n inconnues, où $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$
  Une solution est un vecteur x tel que $\mathbf{f}(x) = 0$

Exemples d'équations non linéaires

- Exemple d'équation non linéaire en dimension 1

$$x^2 - 4\sin(x) = 0$$

pour laquelle $x = 1.9$ est une solution approchée

- Exemple d'un système d'équations non linéaires en dimension 2

$$\begin{aligned} x_1^2 - x_2 + 0.25 &= 0 \\ -x_1 + x_2^2 + 0.25 &= 0 \end{aligned}$$

pour lequel $x = [0.5\ ;\ 0.5]^T$ est une solution

Existence et unicité des solutions

- L'existence et l'unicité de solution sont plus compliquées à montrer dans le cas non linéaire que dans le cas linéaire
- Si $f : \mathbb{R} \to \mathbb{R}$ est continue et si signe$(f(a)) \neq$ signe$(f(b))$ alors le Théorème des Valeurs Intermédiaires nous dit qu'il existe $x^* \in [a, b]$ tel que $f(x^*) = 0$
- Il n'y a pas d'analogue en dimension supérieure.

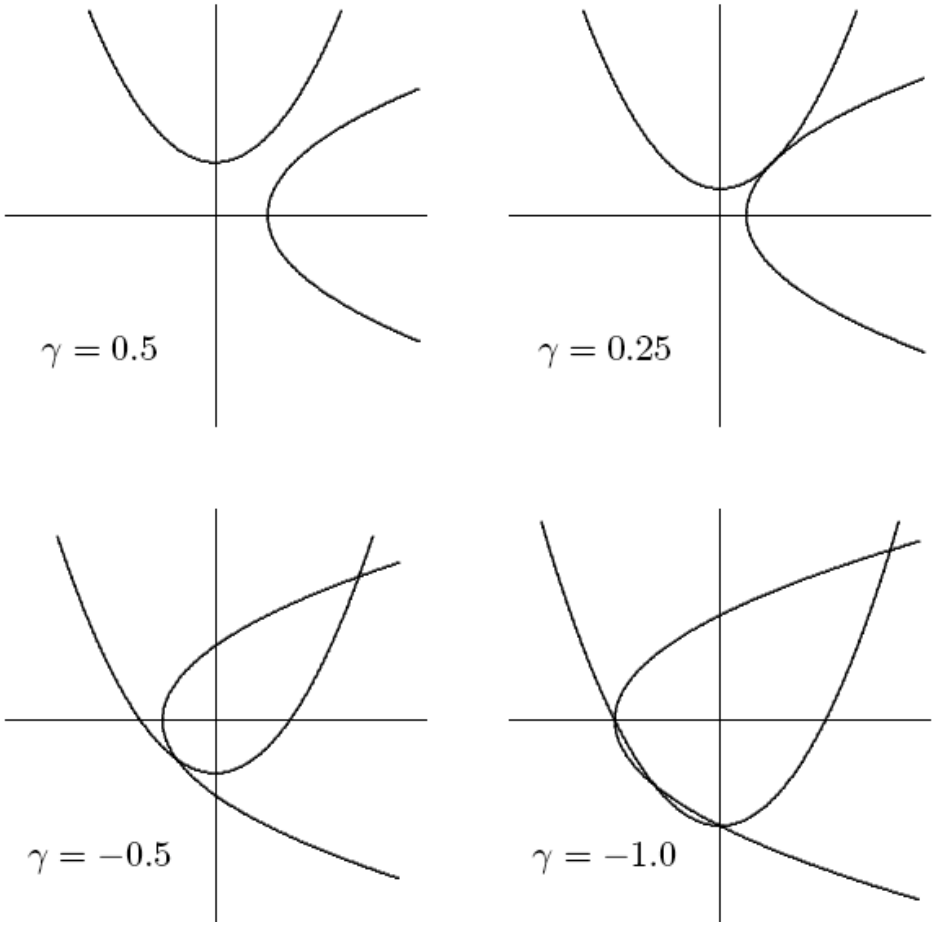
Exemple en dimension 1

Les équations non linéaires peuvent avoir un nombre quelconque de solutions :

