Chapitre 3

Résolution numérique des équations non linéaires

On cherche, souvent, à résoudre un système d'équations non-linéaires de la forme : $F(X) = 0$ où $F : \mathbb{R}^n \to \mathbb{R}^n$, c'est à dire on cherche à determiner les $X^* \in \mathbb{R}^n$ vérifiant $F(X^*) = 0$

3.1 Résolution d'une équation non linéaire à une inconnu

On veut résoudre une équation de la forme $f(x) = 0$ où $f : [a, b] \to \mathbb{R}$ non linéaire cad trouve x^* telle que $f(x^*) = 0$ (si f est linéaire on a évidemment $x^* = 0$). Le problème se pose quand f a une forme analytique compliquée de telle sorte que l'obtention de x^* avec des méthodes directes n'est pas évidente.

Exemple
$f_1(x) = e^x + \cos(x)$ $\quad f_2(x) = \tan(x^3) + 2x^2$

Ce pendant, il existe des méthodes numériques itératives permettent d'approcher x^* avec une suite de réel $(x^{(n)})_{n\in\mathbb{N}}$ telle que

$$\lim_{n\to+\infty} x^{(n)} = x^* \text{ et } f(x^*) = 0$$

Dans ce chapitre on va donner quelques méthodes de construction de ces suites en partant d'une valeur initiale $x^{(0)}$ et suivant l'algorithme général :

Algorithme 14 Algorithme général d'une méthode itérative pour résoudre $f(x) = 0$

Entrée: $x^{(0)}$, F formule de récurrence
Sortie: x^*
  $n \leftarrow 0$
  Tanque (condition d'arret n'est pas valide) **Faire**
    $x^{(n+1)} \leftarrow F(x^{(n)})$
    $n \leftarrow n + 1$
  Fin Tanque
  $x^* \leftarrow x^{(n)}$
  « La solution est x^* et quitter le programme »

Remarque Les méthodes numériques de résolution des équations non linéaires sont pour la majorité à convergence locale (converge si l'estimé initial $x^{(0)}$ est choisi proche de la racine de f)

3.1.1 Localisation de $x^{(0)}$

Le choix du point de départ $x^{(0)}$ joue un rôle important dans la convergence de la méthode itérative en question sur tout si sa convergence est locale. Pour cela il faut choisir soigneusement $x^{(0)}$. Parmi les techniques d'initialisation on peut citer :

- Méthode graphique.
- Tableau de variation.
- Connaissance à priori sur l'ordre de grandeur de l'inconnu x^*

3.1.2 Méthode de dichotomie (ou de la bissection)

Théorème 3.1 *Théorème des valeurs intérimaires*
Soit $f : [a, b] \to \mathbb{R}$ *une fonction continue si* $f(a)f(b) \leq 0$ $(0 \in f([a, b])$ *alors il existe au moins un point* $c \in [a, b]$ *telle que* $f(c) = 0$

Cette méthode est utilisée pour trouver un encadrement grossière de la racine d'une fonction. Son principe est simple, on suppose que $f : [a, b] \to \mathbb{R}$ est continue et qu'elle change de signe cad elle passe par zero au moins une fois $0 \in f([a, b])$. D'après la théorème précédente il existe au moins un point $c \in [a, b]$ où f s'annule. On commence par prendre $c = \frac{a+b}{2}$ si $f(c) = 0$ on a $x^* = c$, sinon il y a deux cas qui se présentent. La premiere, si $f(a)f(c) < 0$ cad $x^* \in [a, c]$, on pose $a = a$ et $b = c$ et on recommence. La deuxième possibilité $(f(a)f(c) > 0)$ on obtient $f^2(a)f(c)f(b) < 0$ comme $f^2(a) > 0$ ca donne que $f(c)f(b) < 0$ et $x^* \in [c, b]$ on pose $a = c$ et $b = b$ et on recommence.

Exercice. 3.1 *Écrire l'algorithme de dichotomie*

Supposons qu'on localiser un intervalle $[a, b]$ ou f s'annule au maximum une fois, L'algorithme de dichotomie est le suivant :

Algorithme 15 Algorithme de dichotomie (ou de la bissection)

Entrée: a, b
Sortie: x^*

$n \leftarrow 0;\ a_n \leftarrow a;\ b_n \leftarrow b$
Si $(f(a_n) == 0)$ **Alors**
« la solution est $x^* = a_n$ et quitter le programme »
Sinon
Si $(f(b_n) == 0)$ **Alors**
« la solution est $x^* = b_n$ et quitter le programme »
Sinon
Si $(f(a_n)f(b_n) > 0)$ **Alors**
« pas de solution et quitter le programme »
Sinon
Tanque (Condition d'arret n'est pas valide) **Faire**
$c_n \leftarrow \frac{a_n+b_n}{2}$
Si $(f(c_n) == 0)$ **Alors**
« la solution est $x^* = c_n$ et quitter le programme »
Sinon
Si $(f(a_n)f(c_n) < 0)$ **Alors**
$a_{n+1} \leftarrow a_n;\ b_{n+1} \leftarrow c_n$
Sinon
$b_{n+1} \leftarrow b_n;\ a_{n+1} \leftarrow c_n$
Fin Si
Fin Si
$n \leftarrow n + 1$
Fin Tanque
Fin Si
Fin Si
Fin Si
« la solution est $x^* = \frac{a_n+b_n}{2}$ et quitter le programme »

