

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITÉ ABDERRAHMANE MIRA-BÉJAIA-ALGÉRIE
FACULTÉ DES SCIENCES EXACTES
DÉPARTEMENT DE MATHÉMATIQUES

*Mémoire présenté pour l'obtention du diplôme de Master 2
en Mathématiques*

OPTION

Analyse Mathématiques

THÈME

*Problèmes-mal posés et méthode du gradient
conjugué*

PAR

AREZKI Samir

Devant le jury:

Président : M. F. BOUHMILA

Promotrice : Mme. H. BECHIR

Examinatrice : Mme. S. MEDJBAR

Soutenu publiquement le :29-09-2021

Année universitaire : 2020/2021

Remerciements

Je tiens tout d'abord à adresser toute ma gratitude à mon encadreur de recherche, Mme **H.BECHIR** Pour avoir orienté et enrichi mon travail. Je la remercie pour sa disponibilité, ses précieux conseils ainsi que son souci du détail, qui ont abouti à la réalisation de ce mémoire.

Mes remerciements vont également aux, **Membres du jury** Pour avoir accepté d'examiner mon travail et de l'enrichir par leurs propositions. Je souhaite aussi adresser mes remerciements au, Corps professoral et administratif de l'université de Béjaia qui a contribué à la réussite de mes études universitaires. Je remercie profondément, Ma famille, mes amis ainsi que les personnes qui m'ont soutenu de près ou de loin au cours de la réalisation de ce mémoire.

Table des matières

Introduction	3
1 problèmes mal posés	6
1.1 Problèmes mal posés	7
1.2 problèmes inverses	13
1.3 Méthodes de résolution	15
1.3.1 Méthode des moindres carrés	16
1.3.2 L'équation normale	16
1.3.3 Opérateur régularisant	16
1.3.4 Méthode de Tikhonov	17
1.3.5 Méthode de Lavrentiev	18
1.3.6 Décomposition en Valeurs singulières(SVD)	20
1.3.7 La méthode de Morozov	22
1.4 Méthodes itératives	25
1.4.1 Méthodes itératives de Tikhonov	25
1.4.2 Méthode itérative de Lavrentièv	26
1.4.3 Méthode de Landweber	26
1.4.4 Méthode du gradient	28
2 La Méthode Gradient conjugué	29
2.1 Introduction en dimension finie	30
2.2 Définition générale dans les espaces de Hilbert	36
2.3 Les algorithmes	37
2.3.1 La méthode du résidu minimal (RM) et la méthode du gradient conjugué (GC).	39
2.3.2.CGEN et CGEM	40
2.4 Théorie de la régularisation pour les méthodes de type gradient conjugué	44
2.4.1. Propriétés de régularisation de RM et CGEN	45
3 Simulation de la méthode du gradient	47
3.1 Débrouillage de l'image	47
3.2 Fonction de Répartition Ponctuelle	51

3.3 Application de FRP Gaussienne	52
3.4 Produit Intérieur	53
3.5 Simulation du GC sur Matlab	54
Conclusion	60
Bibliographie	63

Introduction

Un problème inverse est une situation dans laquelle les valeurs de certains paramètres (où inconnues) d'un modèle doivent être identifiées à partir d'observations (où mesures) du phénomène. C'est également le contraire d'un problème direct. Autrement un problème inverse consiste à déterminer des causes connaissant des effets, ce problème est l'inverse de ce lui appelé problème direct, consistant à déduire les effets, les causes étant connues.

On retrouve les problèmes inverses dans de nombreux domaines scientifiques: l'imagerie médicale, le radar, la mécanique quantique, le traitement d'image (restauration d'images floues), le traitement de signal, l'ingénierie pétrolière, l'acoustique sous-marine ... etc.

En 1923, Hadamard a introduit la notion de problème bien posé. Il s'agit d'un problème dont :

- la solution existe
- la solution est unique
- la solution dépend continûment des données

Un problème qui n'est pas bien posé au sens de la définition ci-dessus est dit mal posé.

La non-existence et non-unicité de la solution d'un problème mal-posé sont sans doute des difficultés sérieuses mais qu'on peut rétablir. Cependant le manque de la continuité est plus problématique, en particulier en vue d'une solution approchée ou numérique. C'est -à-dire il ne sera pas possible (indépendamment de la méthode numérique) d'approcher d'une manière satisfaisante la solution du problème inverse car les données disponibles seront bruitées donc proches, mais différentes des données réelles.

Dans ce travail, nous nous intéressons aux méthodes de résolution des problèmes mal posés et particulièrement à la méthode du gradient conjugué.

Dans le chapitre 1, nous évoquerons les problèmes mal posés et tout ce qui s'y rapporte.

Dans le chapitre 2, nous présenterons la méthode du gradient conjugué, qui sera suivie dans le chapitre 3 d'une simulation correspondant à une situation pratique pour l'application de cette méthode. Nous clorons par une brève conclusion.

problèmes mal posés

1.1 Problèmes mal posés

Définition 1.1 (Problème bien posé)

Soit X et Y deux espaces de Banach, et $A : D(A) \subset X \rightarrow Y$ un opérateur (linéaire ou non linéaire).

$$Ax = y \quad (1.1)$$

x : inconnu qu'on cherche dans X

y : donné dans Y

Le problème (1.1) est dit bien posé si:

1. Le problème (1.1) possède une solution pour tout $y \in Y$.
2. Cette solution est unique.
3. La solution dépend continûment des données.

En général la donnée du problème est obtenue de manière expérimentale et donc n'est pas exacte. Si au lieu de y nous considérons y_δ et si x_δ est la solution qui correspond à la donnée y_δ , $\|x - x_\delta\|$ dépend continûment des données $\|y - y_\delta\|$.

Définition 1.2 Le problème (1.1) est dit bien posé au sens de HADAMARD si les trois conditions suivantes vérifiées:

- Existence d'une solution,

$$\forall y \in Y, \text{ il existe une solution } x \in X \text{ telle que : } Ax = y$$

- unicité de la solution,

$$\forall x_1, x_2 \in X, \text{ si } Ax_1 = Ax_2 = 0 \Rightarrow x_1 = x_2$$

- stabilité: La solution x dépend continûment de y , faible perturbation de y donne faible perturbation de x .

$$i.e : \forall (x_n) \subset X \text{ telle que } y_n = Ax_n \rightarrow y = Ax \text{ lorsque } n \rightarrow \infty \text{ alors } x_n \rightarrow x$$

Exemple 1.1 On considère le problème de Cauchy pour l'équation des ondes

$$(P_1) \begin{cases} \frac{\partial^2 u(x,t)}{\partial t^2} - c^2 \frac{\partial^2 u(x,t)}{\partial x^2} = 0, & x \in \mathbb{R}, t > 0 \\ u(x,0) = \Phi(x), & x \in \mathbb{R} \\ \frac{\partial u(x,0)}{\partial t} = \psi(x), & x \in \mathbb{R} \end{cases}$$

où $\Phi \in C^2$, $\psi \in C^1$

la solution générale du problème (P_1) est donnée par

$$u(x, t) = f(x + ct) + g(x - ct)$$

nous avons la condition :

$$u(x, 0) = \Phi(x), \text{ alors}$$

$$f(x) + g(x) = \Phi(x) \dots\dots\dots(1)$$

et

$$\frac{\partial u(x, t)}{\partial t} = cf'(x + ct) - g'(x - ct)$$

donc

$$\frac{\partial u(x, 0)}{\partial t} = \Psi(x)$$

donc

$$cf'(x) - cg'(x) = \Psi(x) \dots\dots\dots(2)$$

d'où

$$c[\int_0^x f'(s)ds - \int_0^x g'(s)ds] = \int_0^x \Psi(s)ds$$

donc

$$f(x) - g(x) = \frac{1}{c} \int_0^x \Psi(s)ds + k \dots\dots\dots(3)$$

où k est une constante.

$$(1) + (3) \text{ implique que } 2f(x) = \Phi(x) + \frac{1}{c} \int_0^x \Psi(s)ds + k$$

$$(1) - (3) \text{ implique que } 2g(x) = \Phi(x) - \frac{1}{c} \int_0^x \Psi(s)ds - k$$

d'où

$$f(x) = \frac{1}{2}\Phi(x) + \frac{1}{2c} \int_0^x \Psi(s)ds + \frac{k}{2} \dots\dots\dots(4)$$

$$g(x) = \frac{1}{2}\Phi(x) - \frac{1}{2c} \int_0^x \Psi(s)ds - \frac{k}{2} \dots\dots\dots(5)$$

Alors

$$f(x + ct) = \frac{1}{2}\Phi(x + ct) + \frac{1}{2c} \int_0^{x+ct} \Psi(s)ds + \frac{k}{2} \dots\dots\dots(6)$$

$$g(x - ct) = \frac{1}{2}\Phi(x - ct) - \frac{1}{2c} \int_0^{x-ct} \Psi(s)ds - \frac{k}{2} \dots\dots\dots(7)$$

donc (6) + (7) donne

$$f(x + ct) + g(x - ct) = \frac{1}{2}[\Phi(x + ct) + \Phi(x - ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} \Psi(s) ds.$$

donc la solution existe

Unicité:

Supposons que u et v deux solutions du problème (P_1)

i.e;

$$\begin{cases} \frac{\partial^2 u(x)}{\partial t^2} - c^2 \frac{\partial^2 u(x,t)}{\partial x^2} = 0, & x \in \mathbb{R}, t > 0 \\ u(x, 0) = \Phi(x), & x \in \mathbb{R} \\ \frac{\partial u(x,0)}{\partial t} = \psi(x), & x \in \mathbb{R} \end{cases}$$

et

$$\begin{cases} \frac{\partial^2 v(x)}{\partial t^2} - c^2 \frac{\partial^2 v(x,t)}{\partial x^2} = 0, & x \in \mathbb{R}, t > 0 \\ v(x, 0) = \Phi(x), & x \in \mathbb{R} \\ \frac{\partial v(x,0)}{\partial t} = \psi(x), & x \in \mathbb{R} \end{cases}$$

soit $w = u - v$

En substituant w dans le problème (P_1) , on obtient

$$(P_2) \begin{cases} \frac{\partial^2 w(x)}{\partial t^2} - c^2 \frac{\partial^2 w(x,t)}{\partial x^2} = 0, & x \in \mathbb{R}, t > 0 \\ w(x, 0) = \Phi(x), & x \in \mathbb{R} \\ \frac{\partial w(x,0)}{\partial t} = \psi(x), & x \in \mathbb{R} \end{cases}$$

donc la solution du problème (P_1) est donnée par

$$u(x, t) = 0$$

alors $u(x, t) = v(x, t)$.

Ainsi la solution du problème (P_1) est unique

Stabilité: pour $c = 1$

Soit les données $\Phi_1(x) = 0$ et $\Psi_1(x) = 0$ alors la solution du problème (P_1) est

$$u(x, t) = 0$$

Soit la donnée perturbée $\Psi_2(x) = \Psi_1(x) + \varepsilon \sin(\frac{x}{\varepsilon})$, $0 < \varepsilon < 1$ et $\phi_2(x) = 0$ donc

$$\begin{cases} \frac{\partial^2 u_2(x,t)}{\partial t^2} - c^2 \frac{\partial^2 u_2(x,t)}{\partial x^2} = 0, & x \in \mathbb{R}, t > 0 \\ u_2(x, 0) = \Phi(x), & x \in \mathbb{R} \\ \frac{\partial u_2(x,0)}{\partial t} = \varepsilon \sin(\frac{x}{\varepsilon}), & x \in \mathbb{R} \end{cases}$$

la solution du problème (\tilde{P}) est donnée par

$$u_2(x, t) = \frac{1}{2} \int_{x-t}^{x+T} \Psi_2(s) ds$$

$$u_2(x, t) = \frac{1}{2} \int_{x-t}^{x+T} \varepsilon \sin(\frac{s}{\varepsilon}) ds$$

$$u_2(x, t) = -\frac{1}{2} \varepsilon^2 \cos(\frac{s}{\varepsilon}) \Big|_{x-t}^{x+T}$$

$$u_2(x, t) = \varepsilon^2 \sin(\frac{x}{\varepsilon}) \sin(\frac{t}{\varepsilon})$$

On a

$$\|\phi_2 - \phi_1\|_\infty = \sup_{x \in \mathbb{R}} |\phi_2(x) - \phi_1(x)| = 0$$

$$\|\psi_2 - \psi_1\|_\infty = \sup_{x \in \mathbb{R}} |\psi_2(x) - \psi_1(x)| = \sup_{x \in \mathbb{R}} |\varepsilon \sin \frac{x}{2}| = \varepsilon \rightarrow 0 \text{ quand } \varepsilon \rightarrow 0$$

et

$$\begin{aligned} \|u_2 - u_1\|_\infty &= \sup_{x \in \mathbb{R}} |u(x, t)_2 - u_1(x, t)| = \sup_{x \in \mathbb{R}} |\varepsilon^2 \sin(\frac{x}{2}) \sin(\frac{t}{2})| \quad t > 0 \\ &= |\varepsilon^2 \sin(\frac{t}{2})| \leq \varepsilon^2 \rightarrow 0 \text{ quand } \varepsilon \rightarrow 0. \end{aligned}$$

donc une faible perturbation donne une faible perturbation de la solution u la solution dépend continûment des données (u stable).
Ainsi que le problème (P_1) est bien posé

Exemple 1.2 Le calcul de la série de Fourier à coefficients approchés dans la métrique de ℓ^2 . Considérons une série de Fourier que nous écrivons:

$$f(x) = \sum_{n=0}^{+\infty} a_n \cos(nx)$$

et supposons que chaque coefficient a_n soit entaché d'une erreur $\frac{\varepsilon}{n}$ (à l'exception de a_0 évidemment) nous écrivons donc

$$c_n = a_n + \frac{\varepsilon}{n}, c_0 = a_0 \text{ et } \varepsilon > 0$$

c_n donnée perturbée.