- $e^x + 1 = 0$ n'a pas de solution
- $e^{-x} - x = 0$ a une solution
- $x^2 - 4\sin(x) = 0$ a deux solutions
- $x^3 + 6x^2 + 11 - 6 = 0$ a trois solutions
- $\sin(x) = 0$ a une infinité de solutions

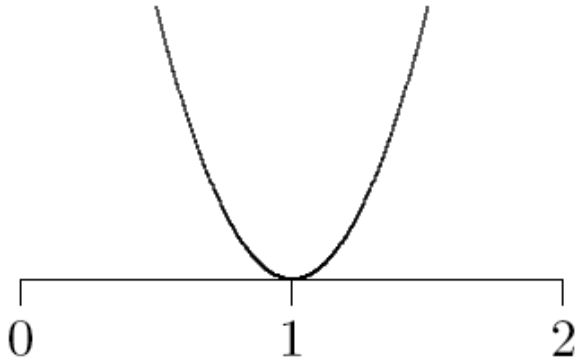
Exemple en dimension 2

$$\begin{aligned} x_1^2 - x_2 + \gamma &= 0 \\ -x_1 + x_2^2 + \gamma &= 0 \end{aligned}$$

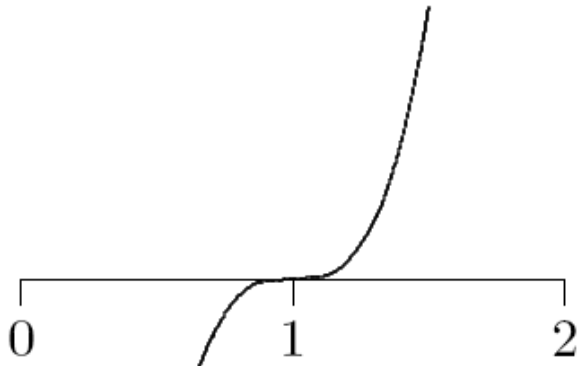


Multiplicité

- Si $f(x^*) = f'(x^*) = f''(x^*) = \cdots = f^{(m-1)}(x^*) = 0$ mais $f^{(m)}(x^*) \neq 0$, alors on dit que x^* est une racine de multiplicité m