L'algorithme produit une suite (a_n, b_n). si on définie $\varepsilon_n = |a_n - b_n|$ avec $\varepsilon_0 = |a_0 - b_0|$ on obtient

$$\forall n > 0,\ \varepsilon_{n+1} = \frac{\varepsilon_n}{2} = \frac{\varepsilon_0}{2^n}$$

Donc pour localiser x^* dans un intervalle de longueur $\epsilon = \frac{1}{10^K}$ il faut $\frac{1}{10^K} = \frac{\varepsilon_0}{2^n} \Rightarrow 2^n = \varepsilon_0 10^K \Rightarrow n \log 2 = K \log 10 + \log \varepsilon_0$

$$n = K \log_2 10 + \log_2 \varepsilon_0 \text{ itérations}$$

Exercice. 3.2 *Soit* $f(x) = e^x + x$ *définie sur* $I = [-1, 1]$

1. *Verifier que* f *admet au moins une solution sur* $I = [-1, 1]$
2. *Combien d'itérations pour trouver un encadrement de* x^* *de* $\varepsilon = 10^{-3}$
3. *Exécuter deux itérations de l'algorithme de la bissection*

3.1.3 Méthode de la point fixe

Rappel Soit E un espace vectoriel, une application $d : E \to \mathbb{R}_+$ est dite distance si et seulement si : pour tous x, y et z des vecteurs de E on a :

- $d(x,y) = d(y,x)$ symétrie
- $d(x,y) = 0 \Leftrightarrow x = y$
- $d(x,y) \leq d(x,z) + d(z,y)$

Rappel Soit E est un espace métrique (muni d'une distance d), et $f : E \to E$ une fonction. f est dite lipschitzienne de rapport $k \in \mathbb{R}_+$ si pour tout $(x,y) \in E \times E$

$$d(f(x), f(y)) \leq k.d(x,y)$$

si $k \in [0,1[$ f est dite contractante ($k \in]0,1[$ strictement contractante). On rappelle que les fonctions lipschitziennes sont toujours continues.

Théorème 3.2 *Point fixe*
Soit E est un espace métrique complet. et $f : E \to E$ une fonction strictement contractante alors il existe un point unique $x^* \in E$ telle que

$$f(x^*) = x^*$$

De plus pour tout point initial $x^{(0)}$ la suite défini par :

$$\forall n \in \mathbb{N},\ x^{(n+1)} = f(x^{(n)})$$

converge vers x^*

unicité : Supposons qu'il existe deux points a et b, $a \neq b$ telle que $f(a) = a$ et $f(b) = b$ comme f est contractante ($k < 1$) $d(f(a), f(b)) = d(a,b)$ donc $\frac{d(f(a),f(b))}{d(a,b)} = 1 > k$ ce qui est contradictoire.

existence : soit $x^{(0)} \in E$ un point quelconque et $(x^{(n)})_{n\in\mathbb{N}}$ une suite définie par $x^{(n+1)} = f(x^{(n)})$. Soit p et q deux entiers telle que $q > p$ On a

$$d[x^{(q)}, x^{(p)}] \leq d[x^{(q)}, x^{(0)}] + d[x^{(p)}, x^{(0)}]$$

$$\begin{aligned} d[x^{(q)}, x^{(0)}] &\leq d[x^{(q)}, x^{(q-1)}] + d[x^{(q-1)}, x^{(0)}] \\ &\leq kd[x^{(q-1)}, x^{(q-2)}] + d[x^{(q-1)}, x^{(0)}] \\ &\leq k^{q-1} d[x^{(1)}, x^{(0)}] + d[x^{(q-1)}, x^{(0)}] \end{aligned}$$

Donc $d[x^{(q)}, x^{(0)}] \leq \sum_{l=0}^{q-1} k^l d[x^{(1)}, x^{(0)}] = d[x^{(1)}, x^{(0)}] \frac{1-k^q}{1-k}$ et par suite

$$d[x^{(q)}, x^{(p)}] \leq \frac{2d[x^{(1)}, x^{(0)}]}{1-k}$$

ce qui éprouve que $(x^{(n)})_{n\in\mathbb{N}}$ est une suite de Cauchy donc convergente et comme E est complet dans elle converge dans E, f est continue

$$\lim_{n\to+\infty} f(x^{(n)}) = f(\lim_{n\to+\infty} x^{(n)}) = f(x^*) = x^*$$

Le problème $f(x) = 0$ peut s'écrire sous forme d'une autre problème équivalente par exemple si on pose $f(x) = g(x) - x$ avec g une fonction bien choisie alors résoudre $f(x) = 0$ revient à résoudre $g(x) = x$ cad trouver le point fixe de g par l'algorithme suivant :