Soit

$$f_2(x) = \sum_{n=0}^{+\infty} c_n \cos(nx)$$

Calculons la distance entre c_n et a_n dans ℓ^2

$$\sum_{n=0}^{+\infty} (c_n - a_n)^2 = \sum_{n=0}^{+\infty} \left(\frac{\varepsilon}{n}\right)^2 = \varepsilon^2 \sum_{n=0}^{+\infty} \frac{1}{n^2}$$

alors

$$\|c_n - a_n\|_{l_2} \rightarrow 0 \text{ quand } \varepsilon \rightarrow 0$$

Si on calcule la distance entre f_1 et f_2 dans $C[0, \pi]$

$$\|f_1 - f_2\| = \sup_{x \in [0, \pi]} \left| \sum_{n=0}^{+\infty} (c_n - a_n) \cos(nx) \right|$$

$$\|f_1 - f_2\| = \sup_{x \in [0, \pi]} \left| \sum_{n=1}^{+\infty} \frac{\varepsilon}{n} \cos(nx) \right|$$

Nous remarquons que:

$$|f_1(0) - f_2(0)| = \varepsilon \sum_{n=1}^{+\infty} \frac{1}{n}$$

Or $\sum_{n=1}^{+\infty} \frac{1}{n}$ est divergente.

donc $|f_1(0) - f_2(0)|$ ne tend pas vers 0 quand $\varepsilon \rightarrow 0$

et puisque $\|f_1 - f_2\| \geq |f_1(0) - f_2(0)|$ alors $\|f_1 - f_2\|$ ne tend pas vers 0 quand $\varepsilon \rightarrow 0$
Donc le problème est mal posé (instable).

Exemple 1.3 Soit le problème

$$(P_2) \begin{cases} u''(x) + u(x) = 0 \\ u(0) = 0 \\ u(\pi) = 1 \end{cases}$$

la solution générale du problème (P_2) est donnée par:

$u(x) = c_1 \cos(x) + c_2 \sin(x)$, en tenant compte des conditions aux limites on trouve:

$u(x) = c_1 \sin(x)$ d'où problème (P_2) admet une infinité de solutions alors le problème (P_2) est mal posé.

1.2 Problèmes inverses

Un problème inverse est une situation dans laquelle les valeurs de certains paramètres (ou inconnues) d'un modèle doivent être identifiées à partir d'observations (ou mesures) du phénomène. En quelques sortes c'est le contraire d'un problème direct. Le problème inverse consiste à remonter le schéma : connaissant les observations, le but est de retrouver les valeurs des paramètres.

La résolution du problème inverse passe donc en général initialement par une modélisation du phénomène, dite problème direct qui décrit comment les paramètres du modèle se traduisent en effets observables expérimentalement. En suite, à partir des mesures obtenues sur le phénomène réel, la démarche va consister à approximer au mieux les paramètres qui permettent de rendre compte de ces mesures. Cette résolution peut se faire par simulation numérique ou de façon analytique. La résolution mathématique est rendue difficile par le fait que les problèmes inverses sont en général des problèmes mal-posés, c'est-à-dire que les seules observations expérimentales ne suffisent pas à déterminer parfaitement tous les paramètres du modèle. Il est donc nécessaire d'ajouter des contraintes ou de l'a priori qui permettent de réduire l'espace des possibilités de façons à aboutir à une unique solution.

On trouve des problèmes inverses dans nombreux domaines scientifiques, nous pouvons entre autres citer :

1. L'imagerie médicale (échographie, scanners,...),
2. L'ingénierie pétrolière (Identification de perméabilité, magnétisme,...)
3. L'hydrologie
4. La chimie (Détermination des constantes de réaction)
5. Le radar (Détermination de la forme d'un obstacle)
6. L'acoustique sous-marine
7. Mécanique quantique (Détermination du potentiel)
8. Traitement d'image (Restauration des images floues)

Les problèmes inverses constituent donc l'un des sujets où le lien entre la théorie mathématique et la pratique est le plus fort.

Exemple 1.4 On s'intéresse à l'estimation de paramètres dans une équation aux dérivées partielles:

$$\begin{cases} \frac{\partial y}{\partial t} - \sum_{i=1}^n \frac{\partial}{\partial x_i} (a \frac{\partial y}{\partial x_i}) = f \text{ dans } E = \Omega \times]0, T[\\ y(x, 0) = y_0(x) \text{ dans } \Omega \\ \frac{\partial y}{\partial n} = g \text{ sur } \partial\Omega \times]0, T[\end{cases}$$

C'est l'équation de la chaleur, y est la température, f est un terme source, a est la conductivité thermique, et g est le flux de la chaleur (entrant ou sortant). On peut utiliser la même équation pour modéliser un écoulement monophasique (comme du pétrole) : y est la pression, f représente les puits le pompage, a est la perméabilité du milieu, et $g = 0$ pour milieu fermé.

Le problème est le suivant : à partir des mesures de y en certains points et à certains instants, il faut identifier a . Le problème direct est évidemment trivial, mais le problème inverse peut être des plus compliqués.

Exemple 1.5 (problème inverse mal-posé).

La différentiation et l'intégration sont deux problèmes inverses l'un de l'autre. Il est plus habituel de penser à la différentiation comme problème direct, et à l'intégration comme problème inverse. En effet, l'intégration possède de bonnes propriétés mathématiques qui conduisent à le considérer comme le problème direct. Et la différentiation est le prototype du problème mal posé, comme nous allons voir.

Considérons l'espace de Hilbert $L^2(\Omega)$, et l'opérateur intégral A défini par :

$$Af(x) = \int_0^x f(t) dt.$$

Il est facile de voir directement que A est un opérateur de $L^2(]0, 1[)$. Cet opérateur est injectif, par contre son image est un sous espace vectoriel.

$$Im(A) = \{f \in H^1(]0, 1[), u(0) = 0\},$$

où $H^1(]0, 1[)$ est l'espace de Sobolev. En effet, l'équation $Af = g$ est équivalente à $f(x) = g'(x)$ et $g(0) = 0$. L'image de A n'est pas fermé dans $L^2(]0, 1[)$ (bien entendu, elle l'est dans $H^1(]0, 1[)$). En conséquence, l'inverse de A n'est pas continu sur $L^2(]0, 1[)$, comme le montre l'exemple suivant. Considérons une fonction $g \in C^1([0, 1])$, et $n \in \mathbb{N}$. Soit

$$g_n(x) = g(x) + \frac{1}{2} \sin(n^2 x)$$

Alors

$$f_n(x) = g'_n(x) = g'(x) + n \cos(n^2 x) = f'(x) + n \cos(n^2 x)$$

De simples calculs montrent que $\|g - g_n\|_2 = O(\frac{1}{n})$ alors que $\|f - f_n\|_2 = O(\frac{1}{n})$.

Ainsi, la différence entre f et f_n peut être arbitrairement grand, alors que la différence entre g et g_n est arbitrairement petit. L'opérateur de dérivation (l'inverse de A) n'est donc pas continue, au moins avec ce choix de norme.

L'instabilité de l'inverse est typique des problèmes mal-posés. Une petite perturbation sur les données (ici g) peut avoir une influence arbitrairement grande sur le résultat (ici f).

1.3 Méthodes de résolution

Dans le but de résoudre un problème mathématique caractérisé par certaines emûches nous appuyons généralement sur des méthodes approximatives.

Dans certains cas ,on remplace le problème de départ par un problème approché.

il y a toujours une possibilité qu'une solution soit trouvée par un algorithme stable,si le problème est bien posé,sauf dans des cas majeurs où le problème est mal posé. On fait alors appel à la reformulation, afin d'arriver à un traitement numérique,ce qui impose des hypothèses supplémentaires, par exemple la régularité de la solution.cette démarche est connue sous le nom de régularisation.

Régulariser un problème mal-posé, c'est le remplacer par un autre bien-posé de sorte que "l'erreur commise" soit compensée par le gain de stabilité.La principale difficulté dans l'application d'une méthode de régularisation à un problème particulier, est la détermination du paramètre de régularisation lui-même.

Afin de pallier aux problèmes de stabilité,on a recours à l'utilisation de méthodes dites de régularisation,ces méthodes permettent de contourner ces problèmes qui empêchent la résolution exacte d'un problème inverse.Elles fournissent une solution approximative la plus proche possible de la solution réelle. Elles permettent de gérer le caractère mal-posé du problème.

Pour régulariser un problème mal-posé, on distingue deux méthodes directes comme la méthode de la décomposition en valeur singuliers (**SDV**) et la méthode de Tikhonov (proposé en 1976 avec Arsenin) et itératives comme la méthode Landweber (1951).

Soit une équation à opérateur compact injectif

$$Ax = u \tag{1.2}$$

où A est un opérateur très souvent linéaire, dans un espace vectoriel normé Y et x l'inconnu recherché dans un espace vectoriel normé X .

Un tel problème est toujours mal-posé puisque un opérateur compact n'est pas continûment inversible en dimension infinie.

1.3.1 Méthode des moindres carrés

Une grande classe de problèmes inverse à résoudre peut s'écrire sous la forme

$$Ax = y \tag{1.3}$$

où A est un opérateur continu d'un espace de Hilbert X dans un espace Hilbert Y . On trouve le caractère mal-posé de (1.3) en écrivant que:

1. A ne peut pas être surjectif.
2. A ne peut pas être injectif.
3. L'inverse de A , s'il existe, peut ne pas être continu.

Le problème (1.3) n'a de solution si $y \in \text{Im}(A)$, mais ceci est trop contraignant du fait des erreurs de mesure. L'idée est alors de remplacer ce problème par un problème de moindre carrés :

$$\min_{x \in X} \|Ax - y\|_Y^2$$

1.3.2 L'équation normale

Théorème 1.1 Soit $A : X \rightarrow Y$, un opérateur linéaire et borné et $y \in Y$, il existe $\hat{x} \in X$ tel que $\|A\hat{x} - y\|_Y \leq \|Ax - y\|_Y$, pour tout $x \in X$, si et seulement si $\hat{x} \in X$, est une solution

de l'équation normale:

$$A^*A\hat{x} = A^*y \tag{1.4}$$

où A^* est l'adjoint de A

Preuve. voir [8] Théorème 2.6 ■

Remarque 1.1 Dans le cas où X et Y sont de dimension infinie et A est compact (A^{-1} n'est pas continu), le problème (1.4) est mal-posé

1.3.3 Opérateur régularisant

Soient X et Y deux espaces de Hilbert et $A : X \rightarrow Y$, un opérateur linéaire injectif. On considère l'équation:

$$Ax = y$$

Soit y^δ un second membre perturbé de l'équation $Ax = y$ tel que:

$$\|y - y^\delta\| < \delta, \delta > 0$$

On souhaite approximer la solution x^δ de $Ax = y$.

si $y \in A(X) = \text{Im}(A) = \{Ax : x \in X\}$, alors il existe une solution unique de $Ax = y$.

Par contre rien ne dit que $y^\delta \in A(X)$!

Connaissant y^δ , on souhaite construire une approximation stable x^δ solution de $Ax^\delta = y^\delta$, il faut approximer l'opérateur inverse non borné $A^{-1} : A(X) \rightarrow X$ par l'opérateur $R : Y \rightarrow X$;

Définition 1.4 On appelle régularisation de (1.3) toute famille d'opérateurs linéaires bornés $R_\alpha : Y \rightarrow X$, $\alpha > 0$, tel que

$$\lim_{\alpha \rightarrow 0} (R_\alpha Ax - x) = 0, \text{ pour tout } x \in X$$

ie l'opérateur R_α converge simplement vers l'identité I .

α : paramètre de régularisation.

Théorème 1.2 Soit R_α une régularisation de (1.3), si A est compact en dimension infinie. Alors la famille d'opérateurs $\{R_\alpha\}$ ne sont pas uniformément bornés: il existe une suite $(\alpha_n)_{n \in \mathbb{N}}$ telle que

$$\lim_{k \rightarrow \infty} \alpha_k = 0 \text{ et } \lim_{n \rightarrow \infty} \|R_{\alpha_n}\| = \infty$$

i.e. $R_\alpha A$ ne converge pas vers l'identité au sens de la norme d'opérateur.

1.3.4 Méthode de Tikhonov

Le principe de la méthode de Tikhonov pour résoudre le problème inverse mal posé $Ax = y$ est de déterminer $x_\alpha \in E$, $\alpha > 0$, comme solution approchée de l'équation 1.3 qui minimise la fonctionnelle de Tikhonov $J_\alpha(x)$ telle que:

$$J_\alpha(x) = \|Ax - y\|_Y^2 + \alpha \|x\|_X^2. \quad (1.5)$$

Théorème 1.3 Soit $A : X \rightarrow Y$, un opérateur linéaire borné et $\alpha > 0$. La fonctionnelle de Tikhonov J_α admet un seul minimum $x_\alpha \in X$, ce minimum est la solution unique de l'équation normal

$$(A^*A + \alpha I)x_\alpha = A^*y. \quad (1.6)$$

Preuve: voir [13] ■

La solution régularisée du problème (1.3) est:

$$x_\alpha = (A^*A + \alpha I)^{-1}$$

qui converge vers la solution exacte x lorsque $\alpha \rightarrow 0$ et la famille d'opérateurs régularisants R_α est donnée par :

$$R_\alpha = (A^*A + \alpha I)^{-1}A^* \quad (1.7)$$

Remarque 1.2 comme l'opérateur $(A^*A + \alpha I)$ admet un inverse borné, alors le problème (1.6) est bien posé.

Théorème 1.4 Soit $A : X \rightarrow Y$ un opérateur linéaire et compact, $\alpha > 0$

a) L'opérateur $(A^*A + \alpha I)$ admet un inverse borné. L'opérateur $R_\alpha : Y \rightarrow X$ défini par 1.7 forme une stratégie de régularisation avec $\|R_\alpha\| \leq \frac{1}{2\sqrt{\alpha}}$. On l'appelle la régularisation de Tikhonov. $R_\alpha y^\delta$ est déterminé comme la solution unique $x_\alpha^\delta \in E$ de l'équation du second espace

$$\alpha x_\alpha^\delta + A^*A x_\alpha^\delta = A^*y^\delta.$$

Chaque choix de $\alpha(\delta) \rightarrow 0$ ($\delta \rightarrow 0$) avec $\frac{\delta^2}{\alpha(\delta)} \rightarrow 0$ ($\delta \rightarrow 0$) est admissible .

b) Soit $x = A^*Az \in \text{Im}(A^*)$ avec $\|z\| \leq E$ (E est une constante) . On choisit $\alpha(\delta) = \frac{c\delta}{E}$ pour $c > 0$, alors l'estimation suivante vérifiée:

$$\|x_{\alpha_\delta^\delta} - x\| \leq \frac{1}{2} \left(\frac{1}{\sqrt{c}} + \sqrt{c} \right) \sqrt{\delta E}$$

c) Soit $x = A^*A \in \text{Im}(A^*A)$ avec $\|z\| \leq E$ (E est une constante). Le choix $\alpha(\delta) = c \left(\frac{\delta}{E} \right)^{\frac{2}{3}}$, pour $c > 0$ donne l'estimation de l'erreur:

$$\|x_{\alpha_\delta^\delta} - x\| \leq \left(\frac{1}{2\sqrt{c}} + c \right) E^{\frac{1}{3}} \delta^{\frac{2}{3}}.$$

Pour cela, la méthode de régularisation de Tikhonov est optimale pour:

$$\|(A^*)^{-1}x\| \leq E \text{ où } \|(A^*A)^{-1}x\| \leq E .$$

Les valeurs propres de A tendent vers zéro et les valeurs propres de $\alpha I + A^*A$ sont bornées loin de zéro pour $\alpha > 0$.