$x^2 - 2x + 1$



$x^3 - 3x^2 + 3x - 1$

- Si $m = 1$, on dit que x^* est une racine simple

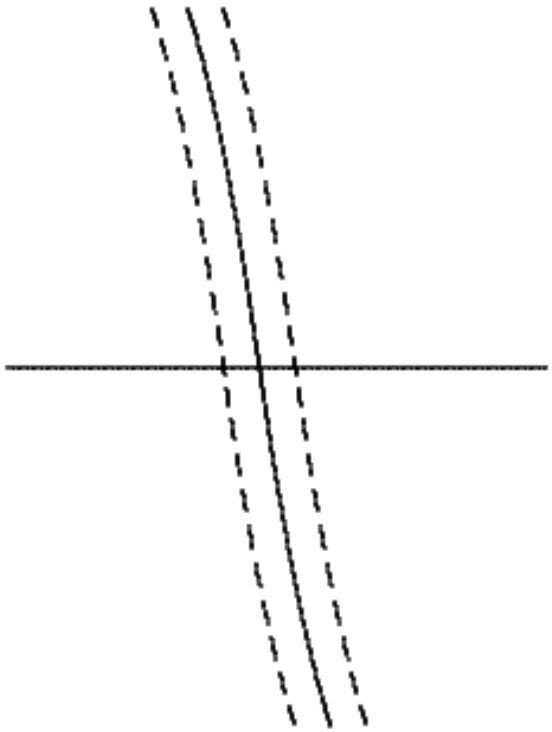
Sensibilité et conditionnement

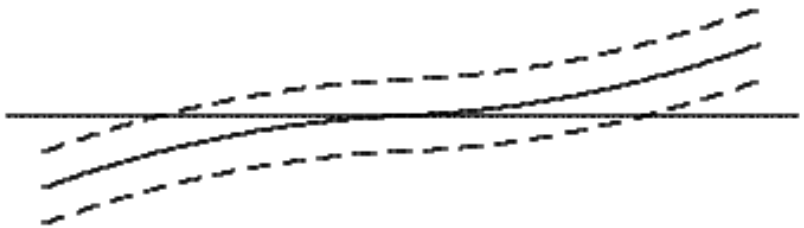
- Le conditionnement du calcul de racines est l'inverse de celui de l'évaluation de fonction
- Le conditionnement absolu du calcul d'une racine x^* de $f : \mathbb{R} \to \mathbb{R}$ est $1/|f'(x^*)|$
- Le calcul est mal conditionné quand la tangente est proche de l'horizontale
- En particulier, les racines multiples sont mal conditionnées
- Le conditionnement absolu du calcul d'une racine x^* de $f : \mathbb{R}^n \to \mathbb{R}^n$ est $\|J_f^{-1}(x^*)\|$ où J_f est la matrice jacobienne de f,

$$J_f(x) = \{\partial f_i(x)/\partial x_j\}$$

- Une racine est mal conditionnée si la matrice jacobienne est presque singulière

Sensibilité et conditionnement (suite)





Sensibilité et conditionnement (suite)

- Que signifie une solution approchée $\widehat{x}$ de l'équation ?

$$\|f(\widehat{x})\| \approx 0 \qquad \text{ou} \qquad \|\widehat{x} - x^*\| \approx 0 ?$$

- La première correspond à un ‘‘petit résidu’’, la seconde mesure la distance à la solution exacte
- Un petit résidu implique une solution approchée avec une bonne précision seulement si le problème est bien conditionné

Taux de convergence

- Pour une méthode itérative générale, on définit l'erreur à l'itération k par

$$e_k = x_k - x^*$$

où x_k est une approximation de la solution et x^* est la solution

- On dit que la suite (x_k) converge avec un taux r si

$$\lim_{k \to +\infty} \frac{\|e_{k+1}\|}{\|e_k\|^r} = C$$

pour une constance $C \geq 0$

Taux de convergence (suite)

Quelques cas particuliers intéressants

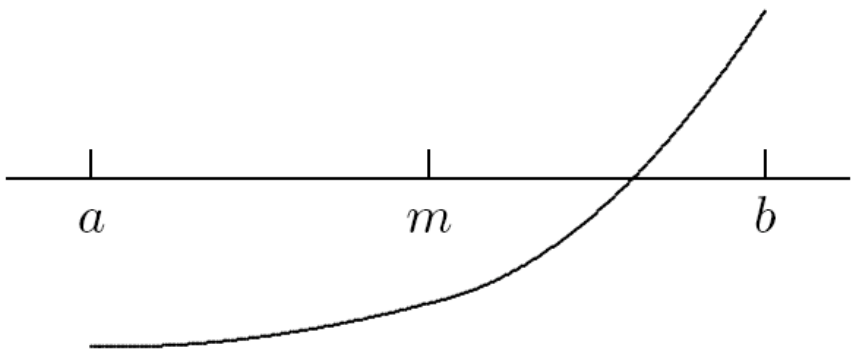
- $r = 1$: linéaire
- $r > 1$: superlinéaire
- $r = 2$: quadratique

Taux de convergence	Gain en chiffre par itération
linéaire	constant
superlinéaire	augmente
quadratique	double