Algorithme 16 Résolution de $f(x) = 0$ par la méthode de point fixe

Entrée: $g, x^{(0)}$
Sortie: x^*
  Tanque (Condition d'arrêt n'est pas valide) **Faire**
    $x^{(n+1)} \leftarrow g(x^{(n)}$
    $n \leftarrow n + 1$
  Fin Tanque
  « la solution est $x^* = x^{(n)}$ et quitter le programme »

Exercice. 3.3 *Soit $a \in \mathbb{R}_+^*$, proposer des fonctions g_i pour résoudre $x^2 = a$ en utilisant la méthode de point fixe.*

On peut choisir $g_1 = \frac{a}{x}$, $g_2(x) = 2x - \frac{a}{x}$ et $g_3(x) = \frac{x}{2} + \frac{a}{2x}$

Théorème 3.3 *Soit $f : [a, b] \to [a, b]$ continûment derivable et supposons que*

$$\forall\ x \in [a, b],\ |g'(x)| \leq K < 1$$

alors la fonction g admet un point fixe $x^* \in [a, b]$ et la suite défini $(x^{(n)})_{n\in\mathbb{N}}$ par $x^{(n+1)} = g(x^{(n)}$ converge vers x^* pour tout $x^{(0)}$

Exercice. 3.4 *Reprenons l'exercice précèdent* (3.3)*, sur un voisinage un voisinage de racine de a quelle est la fonction $(g_1, g_2,\ g_3)$ qu'on doit utiliser pour résoudre $x^2 = a$*

Remarque Pour que l'algorithme de recherche de point fixe converge il suffit que sur un intervalle $[a, b]$, la fonction g' soit borne par une quantité K inférieure à 1. Cette condition est suffisante mais non-nécessaire.

La transformation $f(x) = 0$ en un problème équivalant $g(x) = x$ est toujours possible et celui ci avec une infinite de possibilité. Il est toujours souhaitable de bien choisir celle pour la quelle l'algorithme converge

3.1.4 Méthode de Newton

Soit f une fonction de classe C^1 au voisinage d'un point $x^{(n)}$ donnée donc peut rapprocher la courbe de f par l'equation de la tangente à ce point

$$f(x^{(n)} + h) = f(x^{(n)}) + hf'(x^{(n)})$$

si on pose $x^{(n+1)} = x^{(n)} + h$, $x^{(n+1)}$ le point ou la fonction f s'annule (intersection avec l'axe de abscisse) alors on trouve $h = -f(x^{(n)})/f'(x^{(n)})$, partant de $x^{(0)}$ on défini la suite $(x^{(n)})_{n\in\mathbb{N}}$

$$x^{n+1} = x^n - \frac{f(x^n)}{f'(x^n)} \tag{3.1}$$

Algorithme 17 Algorithme de Newton

Entrée: $x^{(0)},\ f$
Sortie: $x^{(*)}$
  $n \leftarrow 0$
  Tanque (Condition d'arrêt n'est valide) **Faire**
    Si $(f'(x^{(n)}) \neq 0)$ **Alors**
      $x^{(n+1)} \leftarrow x^{(n)} - \frac{f(x^{(n)})}{f'(^{(n)})}$
      $n \leftarrow n + 1$
    Sinon
      « pas de solution et quitter le programme »
    Fin Si
  Fin Tanque

3.1.5 Méthode de la sécante

Dans la méthode de Newton on suppose que f est derivable et que le calcul de $f'(x^{(n)})$ est possible. Pour cela nous utilisons une approximation de sa dérive au point $x^{(n)}$:

$$f'(x^n) = \frac{f(x^n) - f(x^{n-1})}{x^n - x^{n-1}} \tag{3.2}$$

ce qui nécessite la connaissance de $x^{(n)}$ et $x^{(n-1)}$ pour démarre l'algorithme cette fois il faut savoir $x^{(0)}$ et $x^{(1)}$

Algorithme 18 Algorithme de la sécante

Entrée: $x^{(0)}, x^{(1)}$
Sortie: $x^{(*)}$
  $n \leftarrow 1$
  Tanque (Condition d'arrêt n'est valide) **Faire**
    Si $(f(x^n) - f(x^{n-1}) \neq 0)$ **Alors**
      $x^{(n+1)} \leftarrow x^{(n)} - \frac{f(x^{(n)})(x^n - x^{(n-1)})}{f(x^{(n)}) - f(x^{(n-1)})}$
      $n \leftarrow n + 1$
    Sinon
      « pas de solution et quitter le programme »
    Fin Si
  Fin Tanque