Du théorème précédent, on observe que α a été choisit d'une façon à dépendre de δ et qu'il converge vers zéro quand δ tend vers zéro mais pas plus vite que δ^2 .

Preuve. pour la démonstration voir [13] , page 38

1.3.5 Méthode de Lavrentiev

Soit A un opérateur linéaire compact sur un espace de Hilbert X . L'ensemble des valeurs $\text{Im}(A)$ de l'opérateur A est en général non fermé.

Supposons que $\ker(A) = \{0\}$, si X est de dimension fini, l'opérateur A admet un inverse A^{-1} sur $Im(A)$ qui n'est pas borné. Cette circonstance, fait que le problème défini par $Ax = y$ devient instable. La méthode de Lavrentiev consiste à réduire l'équation $Ax = y$ à une équation de deuxième espace ($\alpha x + Ax = y$), $\alpha > 0$

Soit le problème

$$Ax = y, (y \in Im(A)) \quad (1.8)$$

où $A \in L(X)$, un opérateur compact, auto-adjoint et positive ($A = A^*$), dans un espace de Hilbert séparable X . Soit $y^\delta \in X$ une approximation de y telle que $\|y - y^\delta\| \leq \delta$, $\delta > 0$.

Si $y^\delta \notin Im(A)$, le problème

$$Ax = y^\delta \quad (1.9)$$

n'a pas de solution. Même si elle existe une solution x^α , nous avons aucune raison de croire que $x^\alpha \rightarrow x_e$, où x_e est la solution de $Ax = y$, donc il est mal posé.

Proposons-nous de régulariser cette équation.

A cet effet, on peut remplacer l'équation (1.9) par une équation auxiliaire de deuxième espèce.

$$\alpha x + Ax = y^\delta, \alpha > 0 \quad (1.10)$$

pour lequel le problème (1.9) devient bien posé. Dans ce qui suit, nous prouvons dans de nombreux cas l'équation (1.10) a une solution x_α^δ qui tend vers la solution exacte x_e de l'équation $Ax = y$ quand α et l'erreur δ tend vers zéro avec $\frac{\delta}{\alpha} \rightarrow 0$.

Théorème 1.5 Supposons que l'opérateur A vérifie pour tout $\alpha > 0$ la condition $\|A - \alpha I\|^{-1} \leq \frac{C}{\alpha}$, C est une constante.

Supposons aussi que $y \in D(A^{-2})$. Si le paramètre de régularisation $\alpha > 0$ est choisi en fonction de δ de telle sorte que, pour $\delta \rightarrow 0$, $\alpha \rightarrow 0$ et $\frac{\delta}{\alpha^2} \rightarrow 0$, alors $x_\alpha \rightarrow x$ pour $\delta \rightarrow 0$.

Si $\alpha = 0(\delta^{\frac{1}{3}})$ pour $\delta \rightarrow 0$ alors $\|x_\alpha - x_e\| = 0(\alpha^{\frac{1}{3}})$.

Soit x_e la solution de l'équation $Ax = y$, on prend $x_\alpha = (\alpha I + A)^{-1}y$ solution de l'équation $\alpha x + Ax = y$ comme solution approximative de l'équation (1.8).

Soit x_α^δ la solution de l'équation (1.10) telle que $\|y - y^\delta\| \leq \delta$ alors

$$\lim_{\alpha \rightarrow 0} \|x_\alpha - x_e\| = 0$$

et

$$x_\alpha^\delta \rightarrow x_e \text{ quand } \alpha \text{ et } \delta \text{ tend vers } 0 \text{ avec } \frac{\delta}{\alpha} \rightarrow 0$$

Preuve. pour la démonstration voir [12] ■

1.3.6 Décomposition en Valeurs singulières(SVD)

Définition 1.5 (Valeur singulière)

On appelle valeur singulière d'une matrice $A \in M_{m \times n}(\mathbb{R})$, les racines carrés des valeurs propres de $A^T A$, on les note $\delta_j = \sqrt{\lambda_i}$ telle que λ_i est une valeur propre de $A^T A$.

Définition 1.6 (vecteur singulière)

1. On dit que $v \in \mathbb{R}^n$ est un vecteur singulier à gauche, S'il existe un vecteur $w \in \mathbb{R}^m$ unitaire telle que : $Aw = \delta v$.
2. On dit que $w \in \mathbb{R}^m$ est un vecteur singulier à droite, s'il existe un vecteur $v \in \mathbb{R}^n$ unitaire telle que $A^t v = \delta w$.

Définition 1.7 (Matrice orthogonale)

Une matrice $Q \in \mathbb{R}^{m \times m}$ est orthogonale si $Q^T Q = I_m$.

Si $Q = (q_1, \dots, q_m)$ est orthogonal alors les $q_i, i = 1..m$ forment une base orthogonale de \mathbb{R}^m .

la méthode de régularisation par la décomposition en valeurs (**SVD**) consiste en une décomposition de la matrice $A_{m \times n}$ en trois autres matrices $\Sigma_{m \times n}$, $W_{m \times m}$ et $V_{n \times n}$ ayant chacune une structure bien spécifique (particulière).

Pour appliquer cette méthode à la résolution d'un problème inverse linéaire mal conditionné on pose $Ax = y$ avec A est une matrice rectangulaire de dimension $(m \times n)$, x est le vecteur d'état du système et y le vecteur de sortie du système.

La technique de **DVS** consiste en une décomposition de la matrice $A_{m \times n}$ en trois nouvelles matrices $\Sigma_{m \times n}$, $W_{m \times m}$, $V_{n \times n}$ avec $W_{m \times m}$ et $V_{n \times n}$ deux matrices orthogonales et $\Sigma_{m \times n}$ est une matrice "pseudo-diagonale" (tous les éléments hors de la diagonale principale sont nuls, mais la matrice n'est pas carré).

Théorème 1.6 Soit $A \in M_{m \times n}(\mathbb{R})$ une matrice de rang r . Il existe deux matrices orthogonales : $W \in M_{m \times m}(\mathbb{R})$, et $(W^T W = W W^T = I_m)$ et $V \in M_{n \times n}$, $(V^T V = V V^T = I_n)$ telle que :

$$A = W \Sigma V^T \tag{1.11}$$

$$\Sigma = \begin{pmatrix} (\Sigma_1) \in M_{r \times r}(\mathbb{R}) & (0) \in M_{r \times (n-r)}(\mathbb{R}) \\ (0) \in M_{(m-r) \times r}(\mathbb{R}) & (0) \in M_{(m-r) \times (n-r)}(\mathbb{R}) \end{pmatrix} \in M_{m \times n}(\mathbb{R}).$$

Est une matrice dite "pseudo-diagonale", avec $\Sigma_1 = \text{diag}(\delta_1, \delta_2, \dots, \delta_r)$ où $r = \min(m, n)$ et $\delta_1, \delta_2, \dots, \delta_r$ sont des valeurs singulières de A qui sont des membres réels non négatifs et qui respectent la condition : $\delta_1 \geq \delta_2 \geq \dots \geq \delta_r > 0$.

Composante par composante (en comparant les colonnes dans les équations)

$$AV = W\Sigma, A^T W = V\Sigma,$$

l'identité matricielle (1.11) devient :

$$\begin{cases} Av_j = \delta_j w_j, A^T w_j = \delta_j v_j \text{ pour } j = 1..r \text{ avec } r = \min(m, n). \\ A^T w_j = 0, \text{ pour } j = (r + 1) : \max(m, n). \end{cases}$$

Si l'on note $W = (w_1, \dots, w_m)$, $V = (v_1, \dots, v_n)$ les colonnes des matrices W et V , les vecteurs w_j et v_j sont respectivement, les vecteurs singuliers droits et gauches associés à la valeur singulière δ_j .

Théorème 1.7 Soit $A \in M_{m \times n}(\mathbb{R})$ une matrice. notons $A = W\Sigma V^T$ sa décomposition en valeurs singuliers.

1. Les valeurs de la matrice $A^T A$ sont les normes $\delta_j^2, \overline{j = 1, n}$ et ses vecteurs propres sont les vecteurs singuliers à gauche de $A, v_j, \overline{j = 1, n}$.
2. les valeurs propres de la matrice

$$\begin{pmatrix} 0 & A^T \\ A & 0 \end{pmatrix}$$

sont les nombres $\pm \delta_j, \overline{j = 1, n}$ et ses vecteurs propres sont

$$\frac{1}{\sqrt{2}} \begin{pmatrix} v_j \\ \pm w_j \end{pmatrix}$$

Proposition 1.1 Soit $A = W\Sigma V^T$ alors on a les relation suivantes:

1. Le rang de A est égal aux nombres de valeurs singulières non-nulles.
2. $\ker(A) = \text{Vect}(v_{r+1}, \dots, v_n), \text{Im}(A) = \text{Vect}(w_1, \dots, w_r)$.
3. $\ker(A^T) = \text{Vect}(v_1, \dots, v_r), \text{Im}(A) = \text{Vect}(w_{r+1}, \dots, w_m)$
4. $\|A\|_2 = \delta_1$.

Et nous pouvons écrire :

$$A = W_r \Sigma_r V_r^T = \sum_{j=1}^{r=\text{rang}(A)} \delta_j w_j v_j^T.$$

Commentaire

Le système $Ax = y$, telle que $A = W\Sigma V^T$ admet alors l'inversion, nous obtenons ainsi le vecteur entrées par :

$$x = V\Sigma^{-1}W^T y.$$

Le système est alors équivalent à $\Sigma x' = y'$ telle que $x' = V^T x$ et $y' = W^T y$.

Dans la plupart des cas, le système admet une infinité de solutions. On parle des quasi-solutions. Par conséquent, cela devient un problème de minimisation.

1.3.7 La méthode de Morozov

La solution obtenue par une méthode de type Tikhonov est une solution approchée au problème mal-posé, stable par rapports aux données mais dépendante du paramètre de régularisation α utilisé (où d'un critère d'arrêt pour la version itérative). Il existe plusieurs techniques pour choisir le paramètre de régularisation, telles que le critère de Morozov, Le critère de validation croisée [18] où la technique de L-Cuvre [15].

On donne ici un exemple de méthode de choix a posteriori du paramètre de régularisation $\alpha(\delta)$. On expose la plus classique de celles-ci, "The discrepancy principle" de Morozov où le principe de décalage (**D.P**) de Morozov [15].

On suppose que $A : X \rightarrow Y$ est un opérateur compact et injectif entre les deux espaces de Hilbert X et Y avec une image dense $\text{Im}(A) \subset Y$.
On étudie encore l'équation $Ax = y$, $y \in Y$.

D'après Kirch [13], on présente ici un principe basé sur la méthode de régularisation de Tikhonov correspondante :

$$\alpha x^{\alpha, \delta} + A^* A x^{\alpha, \delta} = A^* y^\delta.$$

qui est le minimum de la fonctionnelle :

$$J(x, y) = \|Ax - y^\delta\|_Y^2 + \alpha \|x\|_X^2$$

en plus :

$$\|A^{\alpha, \delta} - y^\delta\| = \delta.$$

On note que le choix de α par "The discrepancy principale" garantit d'une part que l'erreur est δ , d'autre part, α n'est pas trop petit, l'unicité et l'existence de la solution $\|Ax^{\alpha,\delta} - y^\delta\| = \delta$ sont justifiées par le théorème suivant :

Théorème 1.8 Soit $A : X \rightarrow Y$ un opérateur linéaire, compact avec une image dense $Im(A) \subset Y$, et $Ax = y$, $x \in X$, $y \in Y$, $y^\delta \in Y$ tel que :

$$\|y^\delta - y\| \leq \delta < \|y^\delta\|.$$

Soit $x^{\alpha(\delta)}$ la solution de Tikhonov satisfaisant $\|Ax^{\alpha(\delta)} - y^\delta\| = \delta$, pour tout $\delta \in]0, \delta_0[$. Alors :

1. $x^{\alpha(\delta),\delta} \rightarrow x$, i.e "The discrepancy principle" est admissible.

2. Soit $x = A^*x \in A^*(Y)$ avec $\|x\| \leq E$, alors $\|x^{\alpha(\delta),\delta} - x\| \leq 2\sqrt{\delta E}$.

Pour ce "The discrepancy princile" est une stratégie de régularisation sous la condition

$$\|(A^*)^{-1}x\| \leq E.$$

La preuve de ce théorème est dans [8], p.48.

La détermination de $\alpha(\delta)$ est équivalente au problème de recherche de la racine de la fonction monotone :

$$\phi(\alpha) = \|Ax^{\alpha,\delta} - y^\delta\|^2 - \delta^2$$

i.e :

$$\phi(\alpha) = 0 \Leftrightarrow \|Ax^{\alpha,\delta} - y^\delta\| = \delta$$

pour δ fixé, il n'est pas nécessaire de satisfaire l'équation $\|Ax^{\alpha,\delta} - y^\delta\| = \delta$, exactement.

Une inclusion de la forme :

$$C_1\delta \leq \|Ax^{\alpha,\delta} - y^\delta\| \leq C_2\delta$$

est suffisante pour prouver les assertions du théorème précédent.

Dans le théorème suivant, on prouve que l'ordre de convergence $o(\sqrt{\delta})$ est le meilleur pour le principe de décalage de Morozov.