Dichotomie

Le principe de la méthode par dichotomie est de partir d'un intervalle qui contient une solution et l'on divise ensuite par deux sa longueur jusqu'à ce qu'on ait isolé la solution avec une précision suffisante

while $(b-a) > tol$ **do**
    $m = a + (b-a)/2$
    if signe$(f(a))$ = signe$(f(m))$ **then**
        $a = m$
    else
        $b = m$
    end
end



Dichotomie (suite)

- L'algorithme de dichotomie converge tout le temps mais lentement
- Le taux de convergence est $r = 1$ et $C = 0.5$
- On gagne 1 bit de précision à chaque itération

Méthode de Newton

- Développement de Taylor à l'ordre 1

$$f(x+h) \approx f(x) + hf'(x)$$

  Il s'agit d'une fonction linéaire en h approximant f au voisinage de x

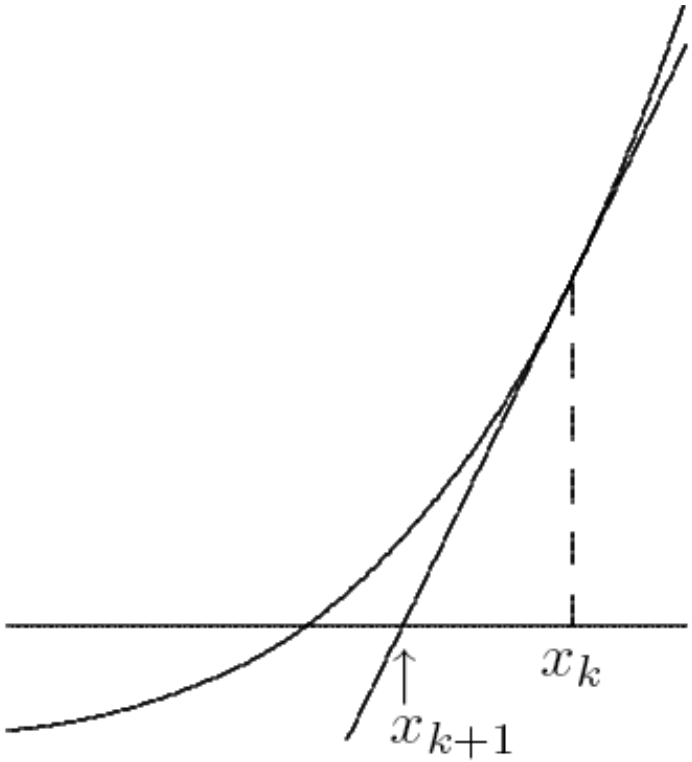
- On remplace la fonction non linéaire f par cette fonction linéaire dont la racine est $h = -f(x)/f'(x)$

- Les racines de la fonction f et de l'approximation linéaire ne sont en général pas identiques donc on itère le processus

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Méthode de Newton (suite)

La méthode de Newton approxime une fonction non linéaire f près de x_k par sa tangente en $f(x_k)$



Convergence de la méthode de Newton

- Si la racine x^* est simple alors la convergence est quadratique ($r = 2$)
- Mais l'itération doit commencer suffisamment près de la racine pour qu'il y ait convergence

Racines de polynômes

- Étant donné un polynôme $p(x)$ de degré n, on veut trouver ses n racines.
- Plusieurs approches existent
  - Utiliser la méthode de Newton pour trouver une racine puis faire de la déflation et continuer
  - Fabriquer la matrice compagnon et calculer ses valeurs propres
  - Utiliser des méthodes spécifiques pour calculer toutes les racines d'un polynôme (Jenkins-Traub, Durand-Kerner, etc.)

