

# **Chapitre 1**

## **Rappel Méthodes de résolution des équations non-linéaires et du système d'équations**

## I. Introduction

La première partie de ce chapitre à pour discuter la résolution des équations non linéaires à une seule variable :

$$f(x) = 0 \quad x \in \mathbb{R}^1$$

La généralisation aux systèmes d'équations à plusieurs variables  $f(x) = 0$ ,  $x \in \mathbb{R}^1$  sera faite en deuxième partie. Ce dernier problème étant relativement plus difficile.

## II. Algorithme de recherche d'un zéro d'une fonction

Un algorithme de recherche d'un zéro d'une fonction est une méthode numérique ou un algorithme de recherche d'une valeur approchée d'un  $x$  vérifiant  $f(x) = 0$ , pour une fonction donnée  $f$ . Ici,  $x$  est un nombre réel appelé zéro de  $f$  ou lorsque  $f$  est polynomiale racine de  $f$ . Lorsque  $x$  est un vecteur, les algorithmes pour trouver  $x$  tel que  $f(x) = 0$  sont généralement appelés "algorithmes de résolution numérique d'une équation". Ces algorithmes sont une généralisation des algorithmes de recherche d'un zéro d'une fonction et peuvent s'appliquer à des équations linéaires ou non linéaires.

Les algorithmes classiques que nous allons étudier sont les suivants :

1. Méthode de Dichotomie (Bissection)
2. Méthode de Newton-Raphson
3. Méthode de la sécante
4. Méthode du point fixe

### II.1. Méthode de la Bissection

On suppose que  $f$  est continue dans  $[a, b]$  et que  $f(a).f(b) < 0$

On pose  $c = (a+b)/2$ ,

- Si  $f(c) = 0$  Alors  $p = c$
- Si  $f(a) * f(c) < 0$  on remplace  $b$  par  $c$
- Si non on remplace  $a$  par  $c$ ,

On continue cette opération jusqu'à ce qu'on trouve  $p$  avec la précision demandée.

#### II.1.1. Algorithme de bissection (ou de dichotomie.)

Cette méthode repose sur le constat que si le produit  $f(a).f(b) < 0$  est négatif, alors la fonction  $f$  s'annule au moins une fois sur l'intervalle  $[a, b]$ . Les différentes étapes de la méthode peuvent être résumées comme suit :

- Trouver une approximation de la solution de  $f(x) = 0$  dans  $[a, b]$ .
- On construit une suite d'intervalles  $([a_n, b_n])_n$  contenant la racine,
- On teste que  $a_n$  ou  $b_n$  est le milieu de l'intervalle  $[a_{n-1}, b_{n-1}]$ .

**Entrées :**  $a, b, \epsilon$  et  $N_0$       **Sortie :** la valeur approchée  $p : f(p) = 0$

1. Si  $f(a) = 0$  imprimer la solution est  $a$ . Si  $f(b) = 0$  imprimer la solution est  $b$ , aller à 10
2. Si  $f(b) * f(a) > 0$ , imprimer (pas de changement de signe). Aller à 10
3. Poser  $N = 1$
4. Tant que  $N \leq N_0$ , faire les étapes 5 à 8
5. Poser  $p = (a+b)/2$

6. Si  $f(p) = 0$  ou  $b-a/2 \leq \epsilon$ , imprimer  $p$ . Aller a 10
7. Poser  $N = N + 1$
8. Si  $f(a) * f(p) > 0$ , alors poser  $a = p$ , sinon poser  $b = p$
9. Imprimer après  $N_0$  itérations l'approximation obtenue est  $p$  et l'erreur maximale est  $b-a/2$
10. Fin

## II.2. Méthode de la sécante

La méthode de la sécante est une méthode dérivée de la méthode de Newton Raphson où l'on remplace la dérivée

$$f'(x_n) \text{ par } \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

On obtient la relation de récurrence

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

### II.2.1. Algorithme de la sécante

Trouver une solution de  $f(x) = 0$

**Entrées :** deux approximations initiales  $p_0$  et  $p_1$   $\epsilon$  (la précision désirée)

$N_0$  (le nombre maximum d'itérations)