Théorème 1.9. Soient A un opérateur compact et $\alpha(\delta)$ choisis par le principe de décalage. On suppose que pour tout $x \in Im(A^*A)$, $y = Ax \neq 0$ et pour toute suite $\delta_n \rightarrow 0$ et $y^{\delta_n} \in Y$ tel que :

$$\|y - y^{\delta_n}\| \leq \delta_n$$

et

$$\|y^{\delta_n}\| > \delta_n, \forall n \in \mathbb{N}$$

La solution de Tikhonov correspondante $x^\delta = x^{\alpha(\delta_n, \delta_n)}$ converge vers x plus rapide que $\sqrt{\delta_n}$ vers zéro, donc :

$$\frac{1}{\delta_n} \|x^n - x\| \longrightarrow 0 \text{ lorsque } n \text{ tend vers l'infini.}$$

Alors :

$$\dim(\text{Im}(A)) < \infty.$$

Preuve :

Nous montrons d'abord que le choix de $\alpha(\delta)$ par le principe de décalage implique la bornitude de $\frac{\alpha(\delta)}{\delta}$.

Abrégeons $x^\delta = x^{\alpha(\delta), \delta}$, On écrit :

$$\begin{aligned} \|y^\delta\| - \delta &= \|y^\delta\| - \|y^\delta - Ax^\delta\| \\ &\leq \|Ax^\delta\| \\ &= \frac{1}{\alpha(\delta)} \|AA^*(y^\delta - Ax^*)\| \\ &\leq \frac{\delta}{\alpha(\delta)} \|A\|^2 \end{aligned}$$

Où nous avons appliqué A à l'équation

$$\alpha x^\alpha + A^*Ax^\alpha = A^*y.$$

Cela permet d'obtenir

$$\alpha(\delta) \leq \frac{\delta \|A\|^2}{(\|y^\delta\| - \delta)}.$$

D'où

$$\|y^\delta\| \geq \|y\| - \|y - y^\delta\| \geq \|y\| - \delta,$$

nous concluons aussi que $\|y\| - \delta$ est borné loin de zéro pour δ suffisamment petit.

Ainsi nous avons montré qu'il existe $c > 0$ avec $\alpha(\delta) \leq c\delta$ pour tout δ suffisamment petit.

A présent nous supposons que $\dim(\text{Im}(A)) = \infty$ et construisons une contradiction.

Soit (γ_j, x_j, y_j) un système singulier par l'opérateur A et définissons :

$$y = y_1 \text{ et } y^{\delta_n} = y_1 + \delta_n y_n \text{ avec } \delta_n = \gamma_n^2$$

Alors $\delta_n \rightarrow 0$ lorsque $n \rightarrow \infty$, $y \in \text{Im}(A(A^*A)^k)$, $\forall k \in \mathbb{N}$ et $\|y^{\delta_n} - y\| \delta_n < \sqrt{1 + \delta_n^2} = \|y^{\delta_n}\|$.
Donc, les hypothèses du principe de décalage sont satisfaites.

Les solutions de $Ax = y$ et $\alpha(\delta_n)x^n + A^*Ax^n = A^*y^{\delta_n}$ sont données par :

$$x = x_1 \text{ et } x^n = \frac{\gamma_1}{\alpha(\delta_n) + \gamma_1^2} x_1 + \frac{\gamma_n \delta_n}{\alpha(\delta_n) + \gamma_n^2} x_n.$$

$\alpha(\delta_n)$ doit être choisi de telle sorte que $\|Ax^n - y^{\delta_n}\| = \delta_n$.

Nous calculons :

$$x^n - x = -\frac{\alpha(\delta_n)}{\gamma_1(\alpha(\delta_n) + \gamma_1^2)} x_1 + \frac{\gamma_n \delta_n}{\alpha(\delta_n) + \gamma_n^2} x_n.$$

et donc pour tout $n \geq 2$

$$\begin{aligned} \|x^n - x\| &\geq \frac{\gamma_n \delta_n}{\alpha(\delta_n) + \gamma_n^2} \\ &= \sqrt{\delta_n} \frac{1}{1 + \frac{\alpha(\delta_n)}{\delta_n}} \\ &\geq \sqrt{\delta_n} \frac{1}{1 + c} \end{aligned}$$

Ce qui contredit $\|x^n - x\| = o(\sqrt{\delta_n})$.

1.4 Méthodes itératives

1.4.1 Méthodes itératives de Tikhonov

On considère l'équation opérateur

$$Ax = y, \quad (y \in \text{Im}(A)) \tag{1.12}$$

où $A \in \mathcal{L}(X, Y)$, et X, Y deux espace de Hilbert, si $\text{Im}(A)$ n'est pas fermé en général alors le problème 1.12 est mal-posé.

Soit $y^\delta \in X$ une approximation de y telle que $\|y - y^\delta\| \leq \delta$, $\delta > 0$, la méthode de régularisation pour résoudre le problème 1.12 est la méthode de Tikhonov

$$x_\alpha = (A^*A + \alpha I)^{-1} A^* y^\delta, \quad \alpha > 0. \tag{1.13}$$

Soit $n \in \mathbb{N}$, fixé, $x_0 = x_\alpha^0 \in X$ une approximation initiale. La n -itéré de Tikhonov $x_\alpha = x_\alpha^n$ est donnée par la relation

$$x_\alpha^i = (A^*A + \alpha I)^{-1}(\alpha x_\alpha^{i-1} + A^*y^\delta), \quad i = \overline{1, n}. \quad (1.14)$$

Si $n = 1$, et $x_0 = 0$, l'équation 1.14 s'écrit

$$x_\alpha^1 = (A^*A + \alpha I)^{-1}(A^*y^\delta),$$

qui correspondant à la régularisation de Tikhonov classique.

1.4.2 Méthode itérative de Lavrentièv

Si l'opérateur A est auto-adjoint positive ($A = A^* > 0$), et $X = Y$, la méthode régularisation pour résoudre le problème (1.12) est la méthode de Lavrentièv

$$(\alpha I + A)x_\alpha = y^\delta \quad (1.15)$$

La méthode n -itéré de Lavrentièv $x_\alpha = x_\alpha^n$ est donné par la relation

$$x_\alpha^i = (A + \alpha I)^{-1}(\alpha x_\alpha^{i-1} - y^\delta), \quad i = \overline{1, n} \quad (1.16)$$

Si $n = 1$, et $x_0 = 0$, l'équation (1.16) s'écrit

$$x_\alpha^1 = (A + \alpha I)^{-1}y^\delta.$$

qui est la régularisation de Lavrentièv classique.

1.4.3 Méthode de Landweber

On examine dans ce paragraphe que la plus simple des méthodes itératives : la méthode de Landweber [6] qui a pour principal avantage de se prêter à une analyse simple. Malheureusement, elle converge trop lentement pour être utilisable en pratique, d'autant plus que des méthodes plus performantes existent. Les deux plus importantes sont méthode de Brakhage voir [6] et surtout la méthode du gradient conjugué et ses deux variantes. Cette dernière méthode est la plus employée. Dans le contexte des problèmes mal posés.

Soit l'équation :

$$Ax = y \quad (1.17)$$

Landweber [6], Fridman [14] et Bialy [2] ont proposé de réécrire l'équation (1.17) sous la forme:

$$x = (I - aA^*A)x + aA^*y$$

pour $a > 0$. Le schéma itérative de cette équation est le suivant :

$$x_0 = 0 \text{ et } x^m = (I - aA^*A)x^{m-1} + aA^*y \quad (1.18)$$

pour $m = \overline{1, n}$

Lemme 1.1

Soit la suite $(x^m)_m$ définie par (1.18) et définit la fonctionnelle $\Psi : X \rightarrow \mathbb{R}$ par :

$\Psi(x) = \frac{1}{2} \|Ax - y\|^2, x \in X$. Alors Ψ est différentiable au sens de Fréchet pour tout $z \in X$ et

$$\Psi'(z)x = \text{Re}(Az - y, Ax) = \text{Re}(A^*(Az - y), x), x \in X \quad (1.19)$$

La fonctionnelle linéaire $\Psi'(z)$ peut être identifié avec $A^*(Az - y) \in X$ sur l'espace de Hilbert X .

Il est facile de voir la forme explicite $x^m = R_m$, où l'opérateur $R_m : Y \rightarrow X$ est défini par :

$$R_m = a \sum_{k=0}^{m-1} (I - \alpha A^*A)^k A^*, m = \overline{1, n} \quad (1.20)$$

Théorème 1.8

1. Soit $A : X \rightarrow Y$, un opérateur compact et soit $0 < a < \frac{1}{\|A\|^2}$. On définit les opérateurs linéaires et bornés $R_m : X \rightarrow Y$ par (1.20). Ces opérateurs définissent une stratégie de régularisation de paramètre $\alpha = \frac{1}{m}$ et $\|R_m\| = \sqrt{am}$. La suite $x^{m,\delta} = R_m y^\delta$ est calculée par les itérations suivantes :

$$x^{0,\delta} = 0 \text{ et } x^{m,\delta} = (I - aA^*A)x^{m-1,\delta} + aA^*y^\delta$$

pour $m = \overline{1, n}$. Toute stratégie $m(\delta) \rightarrow \infty$ $\delta \rightarrow 0$ avec $\delta^2 m(\delta) \rightarrow 0$ $\delta \rightarrow 0$ est admissible.

2. Soit $x = A^*z \in \text{Im}(A^*)$ avec $\|z\| \leq E$ et $0 < c_1 < c_2$, pour chaque choix $m(\delta)$ avec $c_1 \frac{E}{\delta} \leq m(\delta) \leq c_2 \frac{E}{\delta}$, l'estimation suivante est vérifiée :

$$\|x^{m,\delta} - x\| \leq c_3 E^{\frac{1}{3}} \delta^{\frac{2}{3}}$$

pour c_3 qui dépend de c_1, c_2 et a . Pour cela, l'itération de Landweber est aussi optimale pour $\|(A^*A)^{-1}\| \leq E$.

Pour cette méthode, on observe qu'une haute précision demande un nombre large m d'itération mais la stabilité nous force à garder m le plus petit possible.

1.4.4 Méthode du gradient

Cette méthode repose sur le calcul du gradient $\nabla f(x)$ de la fonctionnelle à minimiser. Le comportement de cette méthode est le suivant :

Soit x^0 choisi comme point de départ de la méthode. Le calcul du gradient $\nabla f(x^0)$ en x^0 est réalisé. $\nabla f(x^0)$ indique la direction de plus grande augmentation de f . Afin de trouver le minimum de f , un déplacement dans la direction opposée à celle donnée par le gradient est réalisé avec une profondeur de descente λ_0 , ce qui conduit donc à la formule suivante :

$$x^1 = x^0 - \lambda_0 \nabla f(x^0)$$

La procédure présentée ci-dessus est réalisée itérativement jusqu'à l'obtention d'un minimum local. L'algorithme de la méthode est proposé ci-après.

Algorithme du Gradient

Initialisation de x^0 et calcul de la profondeur de descente initiale λ_0 (pour $k = 0$)

Initialisation du nombre d'itérations $k = 1$

Tant le critère d'arrêt n'est pas satisfait :

- Calcul de la direction de descente $d^k = -\nabla f(x^{k-1})$.
- calcul du nouvel itéré $x^k = x^{k-1} + \lambda^{k-1} d^k = x^{k-1} - \lambda^{k-1} \nabla f(x^{k-1})$
- Calcul de la profondeur de descente λ^k .
- Mis à jour de $k = k + 1$.

Fin Tant que

Le vecteur x^k est jugé satisfaisant lorsque le critère d'arrêt est vérifié .

L'algorithme présenté ci-dessus est l'algorithme général de la méthode de gradient. Plusieurs variantes de cette méthode existent on peut citer par exemple : La méthode du gradient à pas prédéterminé, la méthode de gradient à pas optimal, la méthode du gradient à pas variant ainsi que la méthode du gradient conjugué. Pour plus de détails sur ces différentes méthodes le lecteur peut se rapporter aux ouvrages de (Minoux, 2007) et de (Corriou, 2010). La méthode du gradient conjugué introduite par (Hestenes etiefel, 1952) sera détaillée dans le chapitre 2. Bien évidemment il existe un grand nombre de méthodes régularisantes non présentées dans ce document mais très utilisées notamment en génie thermique comme par exemple la méthode de spécification de fonction introduite par (Beck, et al., 1985).

La Méthode Gradient conjugué

Ce chapitre est entièrement dédié aux méthodes de type gradient conjugué. Ces méthodes sont surtout connues pour être des résolvantes rapides et robustes de grands systèmes d'équations linéaires : par exemple, la méthode classique de gradient conjugué (**CG**), introduite pour la première fois par Hestenes et Stiefel en 1952, trouve la solution exacte d'un système linéaire avec une matrice $N \times N$ définie positive dans au plus N étapes itératives, cf. Théorème 2.1 ci-dessous. Pour cette raison, l'importance de ces méthodes va bien au-delà de la régularisation des problèmes mal posés, bien qu'ici elles soient principalement étudiées de ce point de vue particulier. On peut approcher les méthodes de type gradient conjugué de différentes manières : il est possible de les voir comme des méthodes d'optimisation ou comme des méthodes de projection.

Définition 2.1 Soit V un espace vectoriel et $A : V \rightarrow V$ une application linéaire. Pour un vecteur donné $x_0 \in V$ et $k \in \mathbb{N}$, le $k^{\text{ème}}$ espace Krylov (basé sur x_0) est le sous-espace linéaire de V défini par

$$\mathcal{K}_{k-1}(A; x_0) = \mathbf{Vect} \{x_0, Ax_0, A^2x_0, \dots, A^{k-1}x_0\}.$$

la méthode de gradient conjugué choisit le réducteur d'une fonction particulière dans l'espace décalé $x_0 + \mathcal{K}_{k-1}(A; y - Ax_0)$ par rapport à une mesure particulière. Nous introduirons le sujet dans un cadre dimensionnel fini avec une approche d'optimisation, mais pour comprendre les propriétés de régularisation des algorithmes dans le cadre général l'analyse principale sera développée dans les espaces Hilbert en utilisant des polynômes orthogonaux. La référence principale pour ce chapitre est le livre de M. Hanke [15]. Pour l'introduction dimensionnelle finie, nous allons appuyer sur [15].

2.1 Introduction en dimension finie

Pour $N \in \mathbb{N}$ notons par

$$\langle \cdot, \cdot \rangle : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$$

le produit scalaire usuel sur \mathbb{R}^N munit de la norme euclidienne $\|\cdot\|$.

Pour $A \in \mathbb{M}_{m,N}(\mathbb{R})$, $m \in \mathbb{N}$, notons par $\|A\|$ la norme de A qui un opérateur linéaire de \mathbb{R}^N dans \mathbb{R}^m .