Système d'équations non linéaires

Le cas des systèmes d'équations non linéaires est beaucoup plus complexe

- une large variété de comportement possible (déterminer l'existence, le nombre de solutions ou un bon point de départ est plus complexe)
- Le temps de calcul augmente rapidement en fonction de la dimension du problème

Méthode de Newton

- En dimension n, la méthode de Newton est de la forme

$$x_{k+1} = x_k - J(x_k)^{-1} f(x_k)$$

où $J(x)$ est la matrice jacobienne de f

- En pratique, on ne calcule pas explicitement l'inverse de $J(x_k)$ mais on résout le système linéaire

$$J(x_k)s_k = -f(x_k)$$

et on choisit

$$x_{k+1} = x_k + s_k$$

Convergence de la méthode de Newton

- Le taux de convergence de la méthode de Newton est quadratique, à condition que la matrice jacobienne soit inversible
- Mais il faut commencer les itérations assez près de la solution pour converger

Coût de la méthode de Newton

Le coût par itération de la méthode de Newton pour un problème dense en dimension n est important

- Calculer la matrice jacobienne nécessite n^2 évaluations de fonctions
- Résoudre un système linéaire coûte $\mathcal{O}(n^3)$ opérations

Méthodes de type Newton

- Méthode par différence finie : si J n'est pas disponible, on peut l'estimer par différence finie

$$J(x)_{ij} \approx \frac{f_j(x + te_i) - f_j(x)}{t}$$

pour t petit et e_i le i-ième vecteur unité

- Méthode de Newton inexacte : au lieu de résoudre le système linéaire $J(x_k)s_k = -f(x_k)$ exactement, on utilise une méthode itérative pour obtenir une solution approchée (par exemple la méthode du gradient conjugé)
- Méthode de Quasi-Newton : on utilise une approximation de la jacobienne. Par exemple, la méthode de Broyden

$$B_{k+1} = B_k + \frac{(y - B_k s)s^T}{s^T s}$$

avec $y = f(x_{k+1}) - f(x_k)$ et $s = x_{k+1} - x_k$. On a alors $B_{k+1}s = y$ (équation de la sécante) et $B_{k+1}v = B_k v$ pour $s^T v = 0$

Cas des systèmes d'équations polynomiales

Lorsque l'on a à résoudre un système d'équations polynomiales, il y a d'autres méthodes très efficaces

- les méthodes basées sur le calcul de résultants
- les méthodes basées sur le calcul de bases de Gröbner

Voir le livre : Mathématiques L3 - Mathématiques appliquées, Jacques-Arthur Weil, Alain Yger, Pearson Education, 2009

Méthodes d'optimisation

Supposons que le système $f(x) = 0$ s'écrive sous la forme

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0, \\ f_2(x_1, \ldots, x_n) = 0, \\ \vdots \\ f_n(x_1, \ldots, x_n) = 0. \end{cases}$$

Le système a une solution si et seulement si la fonction

$$g(x_1, \ldots, x_n) := \sum_{i=1}^{n} f_i(x_1, \ldots, x_n)^2$$

admet 0 comme minimum

→ on peut donc utiliser les algorithmes d'optimisation du cours précédent pour minimiser g

Méthodes d'homotopie

Méthodes d'homotopie

- On veut résoudre $f(x) = 0$
- On connaît la solution d'un problème simple $f_a(x) = 0$ (on prendra $f_a(x) = f(x) - f(a)$ pour un vecteur constant a)
- On introduit le problème

$$\rho_a(\lambda, x) = \lambda f(x) + (1-\lambda) f_a(x) = f(x) + (\lambda - 1) f(a)$$

où λ est un nombre réel appartenant à l'intervalle $[0;1]$

- La solution de l'équation $\rho_a(0, x) = 0$ est connue et la solution de l'équation $\rho_a(1, x) = 0$ est la solution de notre problème initial

Méthodes d'homotopie (suite)

Algorithme de base :

Initialiser $\lambda = 0$ et x = solution de l'équation $f_a(x) = 0$
tant que $\lambda < 1$