3.1.6 Méthode de la corde

Cette méthode est obtenue en remplacent $f'(x^{(n)})$ par une constante q et la formule de récurrence sera

$$x^{n+1} = x^n - \frac{f(x^n)}{q} \tag{3.3}$$

On peut choisir $q = f'(x^{(0)})$ ou $q = \frac{f(b)-f(a)}{b-a}$

3.2 Résolution des systèmes d'équations non linéaires

Le but de cette section est de trouver un vecteur $x \in \mathbb{R}^n$

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

qui vérifie pour tout i de 1 à n, $f_i(x) = f_i(x_1, x_2, \ldots, x_n) = 0$, avec les $f_i : \mathbb{R}^n \to \mathbb{R}$ et telle que il existe au moins une fonction f_i non linéaire. On dit que le système à résoudre c'est un système de n équations non-linéaires à n inconnus, on note par $(\mathcal{E})$ ce système :

$$(\mathcal{E}) : \begin{cases} f_1(x) = f_1(x_1, x_2, \ldots, x_n) & = 0 \\ f_2(x) = f_2(x_1, x_2, \ldots, x_n) & = 0 \\ \vdots & \vdots \\ f_n(x) = f_n(x_1, x_2, \ldots, x_n) & = 0 \end{cases}$$

Si on défini une fonction $F \triangleq \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} : \mathbb{R}^n \to \mathbb{R}^n$ par :

$$\forall x \in \mathbb{R}^n,\ F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix}$$

Alors résoudre $(\mathcal{E})$ c'est trouve le vecteur $x \in \mathbb{R}^n$ telle que $F(x) = 0_{\mathbb{R}^n}$

3.2.1 Méthode de Newton-Raphson

Un développement limité à l'ordre 1 de la fonction F au voisinage de $x =^t (x_1, ..., x_n)$

$$F(x + h) = F(x) + DF(x)h$$

avec $h =^t (h_1, \ldots, h_n) \in \mathbb{R}^n$ et

$$DF = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} : \text{Matrice Jacobienne de F}$$

Supposons que $x+h$ est le point de R^n ou la fonction F s'annule c.a.d $F(x) = \overrightarrow{0}_{\mathbb{R}^n}$ $[DF(x)]^{-1}$ existe alors on obtient

$$h = -[DF(x)]^{-1}.F(x)$$

3.2.2 Variantes de la méthode Newton-Raphson

L'inconvénient majeur de la méthode Newton Raphson est d'avoir calculer à chaque itération la matrice Jacobienne de F au point $x^{(k)}$: qui contient n^2 coefficients, ensuite le calcul de son inverse. ce qui augmente la complexité de cette méthode. Pour surmonter ce problème plusieurs variantes de la méthode de Newton existent :

3.2.2.1 Newton-Raphson par différence fini

Cette méthode consiste à approximer la derivée partille de f_i par rapport la j-iemme composante $(i = 1, ..n;\ j = 1, ..n)$ par :

$$\frac{\partial f_i}{\partial x_j}(x_1, ..x_n) = \frac{f_i(x_1, ., x_j + h, ., x_n) - f_i(x_1, ., x_j, ., x_n)}{h}$$

Où h est un réel fixé suffisamment petit

3.2.2.2 Newton-Raphson simplifié

C'est une méthode qui consiste à conserver la matrice jacobienne de F constante pendant un certain nombre d'itérations.

Exemple Supposons qu'on va exécuter $N = p.K$ itérations de l'algorithme de Newton Raphson, Pour $k = 0, p, 2p, 3p, ..., K-1$ on calcule $DF(x^{(k)})$ et on prend ensuite $DF(x^{(k)}) = DF(x^{(k+1)}) = DF(x^{(k+2)}) = .. = DF(x^{(k+p-1)})$

3.2.2.3 Newton-Raphson à itération linéaire

Ce sont les méthodes où on résout à chaque itération k de l'algorithme de Newton-Raphson un système linéaire obtenu à partir du développement limité $F(x^k + h) = F(x^{(k)}) + DF(x^{(k)})h$:

$$(\mathcal{E}^k) : -DF(x^{(k)})h = F(x^{(k)})$$

le résultat d'un algorithme itérative (vu dans le chapitre précèdent) tel que : Jacobi, Gauss-Seidel,.. après « r » itérations sera considère comme la solution exacte du système $(\mathcal{E}^k)$ qu'on l'appelle $h^{(k)}$. $x^{(k+1)}$ sera donnée par :

$$x^{(k+1)} = x^{(k)} + h^{(k)}$$