**Sortie :** la valeur approchée de  $p$  ou un message d'échec

1. 1 poser  $N = 1$ ,  $q_0 = f(p_0)$ ,  $q_1 = f(p_1)$
2. Tant que  $N \leq N_0 + 1$ , faire les étapes 3 à 6
3. poser  $p = p_1 - q_1 (p_1 - p_0) / (q_1 - q_0)$
4. Si  $|p - p_1| \leq \epsilon$  alors imprimer  $p$ , aller à l'étape 8
5. Poser  $N = N + 1$
6. Poser  $p_0 = p_1$ ,  $q_0 = q_1$ ,  $p_1 = p$ ,  $q_1 = f(p)$
7. Imprimer la méthode a échoué après  $N_0$  itérations
8. Fin

## II.3. Méthode de Regula Falsi (fausse position)

On appelle ainsi une **méthode** de résolution algébrique (**regula**) consistant à fournir une solution approchée (**falsi**) conduisant, par un algorithme approprié tirant parti de l'écart constaté, à la solution du problème considéré ; Cette méthode est un algorithme de recherche d'un zéro d'une fonction qui combine les possibilités de la méthode de dichotomie et de la méthode de la sécante.

### II.3.1. Principe de la méthode

La méthode de la fausse position commence par deux points  $a_1$  et  $b_1$  tels que  $f(a_1)$  et  $f(b_1)$  soient de signes opposés, il en résulte d'après le théorème des valeurs intermédiaires que la fonction  $f$  possède au moins un zéro dans l'intervalle  $[a_1, b_1]$ . La méthode consiste à produire une suite décroissante d'intervalles  $[a_k, b_k]$  qui contiennent tous un zéro de  $f$ .

$$c_k = a_k - \frac{a_k - b_k}{f(a_k) - f(b_k)} f(a_k)$$

Cette formule est également employée dans la méthode de la sécante, cependant la méthode de la sécante retient systématiquement les deux derniers points calculés, alors que la méthode de la fausse position retient deux points qui encadrent certainement un zéro. D'autre part, la seule différence entre la méthode de la fausse position et la méthode de dichotomie est l'utilisation de la relation  $c_k = (a_k + b_k) / 2$ .

#### II.4. Méthode de ISSAAC Newton et Joseph Raphson (N-R)

La méthode de Newton –Raphson est une méthode numérique qui permet de trouver la valeur de variable  $x$  pour laquelle  $f(x)=0$  à partir d'une première estimation de la valeur de  $x$

Le principe consiste à construire une suite  $(x_n)_n$ ; telle que  $x_{n+1}$  soit l'intersection de la tangente à la courbe de  $f$  au point  $(x_n; f(x_n))$  avec l'axe horizontal.

La fonction  $f$  en la valeur approchée courante du zéro. Cela donne la relation récurrente suivante :

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

La méthode de Newton peut ne pas converger si on commence trop loin d'un zéro. Cependant, si elle converge, elle est beaucoup plus rapide que la méthode de dichotomie (sa complexité est quadratique). La méthode de Newton est importante parce qu'elle peut aisément se généraliser à des problèmes en dimensions supérieures.

##### II.4.1. Algorithme de Newton –Raphson

**Entrées :** une approximation initiale  $p_0$

$\varepsilon$  (La précision désirée)

$N_0$  (le nombre maximum d'itérations)

**Sortie :** valeur approchée de  $p$  ou un message d'échec

1.  $N = 1$
2. Tant que  $N \leq N_0$ ; faire les étapes 3 à 6.
3. Poser  $p = p_0 - \frac{f(p_0)}{f'(p_0)}$
4. Si  $|p - p_1| \leq \varepsilon$  " alors imprimer  $p$ , aller à l'étape 8.
5. Poser  $N = N + 1$ :
6. Poser  $p_0 = p$ :
7. Imprimer la méthode a échoué après  $N$  itérations.
8. Fin.