Pour plus de facilité, ici et en par la suite un vecteur $x \in \mathbb{R}^N$ sera pensé en tant que vecteur de colonne $x \in \mathbb{M}_{N,1}(\mathbb{R})$ donc x^* sera le vecteur de ligne transposé de x .

Nous considérons le système linéaire

$$Ax = b, \tag{2.1}$$

où $A \in \mathbb{GL}_N(\mathbb{R})$ symétrique et définie positive, $b \in \mathbb{R}^N$, $N \gg 1$.

Définition 2.2 La méthode de gradient conjugué pour la résolution (2.1) génère une suite $\{x_k\}_{k \in \mathbb{N}}$ dans \mathbb{R}^N de telle sorte que pour chaque k l'itération x_k minimise

$$\phi(x) := \frac{1}{2}x^*Ax - x^*b \tag{2.2}$$

sur $x_0 + \mathcal{K}_{k-1}(A; r_0)$, avec $r_0 = b - Ax_0$.

Bien sûr, lorsque la minimisation est faite sur l'ensemble de l'espace, alors minimiser est la solution exacte x^\dagger .

En raison des hypothèses formulées sur la matrice A , il y a une matrice orthogonale $U \in \mathbb{Q}_N(\mathbb{R})$ et matrice diagonale $\Lambda = \text{diag} \{\lambda_1, \dots, \lambda_N\}$, avec $\lambda_i > 0$ pour chaque $i = \overline{1, N}$, de telle sorte que

$$A = U\Lambda U^* \tag{2.3}$$

et (2.3) peut utiliser pour définir une norme sur \mathbb{R}^N

$$\|x\|_A := \sqrt{x^*Ax}.$$

Il s'avère que la propriété de minimisation de x_k peut être lue en termes de cette norme

Proposition 2.1 si $\Omega \subset \mathbb{R}^N$ et x_k minimise la fonction ϕ sur Ω , alors elle minimise aussi $\|x^\dagger - x\|_A = \|r\|_{A^{-1}}$ sur Ω , avec $r = b - Ax$.

Preuve. Puisque $Ax^\dagger = b$ et A est symétrique, nous avons

$$\begin{aligned} \|x^\dagger - x\|_A^2 &= (x^\dagger - x)^*A(x^\dagger - x) = x^*Ax - x^*Ax^\dagger - (x^\dagger)^*Ax + (x^\dagger)^*Ax^\dagger \\ &= x^*Ax - 2x^*b + (x^\dagger)^*Ax^\dagger = 2\phi(x) + (x^\dagger)^*Ax^\dagger. \end{aligned}$$

Ainsi la minimisation de ϕ est équivalent à la minimisation de $\|x^\dagger - x\|_A^2$ (et par conséquent de $\|x^\dagger - x\|_A$).

De plus, en utilisant la symétrie de A ,

$$\begin{aligned}\|x^\dagger - x\|_A^2 &= (A(x - x^\dagger))^* A^{-1} (A(x - x^\dagger)) = (Ax - b)^* A^{-1} (Ax - b) \\ &= \|x^\dagger - x\|_{A^{-1}}^2\end{aligned}$$

C.Q.F.D

■

Remarque 2.1. La proposition 2.1.1 a les conséquences suivantes :

1. L'itération $k^{\text{ème}}$ de **GC** minimise l'erreur d'approximation

$$\varepsilon_k := x_k - x^\dagger$$

avec la norme $\|\cdot\|_A$ dans l'espace décalé Krylov $x_0 + \mathcal{K}_{k-1}(A; r_0)$. Puisque un élément quelconque \check{x} de $x_0 + \mathcal{K}_{k-1}(A; r_0)$ peut s'écrire dans la formule

$$\check{x} = x_0 + \sum_{j=0}^{k-1} \gamma_j A^j r_0 = x_0 + \sum_{j=0}^{k-1} \gamma_j A^{j+1} (x^\dagger - x_0)$$

pour certains coefficients fixant $\gamma_0, \dots, \gamma_{k-1}$, si nous définissons les polynômes

$$\begin{aligned}q_{k-1}(\lambda) &:= \sum_{j=0}^{k-1} \gamma_j \lambda^j, \\ p_k(\lambda) &:= 1 - \lambda q_{k-1}(\lambda),\end{aligned}$$

nous obtenons

$$\begin{aligned}x^\dagger - \check{x} &= x^\dagger - x_0 - q_{k-1}(A)r_0 = x^\dagger - x_0 - q_{k-1}(A)A(x^\dagger - x_0) \\ &= p_k(A)(x^\dagger - x_0).\end{aligned}$$

Par conséquent, la propriété de minimisation de x_k peut également être écrite dans la formule

$$\|x^\dagger - x_k\|_A = \min_{p \in \Pi_k^0} \|p(A)(x^\dagger - x_0)\|_A,$$

Π_k^0 est l'ensemble de tous les polynômes p de degré égal à k tel que $p(0) = 1$.

2. pour tout $p \in \Pi_k = \{\text{polynômes de degré } k\}$ on a

$$P(A) = Up(\Lambda)U^* .$$

De plus, la racine carrée de A est bien définie par

$$A^{\frac{1}{2}} = U\Lambda^{\frac{1}{2}}U^*$$

avec $\Lambda^{\frac{1}{2}} := \text{diag} \left\{ \lambda_1^{\frac{1}{2}}, \dots, \lambda_N^{\frac{1}{2}} \right\}$ et immédiatement

$$\|x\|_A^2 = \|A^{\frac{1}{2}}x\|^2, x \in \mathbb{R}^N .$$

Par conséquent, puisque la norme d'une matrice symétrique, définie positive est égal à sa plus grande valeur propre, nous obtenons facilement :

$$\|p(A)x\|_A = \|p(A)A^{\frac{1}{2}}x\| \leq \|p(A)\| \|x\|_A, \forall x \in \mathbb{R}^N, \forall p \in \Pi_k,$$

$$\|x^\dagger - x_k\|_A \leq \|x^\dagger - x_0\|_A \min_{p \in \Pi_k^0} \max_{\lambda \in \text{spec}(A)} |P(\lambda)|,$$

où $\text{spec}(A)$ désigne le spectre de la matrice A .

La dernière inégalité peut être réinterprétée en termes d'erreur relative :

Corollaire 2.1. Supposons que A est symétrique et défini positif et $\{x_k\}_{k \in \mathbb{N}}$ est la suite d'itérations de la méthode **CG** . Si $k \geq 0$ est fixé et si p est un polynôme de Π_k^0 , alors l'erreur relative est bornée en plus :

$$\frac{\|x^\dagger - x_k\|_A}{\|x^\dagger - x_0\|_A} \leq \max_{\lambda \in \text{spec}(A)} |p(\lambda)| .$$

Cela conduit au résultat le plus important sur la méthode du gradient conjugué dans \mathbb{R}^N .

Théorème 2.1. Si $A \in \text{GL}_N(\mathbb{R})$ est une matrice symétrique et définie positive et b est un vecteur quelconque dans \mathbb{R}^N , alors la méthode du gradient conjugué trouvera la solution x^\dagger de (2.3) en N étapes itératives.

Preuve. Il suffit de définir le polynôme

$$\bar{p}(\lambda) = \prod_{j=1}^N \frac{\lambda_j - \lambda}{\lambda_j},$$

on remarque que \bar{p} appartenant à \prod_N^0 et on utilisant le corollaire 2.1 : puisque \bar{p} disparaît sur le spectre de A , $\|x^\dagger - x_N\|$ doit être égal à 0. ■

Le résultat est bien sûr très agréable, mais pas du tout acceptable : premièrement, si N est très grand, N itérations peuvent être trop nombreuses. Alors, nous devrions nous rappeler que nous devons habituellement traiter les données perturbées et si A est mal conditionné trouver la solution exacte du système perturbé peut mener à de très mauvais résultats. Ce problème sera examiné immédiatement, les informations A-priori sur les données b et le spectre de A peuvent être très importantes, utiles pour améliorer le résultat énoncé dans le théorème 2.1.: nous considérons deux choses différentes situations dans lesquelles le même résultat amélioré peut être montré.

proposition 2.2. Soit $u_j \in \mathbb{R}^N$, $j = 1, \dots, N$ le vecteur colonne de la matrice U dont (2.3) est retenu. Supposons que b soit une combinaison linéaire de vecteurs propre de A ;

$$b = \sum_{l=1}^k \gamma_l u_{i_l}, \quad \gamma_l \in \mathbb{R}, \quad 1 \leq i_1 < \dots < i_k \leq N.$$

Alors, si nous fixons $x_0 = 0$, la méthode du gradient conjugué convergera au plus en k étapes d'itération .

Preuve. Pour tout $l = 1, \dots, k$, soit λ_{i_l} la valeur propre associée à le vecteur propre u_{i_l} . Donc évidemment

$$x^\dagger = \sum_{l=1}^k \frac{\gamma_l}{\lambda_{i_l}} u_{i_l}$$

nous procédons comme la preuve de théorème 2.1 définir

$$\bar{p}(\lambda) = \prod_{l=1}^k \frac{\lambda_{i_l} - \lambda}{\lambda_{i_l}}$$

maintenant \bar{p} appartenant à \prod_k^0 et disparaît sur λ_{i_l} pour tout l , alors

$$\bar{p}(A)x^\dagger = \sum_{l=1}^k \bar{p}(\lambda_{i_l}) \frac{\gamma_l}{\lambda_{i_l}} u_{i_l}$$

et nous utilisons la propriété de minimisation

$$\|x^\dagger - x_k\|_A \leq \|\bar{p}(A)x^\dagger\|_A = 0$$

pour conclure. ■

De la même manière, il est possible de prouver l'énoncé suivant.

Proposition 2.3. Supposons que le spectre de A se compose exactement de k valeurs propres distinctes. **CG** trouvera alors la solution de (2.1) au plus en k itérations.

On peut aussi étudier le comportement de l'erreur relative mesurée avec la norme euclidienne de conditionnement de A :

Proposition 2.4.

1. Soit $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ sont des valeurs propres de A , alors $\forall x \in \mathbb{R}^N$, on a

$$\|x\|_A \lambda_N^{\frac{1}{2}} \leq \|Ax\| \leq \|x\|_A \lambda_1^{\frac{1}{2}}.$$

2. si $k_2(A) := \|A\| \|A^{-1}\|$ est le conditionnement de A , alors

$$\frac{\|b - Ax_k\|}{\|b\|} \leq \sqrt{k_2(A)} \frac{\|r_0\|}{\|b\|} \frac{\|x_k - x^\dagger\|_A}{\|x_0 - x^\dagger\|_A}.$$

Preuve. Soit $u_j \in \mathbb{R}^N$ le vecteur colonne de la matrice U comme dans la preuve de la proposition 2.2. Alors

$$Ax = \sum_{j=1}^N \lambda_j (u_j^* x) u_j,$$

alors

$$\begin{aligned} \lambda_N \|x\|_A^2 &= \lambda_N \|A^{\frac{1}{2}} x\|_A^2 = \lambda_N \sum_{j=1}^N \lambda_j (u_j^* x)^2 \\ &\leq \|Ax\|^2 \leq \lambda_1 n \sum_{j=1}^N \lambda_j (u_j^* x)^2 = \lambda_1 \|A^{\frac{1}{2}} x\|_A^2 = \lambda_1 \|x\|_A^2, \end{aligned}$$

ce qui prouve la première partie.

pour le second énoncé, rappelons que $\|A^{-1}\| = \lambda_N^{-1}$ et en utilisant l'inégalité précédente, on obtient

$$\frac{\|b - Ax_k\|}{\|r_0\|} = \frac{\|A(x^\dagger - x_k)\|}{\|A(x^\dagger - x_0)\|} \leq \sqrt{\frac{\lambda_1}{\lambda_N}} \frac{\|x^\dagger - x_k\|}{\|x^\dagger - x_0\|} = \sqrt{k_2(A)} \frac{\|x_k - x^\dagger\|_A}{\|x_0 - x^\dagger\|_A}.$$



Enfin, nous mentionnons un résultat de J.W. Daniel [4] qui fournit une liaison pour l'erreur relative, qui est, dans un certain sens, aussi nette que possible :

$$\frac{\|x_k - x^\dagger\|_A}{\|x_0 - x^\dagger\|} \leq 2 \left(\frac{\sqrt{k_2(A)} - 1}{\sqrt{k_2(A)} + 1} \right)^k .$$

Exemple 2.1. Supposons que le spectre de A soit contenu dans les intervalles $I_1 =]1; 1,5[$ et $]399; 400[$ et posons $x_0 = 0$.

La meilleure chose que nous puissions dire sur le conditionnement de A est que $k_2(A) \leq 400$ qui est inséré dans la formule de Daniel donne

$$\frac{\|x_k - x^\dagger\|}{\|x^\dagger\|} \leq \left(\frac{19}{21} \right)^k \approx 2 (0.91)^k ,$$

prévoir une convergence lente.

Cependant, si nous prenons

$$\max_{\lambda \in \text{spec}(A)} |\bar{p}_{3k}(\lambda)| \leq \left(\frac{0,25}{1,25} \right)^k = (0,2)^k ,$$

assure une estimation beaucoup plus précise. Plus précisément, afin de réduire l'erreur relative du facteur 10^{-3} en utilisant la norme $\|\cdot\|_A$ Daniel prévoit 83 itérations étapes, car $2(0.91)^k \leq 10^{-3}$ quand $k > -\frac{\log_{10}(2000)}{\log_{10}(0.91)} \approx 82,5$. Au lieu de cela, selon selon l'estimation fondée sur \bar{p}_{3k} , l'erreur relative du facteur 10^{-3} sera réduite après $k = 3i$ itérations quand $(0,2)^i \leq 10^{-3}$ i.e. quand $i > -\frac{3}{\log_{10}(0,2)} \approx 4,3$, il ne prévoit donc que 15 itérations !

En conclusion, en dimension finie, nous avons vu que la méthode du gardient conjugué combine certaines propriétés de minimisation efficace et que l'information a priori peut être utilisée pour prédire la force de sa performance. En outre, les polynômes q_k et p_k peuvent être utilisés pour comprendre son comportement et peut s'avérer très utile dans des cas particuliers.

2.2 Définition générale dans les espaces de Hilbert

Dans cette section, nous définissons la méthodes du gradient conjugué dans les espaces de Hilbert. Pour les étapes sautées consultez [7].