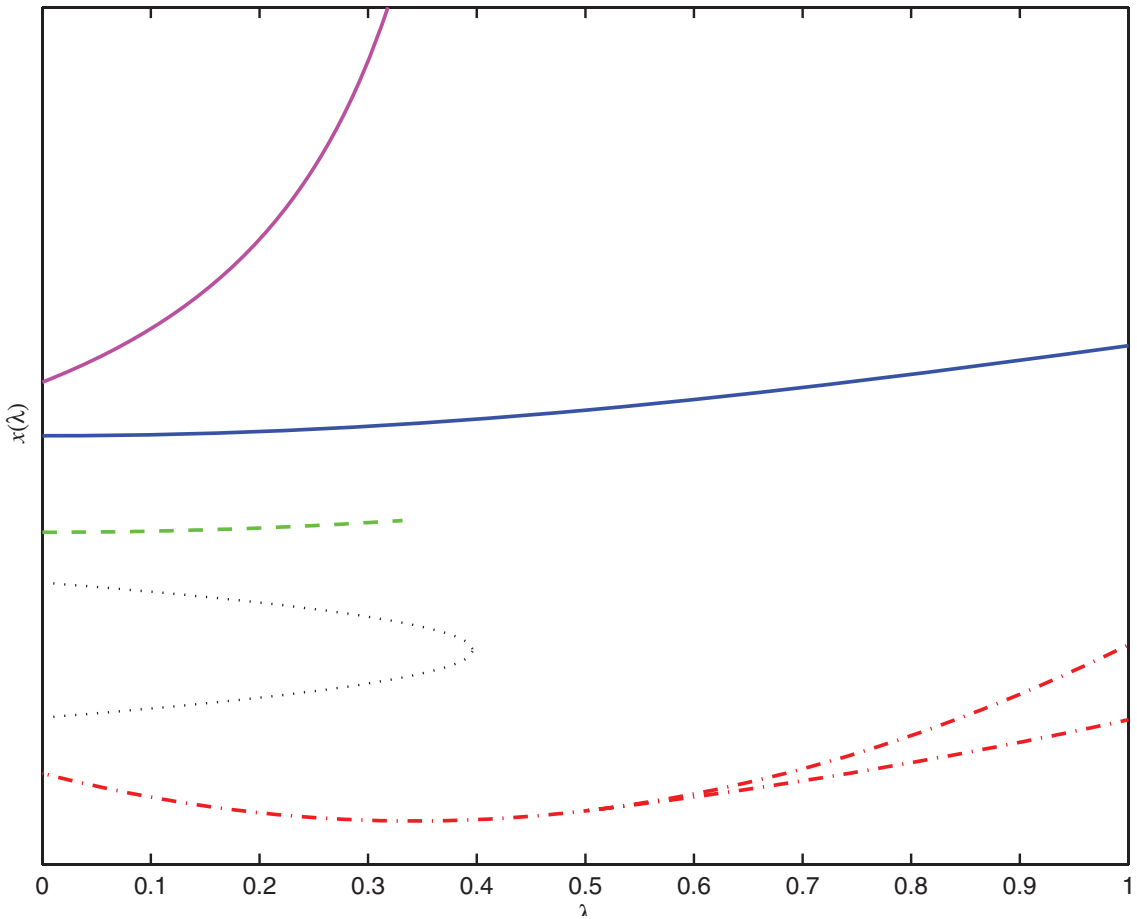
- augmenter légèrement λ
- résoudre l'équation $\rho_a(\lambda, x) = 0$ en utilisant la solution précédente comme point de départ pour résoudre le nouveau problème

fin

- On espère qu'il existe une solution pour chaque problème intermédiaire
- Comme on modifie légèrement λ à chaque étape, on espère aussi que la solution $x(\lambda)$ ne change que légèrement si bien que la nouvelle équation peut être résolue facilement à chaque itération

Méthodes d'homotopie (suite)

Problèmes :



- Points de retournement
- Points de bifurcation
- Points d'arrêt (la solution n'existe plus pour un $\lambda < 1$)
- Divergence vers l'infini