#### II.5. Méthode du point fixe

Il y a plusieurs façons de récrire une équation  $f(x)=0$  sous la forme  $g(x)=x$  sans que cela change les racines. Dans ce cas, on dira que la racine  $x$  de  $f(x)=0$  est un point fixe de  $g(x)$ . Pour chercher les points fixes de  $g(x)$  on peut, sous certaines conditions, utiliser l'algorithme suivant :

Trouver une solution de  $g(x) = x$

**Entrées :** une approximation initiale  $p_0$

$\varepsilon$  (la précision d' désirée)

$N_0$  le nombre maximale d'itérations

**Sortie :** valeur approchée de  $p$  ou un message d' échec

### III. Méthodes pour résoudre les systèmes d'équations

#### III.1 Méthodes directes

##### III.1.1 Méthode de Cramer

La règle de Cramer est un théorème en algèbre linéaire qui donne la solution d'un système de Cramer, c'est-à-dire un système d'équations linéaires avec autant d'équations que d'inconnues et dont le déterminant de la matrice de coefficients est non nul, sous forme de quotients de déterminants ,elle est utilisée surtout pour les systèmes de faible taille et valable pour le  $\det A \neq 0$ . Nous n'allons pas trop insister sur la méthode de Cramer car elle n'est pas très performante du point de vue du calcul effectif

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{Bmatrix}$$

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_n & a_{n2} & \dots & a_{nn} \end{vmatrix}}{\det A}, \quad x_2 = \frac{\begin{vmatrix} a_{11} & b_1 & \dots & a_{1n} \\ a_{21} & b_2 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & b_n & \dots & a_{nn} \end{vmatrix}}{\det A}, \quad \dots, \quad x_n = \frac{\begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & b_n \end{vmatrix}}{\det A}$$

##### III.2.Méthode d'inversion de la matrice

La méthode d'inversion de la matrice est apparue simple mais la recherche de l'inverse de A qui est un travail supplémentaire. ( TOUMI Abdelouaheb 2017 )

$$[A]\{x\} = \{b\} \Rightarrow \{x\} = [A]^{-1}\{b\}$$

On appelle inverse d'une matrice A d'ordre n, la matrice B, si elle existe, telle que  $BA=AB=I_n$ . Où  $I_n$  la matrice unité d'ordre n.

Une matrice est dite inversible (régulière) si, et seulement si, son déterminant est différent de zéro ; l'inverse de A, noté  $A^{-1}$ , est unique.

Pour obtenir l'inverse de A on doit passer par les étapes suivantes :

- a) On prend la transposée de A notée  $\tilde{A}$ .
- b) On en déduit l'ajointe de  $\tilde{A}$  notée  $\text{adj}(\tilde{A})$ .
- c) On divise l'ajointe par le déterminant de A noté  $(\det A)$

$$A^{-1} = \frac{\text{adj}(\bar{A})}{\det(A)}$$

### III.3. Méthode du pivot de Gauss : première approche

Nous allons décrire maintenant sur un exemple le principe de la méthode de Gauss, dans une version un peu simplifiée. On part du système

$$(E) \begin{cases} 2x + y + z = 5 & (L1) \\ x - 2y + z = -5 & (L2) \\ y + 4z = 3 & (L3) \end{cases}$$

Nous allons procéder en agissant sur les lignes du système en utilisant des combinaisons linéaires de celle-ci en vue d'obtenir dans un premier temps un système triangulaire, avec des coefficients égaux à 1 sur la diagonale, i.e. on veut arriver à un système de la forme

$$(E') \begin{cases} x + \dots y + \dots z = \dots \\ y + \dots z = \dots \\ z = \dots \end{cases}$$

A supposer que l'on soit capable d'arriver à ce stade, il suffira alors de «remonter» les calculs par substitution en partant de la valeur de  $z$  qu'on injecte pour obtenir  $y$ , puis  $x$  en injectant  $y$  et  $z$  dans la première ligne. On commence donc par garder la première équation de (E) en la divisant par 2, puis on élimine la variable  $x$  dans la seconde en la remplaçant par (L2) – (L1) ce qui donne :

$$\begin{cases} x + \frac{1}{2}y + \frac{1}{2}z = \frac{5}{2} & (L1) \\ -\frac{5}{2}y + \frac{1}{2}z = -\frac{15}{2} & (L2) \\ y + 4z = 3 & (L3) \end{cases}$$

Il faut alors diviser la nouvelle ligne (L2) par  $-5/2$ , ce qui donne

$$\begin{cases} x + \frac{1}{2}y + \frac{1}{2}z = \frac{5}{2} & (L1) \\ y + \frac{1}{5}z = 3 & (L2) \\ y + 4z = 3 & (L3) \end{cases}$$