Soit l'opérateur A agissant entre les deux espaces Hilbert X et Y seront auto-adjoints et semi-définis positifs avec son spectre contenu dans $[0, 1]$.

pour $n \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$, fixons une approximation $x_0 \in X$ de la solution de $A^\dagger x$ de $Ax = y$ et considérer la forme bilinéaire définie sur l'espace de tous les polynômes Π_∞ par

$$\begin{aligned} [\phi, \psi] &:= \langle \phi(A)(y - Ax_0), A^n \psi(A)(y - Ax_0) \rangle \\ &= \int_0^\infty \phi(\lambda) \psi(\lambda) \lambda^n d\|\varepsilon_\lambda(y - Ax_0)\|^2, \end{aligned}$$

avec $\{\varepsilon_\lambda\}$ est la famille spectrale associée à A .

La théorie des polynômes orthogonaux (voir, par exemple, [7] chapitre II) donne l'existence d'une suite de polynômes orthogonaux bien définie $\{p_k^{[n]}\}$ telle que $p_k^{[n]} \in \Pi_\infty$ et

$$[p_k^{[n]}, p_j^{[n]}]_n = 0, \quad k \neq j.$$

si nous imposons que ces polynômes appartiennent à Π_k^0 , la suite est univoque déterminé et satisfait à une formule de récurrence à trois termes bien connus, donnée par

$$\begin{aligned} p_0^{[n]} &= 1, \quad p_1^{[n]} = 1\alpha_0^{[n]}\lambda, \\ p_{k+1}^{[n]} &= -\alpha_k^{[n]}\lambda p_k^{[n]} + p_k^{[n]} - \alpha_k^{[n]} \frac{\beta_k^{[n]}}{\beta_{k-1}^{[n]}} (p_{k-1}^{[n]} - p_k^{[n]}), \quad k \geq 1, \end{aligned} \tag{2.4}$$

où $\alpha_k^{[n]} \neq 0$ et $\beta_k^{[n]}, k \geq 0$ peut être calculé explicitement (voir ci-dessous). L'itération $k^{\text{ème}}$ d'une méthode de type de gradient conjugué est donnée par

$$x_k^{[n]} := x_0 + q_{k-1}^{[n]}(A)(y - Ax_0),$$

où les polynômes d'itération $\{q_{k-1}^{[n]}\}$ sont liés aux polynômes résiduels $\{q_k^{[n]}\}$ via

$$q_{k-1}^{[n]}(\lambda) = \frac{1 - p_k^{[n]}}{\lambda} \in \Pi_{k-1}$$

L'expression polynômial résiduel pour $p_k^{[n]}$ est justifié par le fait que

$$\begin{aligned} y - Ax_k^{[n]} &= y - A \left(x_0 + q_{k-1}^{[n]}(A)(y - Ax_0) \right) \\ &= y - Ax_0 - Aq_{k-1}^{[n]}(A)(y - Ax_0) \\ &= \left(I - Aq_{k-1}^{[n]}(A) \right) (y - Ax_0) \\ &= p_k^{[n]}(A)(y - Ax_0). \end{aligned}$$

De plus, si $y \in \text{Im}(A)$ et $x \in X$ est tel que $Ax = y$, alors

$$x - x_k^{[n]} = x - x_0 - q_{k-1}^{[n]}(A)A(x - x_0) = p_k^{[n]}(A)(x - x_0).$$

Dans les sections suivantes, afin de simplifier les notations, nous allons omettre l'exposant n et la dépendance de p_k et q_k de y sauf si strictement nécessaire.

2.3 Les algorithmes

Dans cette section, nous décrivons comment les algorithmes du type de gradient conjugué peuvent être dérivées du cadre général de la section précédente. Nous nous référons essentiellement à [7], en ajoutant quelques détails

Proposition 2.3.1 De la formule (2.4), les polynômes d'itération satisfont

$$q_{-1} = 0, q_0 = \alpha_0, \\ q_k = q_{k-1} + \alpha_k \left(p_k + \frac{\beta_k}{\alpha_{k-1}} (q_{k-1} - q_{k-2}) \right), k \geq 1.$$

Preuve. Par la définition des polynômes d'itération, nous avons

$$\lambda q_{-1}(\lambda) - 1 = 0, q_0(\lambda) = \frac{1 - p_1(\lambda)}{\lambda} = \alpha_0$$

et $k \geq 0$, la formule de récurrence de p_k donne

$$\begin{aligned} q_k(\lambda) &= \frac{1 - p_{k+1}(\lambda)}{\lambda} = \frac{1 + \alpha_k \lambda p_k(\lambda) - p_k(\lambda) + \alpha_k \alpha_{k-1}^{-1} \beta_k (p_{k-1}(\lambda) - p_k(\lambda))}{\lambda} \\ &= \frac{\alpha_k \lambda - \lambda^2 \alpha_k q_{k-1}(\lambda) + \alpha_k \alpha_{k-1}^{-1} \beta_k (\lambda q_{k-1}(\lambda) - \lambda q_{k-2}(\lambda))}{\lambda} \\ &= \alpha_k p_k(\lambda) + q_{k-1}(\lambda) + \frac{\alpha_k}{\alpha_{k-1}} \beta_k (q_{k-1}(\lambda) - q_{k-2}(\lambda)). \end{aligned}$$

■

Proposition 2.3.2 L'itéré x_k peut calculé par la méthode de gradient conjugué avec la récursion suivante :

$$\Delta x_0 = y - Ax_0 \qquad x_1 = x_0 + \alpha_0 \Delta x_0, \qquad (2.5)$$

$$\Delta x_k = y - Ax_k + \beta_k \Delta x_{k-1}, \qquad x_{k+1} = x_k + \alpha_k \Delta x_k, k \geq 1. \qquad (2.6)$$

Preuve. Puisque $q_0 = \alpha_0$, la relation entre x_1 et x_0 est évidente. Nous procédons par induction sur k . D'après les définitions de x_k et x_{k+1} , il s'ensuit que

$$x_{k+1} = x_k + (q_k - q_{k-1})(A)(\Delta x_0)$$

et maintenant, en utilisant la proposition 2.1 et l'induction, nous avons :

$$\begin{aligned} (q_k - q_{k-1})(A)(\Delta x_0) &= \alpha_k p_k(A)(\Delta x_0) + \frac{\alpha_k}{\alpha_{k-1}} \beta_k (q_{k-1} - q_{k-2})(A)(\Delta x_0) \\ &= \alpha_k (y - Ax_k) + \alpha_k \beta_k \frac{(q_{k-1} - q_{k-2})(A)(\Delta x_0)}{\alpha_{k-1}} \\ &= \alpha_k (y - Ax_k + \beta_k \Delta x_{k-1}). \end{aligned}$$

■

Proposition 2.3.3 Définissons

$$s_0 := 1, s_k := p_k + \beta_k s_{k-1}, k \geq 1.$$

Alors, pour tout $k \geq 1$, les relations suivantes sont valables :

$$\Delta x_k = s_k(A)(y - Ax_0), \quad (2.7)$$

$$p_{k+1} = p_k - \alpha_k \lambda s_k \quad (2.8)$$

Preuve. Pour $k = 0$, la première relation est évidemment satisfaite. Pour $k \geq 1$, en utilisant à nouveau l'induction, on obtient :

$$\Delta x_k = y - Ax_k + \beta_k \Delta x_{k-1} = p_k(A)(\Delta x_0) + \beta_k s_{k-1}(A)(\Delta x_0) = s_k(A)(\Delta x_0),$$

ce qui prouve (2.5)

Pour avoir (2.6), il suffit de considérer les relations

$$\begin{aligned} \frac{x_{k+1} - x_k}{\alpha_k} &= \Delta x_k = s_k(A)(\Delta x_0) \\ x_{k+1} - x_k &= (q_k(A) - q_{k-1}A)(\Delta x_0), \\ \lambda(q_k(\lambda) - q_{k-1}(\lambda)) &= p_k(\lambda) - p_{k+1}(\lambda) \end{aligned}$$

et de les relier entre eux.

■

Proposition 2.3.4 La suite $\{s_k\} = \{s_k^{[n]}\}_{k \in \mathbb{N}}$ est orthogonale par rapport à du produit scalaire $[\cdot, \cdot]_{n+1}$. Plus précisément, si l désigne le nombre de points non nuls d'accroissement de la fonction $\alpha(\lambda) = \|\varepsilon(\Delta x_0)\|^2$ alors

$$p_k^{[n+1]} = \frac{1}{\pi_{k,n}} \frac{p_k^{[n]} - p_{k+1}^{[n]}}{\lambda}, \text{ avec } \pi_{k,n} := (p_k^{[n]})'(0) - (p_{k+1}^{[n]})'(0) > 0$$

pour tout $0 \leq k \leq l$.

Preuve. voir [7] ■

Proposition 2.3.5 Si la fonction $\alpha(\lambda)$ définie dans la proposition 2.3.4 possède $l = \infty$ points d'accroissement, les coefficients α_k et β_k apparaissant dans les formules (2.5) et (2.6) de la proposition 2.3.2 peuvent être calculés comme suit :

$$\alpha_k = \frac{[p_k, p_k]_n}{[s_k, s_k]_{n+1}}, \quad k \geq 0.$$

$$\beta_k = \frac{1}{\alpha_{k-1}} \frac{[p_k, p_k]_n}{[s_{k-1}, s_{k-1}]_{n+1}} = \frac{[p_k, p_k]_n}{[p_{k-1}, p_{k-1}]_n}, \quad k \geq 1.$$

Sinon, les formules ci-dessus restent valides, mais l'itération doit être arrêtée au cours du $(l + 1)$ -ième pas puisque $[s_l, s_l]_{n+1} = 0$ et α_k est indéfini. Dans ce cas, on distingue les possibilités suivantes :

- si $y \in \text{Im}(A)$, pour tout $n \in \mathbb{N}_0$ $x_l^{[n]} = A^\dagger y$;
- si y a une composante non triviale le long de $\text{Im}(A)^\perp$ et $n = 0$, alors la conclusion $(I - \varepsilon_0)x_l = A^\dagger y$ n'est plus valable .

Preuve. Voir [7] ■

Proposition 2.3.6 Supposons que $n \geq 1$, que x_k soit la k -ième itération de la méthode de type gradient conjugué correspondante. Méthode correspondante de type gradient conjugué et que x soit tout autre élément du sous-espace décalé de Krylov $x_0 + K_{k-1}(A; y - Ax_0)$. Alors

$$\|A^{\frac{n-1}{2}}(y - Ax_k)\| \leq \|A^{\frac{n-1}{2}}(y - Ax)\|$$

et l'égalité tient si et seulement si $x = x_k$.

Si $n = 0$ et $y \in \text{Im}(A)$, alors $\|A^{\frac{n-1}{2}}(y - Ax_k)\|$ est bien défini et le même résultat obtenu dans le cas $n \geq 1$ reste valable.

Preuve. Voir proposition 2.1 dans [7] ■

2.3.1 La méthode du résidu minimal (RM) et la méthode du gradient conjugué (GC).

Dans le cas $n = 1$, la Proposition 2.3.6 nous montre que la méthode correspondante minimise, dans l'espace de Krylov décalé $x_0 + K_{k-1}(A; y - Ax_0)$, la norme résiduelle. Pour cette raison, cette méthode est appelée méthode du résidu minimal (**RM**). Les propositions 2.3.1-2.3.5 conduisent à l'algorithme 1

Dans le cas $n = 0$, en utilisant à nouveau les propositions 2.3.1-2.3.5, nous trouvons (cf. Algorithme 2) la méthode classique du gradient conjugué proposée à l'origine par Hestenes et

Stiefel dans [10] en 1952. Si $y \in \text{Im}(A)$ alors selon la proposition 2.3.6, la k -ième itération x_k de **GC** minimise l'erreur $x^\dagger - x_k$ dans $x_0 + K_{K-1}(A; y - Ax_0)$ en respectant la norme d'énergie $\langle x^\dagger - x_k, A(x^\dagger - x_k) \rangle$.

En regardant les algorithmes, il est important de noter que pour chaque étape itérative, **RM** et **GC** ne doivent calculer qu'une seule fois un produit de type Av avec $v \in X$

Algorithme 1 RM

$r_0 = y - Ax_0;$
 $d = r_0;$
 $Ad = Ar_0;$
 $k = 0;$

Tant que (pas d'arrêt) faire

- $\alpha = \frac{\|r_k\|^2}{\langle d; Ad \rangle}.$
- $x_{k+1} = x_k + \alpha d;$
- $r_{k+1} = r_k - \alpha Ad;$
- $\beta = \frac{\|r_{k+1}\|^2}{\|r_k\|^2};$
- $d = r_{k+1} + \beta d;$
- $k = k + 1;$

Fin Tant que

2.3.2 CGEN et CGEM.

Supposons que l'opérateur A ne soit pas auto-adjoint et semi-défini, . Il est alors toujours possible d'utiliser les méthodes de type gradient conjugué, en recherchant la solution (la plus proche) de l'équation

$$AA^*v = y \tag{2.9}$$

Algorithme 2 CG

$r_0 = y - Ax_0;$
 $d = r_0;$
 $k = 0;$

Tant que (pas d'arrêt) faire

- $\alpha = \frac{\|r_k\|^2}{\langle d, Ad \rangle}.$
- $x_{k+1} = x_k + \alpha d;$
- $r_{k+1} = r_k - \alpha Ad;$
- $\beta = \frac{\|r_{k+1}\|^2}{\|r_k\|^2};$
- $d = r_{k+1} + \beta d;$
- $k = k + 1;$

Fin Tant que

Posons $x = A^*v$.

Dans ce cas plus général, nous désignerons par $\{E_\lambda\}$ la famille spectrale de A^*A et par $\{F_\lambda\}$ la famille spectrale de A^*A . Toutes les définitions du cas auto-adjoint s'appliquent ici, en gardant à l'esprit qu'elles feront toujours référence à A^*A au lieu de A . et les itérés correspondants sont

$$v_k = v_0 + q_{k-1}(AA^*)(y - Ax_0). \quad (2.10)$$

La définition de la première itération v_0 n'est pas importante, puisque nous ne sommes pas intéressés de calculer v_k , mais nous cherchons x_k . Ainsi, nous multiplions les deux côtés de l'équation (2.10) par A^* et nous obtenons

$$x_k = x_0 + A^*q_{k-1}(AA^*)(y - Ax_0) = x_0 + q_{k-1}(A^*A)A^*(y - Ax_0).$$

Comme dans le cas auto-adjoint, le résidu $y - Ax_k$ est exprimé en termes de polynômes résiduels p_k correspondant à l'opérateur AA^* par la formule suivante

$$y - Ax_k = p_k(AA^*)(y - Ax_0)$$

et si $y = Ax$ pour tout $x \in X$, alors

$$x - x_K = pk(AA^*)(x - x_0).$$

Comme dans le cas auto-adjoint, nous considérons les possibilités $n = 1$ et $n = 0$.