Méthodes d'homotopie (suite)

- La difficulté de la méthode est de construire ρ_a tel que nous pouvons suivre la solution de $\lambda = 0$ à $\lambda = 1$ sans la perdre

- Une fonction w est dite transverse à 0 sur un ouvert U si pour tout point $u \in U$ tel que $w(u) = 0$, la matrice jacobienne de w en u est de rang plein

- Théorème de Sard paramétré : Soit $U = \mathbb{R}^n \times [0,1[\times \mathbb{R}^n$ et $\rho : U \to \mathbb{R}^n$ de classe C^2 transverse à 0 sur U. Soit a un point de $\mathbb{R}^n$ et $\rho_a(\lambda, z) = \rho(a, \lambda, z)$. Alors la fonction ρ_a est transverse en à 0 sur $[0,1[\times \mathbb{R}^n$ pour presque tout a

Méthodes d'homotopie (suite)

Théorème : Si en plus des hypothèses précédentes, l'équation $\rho_a(0, z) = 0$ a une unique solution z_0. Alors pour presque tout $a \in \mathbb{R}^n$, il existe une courbe γ solution de $\rho_a(\lambda, z) = 0$ partant de $(0, z_0)$ ayant les propriétés suivantes

- La matrice jacobienne de ρ_a est de rang plein
- La courbe γ est lisse, ne s'intersecte pas avec elle-même et n'intersecte aucune des autres courbes solutions
- La courbe a une longueur finie sur tout compact de $[0, 1[\times \mathbb{R}^n$
- Soit la courbe atteint l'hyperplan $\lambda = 1$ soit elle diverge à l'infini

En résumé, on peut presque toujours suivre la solution de $\lambda = 0$ à $\lambda = 1$

Méthodes d'homotopie (suite)

Première méthode :

Étant donné une fonction ρ_a, on pose $\lambda = 0$ et $\widehat{x} = a$. Par conséquent $\rho_a(\lambda, \widehat{x}) = 0$

Jusqu'à ce que $\lambda = 1$ faire

- augmenter légèrement λ
- résoudre l'équation $\rho_a(\lambda, x) = 0$ en utilisant un algorithme (par exemple la méthode de Newton, etc) en prenant $\widehat{x}$ comme point de départ
- Appeler cette solution $\widehat{x}$

fin

Si l'algorithme termine, alors on a calculé $\widehat{x}$ vérifiant $\rho_a(1, \widehat{x}) = 0$ et donc $f(\widehat{x}) = 0$

Problème : choix du pas d'incrémentation de λ et de la précision de la solution pour le solveur

Méthodes d'homotopie (suite)

Deuxième méthode : via la résolution d'équations différentielles

On a

$$\rho_a(\lambda, x) = \lambda f(x) + (1-\lambda) f_a(x) = f(x) + (\lambda - 1) f(a)$$

- La solution x dépend de λ. Par conséquence $\rho_a(\lambda, x(\lambda)) = 0$
- En différentiant par rapport à λ, on obtient

$$\frac{\partial \rho_a(\lambda, x(\lambda))}{\partial \lambda} + \frac{\partial \rho_a(\lambda, x(\lambda))}{\partial x} x'(\lambda) = 0$$

- Un calcul montre que

$$\frac{\partial \rho_a(\lambda, x(\lambda))}{\partial \lambda} = f(a) \quad \text{et} \quad \frac{\rho_a(\lambda, x(\lambda))}{\partial x} = J_f(x(\lambda))$$

On obtient donc le système d'équations différentielles

$$J_f(x(\lambda)) x'(\lambda) = -f(a) \text{ avec } x(0) = a$$

Conclusion

Méthode de Newton

- Méthode pour calculer les racines d'équations non linéaires
- Utiliser en calcul numérique mais aussi en calcul formel (sera vu en TD)

Méthode d'homotopie

- Méthode globale