On élimine alors  $y$  dans la dernière équation en faisant la combinaison linéaire (L3) – (L2) :

$$\begin{cases} x + \frac{1}{2}y + \frac{1}{2}z = \frac{5}{2} & (L1) \\ y + \frac{1}{5}z = 3 & (L2) \\ \frac{19}{5}z = 0 & (L3) \end{cases}$$

Evidemment, on obtient  $z = 0$  en dernière équation, ce qui donne directement  $y = 3$  (grâce à (L2)), puis  $x = 1$  dans (L1)

### IV. Méthodes itératives pour les systèmes d'équations linéaires

#### IV.1.Méthode de Jacobi

La méthode de Jacobi est une méthode itérative de résolution d'un système matriciel de forme. Pour cela, on utilise une suite  $x(k)$  qui converge vers un point fixe  $x$ , solution du système d'équations linéaires.

Elle utilise le principe du point fixe pour un système d'équations linéaires, soit le système :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n = b_n \end{cases} \tag{1}$$

Transformons le système en supposant que les éléments du pivot sont non nuls

$$\begin{cases} x_1 = (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n)/a_{11} \\ x_2 = (b_2 - a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n)/a_{22} \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ x_n = (b_n - a_{n2}x_2 - a_{n3}x_3 - \dots - a_{nn-1}x_{n-1})/a_{nn} \end{cases} \tag{2}$$

Pour résoudre le système (1) on utilise l'écriture (2) en portant les termes de droite à l'itération (k) et ceux à gauche à l'interaction (k).

$$\begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)})/a_{22} \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ x_n^{(k+1)} = (b_n - a_{n2}x_2^{(k)} - a_{n3}x_3^{(k)} - \dots - a_{nn-1}x_{n-1}^{(k)})/a_{nn} \end{cases} \tag{3}$$

En prenant une estimation initiale  $X(0) = (x_1(0), x_2(0), \dots, x_n(0))$  et en utilisant le système (3) on calcule  $X(1) = (x_1(1), x_2(1), \dots, x_n(1))$  ensuite on remplace le vecteur  $X(1)$  dans le système (3) avec  $k=1$  on calcule  $X(2)$  et continue de la même façon de calculer les vecteurs  $X(3), X(4), X(5), \dots$  jusqu'à la convergence.

#### IV.2.Méthode de Gauss-Seidel

Le principe de la méthode de Gauss-Seidel, on utilise chaque composante des quelle sera calculée. Ainsi, pour calculer la composante  $x_i^{(k+1)}$ , on utilise toutes les composantes de  $x_1^{(k+1)}$  à  $x_{i-1}^{(k+1)}$  déjà calculées à l'itération (k+1) en plus de celles  $x_{i+1}^{(k)}$  à  $x_n^{(k)}$  qui ne sont qu'à l'itération (k). On écrit donc :

$$\begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \dots - a_{1n}x_n^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)})/a_{22} \\ x_3^{(k+1)} = (b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)} - \dots - a_{3n}x_n^{(k)})/a_{33} \\ \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \\ x_n^{(k+1)} = (b_n - a_{n2}x_2^{(k+1)} - a_{n3}x_3^{(k+1)} - \dots - a_{nn-1}x_{n-1}^{(k+1)})/a_{nn} \end{cases} \tag{4}$$

En prenant une estimation initiale  $X^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)})$  et en utilisant le système (4) on calcule  $X^{(1)} = (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)})$  ensuite on remplace le vecteur  $X^{(1)}$  dans le système

(4) avec  $k=1$  on calcule  $X^{(2)}$  et continue de la même façon de calculer les vecteurs  $X^{(3)}$ ,  $X^{(4)}$ ,  $X^{(5)}$ , ... jusqu'à la convergence.

### IV.3. Critère d'arrêt de calcul pour la méthode de Jacobi et Gauss-Seidel

les calculs sont arrêtés par cette méthode lorsque la différence absolue entre deux itérations successives soit inférieure à une certaine précision  $\varepsilon$  donnée.

$$|X^{(n+1)} - X^{(n)}| < \varepsilon$$

Ici, il faut vérifier la différence pour toutes les composantes une par une.

$$|x_1^{(n+1)} - x_1^{(n)}| < \varepsilon, |x_2^{(n+1)} - x_2^{(n)}| < \varepsilon, \dots, |x_n^{(n+1)} - x_n^{(n)}| < \varepsilon$$