◆ Si $n = 1$, selon la proposition 2.3.6, les itérés x_k minimisent la norme résiduelle dans l'espace décalé de Krylov $x_0 + K_{k-1}(A^*A; A^*(y - Ax_0))$, cf. Algorithme 3.

Un fait très important concernant ce cas est que ceci est égal à l'application directe de **CG** à l'équation normale. application directe de **CG** à l'équation normale

$$A^*Ax = A^*y,$$

comme on peut facilement le vérifier en utilisant la proposition 2.3.6 ou en comparant les algorithmes. Cette méthode est de loin la plus célèbre dans la littérature et est généralement appelée **CGEN**, c'est-à-dire **GC** appliquée à l'équation normale.

◆ Il est également possible d'appliquer **GC** à l'équation (2.9), en obtenant l'algorithme 4 : cela correspond au choix $n = 0$ et par la proposition 2.3.6 si $y \in \text{Im}(A)$ les itérés X_K minimisent la norme d'erreur $\|x^\dagger - x_k\|$ dans l'espace de Krylov correspondant.

Nous concluons par une remarque : la formation et la résolution de l'équation (2.9) ne peut conduire qu'à la solution de norme minimale de $Ax = y$, car les itérés $x_k = A^*v_k$ se trouvent dans $\text{Im}(A^*) \subseteq \ker(A)^\perp$, qui est fermé. Ainsi, si l'on cherche des solutions différentes de x^\dagger , alors il ne faut pas se fier à ces méthodes.

Algorithme 3 CGEN

$$r_0 = y - Ax_0;$$

$$d = A^*r_0;$$

$$k = 0;$$

Tant que (pas d'arrêt) faire

- $\alpha = \frac{\|A^*r_k\|^2}{\|Ad\|^2}.$

- $x_{k+1} = x_k + \alpha d;$

- $r_{k+1} = r_k - \alpha Ad;$

- $\beta = \frac{\|A^*r_{k+1}\|^2}{\|A^*r_k\|^2};$

- $d = A^*r_{k+1} + \beta d;$

- $k = k + 1;$

Fin Tant que

Algorithme 4 CGEM

$r_0 = y - Ax_0;$
 $d = A^*r_0;$
 $k = 0;$

Tant que (pas d'arrêt) faire

- $\alpha = \frac{\|r_k\|^2}{\|d\|^2}.$
- $x_{k+1} = x_k + \alpha d;$
- $r_{k+1} = r_k - \alpha Ad;$
- $\beta = \frac{\|r_{k+1}\|^2}{\|r_k\|^2};$
- $d = A^*r_{k+1} + \beta d;$
- $k = k + 1;$

Fin Tant que

2.4 Théorie de la régularisation pour les méthodes de type gradient conjugué

Théorème 2.2.1. L'opérateur semi-défini auto-adjoint A est compact et non dégénéré. . Alors pour toute méthode de type gradient conjugué à paramètre $n \in \mathbb{N}_0$ et pour tout $k \in \mathbb{N}$ l'opérateur $R_k = R_k^{[n]}$ qui fait correspondre la donnée y sur la k -ième itération, $x_k = x_k^{[n]}$ est discontinu dans X . De plus, même dans le cas non compact, R_k est discontinu en y si et seulement si $\varepsilon_0 y$ appartient à un sous-espace invariant de A de dimension au plus $k - 1$.

Chaque règle d'arrêt pour une méthode de type gradient conjugué doit tenir compte de ce phénomène. En particulier, aucune règle d'arrêt a-priori $K(\delta)$ ne peut rendre convergente une méthode de type gradient conjugué (cf. rendre convergente une méthode de type gradient conjugué (cf. [8] et [7]). Sur Au début, cela semble décourageant, mais l'absence de discontinuité de R_k n'est pas vraiment un gros problème, puisqu'il s'agit d'une méthode de type gradient conjugué. n'est pas vraiment un gros problème, puisqu'il est toujours possible de trouver des

règles d'arrêt a-posteriori fiables qui préservent les principales propriétés de la méthode. fiables qui préservent les principales propriétés de convergence et d'optimalité de l'ordre. l'optimalité de l'ordre.

Avant de procéder à l'analyse, nous devons souligner que les méthodes avec le paramètre $n \geq 1$ sont beaucoup plus faciles à traiter que celles avec $n = 0$. cette raison, nous allons considérer les deux cas séparément.

2.4.1. Propriétés de régularisation de RM et CGEN

Nous commençons par considérer le cas non perturbé.

Proposition 2.4.1. Soit $y \in Im(A)$ et soit n_1 et n_2 des entiers avec $n_1 \leq n_2$ et $[1, 1]_{n_1} < +\infty$. Alors $[p_k^{[n_2]}, p_k^{[n_2]}]_{n_1}$ est strictement décroissant lorsque k allant de 0 à l .

Cela a deux conséquences importantes :

Corollaire 2.4.1. Si $y \in Im(A)$ et $x_k = x_k^{[n]}$ sont les itérés d'une méthode de type gradient conjugué correspondant à un paramètre $n \geq 1$ et à une droite y , alors

- La norme résiduelle $\|y - Ax_k\|$ est strictement décroissante pour $0 \leq k \leq l$.
- L'erreur d'itération $\|x^\dagger - x_k\|$ est strictement décroissante pour $0 \leq k \leq l$.

Pour obtenir les résultats de convergence les plus importants, les estimations suivantes jouent un rôle central. Nous devons faire la distinction entre le cas auto-adjoint et le cadre plus général de la section 2.3.2. La preuve de la partie avec l'opérateur opérateur AA^* , qui s'avérera d'une grande importance par la suite, se trouve entièrement dans [8].

Lemme 2.4.1. Soit $\lambda_{1,k} < \dots < \lambda_{k,k}$ sont les zéros de p_k . Alors :

- Dans le cas auto-adjoint, pour $y \in X$,

$$\|y - Ax_k\| \leq \|\varepsilon_{\lambda_{1,k}} \phi_k(A)y\|$$

avec la fonction $\phi_k(\lambda) = \frac{\lambda_{1,k}}{\lambda_{1,k} - \lambda}$ satisfait

$$0 \leq \phi_k(\lambda) \leq 1, \lambda^2 \phi_k^2(\lambda) \leq 4|p_k'(0)|^{-1}, 0 \leq \lambda \leq \lambda_{1,k}.$$

- Dans le cas général avec AA^* , pour $y \in Y$,

$$\|y - Ax_h\| \leq \|F_{\lambda_{1,k}} \phi_k(AA^*)y\|,$$

avec la fonction $\phi(\lambda) = p_k(\lambda) \left(\frac{\lambda_{1,k}}{\lambda_{1,k} - \lambda} \right)^{\frac{1}{2}}$ satisfaisant

$$0 \leq \phi_k(\lambda) \leq 1, \lambda \phi_k^2(\lambda) \leq |p_k'(0)|^{-1}, 0 \leq \lambda \leq \lambda_{1,k}.$$

Cela conduit au théorème de convergence suivant

Théorème 2.4.2. • Supposons que A soit auto-adjoint et semi-défini. Si $y \in \text{Im}(A)$, alors les itérés $\{x_k\}$ d'une méthode de type gradient conjugué. avec le paramètre $n \geq 1$ convergent vers $A^\dagger y$ lorsque k tend vers l'infini. Si $y \notin \text{Im}(A)$ et $l = \infty$, alors $\|x_k\| \rightarrow +\infty$ lorsque $k \rightarrow +\infty$. Si $y \notin \text{Im}(A)$ et $l < \infty$ alors l'itération se termine après l pas, $Ax_l = \varepsilon_0 y$ et $x_l = A^\dagger y$ si et seulement si $l = 0$.

• Soit A satisfaisant les hypothèses de la section 2.3.2 et soit $\{x_k\}$ les itérations d'une méthode de type gradient conjugué avec le paramètre $n \geq 1$ appliquée avec AA^* . Si $y \in D(A^\dagger)$ alors x_k converge vers $A^\dagger y$ lorsque $k \rightarrow +\infty$, mais si $y \notin D(A^*)$, alors $\|x_k\| \rightarrow +\infty$ lorsque $k \rightarrow +\infty$.

Le théorème 2.4.2 implique que l'itération doit être terminée de manière appropriée lorsqu'on traite des données perturbées $y^\delta \notin D(A^\dagger)$, en raison d'instabilités numériques.

Simulation de la méthode du gradient

3.1 Débrouillage de l'image

L'une des applications les plus célèbres de la théorie des problèmes mal posés est la suivante de récupérer une image nette à partir de son observation floue, c'est-à-dire le débrouillage de l'image. Ce problème se pose fréquemment dans les sciences et technologies de l'imagerie, y compris les applications optiques, médicales et astronomiques. caractéristiques et motifs importants, tels que ceux d'une planète lointaine ou d'un tissu microscopique.

En raison de son importance, ce sujet a été largement étudié dans la littérature : sans aucune prétention à l'exhaustivité, nous signalons quelques livres [3], [9], [11], [5], ou des chapitres de livres [1], [17] consacrés à ce problème. Dans la plupart des applications, les flous sont introduits par trois types différents de facteurs physiques : optiques, mécaniques ou de support. facteurs physiques : optiques, mécaniques ou induits par le milieu, qui peuvent conduire à des qui peuvent conduire respectivement à des flous de mise au point, des flous de mouvement ou des flous atmosphériques. Nous renvoyons le lecteur à [3] pour un compte rendu plus détaillé des processus physiques associés.

Mathématiquement, une image continue (analogique) est décrite par une fonction non négative $f = f(x)$ sur \mathbb{R}^2 supportée par un domaine $2D$ (rectangulaire) et le processus de flou est un opérateur linéaire ou non linéaire K agissant sur l'espace fonctionnel quelconque. Puisque nous nous concentrerons uniquement sur les problèmes de flou linéaire, K est supposé être linéaire. Parmi tous les flous linéaires, le type le plus fréquemment rencontré est le flou invariant de décalage, c'est-à-dire un flou linéaire $K = K[f]$ tel que pour tout vecteur de déplacement $y \in \mathbb{R}^2$,

$$g(x) = K[f(x)] \implies g(x - y) = K[f(x - y)], x \in \Omega \quad (3.1)$$

Il est bien connu en traitement du signal ainsi qu'en théorie des systèmes [16] qu'un opérateur linéaire invariant par décalage doit être sous la forme d'une convolution:

$$g(x) = K[f(x)] = m * f(x) = \int_{\Omega} m(x - y)f(y)dy, \quad (3.2)$$

pour une fonction noyau appropriée $m(x)$, ou la fonction de répartition des points (FRP). La fonction $g(x)$ est l'image analogique floue qui est convertie en une image numérique par un processus de numérisation (ou échantillonnage). Une image numérique est généralement enregistrée à l'aide d'un DCC (dispositif à couplage de charge), qui est une matrice d'éléments d'image, qui est un réseau de minuscules détecteurs (puits de potentiel), disposés dans une grille rectangulaire, capables d'enregistrer le temps de propagation de l'image. Rectangulaire, capable d'enregistrer la quantité, ou l'intensité, de la lumière qui frappe chaque détecteur.

Ainsi, une image numérique en niveaux de gris

$$G = (g_{j,l}), j = 1, \dots, J, l = 1, \dots, L \quad (3.3)$$

est un tableau rectangulaire, dont les entrées représentent les intensités lumineuses (non négatives) capturées par chaque détecteur. La FRP est décrit par une matrice $H = (h_{j,l})$ de la même taille que l'image, dont les entrées sont toutes nulles à l'exception d'un très petit ensemble de pixels (j, l) répartis autour d'un certain pixel (j_c, l_c) qui est le centre du flou. Puisque nous supposons que les points de répartitions sont invariants dans l'espace, le centre du FRP correspond au centre de la matrice 2D.

Dans certains cas, la FRP peut être décrite de manière analytique et H peut être construit à partir d'une fonction plutôt que par l'expérimentation (par exemple, les flous de mouvement horizontal et vertical sont construits de cette manière). les flous de mouvement horizontal et vertical sont construits de cette manière). Dans d'autres cas, la connaissance du processus physique qui provoque le flou fournit une formulation explicite de la FRP. Dans ce cas, les éléments du tableau FRP sont donnés par une expression mathématique précise : par exemple, le flou hors-foyer est donné par la formule suivante :

$$h_{j,l} = \begin{cases} \frac{1}{\pi r^2} & \text{si } (j - j_c)^2 + (l - l_c)^2 \leq r^2, \\ 0 & \text{sinon,} \end{cases}$$

où $r > 0$ est le rayon du flou. Pour d'autres exemples, tels que le flou provoqué par la turbulence atmosphérique ou la FRP associée à un télescope astronomique, nous renvoyons à [11] et aux

références qui s'y trouvent.

En conséquence du processus de numérisation, le modèle continu décrit par (3.1) doit être adapté au cadre discret. Pour ce faire, nous considérons d'abord le cas 1D

$$g(t) = \int m(t-s)f(s)ds. \quad (3.4)$$

Pour fixer les idées, nous supposons que J est pair et que la fonction $f(s)$ est définie dans l'intervalle $[-\frac{j-1}{2}, \frac{j-1}{2}]$. Soit

$$s_j = -\frac{J-1}{2} + j - 1, \quad j = 1, \dots, J \quad (3.5)$$

sont les J points en lesquels l'intervalle est subdivisé et discrétiser m et f de telle sorte que $m(s) = m(s_j)$ si $|s - s_j| < 1$ ou $s = s_j + 1$ et de manière analogue pour m . En approchant (3.4) avec la règle du trapèz

$$g(t) \cong \sum_{j'=1}^J m(t-s_{j'})f(s_{j'}) \quad (3.6)$$

et en recalculant dans les points s_j , nous obtenons les composantes de la version discrétisée de la fonction g :

$$g_j(t) = \sum_{j'=1}^J m(s_j - s_{j'})f(s_{j'}), \quad j = 1, \dots, J \quad (3.7)$$

En conséquence des hypothèses que nous avons faites, (3.7) peut être réécrit comme suit :

$$g_j = \sum_{j'=1}^J h_{j-j'+\frac{j}{2}}f_{j'}, \quad j = 1, \dots, J \quad (3.8)$$

qui est l'expression en composantes de la convolution discrète entre les vecteurs colonnes $h = (h_j)$ et $f = (f_j)$. Nous observons que certains termes de la somme du côté droit de (3.8) peuvent ne pas être définis : cela se produit parce que le support de la convolution entre m et f est plus grand que les supports de m et f . Le problème est résolu par l'extension du vecteur h au vecteur plus grand

$$\tilde{h} = \begin{pmatrix} h_{-\frac{J}{2}+1} \\ \dots \\ h_0 \\ h \\ h_{J+1} \\ \dots \\ h_{J+\frac{J}{2}} \end{pmatrix}, h_j = h_{j+J}, \quad j = -\frac{J}{2} + 1, \dots, \frac{J}{2}$$

et en substituant h par \tilde{h} dans (3.8), ce qui revient à prolonger m périodiquement sur la droite réelle. La convolution (3.8) peut aussi être exprimée sous la forme

$$g = Af, \quad A = (a_{i,j}), \quad a_{i,j} = h_{i-j+\frac{J}{2}}, \quad i, j = 1, \dots, J.$$

Dans le cas de la 2D, en procédant de manière analogue, on obtient

$$g = Af, \tag{3.9}$$

Dans le cas d'une image f de 1024×1024 pixels, alors le système (3.9) a plus d'un million d'inconnues. Pour les problèmes génériques de cette taille, le calcul de la décomposition en valeurs singulières n'est généralement pas possible.

La FRP est appliquée à chaque pixel puis ces frottis sont additionnés. Comme il s'agit d'un processus "linéaire", cela revient à multiplier notre vecteur image par une matrice. Ainsi, si x est notre image non floue et A est la matrice correspondant au processus de flou FRP, alors l'image floue est $Ax = b$.

Maintenant, étant donné une image floue b , nous devons "simplement" résoudre le système $Ax = b$ pour x . En algèbre linéaire, nous apprenons à résoudre des systèmes linéaires en utilisant la "réduction de rangée". Il s'avère que cela ne fonctionne pas pour nous pour plusieurs raisons. Le graphe de f est une sorte d'"hyper-surface". Dans Analyse, on apprend à connaître les vecteurs gradients. Le gradient d'une fonction, ∇f , est un vecteur qui pointe dans la direction de l'augmentation maximale. Le négatif du gradient pointe alors dans la direction de la diminution maximale. Il s'avère que le gradient de f est

$$\nabla f = Ax - b$$

Par conséquent,

$$\nabla f = 0 \quad \iff \quad Ax - b = 0 \quad \iff \quad Ax = b$$

Les points où le vecteur gradient est nul sont des points critiques. Si nous voulons résoudre notre système, nous devons trouver un "minimum" pour f . Pour ce faire, il suffit de choisir un point quelconque de notre surface et de le descendre. Pour atteindre le minimum le plus rapidement possible, nous devons descendre le long de $-\nabla f$ puisque c'est la direction de la décroissance maximale. C'est en gros ce que fait la méthode CG.

3.2 Fonction de Répartition Ponctuelle

Algorithme de FRP sur matlab

```
function [v] = FRP(sigma,x,y)
% renvoie la valeur de la fonction de répartition des points
% Nous utilisons une FRP gaussienne (tronquée loin de son centre - l'origine).
% son centre - l'origine).

if x >= -8 & x <= 8 & y >= -8 & y <= 8
v = exp(-sigma * (x.^2 + y.^2));
else
v = 0;
end;
```

3.3 Application de FRP Gaussienne

Application de FRP Gaussienne

```

function [g] = applPSFGauss(sigma, f, x)
% Retourne le graphique flou de f
% si x = 0, le graphique flou est généré,
% sinon il n'est pas tracé.
% sigma détermine le degré de dispersion de la lumière
% (à quel point le flou est mauvais.)
% Obtenir la taille
[mn] = size(f) ;
% Créez des blocs Toeplitz
for i = -8 : 8
  pour k = 1 : n
    c(k) = FRP(sigma, i, k - 1) ;
    r(k) = FRP(sigma, i, 1 - k) ;
  fin ;
  block(i + 9, 1 : n, 1 : n) = toeplitz(c, r);
fin ;
% appliquer le produit de convolution ligne par ligne
  for i = 1 : m ;
    g(i, 1 : n) = zeros(1, n)
    % applique le produit de convolution
    for j = -8 : 8
      if i - j >= 1 & i - j <= m
        b(:, :) = bloc(j + 9, :, :); g(i, 1 : n) = g(i, 1 : n) + f(i - j, 1 : n) * b;
      end;
    end;
  end;
if x == 0, mesh (end); end;

```

3.4 Produit Intérieur

Produit Intérieur

```
fonction [n] = inner_product(x,y)

% transforme x et y en colonnes en transposant chaque ligne
% en une colonne et en concaténant le tout.
% puis trouver le produit interne régulier
% (x et y sont des matrices sz x ? ?)

sz = size(x,1);

n = 0;
for i = 1 : sz
n = n + x(i,:) * y(i,:)' ;
end ;
```

3.5 Simulation du GC sur Matlab

Algorithme de gradient conjugué sur Matlab

```

Function [g, error, niter, flag] = methoGC(sigma, g, f, maxit, tol)
% g en entrée est l'estimation initiale d'une solution, puis
% g en sortie est la solution du système.
% f est l'image floue.
% niter est le nombre d'itérations réellement effectuées.
% error est la différence entre la réponse exacte et notre réponse.
% tol est l'erreur permise
% flag est mis à 1 si la fonction retourne avant que error ≤ tol
  flag = 0;
  niter = 0;
% r est le résidu (erreur).
  r = f - applPSFGauss(sigma,g,1)
% Direction de recherche initiale.
  d = r;
% norme de f (en tant que vecteur : transformer les lignes en colonnes et concaténer)
  norm_f = sqrt(inner_product(f,f)) ;
  if(norm_f == 0.0), norm_f = 1.0; fin;
  erreur = sqrt(inner_product(r,r))/norm_f;
  for niter = 1:maxit
  if(error < tol), return, end;% sortir de l'erreur si elle est suffisamment petite.
% multipliez notre direction de recherche par la matrice de flou.
  Ad = applyGaussianPSF(sigma,d,1);
% le carré de la norme de la matrice de flou de d.
  inp_Ad_d = inner_product(Ad,d);
% Taille du pas (distance à parcourir dans la direction du gradient).
  alpha = inner_product(r,d) / inp_Ad_d;
% Améliorer le
  g = g + alpha * d;
% mettez à jour le résidu (erreur).
  r = f - applPSFGauss(sigma,g,1);
% le facteur conjugué pour forcer nos directions de recherche
% à être orthogonales (dans un certain sens).
  beta = -inner_product(r,Ad) / inp_Ad_d;
% Nouvelle direction de recherche.
  d = r + beta * d;
% mesure de l'erreur relative.
  error = sqrt(inner_product(r,r)) / norm_f;
  end;
  if (error > tol) flag = 1; end;

```

Charger Mon Image sur MATLAB

```
function myImg = Load_My_Image(pic, gry, flag)
% Cette fonction charge le fichier "pic" (qui doit être un jpeg).
% Si gry = 0, l'image est chargée en couleur.
% Si gry = 1, il est transformé en une image en niveaux de gris.
% flag détermine si l'image est affichée.
% flag = 0 non affiché / flag = 1 affiché

myImg = imread(pic) ;

if gry == 1
tmp = .2989 * myImg(:, :, 1) + .5870 * myImg(:, :, 2) + .1140 * myImg(:, :, 3) ;
myImg = tmp ;
end ;

if flag == 1
figure ;
if gry == 1
colormap(gray(256)) ;
fin ;
imshow(myImg) ;
fin ;

% convertir en double pour que l'image puisse être manipulée.
myImg = double(myImg) ;

endfunction
```

Flouer Mon Image

```

fonction B = Blur_Picture(pic, gry, flag)
% Floute "pic" (un jpeg) et renvoie l'image floue.
% L'image est convertie en niveaux de gris si gry = 1 et
% l'image est affichée si flag = 1.

% Chargez l'image. A est une matrice m par n si niveaux de gris
% et un tenseur m par n par 3 si en couleur (pensez à 3 matrices
empilées les unes sur les autres – une pour chaque valeur RVB.
A = Load_My_Image(pic, gry, 0) ;

if gry == 0 % Couleur
% Flouter les rouges, les verts, puis les bleus.
B( :,:,1) = applyGaussianPSF(0.1, A( :,:,1), 1) ;
B( :,:,2) = applyGaussianPSF(0.1, A( :,:,2), 1) ;
B( :,:,3) = applyGaussianPSF(0.1, A( :,:,3), 1) ;
else % Niveaux de gris
B = applyGaussianPSF(0.5, A, 1) ;
fin ;

if flag == 1 % Affiche l'image
figure ;
if gry == 1
colormap(gray(256)) ;

% redimensionner pour faire apparaître l'image.
x = max([max(max(B)),0.000001]) ;
scaleB = (1/x) * B;
sinon
% redimensionner pour que l'image soit visible.
x = max([max(max(max(B))),0.000001]) ; scaleB = (1/x)*B ; fin ;
%image(monImg) ;

imshow(scaleB) ;

fin ;

imwrite(scaleB,'Er_blur.png')

endfunction

```

Défloué Mon Image

```

fonction A = Deblur_My_Image(pic, n, gry, flag)
% Défloue l'image stockée dans la matrice/tensor B
% et renvoie l'image "non floutée".
% L'image est considérée comme étant en niveaux de gris si gry = 1
% et l'image est affichée si flag = 1.

B=double(imread(pic)) ;

% Notre "supposition" initiale pour alimenter la méthode CG est l'image zéro.
A = zeros(size(B)) ;

if gry == 0 % Couleur

% Exécutez la méthode CG sur des valeurs RVB.
% Chaque appel à la méthode CG est itéré n fois.
[A(:, :, 1), error, niter, flag] = CGmethod(0.1, A(:, :, 1), B(:, :, 1), n, 0);
[A(:, :, 2), erreur, niter, flag] = CGmethod(0.1, A(:, :, 2), B(:, :, 2), n, 0);
[A(:, :, 3), error, niter, flag] = CGmethod(0.1, A(:, :, 3), B(:, :, 3), n, 0) ;

else % Niveaux de gris

% Chaque appel à la méthode CG est itéré n fois.
[A, error, niter, flag] = CGmethod(0.1, A, B, n, 0) ;

fin ;

% Affiche l'image lorsque le drapeau est à 1.
if flag == 1
figure ;
if gry == 1
colormap(gray(256)) ;

% redimensionner pour faire apparaître l'image.
x = max([max(max(A)),0.000001]) ;
scaleA = (1/x) * A ;
sinon
% redimensionner pour que l'image soit visible.
x = max([max(max(max(A))),0.000001]) ;
scaleA = (1/x) * A;
fin;
%image(monImg) ;
imshow(scaleA) ;
imwrite(scaleA,'Er_blur_deblured.png') ;

fin ;

```

Exemple 3.1 :



Figure 3.1: Image Originale



Figure 3.2: Image flouée



Figure 3.3: Image déflouée

Conclusion

Dans ce travail, nous nous sommes intéressés aux problèmes inverses et aux méthodes de résolution. Nous nous sommes particulièrement penchés sur la méthode du gradient conjugué pour laquelle nous avons tenté à travers ce modeste manuscrit, de donner les grandes lignes. Nous avons présenté une partie théorique que nous avons appliquée sur un exemple pratique. Les résultats de nos simulations, sont assez encourageants (à défaut d'être parfaits).

Bibliography

- [1] M.BERTERO AND P.BOCCACCI. Introduction to inverse problems in imaging , *IOP Publishing, Bristol* , 1998.
- [2] H. BIALY. Iterative behandlung linearer funktionalg leichungen., *Arch. Rat. Mech. Anal* , 4 : 166, 1959.
- [3] T.F. CHAN. Image processing and analysis: variational, pde, wavelet, and stochastic methods, *SIAM, Philadelphia*, 2005.
- [4] J. W. DANIEL. The Conjugate Gradient Method for linear and non linear operator Equations, *SIAM J.Numer.Anal*, 1967.
- [5] R. G. DRIGGERS AND E. L. JACOBS. Image Deblurring, *Boston, Artech House Publishers*, 2008.
- [6] V. FRIDMAN. A method of successive approximations for Fredholm integral equations of first kind, *Uspeki Mat. Nauk* , 11 :233-234, 1956. In Russian.
- [7] M. HANKE. Conjugate Gradient Type Methods for Ill-Posed Problems, *man House, Harlow*, 1995.
- [8] H.W.ENGL, HANKE, AND A.NEUBAUER. Regularization of Inverse problems.kluwer, *Dordrecht* , 1996.
- [9] P.C. HANSEN. Deblurring Images, Matrices, Spectra and Filtering, *SIAM, Philadelphia*, 2006.
- [10] M. R. HESTENES. Methods of Conjugate Gradients for Solving Linear Systems, *J. Research Nat. Bur. Standards 49*, 1952.
- [11] P. A. JANSSON. Deconvolution of Images and Spectra, *San Diego, Academic*, 1997.
- [12] S. T. KABANIKHIN. Definition and examples of invese and ill-posed problems, *in J. Inv* , 2008.

-
- [13] A. KIRSCH. An Introduction to the Mathematical Theory of inverse problem, *New York, NY*, 1996.
- [14] L.LANDWEBER. An iteration formula for Fredholm integral equations of the first kind, *Amer.J. Math*, 73 :615-624, 1951.
- [15] V.A.MOROZOV. Methodes for solving incorrectly posed problems, *Springer-Techniques de débruitage*, 2008.
- [16] A. V. OPPENHEIM AND R. W. SCHAFER. Discrete-Time Signal Processing, *Prentice Hall Inc, New Jersey*, 1989.
- [17] C. R. VOGEL. Computational Methods for Inverse Problems, *SIAM Frontiers in Applied Mathematics*, 2002.
- [18] G.WAHDA. Partial approximate solution to Linear Operator Equation When the Data are Noisy, *SIAM Journal on Numerical Analysis*, 1977

Bibliography